HAL
open science

Schémas de discrétisation Compatible Discrete Operator pour les équations de Navier–Stokes d'un fluide incompressible en régime instationnaire

Riccardo Milani

To cite this version:

Riccardo Milani. Schémas de discrétisation Compatible Discrete Operator pour les équations de Navier–Stokes d'un fluide incompressible en régime instationnaire. Mathématiques générales [math.GM]. Université Paris-Est, 2020. Français. NNT : 2020PESC1037 . tel-03413250

HAL Id: tel-03413250

https://pastel.hal.science/tel-03413250v1

Submitted on 3 Nov 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

THÈSE

PRÉSENTÉE POUR OBTENIR LE GRADE DE

DOCTEUR DE L'UNIVERSITÉ PARIS-EST

École doctorale MSTIC : mention Mathématiques Appliquées

par **Riccardo Milani**

Compatible Discrete Operator schemes for the unsteady incompressible Navier–Stokes equations

Schémas *Compatible Discrete Operator* pour les équations de Navier–Stokes d'un fluide incompressible en régime instationnaire

---

Soutenue publiquement le 16 décembre 2020 devant le jury de thèse composé de :

Pr. Roland Becker	Université de Pau et des Pays de l'Adour	Examinateur
Dr. Jérôme Bonelle	EDF R&D	Encadrant industriel
Dr. Pierre Cantin	INSA Toulouse	Examinateur
Pr. Alexandre Ern	ENPC - Université Paris Est	Directeur de thèse
Pr. Raphaèle Herbin	Université d'Aix-Marseille, I2M	Rapporteur
Dr. Stella Krell	Université de Nice - Sophia Antipolis	Examinatrice
Pr. Peter D. Minev	University of Alberta	Rapporteur

Thèse préparée à :

CERMICS
École des Ponts ParisTech - INRIA Paris
6 et 8, Av. Blaise Pascal
77455 Marne-la-Vallée cedex 2, France

EDF R&D
6, Quai Watier
78400 Chatou, France

"Considerate la vostra semenza:
fatti non foste a viver come bruti,
ma per seguir virtute e canoscenza"

Divina Commedia, Inferno, canto XXVI,
Dante Alighieri

Did you exchange
A walk-on part in the war
For a lead role in a cage?

Wish you were here, Pink Floyd

Acknowledgments - Remerciements - Ringraziamenti

Je tiens à remercier tout d'abord mon directeur de thèse Alexandre Ern. Cela était un honneur de pouvoir poursuivre ce projet sous sa direction. Je lui suis particulièrement reconnaissant pour sa patience, que j'ai parfois mise à rude épreuve, sa bienveillance qui a permis une bonne réussite de cette thèse, et ses explications, claires et pédagogiques, qui m'ont beaucoup appris.

En deuxième lieu, j'exprime tous mes remerciements à Jérôme Bonelle, mon encadrant industriel à EDF R&D. Je suis extrêmement reconnaissant pour son aide, pour le suivi qu'il m'a accordé, même au détriment de son temps personnel. Sans son aide et son soutien cette thèse n'aurais pas eu la même ampleur. J'espère que le projet CDO que Jérôme a monté et auquel il se consacre depuis des années pourra continuer encore pour autant de temps, au moins.

I would like to thank Raphaële Herbin and Peter Minev for reviewing my work. It is a great honor. I've met Pr Minev on the second day of my PhD: I feel that him being part of the final committee is the logical end of this project. I extend my thanks to Stella Krell, Roland Becker, and Pierre Cantin for kindly accepting to be in the committee.

Ce projet de thèse a été supporté par EDF R&D que je remercie pour son soutien. Je voudrais remercier les chefs de groupe de I8A, Marc Boucker, qui m'a fait confiance en me choisissant pour ce projet, et Anthony Dyan, qui m'a toujours accordé une grande confiance. Un remerciement spécial à toute l'équipe de *Code_Saturne*, et en particulier à Martin Ferrand, Yvan Fournier, Erwan Le Coupanec et Mathieu Guingo qui ont toujours suivi de près mes travaux et avec lesquels j'ai eu des discussions très enrichissantes. J'ajoute à cette liste Jean-Marc Hérard qui a été bienveillant à mon égard ainsi qu'à celui des autres doctorants.

Un remerciement particulier à tous les doctorants que j'ai rencontrés ces trois années, pour les conseils qui m'ont donnés, pour les discussions, plus ou moins sérieuses, que l'on a eues. Cela s'étend évidemment aux doctorants du CERMICS et INRIA, comme Nicolas, Amina, Ani, Frédéric et Karol, tout comme à ceux d'EDF, Benjamin, Germain, Vladimir, Lucie et Li. Une place spéciale méritent Gaëtan et Clément avec lesquels j'ai partagé un bureau, des cafés, des blagues et, surtout, beaucoup de kilomètres avec des baskets aux pieds.

J'adresse aussi une pensé aux *anciens* de la section Basket 2012 et en particulier à François, Vincent, Lisa et Justine.

J'embrasse et remercie Ophélie qui a partagé ces derniers mois avec moi : ils n'étaient pas des plus faciles, ton soutien inconditionnel m'a beaucoup aidé. Sois rassurée : la suite sera sans doute plus sereine.

Venendo alla parte "Italia", ringrazio tutta la compagnia *del comune.* Mi sorprendo ogni volta che torno e ho l'impressione che nulla cambi, come se in realtà non fossero passati mesi dall'ultima volta che ci siamo visti.

Dedico un pensiero speciale a mia sorella Gaia, costantemente presente quando avevo bisogno farmi risollevare il morale. Credo che non siamo mai stati così vicini, nemmeno quando abitavamo insieme (o forse è proprio quella la ragione). Le faccio un enorme "in bocca al lupo" per la nuova fase della sua vita che comincerà tra qualche settimana.

Infine, un enorme ringraziamento a tutta la mia famiglia, soprattutto ai miei genitori, Pinuccia e Fabio. Non è stato un periodo semplice per nessuno, ma il loro sostegno, purtroppo spesso solamente virtuale, non è mai mancato. Se sono qui oggi, lo devo tutto a loro.

Thanks! Merci ! Grazie!

Résumé

Nous développons des schémas dits *face-based Compatible Discrete Operator* (CDO-Fb) pour les équations de Stokes et Navier–Stokes incompressibles en régime instationnaire. Des opérateurs pour la reconstruction du gradient, de la divergence et un autre pour le terme de convection sont proposés. On montre que l'opérateur de divergence discret permet de satisfaire une condition inf-sup, tandis que l'opérateur de convection discret est dissipatif, propriété cruciale pour le bilan d'énergie. Le schéma de discrétisation est d'abord testé dans le cas stationnaire sur des maillages généraux mais aussi déformés, afin d'illustrer la flexibilité et la robustesse de la discrétisation CDO-Fb. Dans un deuxième temps, l'attention est placée sur les techniques de marche en temps. En particulier, nous étudions l'approche monolithique traditionnelle qui consiste à résoudre directement le système de point-selle, et la méthode de Compressibilité Artificielle (AC), qui permet de ne plus avoir un système de point-selle à résoudre au prix d'une relaxation du bilan de masse. Trois stratégies classiques pour le traitement du terme non linéaire dû à la convection sont examinées : l'algorithme de Picard, la linéarisation et l'explicitation. Des résultats numériques utilisant des schémas temporels du premier ordre d'abord, puis du deuxième ordre, montrent que la méthode AC constitue une alternative précise et efficace à l'approche monolithique traditionnelle.

Mot-clés : *schémas CDO, maillages polyédriques, Navier–Stokes, Compressibilité Artificielle, condition inf-sup, convection.*

Abstract

We develop face-based *Compatible Discrete Operator* (CDO-Fb) schemes for the unsteady, incompressible Stokes and Navier–Stokes equations. We introduce operators discretizing the gradient, the divergence, and the convection term. It is proved that the discrete divergence operator allows one to recover a discrete inf-sup condition. Moreover, the discrete convection operator is dissipative, a paramount property for the energy balance. The scheme is first tested in the steady case on general and deformed meshes in order to highlight the flexibility and the robustness of the CDO-Fb discretization. The focus is then moved onto the time-stepping techniques. In particular, we analyze the classical monolithic approach, consisting in solving saddle-point problems, and the Artificial Compressibility (AC) method, which allows one to avoid such saddle-point systems at the cost of relaxing the mass balance. Three classic techniques for the treatment of the convection term are investigated: Picard iterations, the linearized convection and the explicit convection. Numerical results stemming from first-order and then from second-order time-schemes show that the AC method is an accurate and efficient alternative to the classical monolithic approach.

Keywords: *CDO schemes, polyhedral meshes, Navier–Stokes, Artificial Compressibility, inf-sup condition, convection.*

Contents

Résumé de la Thèse **1**

1 Introduction **11**
1.1 Industrial context . . . 11
1.2 Compatible Discrete Operator schemes . . . 13
1.3 Numerical methods for the Navier–Stokes equations . . . 17
1.4 Document overview . . . 34

2 Discrete face-based CDO setting **37**
2.1 The mesh . . . 37
2.2 Functional discrete setting and degrees of freedom . . . 40
2.3 Velocity gradient reconstruction . . . 43
2.4 Velocity-pressure coupling . . . 47
2.5 Scalar-valued advection and vector-valued convection . . . 50
2.6 Source term . . . 60

3 The steady Navier–Stokes equations **61**
3.1 Stokes equations with face-based CDO . . . 62
3.2 Navier–Stokes equations with face-based CDO . . . 68
3.3 Preliminary numerical setting . . . 71
3.4 Numerical results: Stokes equations . . . 75
3.5 Numerical results: Navier–Stokes equations . . . 80

4 First-order time-stepping for the Navier–Stokes equations **93**
4.1 Preliminary notions . . . 94
4.2 Velocity-pressure couplings and time-stepping techniques . . . 96
4.3 Convection treatments . . . 101
4.4 Numerical results: Stokes equations . . . 105
4.5 Numerical results: Navier–Stokes equations . . . 113
4.6 Detailed results . . . 122

5 Extension to second-order time-stepping **131**
5.1 Second-order time-schemes . . . 131
5.2 Numerical results: Stokes equations . . . 136
5.3 Numerical results: Navier–Stokes equations . . . 141
5.4 Detailed results . . . 148

6 Conclusions and perspectives **155**

Acronyms **159**

Bibliography **161**

Résumé de la Thèse

Contexte industriel

Les applications de la mécanique des fluides dans un contexte industriel sont nombreuses et concernent à la fois les échelles macroscopiques (e.g. la météorologie), moyennes (e.g. aéronautique) et microscopiques (e.g. écoulements dans des fissures). Le département MFEE (*Mécanique des Fluides, Énergie et Environnement*) fait partie de l'unité Recherche et Développement (R&D) de l'entreprise EDF. Les applications industrielles traitées au sein du département MFEE sont nombreuses et peuvent être regroupées dans deux catégories principales : l'optimisation de la production d'énergie et les études de sureté. Par exemple, des études sur les écoulements atmosphériques sont menées pour prévoir le productible d'un champ d'éoliennes, ou dans un autre contexte pour mieux évaluer et minimiser les conséquences de potentiels rejets industriels sur les zones limitrophes. Un autre exemple concerne les études de sureté à long terme dans lesquelles on évalue les écoulements dans les couches géologiques entourant un centre de stockage de déchets nucléaires. De plus, une grande attention est dédiée à la compréhension des phénomènes physiques complexes qui se développent dans un réacteur nucléaire. En particulier, les codes de calcul développés au sein du département MFEE sont utilisés pour simuler la thermohydraulique des composants principaux d'un réacteur, comme notamment les générateurs de vapeur, les pompes, la cuve et les assemblages combustibles. Le but de ces simulations est d'optimiser l'efficacité de ces composants tout en assurant un niveau de sureté maximum.

Afin de traiter ces applications, EDF R&D développe depuis 1998 *Code_Saturne*[1], un logiciel libre (open-source) pour la simulation d'écoulements tridimensionnels monophasiques, reposant sur une description eulérienne et/ou lagrangienne des écoulements. *Code_Saturne* est un code efficace, flexible et bien adapté à la variété des besoins industriels rencontrés. Il est également massivement parallèle, ce qui lui assure de bonnes performances sur des études de grande taille. Dans une optique d'amélioration continue de *Code_Saturne*, deux axes de travail sont privilégiés : l'amélioration (i) de la robustesse par rapport à la qualité des maillages et (ii) de la représentativité physique de la solution. D'une part, les maillages générés à partir de géométries complexes abordées dans les études d'ingénierie sont parfois de mauvaise qualité ou avec des formes non triviales (des exemples sont reproduits dans la Fig. 1). En effet, la génération de maillages suit souvent une stratégie de type *divide-and-conquer* : les géométries sont d'abord coupées en plusieurs sous-domaines qui sont ensuite maillés indépendamment avant d'être recollés pour obtenir le domaine de calcul global. Ceci conduit parfois à la génération de mailles ayant un mauvais rapport d'aspect ou avec des nœuds qui ne coïncident pas entre eux. Il est donc souhaitable de disposer d'une discrétisation spatiale pouvant traiter tout type de maille sans perte de précision. D'autre part, on souhaite aussi améliorer la qualité des solutions générées. Pour ce faire, la conservation au

[1] `https://github.com/code-saturne/code_saturne`, `http://code-saturne.org`

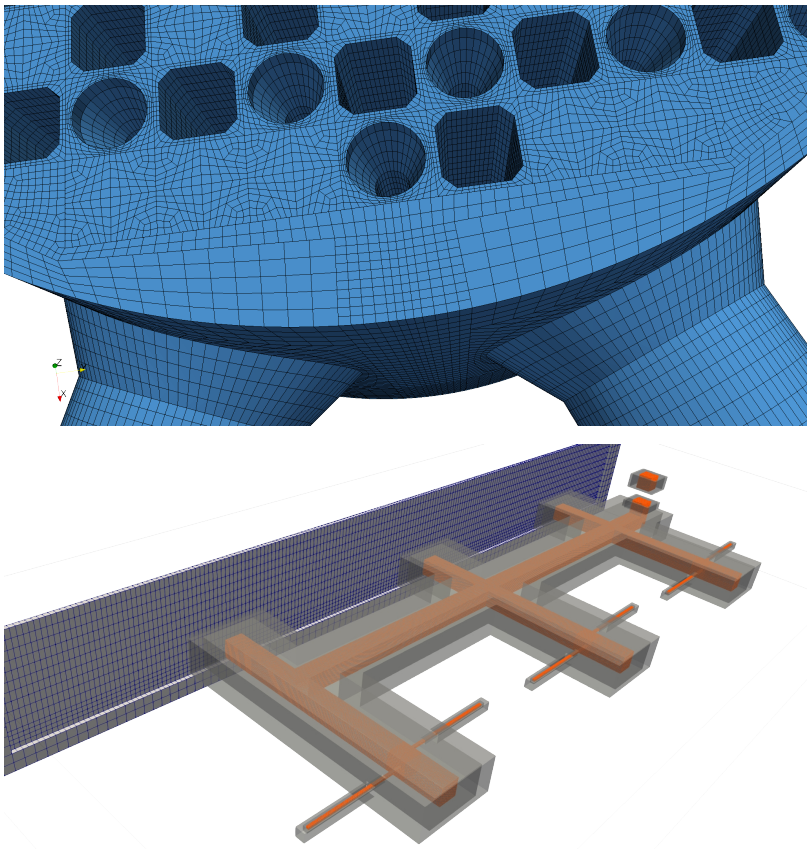
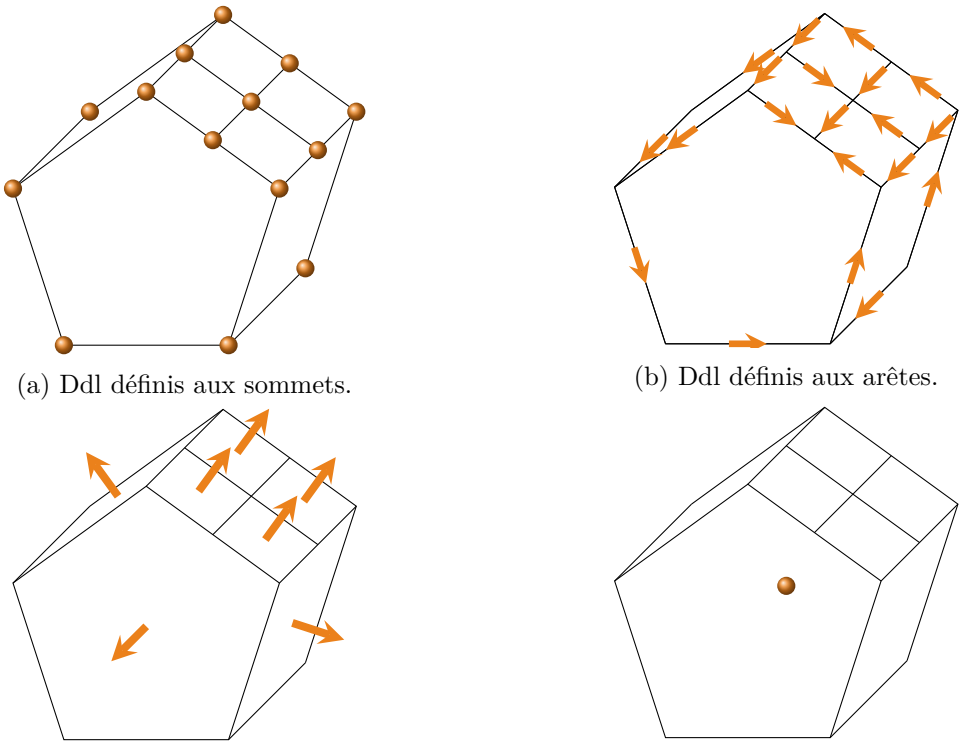


FIGURE 1 – Exemples de maillages avec recollement non conforme utilisés pour des applications industrielles. Haut : écoulement dans une cuve simplifiée de réacteur nucléaire. Bas : partie d'un site de stockage souterrain.

niveau discret des propriétés valables dans le cas continu, comme les relations entre opérateurs différentiels ($\underline{\nabla} \cdot (\underline{\nabla} \times) = 0$ ou $\underline{\nabla} \times (\underline{\nabla}\,) = 0$) et l'absence de modes parasites, serait bénéfique.

La discrétisation actuelle de *Code_Saturne* repose sur une méthode de Volumes Finis (FV) où les degrés de liberté (ddl) des variables vitesse et pression sont co-localisés aux mailles. De plus, *Code_Saturne* traite les maillages polyédriques. Un algorithme de projection-correction est utilisé pour la marche en temps. Les objectifs ont conduit au développement des schémas *Compatible Discrete Operator* (CDO, en français *Opérateurs Compatibles Discrets*) depuis 2011. Des schémas faisant partie de la famille CDO sont actuellement disponibles dans *Code_Saturne* comme alternative à la discrétisation FV. Les principaux domaines d'application actuelle des schémas CDO sont les écoulements souterrains et les problèmes en régime diffusif.

Dans cette Thèse, nous étendons les schémas CDO aux équations de Navier–Stokes (NS) pour un fluide incompressible en régime stationnaire ou instationnaire. Ces équations régissent la dynamique des fluides. Savoir les résoudre ouvre donc la porte à l'utilisation des schémas CDO à un vaste domaine d'applications. Le deuxième axe de la Thèse se concentre sur l'exploration de deux techniques de couplage vitesse-pression. Nous examinons l'utilisation d'un couplage traditionnel obtenu avec l'approche dite monolithique (totalement couplée) et de la méthode de Compressibilité Artificielle (AC). Lorsqu'on traite un cas stationnaire ou lorsqu'une grande qualité d'approximation est demandée, on privilégie l'approche monolithique. Néanmoins, cette approche peut nécessiter un important effort numérique



(a) Ddl définis aux sommets.

(b) Ddl définis aux arêtes.

(c) Ddl définis aux faces.

(d) Ddl défini à la cellule.

FIGURE 2 – Positionnement possible des ddl dans une discrétisation CDO. Seuls les ddl visibles sont montrés (à l'exception de ceux définis aux mailles qui sont toujours montrés). Chaque symbole (sphère, flèche ou cercle) représente une valeur scalaire, même pour les quantités vectorielles (comme la circulation d'un champ de vecteurs dans la discrétisation définie aux arêtes).

pour la résolution des systèmes linéaires associés. Si, au contraire, l'efficacité (i.e. le temps de résolution) est primordiale, la méthode AC est à privilégier. Ces deux stratégies ont été retenues au vu de leurs propriétés complémentaires qui assurent une grande flexibilité aux schémas CDO dans le cadre des équations de NS.

Les schémas Compatible Discrete Operator

Nous introduisons ici les schémas *Compatible Discrete Operator* (CDO). Après une revue des principes de conception de CDO et des discrétisations possible, nous nous concentrons sur les schémas CDO dits *face-based* (i.e. définis aux faces) que nous utilisons dans cette Thèse.

Une introduction aux schémas CDO

Sous la dénomination CDO on retrouve des schémas d'ordre bas pouvant être utilisés sur des maillages polyédriques. Selon les schémas CDO considérés, les reconstructions des champs discrets sont soit conformes ou non-conformes. À la base des schémas CDO, on trouve une structure flexible qui permet de choisir un type de discrétisation en fonction du problème à traiter et des variables qui y sont associées. Les schémas CDO font partie de la famille des schémas dits *mimétiques*. La construction de ces schémas repose sur des opérateurs différentiels dont le noyau est préservé aussi au niveau discret.

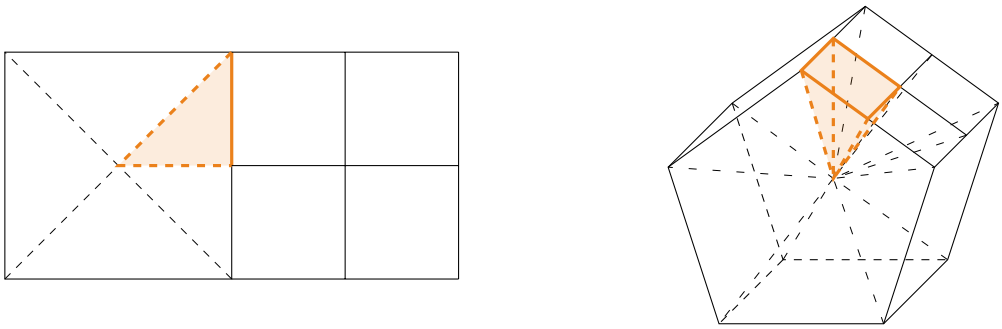


FIGURE 3 – Exemples d'une subdivision d'une cellule avec un *hanging node* en 2D (gauche) et en 3D (droite). À cause du *hanging node*, le carré à gauche de la cellule en 2D est considéré comme un pentagone avec deux faces coplanaires. Pour la même raison, la cellule en 3D (droite) a 10 faces (au lieu de 7), dont 4 coplanaires. Un sous-triangle (gauche) et une sous-pyramide (droite) obtenus en considérant une face de la cellule comme base et le barycentre de la cellule comme sommet sont mis en évidence.

La première notion clé de la discrétisation CDO concerne la définition des ddl, choix qui découle directement de la nature physique de la variable considérée. Prenons un maillage 3D et définissons les entités géométriques associées : cellules, faces, arêtes et sommets. Les potentiels sont associés aux sommets, les circulations aux arêtes, les flux aux faces et, enfin, les densités aux cellules. Les différents positionnements des ddl sont illustrés dans la Fig.2. La compréhension des structures physique et différentielle sous-jacentes au problème traité est aussi à la base des schémas CDO. L'outil principal pour mener une telle analyse est la géométrie différentielle. Le lecteur peut se référer aux travaux précurseurs de KRON (1945, 1953), TONTI (1975), BRANIN (1966) et BOSSAVIT (1988) pour une présentation des principes fondamentaux ou à BONELLE (2014, Section 2.1) pour une revue récente. En particulier, on peut construire un ensemble d'opérateurs différentiels discrets qui satisfont leurs contreparties continues respectives. Par exemple, la divergence d'un rotationnel et le rotationnel d'un gradient sont identiquement nuls au niveau discret.

La deuxième notion clé des schémas CDO est l'utilisation d'un opérateur de Hodge à chaque fois qu'une loi phénoménologique entre en jeu. Ce type d'opérateur met en relation les ddl de natures différentes. Par exemple, il associe une circulation à un flux. Les schémas CDO utilisent souvent une sous-partition du maillage (par exemple en considérant les pyramides formées par les faces et les barycentres des cellules) pour construire ces opérateurs de Hodge. Cette nouvelle partition est fictive : elle est cachée à l'utilisateur et n'a pas besoin d'être construite explicitement. Les deux types d'éléments géométriques, les primaux (définis sur le maillage originel) et duaux (sur la nouvelle subdivision), sont en relation : les sommets, arêtes, faces et cellules primaux sont associés respectivement aux cellules, faces, arêtes et sommets duaux. Les ddl et les opérateurs différentiels peuvent être définis à la fois sur le maillage primal et dual. Dans le cadre CDO, les opérateurs différentiels discrets sont exacts et n'introduisent pas d'erreur de consistance, au contraire des opérateurs de Hodge. De cette façon l'erreur est portée par les opérateurs de Hodge et se positionne au même niveau que l'erreur due au modèle physique, ce qui est l'une des caractéristiques des schémas CDO.

Le développement et l'analyse des schémas CDO ont été initiés au cours de deux Thèses. La méthode a été introduite par BONELLE (2014) avec un intérêt particulier aux problèmes elliptiques et de Stokes (voir également BONELLE et ERN (2014, 2015)). Les équations de transport ont été abordées par CANTIN (2016) (voir également CANTIN et ERN (2016), CANTIN et al. (2016) et CANTIN et ERN (2017)). Pour un problème donné, on choisit la discrétisation CDO la plus adaptée parmi celles disponibles en fonction de la nature physique

de la variable principale. Ainsi, on retrouve les situations suivantes :

- Pour un problème elliptique (voir Bonelle (2014) et Bonelle et Ern (2014)), la variable principale est un potentiel. Ainsi, les ddl sont positionnés aux sommets, primaux ou duaux. Quand les sommets primaux sont retenus, on dit que la discrétisation est *vertex-based* (définie aux sommets) : une telle discrétisation est proche des méthodes éléments finis conformes. Si les sommets duaux sont choisis (associés aux cellules primales), la discrétisation est dite *cell-based* (définie aux cellules). Elle conduit à un problème de type point-selle, une structure proche des méthodes éléments finis mixtes, comme l'élément de Raviart-Thomas (à l'ordre bas). La discrétisation cell-based a été étendue au moyen d'une procédure d'hybridation dans Bonelle (2014, Section 8.3) en ajoutant des ddl additionnels aux faces. Ce nouveau type, dont on parlera plus bas, est dit *face-based* (défini aux faces).

- Pour un problème de Stokes en formulation rotationnelle (Bonelle, 2014 ; Bonelle et Ern, 2015), nous nous intéressons au potentiel dérivé de la pression (c'est-à-dire, la pression divisée par la masse volumique). Dans ce cas, on peut considérer une discrétisation soit vertex-based soit cell-based. Ainsi, la vitesse est vue comme une circulation sur les arêtes primales dans la version vertex-based, ou comme un flux sur les faces duales dans la version cell-based.

- Pour des problèmes scalaires d'advection-diffusion (Cantin, 2016 ; Cantin et Ern, 2016 ; Cantin et al., 2016), la variable est de nouveau un potentiel, conduisant à une discrétisation vertex- ou cell-based. Pour un problème vectoriel, la variable est une circulation, donc définie aux arêtes (Cantin et Ern, 2017).

Les positionnements possibles des ddl sont rappelés dans la Fig. 2. Parmi les applications qui sont aujourd'hui traitées à l'aide des schémas CDO dans *Code_Saturne*, une discrétisation aux sommets est choisie pour les écoulements souterrains (qui peuvent être considérés comme des problèmes de transport scalaire). Tandis que dans le cadre de l'électromagnétisme, une discrétisation aux arêtes est retenue. Enfin, pour des applications de dynamique des fluides, la discrétisation aux faces est retenue : cela constitue une des contributions majeures de cette Thèse.

La discrétisation face-based

Nous nous intéressons dans cette Thèse à la discrétisation face-based (CDO-Fb) présentée brièvement ici. Pour plus de précisions, le lecteur peut se référer à Bonelle (2014, Section 8.3) pour la conception de ces schémas et l'application aux problèmes de diffusion scalaire. Plus de détails sur le cadre discret seront donnés dans la Section 2.2. Pour les équations de NS, la vitesse est hybride et considérée comme un potentiel vectoriel défini aux faces et aux cellules (contrairement à la forme rotationnelle choisie dans Bonelle (2014) et Bonelle et Ern (2015)).

La discrétisation aux faces est obtenue à partir d'une formulation mixte avec une classique procédure d'hybridation, comme décrit dans Boffi et al. (2013). En effet, on ajoute au système cell-based des ddl aux faces. Un exemple est donné dans la Fig. 4. Il est important de remarquer que le potentiel a une unique valeur sur une face interne : cette valeur est la même pour les deux cellules partageant cette face. Pour un problème de diffusion scalaire, l'hybridation permet d'éviter la résolution d'un système de type point-selle (à la différence des éléments finis mixtes classiques). De plus, avec une procédure de condensation statique (ou complément de Schur), les ddl définis aux cellules sont éliminés, réduisant la taille du système final. Les ddl aux cellules peuvent être calculés après la résolution dans une phase

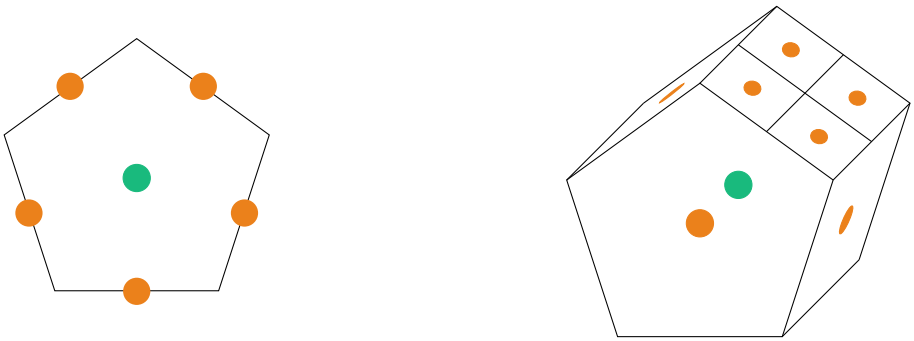


FIGURE 4 – Cadre discret pour un problème scalaire traité avec les schémas CDO-Fb en 2D (gauche) et 3D (droite). Les quantités définies aux faces sont dessinées en orange, celles aux cellules en vert. Seuls les ddl visibles sont montrés.

de post-traitement. Par ailleurs, la discrétisation face-based s'appuie sur une reconstruction non-conforme du potentiel.

L'étape suivante est l'écriture d'une formulation variationnelle. Comme c'est souvent le cas pour les méthodes hybrides, la partie du problème discret associée aux fonctions test définies aux cellules traduit la loi de conservation de l'équation considérée au niveau de la cellule. Les fonctions test définies aux faces, quant à elles, sont associées à l'équilibre des flux aux faces.

Au cœur de la méthode face-based correspond une reconstruction du gradient discret qui utilise une subdivision pyramidal des cellules. Une première partie de la reconstruction prend en compte les différences entre la valeur à la cellule et celles aux faces. Cette partie conduit à un gradient constant sur la cellule et consistant, c'est-à-dire exact pour des fonctions affines. À cela on ajoute une stabilisation dérivée d'une approximation de Taylor et constant par morceaux sur les sous-pyramides (un exemple est donné dans la Fig. 4).

Les schémas CDO-Fb ont des caractéristiques communes avec d'autres méthodes hybrides capables de traiter des maillages généraux. En particulier, dans BONELLE (2014, Prop. 8.38) il est démontré que dans le cas d'un problème scalaire elliptique les schémas CDO-Fb sont équivalents (à un coefficient de stabilisation près) à la méthode Hybrid Finite Volume (EYMARD et al., 2010). Les schémas Hybrid High-Order (HHO), introduits par DI PIETRO et al. (2014) et DI PIETRO et ERN (2015), sont très proches des schémas CDO-Fb lorsque l'on considère l'ordre le plus bas, $k := 0$. Prenons à nouveau un problème scalaire elliptique. Les deux schémas, CDO-Fb et HHO($k = 0$), considèrent un ddl par face et par cellule,[2] et utilisent un opérateur de reconstruction du gradient. La différence est que, d'une part, le schéma CDO-Fb produit un gradient constant par morceaux sur une subdivision de la cellule, tandis que le gradient reconstruit par HHO est constant sur une cellule (cela correspond au gradient consistant mentionné ci-dessus) auquel est ajoutée une stabilisation. Cette stabilisation est une pénalisation aux moindres-carrés sur la différence entre les valeurs définies aux faces et aux cellules. Le principe est synthétisé dans la Fig. 5. Nous utilisons souvent dans cette Thèse les similitudes entre HHO et CDO-Fb pour profiter des résultats d'analyse démontrés dans le cadre HHO.

[2]Ces ddl sont vus par HHO($k = 0$) comme des polynômes d'ordre zéro $p \in \mathbb{P}^0$, où, ayant fixé $k \geq 0$, $\mathbb{P}^k$ est l'espace vectoriel des polynômes d'ordre k au plus, définis aux faces et aux cellules

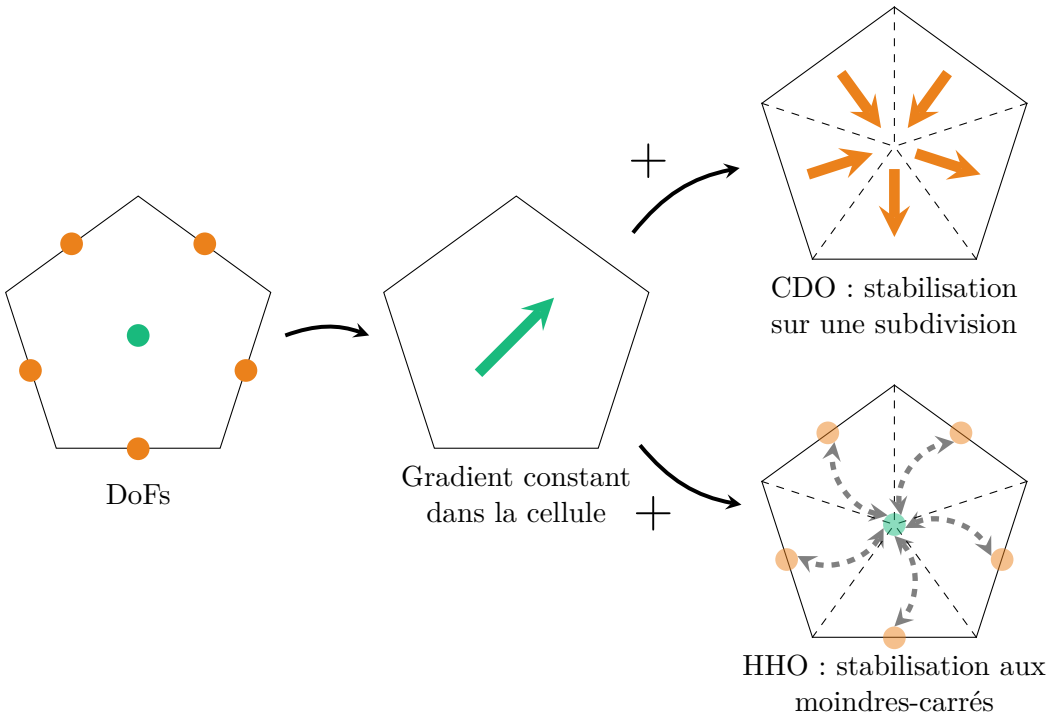


FIGURE 5 – Comparaison entre la reconstruction du gradient d'une fonction scalaire en 2D dans le cadre des schémas CDO-Fb et HHO($k = 0$). Les quantités définies aux faces sont dessinées en orange, celles aux cellules en vert. Les deux reconstructions ont en commun le même gradient consistant (exact pour des fonctions affines). Le gradient CDO est enrichi par une stabilisation constante par morceaux sur une subdivision de la cellule. Tandis que dans le cadre HHO, le gradient constant dans la cellule est complété par un terme de pénalisation.

Principales contributions

Nous présentons ici les sujets développés dans les chapitres suivants de la Thèse. Une partie de la présentation faite au Chapitre 2 et certains résultats du Chapitre 3 sont déjà parus dans

- BONELLE, J., ERN, A. et MILANI, R. (2020). "Compatible Discrete Operator schemes for the steady incompressible Stokes and Navier–Stokes equations". In : *Finite Vol. Complex Appl. IX ; Methods Theor. Aspects*. Sous la dir. de R. KLÖFKORN et al. T. 323. Springer Proc. Math. Stat. Bergen : Springer International Publishing, p. 93–101.

Dans le Chapitre 2, le cadre discret des schémas numériques utilisés dans cette Thèse est défini. Nous donnons les hypothèses de régularité du maillage attendues et nous introduisons la discrétisation CDO-Fb et, en particulier, ses opérateurs différentiels discrets. La reconstruction du gradient introduite par BONELLE (2014, Section 8.3) est étendue au cas vectoriel. Nous prouvons qu'elle est stable et exacte pour des fonctions affines. Ensuite, nous proposons un opérateur divergence qui assure le couplage vitesse-pression et satisfait une condition inf-sup discrète. Enfin, en se basant sur des opérateurs déjà présents dans la littérature, nous introduisons un opérateur d'advection scalaire qui sert de base pour la définition d'un opérateur de convection pour le traitement des équations de NS. Nous démontrons que l'erreur de consistance de l'opérateur scalaire est bornée, que l'opérateur de convection satisfait les équivalents discrets de certaines identités continues classiques d'intégration par parties et que ce dernier est aussi dissipatif.

Nous abordons dans le Chapitre 3 la discrétisation des problèmes de Stokes et NS en régime incompressible et stationnaire. Tout d'abord, nous donnons leurs formulations variationnelles discrètes à l'aide de la méthode CDO-Fb et les comparons aux versions continues classiques. Nous montrons ensuite comment les opérateurs intervenant dans ces formulations discrètes sont construits algébriquement. Enfin, nous les testons sur des cas test bien connus dans la littérature, comme le problème de Bercovier–Engelman (solution d'un problème de Stokes en 2D) ou la cavité entraînée. Lorsqu'une solution analytique est connue, nous nous intéressons aux ordres de convergence en espace pour la vitesse et la pression.

Le Chapitre 4 analyse la version instationnaire des problèmes de Stokes et NS. D'abord, un schéma de discrétisation temporelle implicite à l'ordre un est introduit. Nous procédons ensuite à la discussion de deux stratégies de couplage vitesse-pression courantes dans la littérature : l'approche monolithique, retenue pour sa bonne qualité d'approximation de la contrainte d'incompressibilité, et la méthode de la Compressibilité Artificielle (AC), retenue pour son efficacité en termes de coût de calcul. Nous rappelons les avantages et inconvénients des deux stratégies et établissons également un bilan d'énergie cinétique. Dans un second temps, nous abordons les équations de NS et nous discutons en particulier trois techniques classiques pour le traitement du terme convectif, à savoir, l'algorithme de Picard, la linéarisation et l'explicitation de la convection. Enfin, des résultats numériques validant le cadre introduit dans le chapitre sont présentés. Un problème instationnaire de Stokes de grande taille nous permet de comparer la précision et l'efficacité des deux stratégies de couplage. Avec le cas du vortex de Taylor–Green, nous analysons les conséquences du choix du traitement de la convection.

Le cadre des équations de NS en régime instationnaires est étendu aux schémas temporels de second ordre dans le Chapitre 5. La méthode dite *Backward Differentiation Formula* à l'ordre deux est considérée pour le couplage monolithique. Quant au cadre AC, nous considérons une procédure dite de *bootstrapping* introduite dans Guermond et Minev (2015). Les deux stratégies sont considérées avec les traitements de la convection évoqués plus haut. Les mêmes cas que ceux considérés au Chapitre 4 sont analysés ici pour valider l'approche au second ordre. Les résultats sont ensuite mis en perspective avec ceux obtenus à l'ordre un afin de mettre en évidence l'intérêt potentiel de l'ordre deux en temps.

Perspectives

Un premier axe d'amélioration identifié concerne la robustesse des schémas numériques vis-à-vis du nombre de Reynolds. Tout d'abord, différentes techniques peuvent être considérées pour rendre *pressure-robust* les schémas CDO-Fb, c'est-à-dire rendre l'erreur en vitesse indépendante de celle en pression. En effet, on démontre (voir Lemme 3.3) que l'erreur en vitesse dépend de la norme de la pression via un coefficient proportionnel à l'inverse de la viscosité. Pour éviter cela, une possibilité serait d'adopter des reconstructions adaptées du terme source du bilan de quantité de mouvement, comme proposé par Linke (2014), John et al. (2017) et Lederer et al. (2017). Ces reconstructions ont été considérées dans le cadre des schémas HHO. Ainsi, dans Di Pietro et al. (2016), les auteurs utilisent un opérateur de reconstruction pour des maillages de simplexes en se basant sur les méthodes de Raviart–Thomas.

Les difficultés rencontrées pour des nombres de Reynolds élevés s'appliquent également au traitement de la non-linéarité. Nous avons choisi l'algorithme de Picard pour sa robustesse, tout en connaissant ses limites en termes de performance. Le développement d'une méthode alternative, basée sur la méthode de Newton ou l'accélération de Anderson, pourrait être très bénéfique. De plus, contrairement au cas des équations de Stokes où l'algorithme de Bidiagonalisation de Golub–Kahan couplé à un Gradient Conjugué préconditioonné par

une approche multigrille forme une approche efficace, la performance globale de la présente méthode peut sans doute profiter de solveurs linéaires plus adaptés au cadre des équations de NS (pour l'instant nous utilisons seulement une simple factorisation LU). L'utilisation de solveurs linéaires itératifs robustes (par exemple le *flexible GMRES*, voir par exemple Saad (1996), ou ceux détaillés dans Benzi et al. (2005)) devra s'accompagner de préconditionneurs efficaces (comme ceux proposés dans Benzi et Olshanskii (2006) et Olshanskii et Benzi (2008)).

Les résultats numériques (voir par exemple la Section 4.4) confirment que la méthode AC constitue une alternative fiable et efficace (en termes de coût de calcul) à l'approche monolithique. L'influence du paramètre arbitraire intervenant dans AC pourrait faire l'objet d'analyses ultérieures plus approfondies. D'une part, ce paramètre peut améliorer la précision de la méthode. D'autre part, il peut détériorer le conditionnement des systèmes considérés et ainsi impacter les performances de la méthode. Il serait pertinent de pouvoir identifier de façon automatique l'intervalle dans laquelle se trouve la valeur optimale de ce paramètre. La perturbation de l'équation du bilan masse dans le cadre de la méthode AC mérite également une étude plus approfondie. D'une part, dans le cadre des schémas CDO, pour que l'opérateur de convection soit anti-symétrique, il est nécessaire que la vitesse discrète soit à divergence nulle. Cela n'est pas vérifié en général dans AC. Une première parade consisterait à considérer l'anti-symétrisation de Temam. En particulier, cela peut assurer la conservation de l'énergie cinétique, propriété importante lorsque l'on traite des nombres de Reynolds élevés. D'autres problématiques pourraient surgir lorsqu'un champ à divergence non nulle transporte un soluté. Plusieurs auteurs mettent en garde contre ce non-respect de l'incompressibilité (voir, par exemple, Chippada et al. (1997), Wheeler et al. (2002), Olshanskii et Reusken (2004), Linke (2009) et Galvin et al. (2012)). Pour éviter cela, on pourrait envisager d'introduire une étape de post-traitement dans laquelle un champ à divergence nulle est reconstruit à partir de la solution donnée par la méthode AC.

Dans les cas test que nous considérons, seules les conditions aux limites de Dirichlet ont été abordées. Pourtant, plusieurs types sont acceptables. Par exemple, les conditions aux limites de type symétrie ou de Neumann homogènes nous permettraient d'étendre la portée des schémas CDO-Fb, en particulier dans un but plus applicatif. Dans ce but, la prise en compte de la turbulence via l'intégration de modèles sera un pré-requis.

Tous les développements illustrés dans cette Thèse ont été entièrement intégrés au logiciel industriel de dynamique des fluides *Code_Saturne*. Ces développements sont librement disponibles. La discrétisation CDO-Fb pour des équation de NS (avec l'approche monolithique et une convection implicite traitée à l'aide de l'algorithme de Picard) est actuellement testée à EDF R&D dans le cadre de simulations de solidification, où les nombres de Reynolds restent modérés (une approximation de Boussinesq, une équation d'énergie, ainsi qu'une équation de transport de concentration de soluté sont également considérées). Les comparaisons faites entre les schémas CDO-Fb et la méthode standard actuelle de *Code_Saturne* (une discrétisation spatiale basée sur une méthode de volumes finis co-localisés et une technique de prédiction-correction) montrent un gain en robustesse et en précision en faveur des schémas CDO-Fb. L'utilisateur peut en effet choisir des pas de temps plus grands et obtenir ainsi des meilleures performances.

Chapter 1

Introduction

1.1 Industrial context

The applications of fluid mechanics in an industrial context are numerous and span from the macroscale, e.g. meteorology, through the midscale, e.g. aeronautics, to the microscale, e.g. flows in fractures. The MFEE (*Fluid Mechanics, Energy, and Environment*) department is part of EDF R&D, the research branch of the French electrical utility. The domains of interest for this department within Computational Fluid Dynamics (CFD) are various. Informally, they could be classed into two main categories: optimization of the energy production and safety studies. For instance, atmospheric studies are performed in order to foresee the production of a wind-turbine field, and also to better evaluate and minimize the consequences of potential airborne industrial rejects of a power plant to its surrounding area. Another example is long-term safety studies related to the disposal of nuclear wastes implying groundwater flow simulations. Moreover, a great effort is deployed towards the comprehension of the complex physical phenomena at stake in nuclear power plants. In particular, thermohydraulic simulations are performed on the key components of a nuclear power plant such as, among many others, steam-generators, pumps, the nuclear vessel and fuel assemblies, in order to optimize their efficiency and their lifetime, while still maintaining the highest safety standards.

In order to address CFD applications, EDF R&D has been developing since 1998 the software *Code_Saturne*[1], which is an open-source, multi-purpose solver for single-phase flows. *Code_Saturne* is an efficient and flexible solver, which is thus well adapted to the aforementioned industrial needs, and which enjoys a High-Performance-Computing–oriented parallelized implementation. In a process of continuous development and improvement of *Code_Saturne*, two main axes have been identified: (i) increasing the robustness of the solver with respect to poor quality meshes, and (ii) improving the physical fidelity of the numerical solutions. On the one hand, the complex geometries encountered in engineering studies lead to meshes which sometimes have locally a poor quality and/or complex shapes, see the examples given in Fig. 1.1. As a matter of fact, a divide-and-conquer strategy is often used in order to ease and speed up the process of mesh generation. The geometries are cut into simpler parts which are meshed separately and then joined together, thus generating cells which have a strong aspect-ratio or hanging nodes, for instance. Using a discretization which is able to deal with such kind of meshes without loosing in accuracy is then of paramount importance. On the other hand, one aims at increasing the quality of the solution. An example of this requirement is to preserve at the discrete level the main features

[1] `https://github.com/code-saturne/code_saturne`, `http://code-saturne.org`

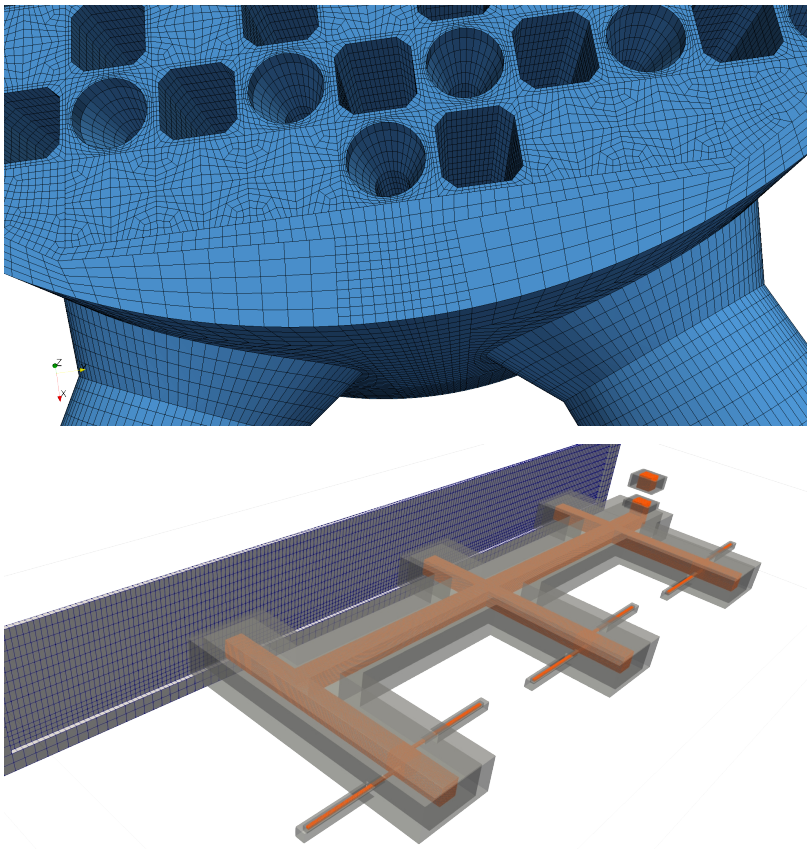


Figure 1.1 – Examples of meshes with nonconforming joining used for industrial applications. Top: flow in a simplified nuclear vessel. Bottom: part of an underground storage facility.

of the continuum system, such as relations between differential operators or the absence of unphysical spurious modes.

The underlying discretization of *Code_Saturne* hinges on a Finite Volume scheme handling polyhedral meshes in which velocity and pressure Degrees of Freedom (DoFs) are colocated at the cells. In *Code_Saturne*, the velocity-pressure coupling within time-marching schemes relies on a projection algorithm (see Archambeau *et al.* (2004) for more details). In this context and with the above goals in mind, the development (design, analysis and implementation) of *Compatible Discrete Operator* (CDO) schemes has been undertaken since 2011. Some schemes belonging to the CDO family are currently available as an alternative discretization in *Code_Saturne*. Today, CDO schemes have become the standard discretization in *Code_Saturne* for industrial applications such as groundwater flows or diffusion dominated-problems.

In this Thesis, CDO schemes are extended to the unsteady, incompressible Navier–Stokes equations (NSE). The ability to deal with such equations which are at the very core of fluid dynamics will allow the CDO framework to extend its domain of application even further. In addition to the development of CDO schemes for spatial discretization, the other main axis of this Thesis is the exploration of two velocity-pressure coupling techniques in order to provide some alternative strategies to the standard projection method of *Code_Saturne*. In particular, we explore the use of the (fully coupled) monolithic approach and of the so-called Artificial Compressibility (AC) approach within the CDO framework. On the one hand, when accuracy is the paramount concern or a steady problem is addressed, one prefers the monolithic approach, which however may need a significant computational effort to solve the

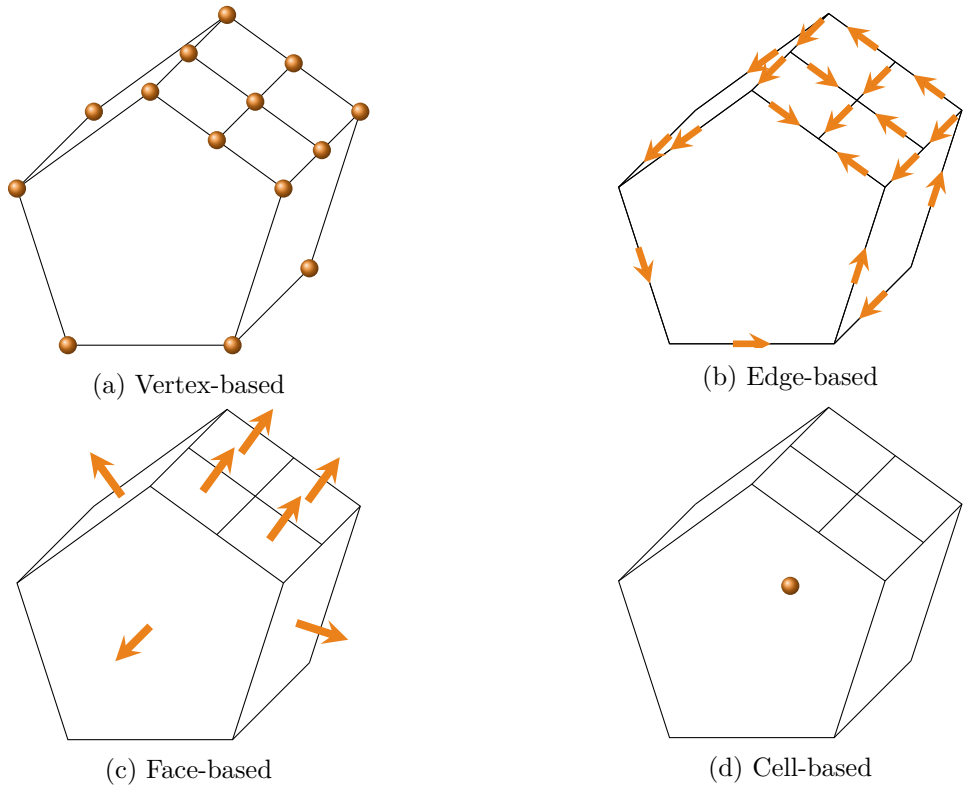


Figure 1.2 – Possible DoFs for a CDO discretization. Only visible DoFs are shown (except for the cell-based ones which are always shown). Every graphical element (ball, arrow or circle) represents a scalar value, even for a vector-valued variable (e.g. the circulation of a vector field in the edge-based discretization).

resulting linear systems. On the other hand, when efficiency is of order, the AC approach is preferable. These two strategies have been identified because of their complementary properties which ensure a great flexibility and versatility to the CDO approach within the context of the NSE.

1.2 Compatible Discrete Operator schemes

Compatible Discrete Operator schemes (CDO) are introduced in this section. After an overview of the driving design principles and the available discretizations within the CDO framework, the face-based schemes, which are the ones considered in this Thesis, are detailed and compared to other face-based methods available in the literature.

1.2.1 A brief introduction to CDO schemes

The CDO framework gathers low-order, conforming or nonconforming schemes for polyhedral meshes. It consists in a flexible structure that allows one to choose a discretization principle adapted to the problem at hand and its related variables. The CDO schemes are part of the broad family of so-called *mimetic* (or *structure-preserving*) schemes. For instance, their design enables them to preserve the kernel of differential operators at the discrete level.

The first key step in the design of CDO schemes is the definition of the DoFs which stems directly from the physical nature of the fields under investigation. Let us consider a three-dimensional mesh and identify its geometric entities as cells, faces, edges, and vertices. Then,

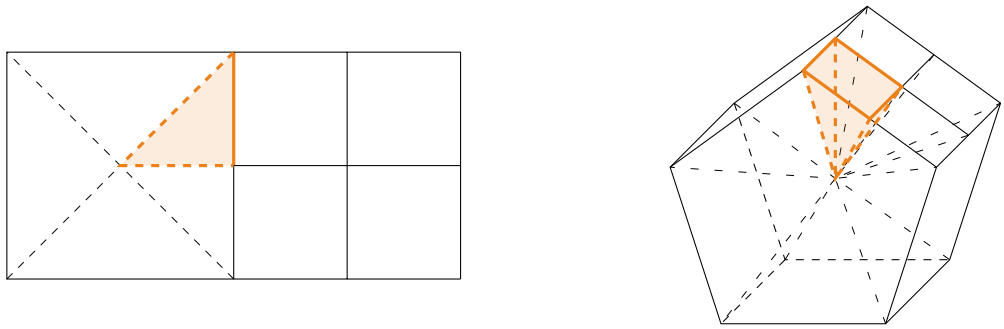


Figure 1.3 – Examples of a face-based subdivision of a 2D (left) and 3D cell (right) with a hanging node. Because of the hanging node, the leftmost square on the left panel is viewed as a pentagon with two coplanar faces, and the pentagon-prismatic cell on the right panel is considered to have 10 faces (instead of 7), among which 4 are coplanar. A subtriangle and a subpyramid obtained by considering a face as basis and the cell barycenter as apex are highlighted.

potentials are defined at vertices, circulations along edges, fluxes across faces, and finally densities in cells. The different discretizations are illustrated in Fig. 1.2. The understanding of how the underlying physical and differential structures of the problem are linked is at the core of CDO schemes. The main tool for such an analysis is differential geometry. The reader is referred to the works of Kron (1945, 1953), Tonti (1975), Branin (1966), and Bossavit (1988) for a presentation of the founding principles of differential geometry and to Bonelle (2014, Section 2.1) for a more recent overview. In particular, among desirable properties, one is able to design a set of discrete differential operators which satisfy the same key features as their continuous counterparts. For instance, the divergence of a curl or the curl of a gradient is always identically zero at discrete level as well.

The second key notion in CDO schemes is the Hodge operator, used whenever a phenomenological law is involved. Such an operator relates DoFs of different nature: for instance, a circulation to a flux. The distinctive feature of CDO schemes is the usage of a partitioning of the mesh in order to locally build the needed Hodge operators. Several ways are available to build such a partitioning, for instance, by considering a face-based subdivision of the cells, see Fig. 1.3. This additional geometric discretization is fictitious: the end user does not see it and indeed he does not need to build it. The two kinds of geometric entities, primal (of the original mesh) and dual (defined when considering the additional subdivision), are in a one-to-one pairing: primal vertices, edges, faces, and cells are associated with, respectively, dual cells, faces, edges and vertices. DoFs and discrete differential operators can be defined on both the primal mesh and the dual mesh. If, on the one hand, the discrete differential operators are exact and do not introduce any consistency error, on the other hand, the Hodge operators do introduce such an error. In doing so, the error occurs at the same level as the physical modeling error, which can be viewed as an attractive feature of CDO schemes.

Two previous PhDs dealt with the development and analysis of CDO schemes: in Bonelle (2014), the method has been deployed for elliptic and the Stokes equations (see also Bonelle and Ern (2014, 2015)), whereas in Cantin (2016) scalar and vector transport equations have been addressed (see also Cantin and Ern (2016), Cantin *et al.* (2016), and Cantin and Ern (2017)). Broadly speaking, given a problem and the associated Partial Differential Equation (PDE), the discretization is chosen depending on the physical nature of the main variable. Let us give some details.

- In elliptic problems (see Bonelle (2014) and Bonelle and Ern (2014)), one seeks a po-

tential: therefore, the DoFs are based on vertices, and either primal or dual entities can be chosen. The former choice leads to the so-called CDO vertex-based discretization which is similar in spirit to the conforming $\mathbb{P}^1$ (where $\mathbb{P}^1$ stands for the space of affine functions) finite element methods on simplicial meshes. Concerning the latter choice, since dual vertices are associated with primal cells, the name cell-based is employed, and the discretized problem has a saddle-point structure similar in spirit to the lowest-order Raviart-Thomas mixed finite element method on simplicial meshes. Furthermore, the cell-based discretization has been extended by means of a hybridization procedure which introduces additional DoFs at the faces. This discretization, derived in Bonelle (2014, Section 8.3) and about which more details are given in Section 1.2.2, is called face-based.

- In the Stokes problem considered in Bonelle (2014) and Bonelle and Ern (2015), the variable of interest is the pressure potential (the pressure divided by the mass density) and one can consider either vertex- or cell-based schemes. Moreover, the so-called curl formulation of the Stokes problem (hinging on the differential identity $-\underline{\Delta} = \underline{\nabla} \times (\underline{\nabla}\times) - \underline{\nabla}(\underline{\nabla}\cdot)$) is considered, so that it is convenient to think of the velocity as a circulation or a flux. Thus, in the vertex-based (respectively cell-based) discretization, the velocity is considered as a circulation (resp. flux) located along primal edges (resp. across primal faces). It is interesting to observe that in the cell-based discretization, a three-field formulation of the Stokes problem is obtained. The reason is that the curl of the velocity has to be considered as an additional independent unknown, so that the final linear system has a double saddle-point structure.

- In the scalar advection-reaction problem (Cantin, 2016; Cantin and Ern, 2016; Cantin *et al.*, 2016), the main variable is again a potential, leading to vertex-based discretizations. In the vector-valued advection-reaction problem, the main variable is a circulation which is treated with edge-based schemes (Cantin and Ern, 2017).

The DoFs considered in the above-mentioned discretizations are shown in Fig. 1.2. Among the applications currently addressed by means of a CDO scheme in *Code_Saturne*, groundwater flows and thermal problems (which, loosely speaking, can be thought of as scalar transport equations) have been treated with a vertex-based discretization, whereas in the context of electromagnetism, the edge-based discretization is considered. For fluid dynamics applications, the face-based version has been retained and constitutes one of the major contributions of the Thesis.

1.2.2 The face-based CDO discretization

We now give some more details on the CDO face-based discretization (CDO-Fb for short), since this is the discretization considered in this Thesis. Indeed, CDO-Fb naturally allows us to address the Navier–Stokes problem by looking at the velocity as a vector-valued potential defined at faces and cells (in contrast with the curl form of the Stokes problem of Bonelle (2014) and Bonelle and Ern (2015)). Here we limit ourselves to recalling the main features of the CDO-Fb method, and we give more details in Section 2.2. The reader is referred to Bonelle (2014, Section 8.3) for the derivation and analysis of the scheme, and for error results concerning scalar diffusion problems.

The CDO-Fb discretization stems from the cell-based discretization considered for scalar diffusion problems to which a classical hybridization procedure is applied as described for instance in Boffi *et al.* (2013). In addition to the cell-based potential DoFs, the CDO-Fb discretization introduces a second set of potential DoFs attached to the faces. An example of the resulting discretization is shown in Fig. 1.4. It is important to remark that these

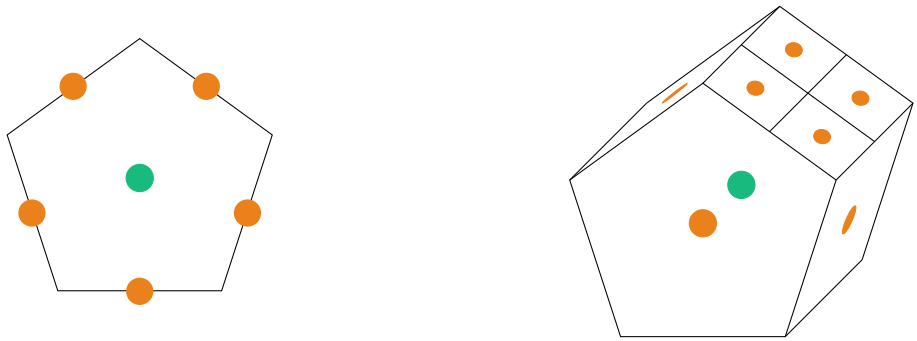


Figure 1.4 – Discrete setting for a scalar problem with a face-based CDO discretization in 2D (left) and 3D (right). Face- and cell-based quantities are shown, respectively, in orange and green. Only visible face-based DoFs are shown.

face-based DoFs are single-valued on internal faces: their value is the same irrespective of the two adjacent cells. The hybridization allows one to avoid the saddle-point structure of the final system resulting from the mixed form of the CDO cell-based discretization. Indeed, owing to the static condensation (or Schur complement) technique, the cell-based DoFs are eliminated from the linear system, thus significantly reducing its size: for a scalar elliptic problem, the final size is equal to the number of (primal) faces (instead of faces and cells). The cell-based DoFs are recovered afterwards, as a postprocessing step. Further details are given later in Section 3.1.3. Incidentally, we notice that the CDO-Fb scheme does not provide an H^1-conforming potential reconstruction (but only a reconstruction with zero mean-value jumps across internal faces). Thus, the CDO-Fb scheme is nonconforming.

A variational formulation is then built. As usual in a hybrid method, the part of the discrete problem resulting from the test functions associated with the cell-based DoFs expresses the conservation principle of the PDE at the cell level, whereas the part resulting from the test functions associated with the faces expresses the equilibrium of the fluxes from the two adjacent cells at the given face.

A key tool of the CDO-Fb method is a stabilized discrete gradient reconstruction which takes advantage of the pyramidal subdivision of the cells. Consider, for instance, a face as basis of the pyramid and the cell barycenter as the apex as shown in Fig. 1.3. The key idea is to approximate the gradient of the variable at hand as piecewise constant on this pyramidal subdivision. More precisely, a so-called consistent gradient is evaluated in each cell by combining for all of its faces the difference of the face- and cell-based DoFs normalized by an appropriate length scale and multiplied by the normal to the face. This allows one to define a gradient reconstruction which is exact for affine functions. However, the resulting gradient is not stable meaning that if the reconstructed gradient is zero, this does not imply that all the underlying DoFs are constant. The gradient reconstruction operator is then enriched by a piecewise constant stabilization term in each subpyramid which is derived from a Taylor expansion and which does not affect the exactness for affine functions. Further details about this gradient reconstruction are given in Section 2.3, and more generally in Bonelle *et al.* (2015).

CDO-Fb schemes share some features with other polyhedral hybrid methods developed in the last decade. In particular, it is shown in Bonelle (2014, Prop. 8.38) that, for a scalar elliptic problem, the CDO-Fb scheme is equivalent (up to mesh regularity assumptions and to a user-defined stabilization parameter) to the Hybrid Finite Volume scheme (Eymard *et al.*, 2010). The Hybrid High-Order (HHO) schemes introduced in Di Pietro *et al.* (2014) and Di Pietro and Ern (2015), are very close to CDO-Fb schemes as well when considering the lowest-order case $k := 0$. Taking again a scalar elliptic problem as example, both

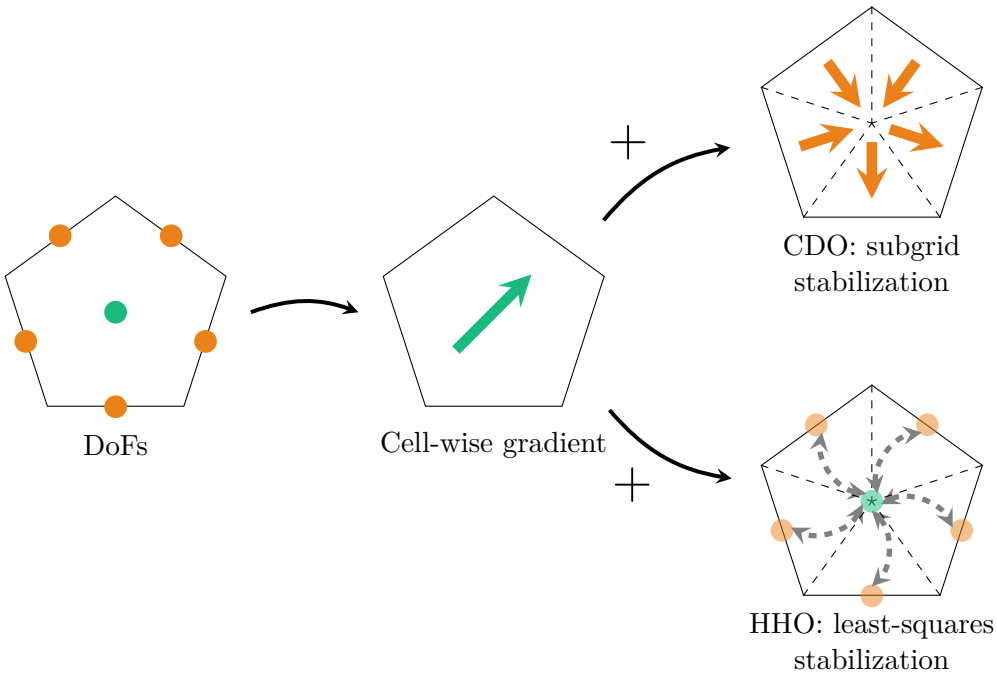


Figure 1.5 – Comparison between gradient reconstructions of a scalar function in 2D for CDO and HHO($k = 0$). Face- and cell-based quantities are shown, respectively, in orange and green. The two reconstructions share a common consistent (exact for affine functions) gradient. However, in CDO the gradient is enriched with a piecewise constant stabilization, constant on a subdivision of the cell. In HHO($k = 0$), the gradient is constant on the whole cell and it is sought among the gradients of affine functions, and a penalty is added to the bilinear formulation.

methods consider one DoF per face and per cell,[2] and both methods hinge on a gradient reconstruction operator. The difference is that, on the one hand, the CDO-Fb scheme produces a piecewise constant gradient on a subdvision of the cell, whereas, on the other hand, the gradient reconstructed in HHO is constant on the cell (it coincides with the above-mentioned consistent gradient) and a stabilization is added as a least-squares penalty on the difference between the cell- and the face-based values. The general idea is outlined in Fig. 1.5. In this Thesis, we will often exploit the similarities between HHO and CDO-Fb to benefit from the analysis results for HHO schemes. A further discussion on other schemes similar to CDO-Fb is postponed to Section 1.3.3 with a focus on CFD applications.

1.3 Numerical methods for the Navier–Stokes equations

The main subject of this Thesis is the unsteady Navier–Stokes equations (NSE) for incompressible Newtonian fluids. In this section, popular strategies used to address numerically the NSE are presented. The first part deals with spatial discretizations. In particular, we discuss schemes close to the CDO-Fb discretization. The second part discusses some time-marching techniques. For the sake of completeness, we also provide a short overview on linear and nonlinear solvers for the NSE.

[2]Viewed for HHO($k = 0$) as lowest-order polynomials $p \in \mathbb{P}^0$, where, for $k \geq 0$, $\mathbb{P}^k$ is the vector space of polynomials of degree k at most on faces and cells

1.3.1 Model problem

Let $\underline{u}$ and p denote respectively the velocity and the pressure. The density of the fluid is supposed to be uniform and equal to 1: it will not appear in our discussion. The viscosity is denoted by $\nu > 0$. Take $\Omega \times [0, T]$, with $\Omega \subset \mathbb{R}^d$, $d = 2, 3$, and $T > 0$. Then the Dirichlet initial-boundary value problem for the incompressible NSE, posed on the space-time cylinder $\Omega \times (0, T)$, reads: Find $\underline{u} : \Omega \times (0, T) \to \mathbb{R}^d$ and $p : \Omega \times (0, T) \to \mathbb{R}$ such that

$$\begin{cases} \dfrac{\partial \underline{u}}{\partial t} + (\underline{u} \cdot \underline{\nabla})\underline{u} - \nu \underline{\Delta}\, \underline{u} + \underline{\nabla}\, p = \underline{f} & \text{in } \Omega \times (0, T)\,, & (1.1a) \\ \underline{\nabla} \cdot \underline{u} = 0 & \text{in } \Omega \times (0, T)\,, & (1.1b) \\ \underline{u} = \underline{u}_\partial & \text{on } \partial\Omega \times (0, T)\,, & (1.1c) \\ \underline{u}_{|t=0} = \underline{u}_0 & \text{in } \Omega\,, & (1.1d) \end{cases}$$

with source term $\underline{f} : \Omega \times (0, T) \to \mathbb{R}^d$, Dirichlet boundary datum $\underline{u}_\partial : \partial\Omega \times [0, T] \to \mathbb{R}^d$, and initial datum $\underline{u}_0 : \Omega \to \mathbb{R}^d$. Other Boundary Conditions (BCs) can be considered as well (see, e.g., Becker *et al.* (2015)). Equation (1.1a) expresses the momentum balance, and (1.1b) the mass balance. The steady version of (1.1) will be treated as well (see Chapter 3): in this case, the time derivative $\frac{\partial \underline{u}}{\partial t}$ is dropped. One of the key features of the NSE is the convection term $(\underline{u} \cdot \underline{\nabla})\underline{u}$ which is the source of the nonlinearity in the problem.

Sometimes, in order to focus on the incompressibility constraint, we will drop the convection term in (1.1a). Then we recover the well-known unsteady Stokes problem: Find $\underline{u} : \Omega \times (0, T) \to \mathbb{R}^d$ and $p : \Omega \times (0, T) \to \mathbb{R}$ such that

$$\begin{cases} \dfrac{\partial \underline{u}}{\partial t} - \nu \underline{\Delta}\, \underline{u} + \underline{\nabla}\, p = \underline{f} & \text{in } \Omega \times (0, T)\,, & (1.2a) \\ \underline{\nabla} \cdot \underline{u} = 0 & \text{in } \Omega \times (0, T)\,, & (1.2b) \\ \underline{u} = \underline{u}_\partial & \text{on } \partial\Omega \times (0, T)\,, & (1.2c) \\ \underline{u}_{|t=0} = \underline{u}_0 & \text{in } \Omega\,. & (1.2d) \end{cases}$$

Here, we kept the time dependency, but the steady version will be considered as well.

Remark 1.1 - Alternative forms of (1.1). Owing to classical differential identities, alternative reformulations of the momentum equation (1.1a) can be found in the literature. We have already mentioned in Section 1.2.1 that a curl operator can be recovered from the vector Laplacian using $-\underline{\Delta} = \underline{\nabla} \times (\underline{\nabla}\times) - \underline{\nabla}\,(\underline{\nabla}\cdot)$ (the second term is zero if one further considers the incompressibility constraint). On the other hand, one may address the convection term $(\underline{u}\cdot\underline{\nabla})\underline{u}$ as it was done in Charnyi *et al.* (2017), where different formulations are compared. Firstly, let us recall that $(\underline{u} \cdot \underline{\nabla})\underline{u} = (\underline{\underline{\nabla}}\, \underline{u})\, \underline{u}$. A first possibility is to consider the conservative term $\underline{\nabla} \cdot (\underline{u} \otimes \underline{u})$, and observing that $\underline{\nabla} \cdot (\underline{u} \otimes \underline{u}) = (\underline{\underline{\nabla}}\, \underline{u})\underline{u} + (\underline{\nabla} \cdot \underline{u})\underline{u}$. A second possibility is the so-called skew-symmetric form which writes the convection term as $(\underline{\underline{\nabla}}\, \underline{u})\underline{u} + \frac{1}{2}(\underline{\nabla} \cdot \underline{u})\underline{u}$. The equivalence with the classical form is evident after considering the incompressibility constraint. The name comes from the fact that this latter writing leads to a skew-symmetric trilinear form. Finally, another popular writing is the so-called rotational form which hinges on the differential identity $(\underline{\underline{\nabla}}\, \underline{u})\underline{u} = \underline{\nabla}\, \left(\frac{|\underline{u}|^2}{2}\right) - \underline{u} \times (\underline{\nabla} \times \underline{u})$ (the first term on the right-hand side is usually integrated to the pressure to form the so-called Bernoulli pressure). ◇

Remark 1.2 - Oseen problem. We can keep the convection term but consider a known convection field. A reaction term may also be added. This leads to the so-called Oseen

problem, often treated in its steady form: Find $\underline{u}:\ \Omega \to \mathbb{R}^d$ and $p:\ \Omega \to \mathbb{R}$ such that

$$\begin{cases} -\nu \underline{\Delta}\, \underline{u} + (\underline{w} \cdot \underline{\nabla})\underline{u} + \mu \underline{u} + \underline{\nabla}\, p = \underline{f} & \text{in } \Omega\,, \quad (1.3a) \\ \underline{\nabla} \cdot \underline{u} = 0 & \text{in } \Omega\,, \quad (1.3b) \\ \underline{u} = \underline{u}_\partial & \text{on } \partial\Omega\,, \quad (1.3c) \end{cases}$$

with $\mu \geq 0$ and $\underline{w}\colon \Omega \to \mathbb{R}^d$ (possibly such that $\underline{\nabla} \cdot \underline{w} = 0$). ◇

Let us start by giving a minimal functional setting which will be useful in the discussion that will follow. For the current goal, a steady Stokes problem suffices:

$$\begin{cases} -\nu \underline{\Delta}\, \underline{u} + \underline{\nabla}\, p = \underline{f} & \text{in } \Omega\,, \quad (1.4a) \\ \underline{\nabla} \cdot \underline{u} = 0 & \text{in } \Omega\,, \quad (1.4b) \\ \underline{u} = \underline{0} & \text{on } \partial\Omega\,, \quad (1.4c) \end{cases}$$

where we consider homogeneous Dirichlet BCs for the sake of simplicity. Usually, one seeks the solution such that $\underline{u} \in \underline{H}^1_0(\Omega) := \{\underline{v} \in \underline{H}^1(\Omega) \,|\, \underline{v}_{|\partial\Omega} = \underline{0}\}$ and $p \in L^2_*(\Omega) := \{q \in L^2(\Omega) \,|\, \int_\Omega q = 0\}$, where the underline notation indicates a vector-valued field or quantity or a linear space composed of such objects, for instance, $\underline{H}^1(\Omega) := [H^1(\Omega)]^d$. Finally, the body force is supposed to belong to $\underline{L}^2(\Omega)$. Notice that the BC (1.4c) is directly taken into account in the velocity space. To derive the variational formulation of (1.4a), one multiplies the equation by a test function $\underline{v} \in \underline{H}^1_0(\Omega)$, and integrates by parts over Ω. The mass balance (1.4b) receives a similar treatment after having chosen test functions in $L^2_*(\Omega)$, but the integration by parts is not needed. One thus obtains the following problem: Find $(\underline{u}, p) \in \underline{H}^1_0(\Omega) \times L^2_*(\Omega)$ such that

$$\begin{cases} \nu \displaystyle\int_\Omega \underline{\underline{\nabla}}\, \underline{u} \colon \underline{\underline{\nabla}}\, \underline{v} - \int_\Omega p \underline{\nabla} \cdot \underline{v} = \int_\Omega \underline{f} \cdot \underline{v} & \forall \underline{v} \in \underline{H}^1_0(\Omega)\,, \quad (1.5a) \\ \displaystyle\int_\Omega q \underline{\nabla} \cdot \underline{u} = 0 & \forall q \in L^2_*(\Omega)\,. \quad (1.5b) \end{cases}$$

Defining the following bilinear and linear forms

$$\begin{aligned} a(\cdot,\cdot) &:\ \underline{H}^1_0(\Omega) \times \underline{H}^1_0(\Omega) \to \mathbb{R} & a(\underline{u}, \underline{v}) &:= \int_\Omega \underline{\underline{\nabla}}\, \underline{u} \colon \underline{\underline{\nabla}}\, \underline{v}\,, \\ b(\cdot,\cdot) &:\ \underline{H}^1_0(\Omega) \times L^2_*(\Omega) \to \mathbb{R} & b(\underline{v}, q) &:= -\int_\Omega q \underline{\nabla} \cdot \underline{v}\,, \\ l(\cdot) &:\ \underline{H}^1_0(\Omega) \to \mathbb{R} & l(\underline{v}) &:= \int_\Omega \underline{f} \cdot \underline{v}\,, \end{aligned} \tag{1.6}$$

the variational formulation (1.5) can be recast as follows: Find $(\underline{u}, p) \in \underline{H}^1_0(\Omega) \times L^2_*(\Omega)$ such that

$$\begin{cases} \nu a(\underline{u}, \underline{v}) + b(\underline{v}, p) = l(\underline{v}) & \forall \underline{v} \in \underline{H}^1_0(\Omega)\,, \quad (1.7a) \\ b(\underline{u}, q) = 0 & \forall q \in L^2_*(\Omega)\,. \quad (1.7b) \end{cases}$$

Notice that we have changed the sign in (1.5b) with respect to (1.4c).

A key feature of (1.5) is the so-called inf-sup condition which ensures the well-posedness of the problem. It is often known under the name Ladyzhenskaya–Babuška–Brezzi (or LBB) condition (Babuška, 1973; Brezzi, 1974). We quote the following results from Ern and Guermond (2004).

Proposition 1.3 - Inf-sup condition. *Problem* (1.5) *is well-posed if and only if*

$$\exists \beta > 0 \text{ such that } \inf_{q \in L^2_*(\Omega)} \sup_{\underline{v} \in \underline{H}^1_0(\Omega)} \frac{b(\underline{v}, q)}{\|q\|_{L^2_*(\Omega)} \|\underline{v}\|_{\underline{H}^1_0(\Omega)}} \geq \beta\,. \tag{1.8}$$

Theorem 1.4 - de Rham. *Assume that Ω is a Lipschitz domain. Then the inf-sup condition* (1.8) *holds true.*

1.3.2 Review of classical spatial discretizations

In this section, we give a short overview on popular schemes used when dealing with the Stokes problem.

Finite Element Methods (FEM) form a broad class of spatial discretization methods commonly used when addressing a wide range of PDEs including the Stokes and NSE. We give here a brief and non-exhaustive overview on the most common discretizations using FEM for the Stokes and NSE. The reader can find a more in-depth analysis of FEM, for instance, in Girault and Raviart (1986), Boffi *et al.* (2013), and Ern and Guermond (2004). A simple way to discretize (1.7) is to use continuous finite elements for both the velocity and the pressure. Let $\underline{V}_\mathrm{h} \subset \underline{H}^1_0(\Omega)$ and $P_\mathrm{h} \subset L^2_*(\Omega)$ be two finite-dimensional functional spaces for the velocity and the pressure, respectively, equipped with the norms of $\underline{H}^1_0(\Omega)$ and $L^2_*(\Omega)$. Then, the discrete version of (1.7) rewrites: Find $(\underline{u}_\mathrm{h}, p_\mathrm{h}) \in \underline{V}_\mathrm{h} \times P_\mathrm{h}$ such that

$$\begin{cases} \nu a(\underline{u}_\mathrm{h}, \underline{v}_\mathrm{h}) + b(\underline{v}_\mathrm{h}, p_\mathrm{h}) = l(\underline{v}_\mathrm{h}) & \forall \underline{v}_\mathrm{h} \in \underline{V}_\mathrm{h}\,, \\ b(\underline{u}_\mathrm{h}, q_\mathrm{h}) = 0 & \forall q_\mathrm{h} \in P_\mathrm{h}\,. \end{cases} \tag{1.9}$$

However, not all the pairs $\underline{V}_\mathrm{h}/P_\mathrm{h}$ lead to a stable discretization. The reason lies in the fact that they do not satisfy the discrete version of the inf-sup condition (1.8). Indeed, since $\underline{V}_\mathrm{h} \subset \underline{H}^1_0(\Omega)$, (1.9) is well-posed if and only if there exists $\beta_* > 0$ independent of the mesh size such that

$$\inf_{q \in P_\mathrm{h}} \sup_{\underline{v} \in \underline{V}_\mathrm{h}} \frac{b(\underline{v}_\mathrm{h}, q_\mathrm{h})}{\|q_\mathrm{h}\|_{L^2_*(\Omega)} \|\underline{v}_\mathrm{h}\|_{\underline{H}^1_0(\Omega)}} \geq \beta_*\,. \tag{1.10}$$

Note that, in general, $\beta_* \leq \beta$. Lacking property (1.10) means that there exist nonzero pressure fields q_h which satisfy $b(\underline{v}_\mathrm{h}, q_\mathrm{h}) = 0$ for all $\underline{v}_\mathrm{h} \in \underline{V}_\mathrm{h}$. Such fields are often called *spurious modes*, see Ern and Guermond (2004, Section 4.2.3) or Boffi *et al.* (2013, Section VI.5) for further details.

In the so-called *equal-order interpolation methods*, the same polynomial approximation is used both for the velocity and the pressure. Unfortunately the choices $\mathbb{P}^k/\mathbb{P}^k$ (the notation is such that the first space refers to the velocity, as in $\underline{V}_\mathrm{h}/P_\mathrm{h}$) and $\mathbb{Q}^k/\mathbb{Q}^k$ (where $\mathbb{Q}^k$ collects the polynomials of degree up to k in each component) fail to satisfy (1.10). Reducing the order of the pressure space usually enables one to recover (1.10): this is the case for the well-known Taylor–Hood mixed finite element (Taylor and Hood, 1973), also denoted by $\mathbb{P}^2/\mathbb{P}^1$, and which is one of the most popular techniques for the Stokes problem. Although initially devised only on simplexes, it has been extended to quadrangular meshes using $\mathbb{Q}^2/\mathbb{Q}^1$. Its higher-order generalizations to $\mathbb{P}^k/\mathbb{P}^{k-1}$ and $\mathbb{Q}^k/\mathbb{Q}^{k-1}$ with $k \geq 2$ also satisfy the LBB condition. Another strategy consists in enriching the velocity space by including additional so-called bubble functions, as initially proposed in Crouzeix and Raviart (1973). Choosing $k = 1$, for instance, one possibility is to consider the well-known *mini* element (or $\mathbb{P}^1$-bubble/$\mathbb{P}^1$) (Arnold *et al.*, 1984).

One may choose to relax the global regularity properties (i.e. continuity) for the pressure approximation space only, or for both velocity and pressure approximation spaces. Of the latter kind is the popular (nonconforming) Crouzeix–Raviart mixed finite element (CR) (Crouzeix and Raviart, 1973). Since the velocity lacks continuity at the mesh interfaces (only the mean-value of the velocity is continuous), one should slightly modify the bilinear forms in (1.6). In particular, one sets

$$a_\mathrm{h}(\underline{u}_\mathrm{h}, \underline{v}_\mathrm{h}) := \sum_{\mathrm{c} \in \mathrm{C}} \int_\mathrm{c} \underline{\underline{\nabla}}\, \underline{u}_\mathrm{h} : \underline{\underline{\nabla}}\, \underline{v}_\mathrm{h}\,, \qquad b_\mathrm{h}(\underline{v}_\mathrm{h}, q_\mathrm{h}) := -\sum_{\mathrm{c} \in \mathrm{C}} \int_\mathrm{c} q_\mathrm{h} \underline{\nabla} \cdot \underline{v}_\mathrm{h}\,, \tag{1.11}$$

where C is the set of the cells of the mesh and a generic cell is denoted by c. The CR mixed finite element has been initially developed for simplices, but an extension to quadrangular meshes has been proposed in Rannacher and Turek (1992) and Turek (1999). Insights about the generalization to polyhedral meshes are given in Section 1.3.3.

The inf-sup condition may be traded against an additional stabilization to be considered in the incompressibility constraint. This is for instance the case with the equal-order discontinuous Galerkin (dG) methods where the discrete velocity does not satisfy global continuity properties, i.e. it is not H^1-conforming. Then, considering piecewise polynomial spaces of the same order for the velocity and the pressure is possible if a least-squares penalty on the pressure jumps is added to the discrete problem. Introduced in the early seventies (Reed and Hill, 1973; Lesaint and Raviart, 1974) for transport problems, dG methods gained popularity in diffusion-related PDEs when interior penalty techniques were developed, see for instance Arnold (1982). Another important step in the development of dG methods is the reformulation via numerical fluxes, applied to the NSE in Bassi and Rebay (1997), see also the unified treatment of elliptic PDEs in Arnold *et al.* (2001). A unified approach to the wide class of Friedrichs' systems is presented in Ern and Guermond (2006a,b, 2008), whereas a general presentation of dG methods for several problems is given in Di Pietro and Ern (2011), see in particular Ch. 6 therein for a discussion on incompressible flows.

We close this review by considering the Finite Volume (FV) schemes (see, for instance, Eymard *et al.* (2000) for a detailed review), which hinge on a different approach than the one used in FEM. As a matter of fact, in FV schemes, the PDE is usually cast into a conservative form involving the divergence operator. The PDE is then integrated over so-called *control volumes* (polygonal shapes are admissible for many FV methods) which pave the geometrical domain. The Gauss theorem is then invoked, making normal fluxes to the interfaces of the control volumes appear. How to compute these numerical fluxes is the central step in the design of a FV scheme. In classical FV schemes, one associates each control volume with a DoF, so that the underlying discretization can be viewed as piecewise constant. For flow problems, one can choose to define both velocity and pressure on the same control volumes, and this leads to so-called *colocated* schemes. Due to this ease of implementation, FV methods are popular in industrial codes, and indeed *Code_Saturne* uses this discretization. The inf-sup condition is not satisfied, so that a stabilization of the pressure is necessary in order to avoid spurious pressure modes, see Eymard *et al.* (2006) for an analysis of one of such schemes for the Stokes problem, extended in Eymard *et al.* (2007) to the NSE (with a Crank–Nicholson time scheme). Associating velocity and pressure with different control volumes (for instance, on a typical 3D mesh, velocity with faces and pressure with cells) leads to so-called *staggered* schemes. These schemes have been widely employed in the engineering literature, see Patankar (1980, Ch. 6) for an introduction.

The well-known *Marker and Cell* scheme (MAC), introduced in Harlow and Welch (1965) for Cartesian meshes, hinges on staggered grids. The pressure is defined on the cells, whereas the velocity at the edges (in 2D), and only the normal component is retained (so that on a edge parallel to the x-axis, only the y-component of the velocity is used, and vice versa). The MAC scheme enjoyed a great popularity and was first analyzed in Nicolaides (1992) and Nicolaides and Wu (1996); it was generalized to Delaunay meshes by means of the covolume technique in Nicolaides (1989). In the context of the Stokes problem, the inf-sup condition for the classical Cartesian scheme has been proved in Shin and Strikwerda (1997) and extended to non-uniform Cartesian meshes in Blanc (1999). Recently, a generalization of the MAC scheme to grids with hanging nodes has been proposed and analyzed to the context of NSE, see Chénier *et al.* (2015), while Gallouët *et al.* (2016) extends the analysis on non-uniform Cartesian meshes with a more general functional setting (with respect to the work of Nicolaides).

1.3.3 Review of face-based spatial discretizations

We focus here on discretization methods which share some features with CDO-Fb schemes and that have been deployed for the (Navier–)Stokes problem as well. With face-based we refer to spatial discretizations which, loosely speaking, involve DoFs defined at the faces. Hence, with an abuse of notation, we will consider both so-called hybrid methods and schemes developed for the mixed formulation of the problem at hand.

Lowest-order methods

The Crouzeix–Raviart (Crouzeix and Raviart, 1973) mixed finite element has already been mentioned. Among the most well-known mixed FEM for diffusion PDEs stands the Raviart–Thomas (Raviart and Thomas, 1977) mixed finite element. Raviart–Thomas finite elements can be used to approximate the velocity in fluid dynamics applications but the velocity is not H^1-conforming. In the lowest-order case, the Raviart–Thomas element uses as Degrees of Freedom (DoFs) the mean-value of the normal component on the face.[3]

The Mimetic Finite Difference (MFD) methods (see Beirão da Veiga *et al.* (2014b) for a thorough review on elliptic problems and Beirão da Veiga *et al.* (2009b) for the application to the Stokes problem) have been among the first structure-preserving methods to address the Stokes equations on polyhedral meshes. MFD is a low-order discretization stemming from the Support Operator method (Shashkov and Steinberg, 1995; Hyman and Shashkov, 1997), initially devised for simplicial and quadrangular meshes (Hyman *et al.*, 2002), and later extended to general meshes (Brezzi *et al.*, 2005a,b; Kuznetsov *et al.*, 2004). Although nodal discretizations are possible (see Brezzi *et al.* (2009)), the first polyhedral MFD method considered face-based DoFs, and indeed such MFD schemes are related to the lowest-order Raviart–Thomas FEM when applied to simplicial meshes. A link between MFD methods and (cell-based) CDO schemes has been established in Bonelle (2014, Section 8.2.3) and Bonelle and Ern (2014).

Several low-order schemes have been derived in the Finite Volume (FV) literature. A class of methods, called Hybrid Mixed Mimetic family (HMM) was identified in Droniou *et al.* (2010) (see, for instance, Cheng and Droniou (2019) for an application to Darcy flows). Two previously developed methods belong to this class: the Mixed Finite Volume (MFV) schemes (see Droniou and Eymard (2006)) and the Hybrid Finite Volume (HFV) (see Eymard *et al.* (2010)) schemes. It is worth mentioning that the alternative formulation of HFV schemes called SUSHI has been proved to be equivalent to CDO-Fb for diffusion problems, up to the scaling of the stabilization part of the gradient reconstruction, see Bonelle (2014, Prop. 8.38). MFV has been successfully applied to the NSE (Droniou and Eymard, 2009). Moving from the HFV, a Generalized CR (lowest-order) scheme for polyhedral meshes has been designed and applied to diffusion and Stokes problems, see Di Pietro and Lemaire (2015). Gradient schemes (also known as Gradient Discretization Methods) provides a framework which encompasses HMM, MFD, the MAC scheme and some high-order methods discussed below, devised in Droniou *et al.* (2013) (see also Droniou *et al.* (2018) for an extensive analysis). These schemes are applied, for instance, to nonlinear problems such as the NSE in Feron (2016), Droniou *et al.* (2015), and Eymard *et al.* (2018).

The Discrete Duality Finite Volume (DDFV) schemes stem from the work of Hermeline (1998, 2000) and have been introduced in Domelevo and Omnes (2005) (see also Andreianov *et al.* (2012, 2013) for a recent overview). DDFV schemes are of lowest-order and can handle polyhedral meshes. They take advantage of a dual mesh like CDO, but also of a third mesh, called *diamond* mesh, obtained in 2D by considering the two subtriangles related to the same internal face on the two adjacent cell (extend the highlighted triangle in the left panel of

[3]This is in the same spirit as CDO when one is dealing with fluxes

Fig. 1.3). In the Stokes problems, the discretization is staggered with velocity DoFs defined at the centers of primal and dual cells and pressure DoFs at the centers of the diamond cells (see Delcourte (2007)). However, a stabilization is needed in order to ensure the well-posedness of the problem on any type of mesh (see Krell (2011a), Krell and Manzini (2012), and Boyer *et al.* (2015) and also Krell (2011b) and Goudon *et al.* (2019) for the NSE).

High-order methods

Taking inspiration from the mixed FEM, the hybridization of discontinuous Galerkin methods led to the design of the so-called Hybridizable Discontinuous Galerkin methods (HDG) (Cockburn *et al.*, 2009b). In HDG methods, the DoFs are polynomials defined both on cells and faces and the dual variable is part of the discretization as well. A static condensation technique allows one to reduce the global size of the problem to a global transmission problem involving only the face unknowns. HDG methods were extended to convection-diffusion-reaction (Cockburn *et al.*, 2009a; Nguyen *et al.*, 2009), Stokes (Cockburn *et al.*, 2010) and Navier–Stokes (Nguyen *et al.*, 2011) problems. In these cases, one usually chooses to hybridize the velocity only, whereas the pressure stays cell-based.

Combining the MFD framework with tools typical of classical FEM (for instance, considering variational formulations instead of a fully discrete setting) led to the development of several high-order methods. As a matter of fact, the MFD framework itself has been extended to higher orders, see Gyrya and Lipnikov (2008) and Beirão da Veiga *et al.* (2009a). A salient example of such an approach are the Virtual Element Methods (VEM). Initially devised for 2D diffusion-like problems (Beirão da Veiga *et al.*, 2013a,b), the framework has been extended to 3D (Beirão da Veiga *et al.*, 2014a) and many other applications, for instance, the Stokes (Antonietti *et al.*, 2014; Beirão da Veiga *et al.*, 2017) and NSE (Beirão da Veiga *et al.*, 2018), see also Beirão da Veiga *et al.* (2016a) for an $H(div)$-conforming version. The original setting used spaces which ensured the conformity of the method and whose basis functions are non-polynomial. However, the analytical form of the basis functions does not need to be known (whence the name *virtual*) and this lack of knowledge is traded against a stabilization. The usual discretization of the VEM is nodal-based with DoFs attached to edges, faces, and cells as well in high-order discretizations. A face-based VEM has also been proposed, leading to the so-called *nonconforming* VEM ($\mathrm{VEM}_{\mathrm{nc}}$), see, for instance Ayuso de Dios *et al.* (2016) for elliptic problems and Cangiani *et al.* (2016) for the Stokes equations. Similar to the nonconforming VEM are the High-Order Mimetic methods (Lipnikov and Manzini, 2014) which use a hybrid (face and cell) discretization. A similar approach owing this time to mixed FEM is the mixed VEM, see Brezzi *et al.* (2014) and Beirão da Veiga *et al.* (2016b) for an introduction and Gatica *et al.* (2018) for the application to NSE.

The extension of the Raviart–Thomas element to higher orders and general meshes led to the Mixed High-Order methods (Di Pietro and Ern, 2017), later developed for Stokes (Aghili *et al.*, 2015) and Oseen (Aghili and Di Pietro, 2018) problems as well. Taking inspiration this time from the CR element, Di Pietro *et al.* (2014) and Di Pietro and Ern (2015) devised the Hybrid High-Order (HHO) schemes. As the name suggests, the HHO method consists in a hybrid (face and cell) nonconforming discretization which ensures arbitrary order of convergence on general meshes. Based on a reconstruction of the gradient and a stabilization operator, the HHO framework shares many features with HDG, from which, as a matter of fact, it differs essentially in the devising of the stabilization and the adoption of a primal formulation right from the start (no mixed formulation is considered and the variable of interest is directly hybridized), see Cockburn *et al.* (2016) for further details. In HHO methods, the DoFs are polynomials attached to the cells and the faces: letting k (respectively l) be the degree of the face (resp. cell) polynomials, then the cell degree may be chosen as $l \in \{k-1, k, k+1\}$, with $l \geq 0$. For a diffusion problem, all the three

Table 1.1 – Summary of hybrid discretizations available for the Stokes and NSE. The rows give insights on the discrete setting and the techniques used in the discretization.

	CR	CDO	HMM	$\mathrm{VEM}_{\mathrm{nc}}$	HDG	HHO
Order	Lowest	Lowest	Lowest	High	High	High
Mesh	Simplex/Tensor	General	General	General	General	General
Stability	Stable	Sub-grid	Sub-grid	Penalty	Penalty	Penalty

choices ensure a $(k+1)$-th order of convergence in the energy norm of the error for smooth solutions (Cockburn *et al.*, 2016). Initially devised for diffusion problems (Di Pietro *et al.*, 2014) and linear elasticity (Di Pietro and Ern, 2015), HHO methods have been extended to numerous applications, such as advection-reaction (Di Pietro *et al.*, 2015), Stokes (Di Pietro *et al.*, 2016), and Navier–Stokes (Di Pietro and Krell, 2018; Botti *et al.*, 2019) equations. Following Di Pietro *et al.* (2016), the velocity has hybrid (face- and cell-based) DoFs, whereas the pressure is cell-based only. The velocity and pressure polynomial spaces are typically of equal order. A static condensation is available and all but one pressure DoFs per cell can be eliminated as well.

We give in Table 1.1 a one-glance overview of the salient features of a few selected (nonconforming) discretizations mentioned above and available for the Stokes and NSE. Usually, lowest-order methods such as CDO or HMM hinge on a stabilized reconstruction of the gradient and this additional stabilization is not needed on simple meshes (cf. CR FEM on simplicial meshes). A setting defining a stabilized gradient in the context of higher-order methods is proposed in Di Pietro *et al.* (2018) (see also Abbas *et al.* (2018) for an application to hyperelasticity). However, higher-order schemes often consider a bilinear formulation based on a gradient reconstruction and an additional stabilization penalizing the gap in the hybrid DoFs.

1.3.4 Linear and nonlinear solvers for the Navier–Stokes equations

The model problems for this section are the steady Stokes or NSE. We also suppose that a spatial discretization has been chosen so that the next step is the numerical resolution of the discrete problem.

The two key features of the NSE, namely, the incompressibility constraint and the convection term, lead, respectively, to the saddle-point structure and the nonlinearity. In this section, procedures commonly used to tackle saddle-point and nonlinear problems are presented. In a nutshell:

- In order to deal with saddle-point problems, Augmented-Lagrangian techniques (see Hestenes (1969) and Fortin and Glowinski (1983)) combined with an Uzawa (Arrow *et al.*, 1958) or a Golub–Kahan Bidiagonalization technique (Arioli, 2013), will be mainly considered. Direct solvers, in particular an LU decomposition, will be used as well, whenever the size of the problem makes it possible.

- In order to deal with the nonlinear convection term, Picard iterations will be performed. We chose this algorithm for its simplicity in implementation and its robustness, which, although not optimal, lead to fairly good results most of the times.

The strategies presented here are classical and they are given for the sake of completeness. The reader familiar with the subject may skip this section. Additional details specific to CDO schemes will be given in Section 3.1.3 as well.

Linear solvers for saddle-point problems

Let us write the algebraic system resulting from the discretization of Stokes or NSE as follows:

$$\begin{bmatrix} \mathbf{A} & \mathbf{B}^T \\ \mathbf{B} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{p} \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \mathbf{0} \end{bmatrix} , \tag{1.12}$$

One may recognize in (1.12) the same structure as in (1.9). The matrix $\mathbf{A}$ is sometimes defined as the velocity-velocity block, whereas the matrix $\mathbf{B}$ corresponds to the (negative) divergence operator. We have supposed here that the spatial discretization enables one to write the gradient of the pressure as the transpose of the (negative) divergence (it is indeed the case for CDO-Fb, see Section 2.4). For the Stokes problem, $\mathbf{A}$ contains the diffusion term and it is symmetric definite positive. For the NSE, $\mathbf{A}$ contains the diffusion and the convection terms, it is still positive but it is not symmetric anymore.

A thorough review on solvers for saddle-point problems is given in Turek (1999) and Benzi *et al.* (2005). Let us briefly address below some common procedures to treat a saddle-point problem like (1.12) by means of iterative solvers. The reader is referred to Saad (1996) for an in-depth general discussion on linear solvers and to Elman *et al.* (2014) for an analysis focused on flow problems.

Uzawa algorithm The central principle of the well-known Uzawa algorithm (Arrow *et al.*, 1958) is to break the saddle-point structure by considering an iterative procedure where at each step a known approximation of the pressure is used so that its gradient becomes explicit. Thus, at each step, one solves a linear system involving the velocity only and then the pressure is updated using the new velocity. The procedure is stopped once a prescribed tolerance is reached, for instance on the pressure increment or on the divergence of the velocity. The simplest version of the Uzawa algorithm reads as follows: Given $\mathbf{p}^0$, iterate on $k \geq 1$ until convergence

$$\begin{aligned} \mathbf{A}\mathbf{u}^k + \mathbf{B}^T\mathbf{p}^{k-1} &= \mathbf{f} \\ \mathbf{M}\mathbf{p}^k &= \mathbf{M}\mathbf{p}^{k-1} - \omega\mathbf{B}\mathbf{u}^k \end{aligned} \tag{1.13}$$

where $\omega > 0$ is an arbitrary parameter. $\mathbf{M}$ is the mass matrix associated with the pressure discretization. Most of the times, this matrix is diagonal and, even more, proportional to the identity matrix, $\mathbf{I}$: thus, without loss of generality, we consider $\mathbf{M} = \mathbf{I}$ and omit it. For the Uzawa algorithm to be stable, an upper bound is available on ω (Glowinski and Le Tallec, 1989; Benzi *et al.*, 2005): supposing $\mathbf{A}$ is symmetric, one needs to enforce $0 < \omega < \frac{2}{\rho_{\mathrm{M}}(\mathbf{B}\mathbf{A}^{-1}\mathbf{B}^T)}$ where $\rho_{\mathrm{M}}(\cdot)$ denotes the spectral radius of a matrix. We stress that, at each iteration, one has to invert $\mathbf{A}$, which can be achieved with standard iterative solvers.

Augmented Lagrangian The main drawback of the procedure (1.13) is that usually it takes many iterations to reach the solution, or indeed, the prescribed tolerance. In order to speed up the convergence, an Augmented Lagrangian technique (Hestenes, 1969; Glowinski and Le Tallec, 1989; Benzi *et al.*, 2005, 2011) can be deployed. Equation (1.12) then becomes

$$\begin{bmatrix} \mathbf{A} + \lambda\mathbf{B}^T\mathbf{B} & \mathbf{B}^T \\ \mathbf{B} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{p} \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \mathbf{0} \end{bmatrix} , \tag{1.14}$$

where $\lambda \geq 0$ is a user-defined parameter. Let us set $\mathbf{A}_\lambda := \mathbf{A} + \lambda\mathbf{B}^T\mathbf{B}$.

By applying the iterative procedure (1.13) to (1.14), one obtains the so-called Augmented Lagrangian-Uzawa algorithm (ALU), which reads

$$\begin{aligned} \mathbf{A}_\lambda\mathbf{u}^k + \mathbf{B}^T\mathbf{p}^{k-1} &= \mathbf{f} , \\ \mathbf{p}^k &= \mathbf{p}^{k-1} - \omega\mathbf{B}\mathbf{u}^k , \end{aligned} \tag{1.15}$$

where, in order to ensure stability, the parameter ω must satisfy $0 < \omega < \frac{2}{\rho_{\mathrm{M}}(\mathbf{B}\mathbf{A}_\lambda^{-1}\mathbf{B}^T)}$. The interval becomes larger and larger for $\lambda \to \infty$. However, the conditioning of $\mathbf{A}_\lambda$ grows linearly with λ (Fortin and Glowinski, 1983; Glowinski and Le Tallec, 1989; Benzi *et al.*, 2005), hence making the inversion of $\mathbf{A}_\lambda$ by means of linear solvers harder. It is often suggested to use $\omega = \lambda$ (see for instance Glowinski and Le Tallec (1989, Section 2.3.6)): this is the choice considered in this Thesis. The resulting problem then reads: Given $\mathbf{p}^0$, iterate on $k \geq 1$ until convergence

$$\begin{aligned} &\mathbf{A}_\lambda \mathbf{u}^k + \mathbf{B}^T \mathbf{p}^{k-1} = \mathbf{f}\,, \\ &\mathbf{p}^k = \mathbf{p}^{k-1} - \lambda \mathbf{B}\mathbf{u}^k\,. \end{aligned} \tag{1.16}$$

Notice that the continuous counterpart to the additional block $\mathbf{B}^T\mathbf{B}$ is the grad-div operator (under appropriate boundary conditions).

Golub–Kahan Bidiagonalization In Arioli (2013) and Arioli *et al.* (2018) a change of variables is proposed leading to a matrix to which a Golub–Kahan Bidiagonalization (Golub and Kahan, 1965) (GKB) procedure can be efficiently applied. The procedure can be applied to either (1.12) or (1.14). However, the GKB procedure requires $\mathbf{A}$ to be symmetric, and as such cannot be applied to NSE. The GKB is an iterative procedure which consists in factorizing matrix into a bidiagonal one. It requires an inversion of $\mathbf{A}_\lambda$ or $\mathbf{A}$ at each iteration (standard iterative solvers may be deployed to achieve this). The efficiency of the GKB procedure depends on the augmentation parameter λ (which we shall rename γ for convenience), although it is not as sensitive as in the ALU algorithm. We are sometimes going to denote this procedure by GKB(γ). Choosing the optimal stopping criterion may not be trivial (see, for instance, Arioli (2013)); in practice we are going to keep track of the divergence and of the residual at the current velocity.

Preconditioning A key element of the iterative resolution of any linear system is to find an appropriate preconditioner. This is investigated in Benzi and Olshanskii (2006), Olshanskii and Benzi (2008), and Benzi *et al.* (2011), which focus on preconditioners for the Augmented Lagrangian systems resulting from the Oseen and NSE. Once the system has been preconditioned, standard linear solvers, such as Algebraic Multi-Grid or iterative Krylov-based solvers, can be deployed efficiently (in Benzi and Olshanskii (2006), Olshanskii and Benzi (2008), and Benzi *et al.* (2011), GMRES solvers are considered). For further details, the reader is referred to Benzi *et al.* (2005, Section 10), Benzi *et al.* (2011), and the references therein.

Summary: Stokes equations Due to its favorable performance, the GKB algorithm is preferred to the ALU method. When the GKB algorithm is chosen, the standard formulation (1.12) seems to lead to a better performance than the augmented one, (1.14), see for instance the efficiency results in Section 4.4 (although, they have been observed for unsteady problems). At each iteration of the procedures, the internal linear systems are solved with, typically, a Conjugate Gradient method. We consider rather simple preconditioners: only standard Jacobi or Algebraic Multi-Grid techniques are applied to the internal (decoupled) systems. However, we will also use direct solvers (mainly the sparse ones available through MUMPS (Amestoy *et al.*, 2001) if the size of the problem allows it.

Summary: Navier–Stokes equations As mentioned above, the matrix $\mathbf{A}$ is nonsymmetric for the NSE, thus the GKB procedure cannot be used anymore. It is replaced by the ALU algorithm. Since we lack of efficient and reliable iterative solvers for nonsymmetric matrices, we consider direct solvers. In order to temper the memory consumption of

such methods, we still consider the iterative ALU algorithm, instead of solving directly the saddle-point problem (task that a direct solver should be able to perform).

Nonlinear solvers for the convection term

The Picard algorithm is a classical, iterative, fixed-point algorithm used to tackle nonlinear problems in general. It is sometimes called *successive substitutions*. The principle on which it hinges is to approximate the solution to the nonlinear problem with a sequence of solutions to linearized auxiliary problems. The iterate computed at each step is used to build the problem to be solved at the next step. Let the superscript k denote a given iteration: for any $k \geq 0$, $\underline{u}^k$ is the velocity approximation obtained at the k-th iteration of the algorithm, $\underline{u}^0$ being the initial guess. Typically, for the NSE, one chooses $(\underline{u} \cdot \underline{\nabla})\underline{u} \approx (\underline{u}^{k-1} \cdot \underline{\nabla})\underline{u}^k$. Hence, the procedure applied to the steady version of (1.1) reads: Given an initial guess $\underline{u}^0$, iterate on $k \geq 1$ until convergence: Find $(\underline{u}^k, p^k)$ such that:

$$\begin{cases} -\nu \underline{\Delta}\, \underline{u}^k + (\underline{u}^{k-1} \cdot \underline{\nabla})\underline{u}^k + \underline{\nabla}\, p^k = \underline{f}\,, \\ \underline{\nabla} \cdot \underline{u}^k = 0\,, \end{cases} \tag{1.17}$$

where for the sake of brevity, we have dropped the boundary conditions as well as the domains. Notice that, at each time step, one has to solve an Oseen-like problem: compare (1.17) with (1.3) taking $\mu = 0$. A stopping criterion should also be chosen. In this Thesis we will consider an approximation of the L^2-norm of the increment on the velocity.

Remark 1.5 - Newton method. Another common technique for solving the discrete NSE is the Newton method, which is a well-known iterative root-finding strategy involving the Jacobian matrix of the considered nonlinear functional. The Newton method usually exhibits a better rate of convergence than the Picard iterations: theoretically, the former converges quadratically and the latter linearly. A drawback of the Newton method is that its radius of convergence (actually, the set in which the initial guess can be chosen for the method to converge) might be quite small. The Picard algorithm is, in general, more robust with respect to the initialization: in fact, it is usually advised to initialize the Newton method with some Picard steps and then possibly switch to the Newton method.

A variant of the Newton method is to keep the Jacobian computed at the first iteration. Moreover, if available, a factorization of the matrix might be performed only once and then stored to be used at all the following iterations, resulting in significant computational savings at the price of extra memory storage. This technique is well-known and sometimes called the *secant method*, or modified Newton. However, the secant method is often less efficient and robust than the original Newton method. Another variant consists in updating the inverse of the Jacobian at each step by using known quantities: this leads to the so-called *quasi-Newton* methods. The reader is referred to Engelman *et al.* (1981) for a more detailed discussion on nonlinear methods for the NSE and an introduction to the quasi-Newton methods. ◇

1.3.5 Time-stepping and velocity-pressure coupling

In this section, for the sake of simplicity, we consider (i) the Stokes equations, (ii) homogeneous Dirichlet boundary conditions, (iii) first-order implicit time discretizations (Backward Euler), (iv) whereas the spatial variable is not discretized. The resulting semi-discrete prob-

lem then reads:

$$\begin{cases} \dfrac{\underline{u}^n - \underline{u}^{n-1}}{\Delta t} - \nu \underline{\Delta}\, \underline{u}^n + \underline{\nabla}\, p^n = \underline{f}^n\,, & \text{in } \Omega\,, \\ \underline{\nabla} \cdot \underline{u}^n = 0\,, & \text{in } \Omega\,, \\ \underline{u}^0 = \underline{u}_0\,, & \text{in } \Omega\,, \\ \underline{u}^n = 0\,, & \text{on } \partial\Omega\,, \end{cases} \tag{1.18}$$

where $\Delta t > 0$ is the time-step, taken constant for simplicity. Popular time-stepping techniques addressing the velocity-pressure coupling for problem (1.18) are now reviewed.

Monolithic approach

The *monolithic* approach consists in addressing directly the saddle-point problem (1.18). No further approximation is considered. The solution hence satisfies many interesting properties: it is divergence-free and verifies the correct BCs. However, solving the saddle-point problem may be demanding as mentioned above. In particular, the conditioning of the pressure Schur complement matrix associated with (1.18) degrades with the time step (see, e.g. Ern and Guermond (2004)).

Projection method

The projection method is undoubtedly one of the most used strategies to avoid having to deal with a strong velocity-pressure coupling and, consequently, a saddle-point problem. The reader is referred to Guermond *et al.* (2006) for a more detailed overview of the projection method, so that we only give here a brief introduction.

The projection method is sometimes regarded as a fractional-step method since (at least) two problems are solved per time step, leading to two approximations of the velocity. The projection method is based on the well-known Helmholtz–Hodge decomposition: the (first) velocity, solution to a relaxed momentum equation, is projected onto a convenient space by ensuring that the final result is divergence-free. This explains why the method is also known as prediction-correction. Several possibilities have been developed to perform the relaxation. In the first and simplest version, the one derived in the late sixties in Chorin (1968, 1969) and Temam (1969b), one simply drops the pressure in the momentum equation and solves sequentially the following problems:

$$\begin{cases} \dfrac{\widetilde{\underline{u}}^n - \underline{u}^{n-1}}{\Delta t} - \nu \underline{\Delta}\, \widetilde{\underline{u}}^n = \underline{f}^n & \text{in } \Omega\,, \\ \widetilde{\underline{u}}^n = 0 & \text{on } \partial\Omega\,, \end{cases} \tag{1.19a}$$

$$\begin{cases} \dfrac{\underline{u}^n - \widetilde{\underline{u}}^n}{\Delta t} + \underline{\nabla}\, p^n = 0 & \text{in } \Omega\,, \\ \underline{\nabla} \cdot \underline{u}^n = 0 & \text{in } \Omega\,, \\ \underline{u}^n \cdot \underline{n}_{\partial\Omega} = 0 & \text{on } \partial\Omega\,. \end{cases} \tag{1.19b}$$

Thus instead of a saddle-point problem, one solves a vector-valued diffusion-reaction problem, and then a scalar Poisson problem with Neumann BCs for the pressure (obtained by taking the divergence of the first equation in (1.19b) and owing to the incompressibility constraint). Notice that both approximations of the velocity, $\widetilde{\underline{u}}^n$ or $\underline{u}^n$, present a drawback: only the latter is divergence-free, whereas only the former fully satisfies the BCs.

Remark 1.6 - Continuous counterpart. It was mentioned that the method hinges on a Helmholtz–Hodge decomposition. We follow Chorin (1968) (see also Gresho (1990)). One rewrites (1.2a) as $\frac{\partial \underline{u}}{\partial t} + \underline{\nabla}\, p = \underline{F}(t, \underline{u}) := \underline{f} + \nu \underline{\Delta}\, \underline{u}$ and observes that, on the left-hand side,

$\underline{\nabla} \cdot (\frac{\partial \underline{u}}{\partial t}) = 0$ and $\underline{\nabla} \times \underline{\nabla} p = 0$. Hence, the left-hand side constitutes a Helmholtz–Hodge decomposition of the right-hand side. On this basis, one drops the pressure and seeks the approximation $\widetilde{\underline{u}}$ such that $\frac{\partial \widetilde{\underline{u}}}{\partial t} = \underline{F}(t, \underline{u})$ and then one solves $\frac{\partial \underline{u}}{\partial t} + \underline{\nabla} p = \frac{\partial \widetilde{\underline{u}}}{\partial t} = \underline{F}(t, \underline{u})$ with $\underline{\nabla} \cdot \underline{u} = 0$. Once a time discretization is applied, the two systems in (1.19) are recovered ◇

It has been shown that, under appropriate regularity assumptions, the solution of (1.19) satisfies (see Shen (1992a, Thm. 1) or Guermond *et al.* (2006, Thm. 3.1)):

$$\begin{aligned} \sum_{n=0}^{N} \Delta t \left(\|\underline{u}(t^n) - \underline{u}^n\|^2_{\underline{L}^2(\Omega)} + \|\underline{u}(t^n) - \widetilde{\underline{u}}^n\|^2_{\underline{L}^2(\Omega)} \right) \leq C \Delta t^2 \,, \\ \sum_{n=0}^{N} \Delta t \, \|p(t^n) - p^n\|^2_{L^2(\Omega)} + \sum_{n=0}^{N} \Delta t \, \|\underline{u}(t^n) - \widetilde{\underline{u}}^n\|^2_{\underline{H}^1(\Omega)} \leq C \Delta t \,, \end{aligned} \tag{1.20}$$

where $C > 0$ is a constant which depends on suitable norms of $\underline{u}$ and p and the final time T, but not on Δt.

The projection method suffers from some well-known drawbacks. First of all, it has an intrinsic error due to the velocity-pressure splitting that limits the accuracy in time; see, for instance, Shen (1992b, 1993), Actually, the stability of some high-order time discretization scheme has not been verified yet, see Guermond *et al.* (2006, Section 11.1). A weak point of the projection method is the boundary conditions. For one, the final velocity, the one obtained in (1.19b), does not satisfy the tangential Dirichlet BCs. Furthermore, a consequence of (1.19b) is an unphysical boundary condition. This is partly overcome by considering the rotational formulation, in Remark 1.7, see Guermond *et al.* (2006, Remark 3.2 and Section 3.3). Despite these drawbacks, projection methods still remain among the most popular strategies to deal with the NSE owing to their efficiency

Remark 1.7 - Alternative formulations. The formulation (1.19), and (1.19a) in particular, can be improved in several ways. We briefly address two of the most common alternatives. First, instead of neglecting the pressure, a known approximation of the pressure can be included in (1.19a) for better accuracy. For instance, it has been proposed in Goda (1979)[4] to subtract $\underline{\nabla} p^{n-1}$ on the left-hand side of (1.19a), thus making $\underline{\nabla}(p^n - p^{n-1})$ appear in (1.19b), whence the name incremental formulation. Another common formulation has been introduced in Timmermans *et al.* (1996) and is often called the rotational formulation, since it takes advantage of the differential identity $-\underline{\Delta} = \underline{\nabla} \times (\underline{\nabla} \times) - \underline{\nabla}(\underline{\nabla} \cdot)$ to improve the order of convergence in time. Less commonly, the roles of velocity and pressure can be exchanged, so that the velocity is first dropped and then corrected: this leads to the so-called velocity-correction methods, see Guermond and Shen (2003) or Guermond *et al.* (2006, Section 4). Finally, for the applications to the NSE, see, for instance, Kim and Moin (1985) and Shen (1992b). ◇

Remark 1.8 - Single-velocity version. As shown in Guermond *et al.* (2006, Section 3.5), the field $\underline{u}$ in (1.19) can be eliminated, leaving $\widetilde{\underline{u}}$ only. As a matter of fact, one recovers $\widetilde{\underline{u}}$ from the first equation in (1.19b) and use it in (1.19a). Then, writing (1.19b) as a Poisson problem in primal form as usual, one obtains

$$\begin{cases} \dfrac{\widetilde{\underline{u}}^n - \widetilde{\underline{u}}^{n-1}}{\Delta t} - \nu \underline{\Delta}\, \widetilde{\underline{u}}^n + \underline{\nabla} p^{n-1} = \underline{f}^n & \text{in } \Omega \,, \\ \widetilde{\underline{u}}^n = 0 & \text{on } \partial\Omega \,, \end{cases} \tag{1.21a}$$

$$\begin{cases} \Delta p^n = \dfrac{1}{\Delta t} \underline{\nabla} \cdot \widetilde{\underline{u}}^n & \text{in } \Omega \,, \\ \underline{\nabla} p \cdot \underline{n}_{\partial\Omega} = 0 & \text{on } \partial\Omega \,. \end{cases} \tag{1.21b}$$

[4]The projection method used in *Code_Saturne* is inspired by the one proposed in Goda (1979)

Finally, notice from (1.20) that both $\underline{u}$ and $\underline{\widetilde{u}}$ satisfy the same error estimate, hence there is no loss in considering the latter instead of the former. Recall that $\underline{\widetilde{u}}$ is not divergence-free, so that one can look at (1.21) as a problem where the incompressibility constraint has been relaxed in order to break the velocity-pressure coupling. For this reason, (1.21) shares some similarities with schemes presented below. ◇

Penalty method

The penalty method has been introduced in Temam (1968) and hinges on a perturbation of the incompressibility constraint with which one manages to decouple velocity and pressure. The resulting semi-discrete system reads

$$\begin{cases} \dfrac{\underline{u}^n - \underline{u}^{n-1}}{\Delta t} - \nu \left(\underline{\Delta}\, \underline{u}^n + \dfrac{1}{\epsilon} \underline{\nabla}\, \underline{\nabla} \cdot \underline{u}^n \right) = \underline{f}^n \,, \\ \qquad\qquad\qquad\qquad\qquad\qquad p^n = -\dfrac{\nu}{\epsilon} \underline{\nabla} \cdot \underline{u}^n \,, \end{cases} \tag{1.22}$$

where $\epsilon > 0$ is a nondimensional user-defined parameter. Notice that the pressure does not appear in the relaxed momentum equation. Actually, it can be eliminated from the system if one is not interested in it.

Remark 1.9 - Continuous counterpart. Equation (1.22) is obtained by adding at the time-continuous level a perturbation proportional to the pressure in the mass balance equation (1.4b) as follows:

$$\begin{cases} \dfrac{\partial \underline{u}}{\partial t} - \nu \underline{\Delta}\, \underline{u} + \underline{\nabla}\, p = \underline{f} \,, \\ \qquad \underline{\nabla} \cdot \underline{u} + \dfrac{\epsilon}{\nu} p = 0 \,. \end{cases} \tag{1.23}$$

It has been proved in Temam (1968) that the solution of (1.23) converges to the solution of the unperturbed problem (1.4) for $\epsilon \to 0$. ◇

It has been shown that (adapt from Thm. 5.1 and Remark 5.1 of Shen (1995) where the NSE are considered; notice also that no spatial discretization has been considered), under appropriate regularity assumptions (see hypotheses therein), one has

$$\begin{aligned} \sqrt{t^n}\, \|\underline{u}(t^n) - \underline{u}^n\|_{\underline{L}^2(\Omega)} + t^n\, \|\underline{u}(t^n) - \underline{u}^n\|_{\underline{H}^1(\Omega)} \leq & C(\Delta t + \epsilon) \,, \qquad \forall n = 1, \ldots, N \,, \\ \Delta t \sum_{n=0}^{N} (t^n)^2\, \|p(t^n) - p^n\|^2_{L^2(\Omega)} \leq & C(\Delta t^2 + \epsilon^2) \,, \end{aligned} \tag{1.24}$$

where $C > 0$ is a constant which depends on $\underline{u}$, p, and the final time T but not on Δt and $t^n := n\Delta t$.

Remark 1.10 - Choice of ϵ. The decoupling of velocity and pressure makes the additional term $-\frac{1}{\epsilon}\underline{\nabla}\, \underline{\nabla} \cdot \underline{u}$ appear in the momentum equation. Owing to what was stated in Remark 1.9, one tends to choose small values for ϵ in order for the solution of the penalized system to be as close as possible to the one of the unperturbed system (furthermore, owing to (1.24), one is led to choose $\epsilon = \mathcal{O}(\Delta t)$). However, this will make the conditioning of the related system increase, thus leading to stiffer matrices and harder inversions. Attention must be paid when choosing ϵ, so that a fair trade-off between accuracy and efficiency is found. ◇

Artificial Compressibility

The Artificial Compressibility (AC) method first appeared in the Western literature in Chorin (1967) and Temam (1969a), but it can be traced independently in the Russian scientific community, see for instance Vladimirova *et al.* (1966), Yanenko (1971), and Ladyzhenskaya (1969). AC stems from principles similar to those at the origin of the Penalty method (Section 1.3.5). As a matter of fact, both schemes consider a relaxed version of the mass balance equation. The semi-discretized in time problem reads:

$$\begin{aligned} \frac{\underline{u}^n - \underline{u}^{n-1}}{\Delta t} - \nu \left(\underline{\Delta}\, \underline{u}^n + \eta \underline{\nabla}\, \underline{\nabla} \cdot \underline{u}^n \right) &= \underline{f}^n - \underline{\nabla}\, p^{n-1} \\ p^n &= p^{n-1} - \nu\eta \underline{\nabla} \cdot \underline{u}^n \,, \end{aligned} \tag{1.25}$$

where $\eta > 0$ is a nondimensional arbitrary parameter.

Remark 1.11 - Continuous counterpart. If in the Penalty method one considers a relaxation of the mass balance equation with a perturbation proportional to the pressure, in the AC method the relaxation is proportional to the time-derivative of the pressure:

$$\begin{cases} \dfrac{\partial \underline{u}}{\partial t} - \nu \underline{\Delta}\, \underline{u} + \underline{\nabla}\, p = \underline{f} \,, \\ \dfrac{\tau}{\nu\eta} \dfrac{\partial p}{\partial t} + \underline{\nabla} \cdot \underline{u} = 0 \,, \end{cases} \tag{1.26}$$

where τ is a time scale. For instance, considering a first-order time discretization and setting $\tau := \Delta t$, one recovers (1.25). The system (1.26) highlights where the name Artificial Compressibility comes from: the incompressibility assumption is traded for a low-Mach number assumption (cf. Guermond and Minev (2015, Section 2.2)). ◇

The AC method was classically used for approximating steady problems (for instance this was the principal goal of Chorin (1967)). But it can be applied to unsteady problem as well. In this case, error estimates for (1.25) (notice that no spatial discretization has been considered), under appropriate regularity assumptions, are as follows (see Shen (1995, Prop. 5.1), see also Ern and Guermond (2020, Prop. 75.3)):

$$\begin{aligned} \left\| \underline{u}(t^N) - \underline{u}^N \right\|^2_{\underline{L}^2(\Omega)} + \sum_{n=0}^{N} \Delta t \left\| \underline{u}(t^n) - \underline{u}^n \right\|^2_{\underline{H}^1(\Omega)} &\leq C \left(\Delta t^2 + \left(\frac{1}{\eta} \right)^2 \right) \,, \\ \sum_{n=0}^{N} \Delta t \left\| p(t^n) - p^n \right\|^2_{L^2(\Omega)} &\leq C \left(\Delta t + \frac{1}{\eta} \right) \,, \end{aligned} \tag{1.27}$$

where $C > 0$ depends on $\underline{u}$, p, and the final time T, but not on Δt.

The main difference behavior of the AC method in (1.25) and the Penalty method in (1.22) is that in the AC method the pressure (at the previous time step) is taken into account in the momentum equation, while one can drop the pressure in the Penalty method. However, the most flagrant similarity between the two methods is the usage of the grad-div operator (compare also the role of η in (1.25) and $\frac{1}{\epsilon}$ in (1.22)). The remarks about the grad-div operator made in (1.10) are valid in this case as well. In particular, if one chooses high values of η in order to be accurate, one will ends up with an ill-conditioned linear system.

Remark 1.12 - Vector Penalty Projection method. A strategy to temper the numerical difficulties which may come from considering a strong grad-div term in the momentum balance could be the Vector Penalty Projection (VPP) method. It consists in a splitting

technique initially devised for general saddle-point problems (Angot *et al.*, 2012). When applied to Stokes or NSE (see Angot *et al.* (2008), Angot and Fabrie (2012) and Angot *et al.* (2011) for the Darcy problem) it becomes:

$$\begin{aligned}
&\frac{\underline{\widetilde{u}}^n - \underline{u}^{n-1}}{\Delta t} - \nu \underline{\Delta}\, \underline{\widetilde{u}}^n = \underline{f}^n - \underline{\nabla}\, p^{n-1}\,, \\
&\frac{\underline{\breve{u}}^n}{\Delta t} - \nu \left(\underline{\Delta}\, \underline{\breve{u}}^n + \frac{1}{\widetilde{\epsilon}} \underline{\nabla}\, \underline{\nabla} \cdot \underline{\breve{u}}^n \right) = \frac{\nu}{\widetilde{\epsilon}} \underline{\nabla}\, \underline{\nabla} \cdot \underline{\widetilde{u}}^n\,, \\
&\underline{u}^n = \underline{\widetilde{u}}^n + \underline{\breve{u}}^n\,, \\
&p^n = p^{n-1} - \frac{\nu}{\widetilde{\epsilon}} \underline{\nabla} \cdot \underline{u}^n\,.
\end{aligned} \tag{1.28}$$

System (1.28) is equivalent to the AC method (see Angot and Fabrie (2012) or Guermond and Minev (2015, Section 2.4)). Indeed, set $\eta := \frac{1}{\widetilde{\epsilon}}$ and sum the first two equations of (1.28) to recover the first line in (1.25). From an algorithmic standpoint, in VPP one solves two vector-valued systems per time step (with respect to only one in AC). However, the grad-div operator appears only in the second equation of (1.28) and, most importantly both at left- and right-hand side. In doing so, the right-hand side should lie closer to the range of the operator on the left-hand side. Thus one expects iterative solvers to have better performances than in the standard case. The version of the VPP method outlined in (1.28) is often denoted by $\mathrm{VPP}_{\widetilde{\epsilon}}$, to distinguish it from a variant where two user-defined parameters are considered and which involves a grad-div operator in the first equation as well. This latter variant is denoted by $\mathrm{VPP}_{r,\widetilde{\epsilon}}$, and it is not equivalent to the AC method. In fact, it is inspired by the Augmented Lagrangian technique combined with a splitting which borrows some similarities to the Penalty method, see Angot *et al.* (2012) for more details about the derivation of both $\mathrm{VPP}_{\widetilde{\epsilon}}$ and $\mathrm{VPP}_{r,\widetilde{\epsilon}}$. ◇

Remark 1.13 - Higher order. It is shown in Guermond and Minev (2015) that arbitrary orders of convergence in time can be attained in the context of the AC method by means of a bootstrapping or defect correction technique. In particular, if one wants to achieve k-th-order of accuracy in time, k linear systems similar to (1.25) (meaning that they are composed of similar operators, in particular the grad-div operator always appears) have to be solved per time step. This is a sizeable advantage with respect to the other two segregated methods presented above, namely the Projection and the Penalty methods, for which orders higher than second cannot be proved.

The AC method for the NSE has been considered in Guermond and Minev (2015) and, in order to remain in the spirit of the method (namely, being efficiency-oriented), it is suggested to use an explicit convection term. If one wants to avoid the stability limitation that such a choice induces on the admissible values of the time step, one could choose to consider a linearized convection operator. ◇

Remark 1.14 - Algebraic consequences of the grad-div operator. Solving linear systems with the additional term $-\eta \underline{\nabla}\, \underline{\nabla} \cdot \underline{u}$ might not be so easy, hence reducing the advantages the AC method not as much convenient with respect to the coupled system (i.e. the saddle-point problem of the monolithic approach). On the one hand, owing to (1.27), one tends to choose high values for η which could lead to ill-conditioned systems. On the other hand, the grad-div operator couples all the Cartesian components of the velocity, hence increasing the filling of the matrix with extra-diagonal entries. A solution to this latter problem consists in considering direction-splitting schemes as proposed in Guermond and Minev (2017). For instance, in two dimensions, a Gauss–Seidel-like splitting method

Table 1.2 – Main features of the time-stepping strategies for the unsteady NSE discussed in Section 1.3.5. The methods that will be considered in this Thesis are highlighted ("HH" stands for "Helmholtz-Hodge").

	Monolithic	Projection	Penalty	AC
Key feature	Saddle-point	HH splitting	grad-div	grad-div
Pros	Accuracy Arbitrary order	Simplicity	Efficiency	Efficiency Arbitrary order
Cons	Numerical effort	BCs Limited order	Relaxed mass Adjust parameter No p feedback	Relaxed mass Adjust parameter

could rely on replacing

$$\underline{\nabla}\,\underline{\nabla}\cdot\underline{v}^n = \begin{bmatrix} \frac{\partial}{\partial x_1}\left(\frac{\partial v_1^n}{\partial x_1} + \frac{\partial v_2^n}{\partial x_2}\right) \\ \frac{\partial}{\partial x_2}\left(\frac{\partial v_1^n}{\partial x_1} + \frac{\partial v_2^n}{\partial x_2}\right) \end{bmatrix} \quad \text{with} \quad \begin{bmatrix} \frac{\partial}{\partial x_1}\left(\frac{\partial v_1^n}{\partial x_1} + \frac{\partial v_2^{n-1}}{\partial x_2}\right) \\ \frac{\partial}{\partial x_2}\left(\frac{\partial v_1^n}{\partial x_1} + \frac{\partial v_2^n}{\partial x_2}\right) \end{bmatrix}. \tag{1.29}$$

The 2D unsteady Stokes AC problem modified by taking the right-hand side of (1.29) is proved to be unconditionally stable. However, to our knowledge there is no theoretical proof in 3D, although the numerical results in Guermond and Minev (2017) suggest a stable a stable behavior. ◇

Summary and retained techniques

We give in Table 1.2 a one-glance summary of the time-stepping techniques discussed in this section. Among these methods, two will be retained and used in this Thesis, namely the monolithic approach and the AC method. The monolithic approach has been retained because it is the only one that does not introduce an additional error: although it may require an important numerical effort, the monolithic approach will be our choice when accuracy is of order. On the other end of the spectrum, one can find the AC method which is oriented towards efficiency. One can choose this method when some accuracy might be traded in favor of quick results. We prefer the AC method over the Penalty technique although the two are somewhat similar because we think it lies on more sound basis and, furthermore, the possibility of devising time schemes with arbitrary order of convergence is very promising (although, no order higher than the second is considered in this Thesis).

Remark 1.15 - Stabilization by grad-div. We have encountered the grad-div operator several times in our discussion. As a matter of fact, it is known to have a stabilizing effect on the system and to help in recovering divergence-free discrete solutions. The grad-div operator is at the core of the Augmented Lagrangian technique (Hestenes, 1969; Fortin and Glowinski, 1983): recall that in $\mathbf{A}_\lambda := \mathbf{A} + \lambda\mathbf{B}^T\mathbf{B}$ from (1.14), the term $\mathbf{B}^T\mathbf{B}$ is a discrete vision of the continuous grad-div operator. Furthermore, several authors advocate adding the grad-div operator even when the monolithic approach is retained, see, for instance, Olshanskii (2002), Olshanskii and Reusken (2004), Olshanskii and Benzi (2008), Layton *et al.* (2009), and Galvin *et al.* (2012). Finally, in the Penalty and AC methods, the grad-div operator is the result of the decoupling of the velocity and the pressure. A drawback of the grad-div operator is that it couples all the Cartesian components of the velocity (which is not the case for the diffusion operator, for instance) and thus increases the filling of the matrix. However, see Remark 1.14 for some insight on how to deal with the grad-div operator. ◇

1.4 Document overview

The topics of the four chapters which constitute the rest of this Thesis are summarized here. Part of Chapter 2 and some numerical results given in Chapter 3 have been presented in

- Bonelle, J., Ern, A., and Milani, R. (2020). "Compatible Discrete Operator schemes for the steady incompressible Stokes and Navier–Stokes equations". In: *Finite Vol. Complex Appl. IX; Methods Theor. Aspects.* Ed. by R. Klöfkorn *et al.* Vol. 323. Springer Proc. Math. Stat. Bergen: Springer International Publishing, pp. 93–101.

Chapter 2 sets the discrete setting that is considered in this Thesis. It opens with the definition of some useful notation, the mesh regularity assumptions are stated, and the CDO-Fb discretization (a low-order hybrid discretization with cell- and face-based DoFs) for the NSE is introduced. Next, the discrete operators used in this Thesis are presented. We start by extending to the vector-valued case the stabilized gradient reconstruction introduced in Bonelle (2014, Section 8.3); consistency (i.e. exactness on affine functions) and stability properties are proved. The second operator is the divergence, which ensures the velocity-pressure coupling; a discrete inf-sup condition is proved. Lastly, taking inspiration from the HHO framework, an advection operator for scalar problems is introduced and then extended to the vector-valued case in order to recover a suitable operator for the convection term in the NSE. We prove a bound on the consistency error for the scalar operator, recover a discrete counterpart of a well-known integration-by-parts result and, finally, show that the convection operator is non-dissipative under appropriate conditions.

The CDO-Fb discretization of the incompressible steady Stokes and NSE is addressed in Chapter 3. Firstly, we derive the discrete variational formulation of the Stokes equations using the CDO-Fb operators introduced in Chapter 2. It is then shown how these operators are implemented in practice, and an algebraic point of view of the static condensation (procedure which enables one to temporarily eliminate the cell-based DoFs) is given as well. The same steps are then followed for the NSE. The setting is tested on classical test cases, such as the 2D Bercovier–Engelman flow for the steady Stokes problem, or the 2D lid-driven cavity problem for the NSE. When reference solutions are available, the expected orders of convergence in space are investigated for discrete L^2- and H^1-like norms of the velocity and the L^2-like norm of the pressure.

Chapter 4 deals with the unsteady versions of the Stokes and NSE introduced in the previous chapter. A simple setting consisting in an implicit first-order time discretization is considered. In the first part, two common strategies used to deal with the velocity-pressure coupling are discussed: the monolithic approach, chosen for its accuracy, and the Artificial Compressibility (AC) method, chosen for its efficiency. For each strategy, advantages and drawbacks are discussed, discretized systems by means of CDO-Fb schemes are presented, and a kinetic energy balance is given. In the second part of the chapter, we move to the NSE and we compare three classical techniques to address the nonlinear term resulting from the convection term. We consider, in particular, a Picard algorithm as well as a linearized and an explicit treatment. Some insights are given in order to recover a discrete kinetic energy balance. Finally, numerical tests are presented, such as the 2D Taylor–Green vortex, in order to assess the implementation and the accuracy of our time-stepping techniques. Moreover, taking advantage of a large-size test case on an unsteady Stokes problem, we compare the performances of the monolithic approach and the AC method. We also study numerically the stability limit on the time-step when treating the convection term explicitly.

The analysis of a second-order time-discretization of the unsteady NSE with CDO-Fb is the subject of Chapter 5. If, on the one hand, a classical method such as the Backward Differentiation Formulae (BDF) suffices for the monolithic approach, a more sophisticated technique has to be considered with the AC strategy, that is, a bootstrapping procedure

introduced in Guermond and Minev (2015). In the case of the NSE, we couple both strategies with the convection treatments presented in Chapter 4. Finally, considering again the same numerical examples as in Chapter 4 allows us, on the one hand, to show that the expected orders of convergence in time are recovered and, on the other hand, to compare not only the two velocity-pressure couplings (monolithic and AC), but also the two time-schemes (first- and second-order) in order to see which strategies are the most efficient.

Chapter 2

Discrete face-based CDO setting

Contents

2.1 The mesh **37**
2.1.1 Basic definitions 38
2.1.2 Mesh regularity 39
2.2 Functional discrete setting and degrees of freedom **40**
2.2.1 Degrees of freedom in a CDO-Fb discretization 41
2.2.2 Reduction maps 42
2.3 Velocity gradient reconstruction **43**
2.4 Velocity-pressure coupling **47**
2.4.1 Main definitions and basic properties 47
2.4.2 Inf-sup condition 49
2.5 Scalar-valued advection and vector-valued convection **50**
2.5.1 Scalar-valued advection 50
2.5.2 Vector-valued convection 55
2.6 Source term **60**

This chapter introduces the discretization tools on which the face-based CDO (CDO-Fb) problem is built. Firstly, in Section 2.1, the notion of mesh is introduced along with the needed regularity assumptions. The discrete functional spaces and related degrees of freedom (DOFs) are presented in Section 2.2. Then, the CDO operators needed to build the Navier–Stokes equations (NSE) are discussed: the gradient reconstruction in Section 2.3, velocity-pressure coupling in Section 2.4, the convection operator in Section 2.5, and finally the body force treatment in Section 2.6.

2.1 The mesh

We now introduce the notation and the geometric setting on which the design of CDO-Fb hinges. Let $\Omega \subset \mathbb{R}^d$ be an open, polytopal, bounded, Lipschitz domain on which the NSE are posed. The space dimension may take the values $d \in \{2, 3\}$. The boundary of the domain is denoted by $\partial\Omega$, and its outward unit normal, which can be defined (almost) everywhere on $\partial\Omega$, is denoted by $\underline{n}_{\partial\Omega}$. This section deals with the way of discretizing the domain Ω. We notice that even though the vocabulary is related to three-dimensional settings, the discussion can be carried out in two dimensions as well.

2.1.1 Basic definitions

Definition 2.1 - Mesh. A *mesh* $\mathrm{M} := \{\mathrm{F}, \mathrm{C}\}$ is a discretization of Ω composed of elements of dimension d, the *cells*, and $d-1$, the *faces*. The set of the cells is a finite collection $\mathrm{C} := \{\mathrm{c}\}$ of nonempty, disjoint, open polytopes covering Ω exactly. A generic mesh element is denoted by c. A generic mesh face is denoted by f and the mesh faces are collected in the set F. If $d = 2$, faces are also called edges. The mesh also contains vertices and edges (which differ from faces if $d \geq 3$), but these additional mesh entities are not needed in what follows. ∘

Definition 2.2 - Internal and boundary faces. The set of the faces is partitioned into boundary and internal faces. The former are subsets of $\partial\Omega$, and collected in $\mathrm{F}^b := \{\mathrm{f} \in \mathrm{F} \,|\, \mathrm{f} \subset \partial\Omega\}$. The latter are those faces shared by two distinct cells, and they are collected in the set is $\mathrm{F}^i := \mathrm{F} \setminus \mathrm{F}^b$. ∘

The faces are supposed to be planar. This property is assumed so that one can define a constant unit normal vector, denoted by $\underline{n}_\mathrm{f}$, valid for the whole face. It is chosen to point outward of Ω if $\mathrm{f} \in \mathrm{F}^b$, and its orientation is left arbitrary but fixed once and for all if $\mathrm{f} \in \mathrm{F}^i$.

Let us set some convenient notation and definitions. The hash symbol # will stand for the cardinality of a set: for instance, $\#\mathrm{F}^b$ is the number of boundary faces. The classical Hausdorff measure of sets in $\mathbb{R}^{d'}$, $1 \leq d' \leq d$, is denoted by $|\cdot|$ and the subscript is dropped if there is no ambiguity. Hence, $|\mathrm{c}|$ is the volume of $\mathrm{c} \in \mathrm{C}$ and $|\mathrm{f}|$ is the surface of $\mathrm{f} \in \mathrm{F}$. Moreover, $|\underline{x}_1 - \underline{x}_2|$ is the distance between $\underline{x}_1$, $\underline{x}_2 \in \mathbb{R}^d$.

Definition 2.3 - Barycenter of a mesh entity. The barycenter of a generic mesh entity $z = \mathrm{c}$ or $z = \mathrm{f}$ is denoted by $\underline{x}_z$:

$$\underline{x}_z := \frac{1}{|z|} \int_z \underline{x} \,. \quad \circ \tag{2.1}$$

Definition 2.4 - Diameter of a mesh entity. One defines the diameter of a generic mesh entity $z = \mathrm{c}$ or $z = \mathrm{f}$ as follows:

$$\mathrm{h}_z := \max_{\underline{x}_1, \underline{x}_2 \in z} |\underline{x}_1 - \underline{x}_2| \,. \quad \circ \tag{2.2}$$

Definition 2.5 - Mesh size. $\mathrm{h} := \max_{\mathrm{c} \in \mathrm{C}} \mathrm{h}_\mathrm{c}$ is called the size of the mesh M. ∘

The following property is assumed to hold for all the meshes considered in this Thesis.

Definition 2.6 - Barycentric star-shapedness. Given a mesh $\mathrm{M} = \{\mathrm{C}, \mathrm{F}\}$, every cell $\mathrm{c} \in \mathrm{C}$ is star-shaped with respect to its barycenter $\underline{x}_\mathrm{c}$. The same holds for every face $\mathrm{f} \in \mathrm{F}$ with respect to its barycenter $\underline{x}_\mathrm{f}$. ∘

Let us also define some convenient local objects which will be used in what follows. Consider a cell $\mathrm{c} \in \mathrm{C}$. The faces composing its boundary are collected in the set

$$\mathrm{F}_\mathrm{c} := \{\mathrm{f} \in \mathrm{F} \,|\, \mathrm{f} \subset \partial\mathrm{c}\} \,. \tag{2.3}$$

For each face $\mathrm{f} \in \mathrm{F}_\mathrm{c}$, a unit normal vector pointing outward c is denoted by $\underline{n}_{\mathrm{fc}}$. Hence, $\underline{n}_{\mathrm{fc}} = \iota_{\mathrm{f,c}} \underline{n}_\mathrm{f}$, where $\iota_{\mathrm{f,c}} = \pm 1$ according to the orientation which was chosen for $\underline{n}_\mathrm{f}$. We are going to make use of a subdivision of the cell c as $\mathfrak{P}_\mathrm{c} := \{\mathfrak{p}_{\mathrm{f,c}}\}_{\mathrm{f} \in \mathrm{F}_\mathrm{c}}$, where the subsets $\mathfrak{p}_{\mathrm{f,c}}$ are the nonempty, disjoint subpyramids (or subtriangles if $d = 2$) obtained by considering the cell barycenter $\underline{x}_\mathrm{c}$ as apex, and a face $\mathrm{f} \in \mathrm{F}_\mathrm{c}$ as basis. The set $\mathfrak{P}_\mathrm{c}$ indeed induces a partition of c if c is star-shaped with respect to its barycenter, which is ensured by Definition 2.6. An example of a mesh cell and of one subpyramid is shown in Fig. 2.1.

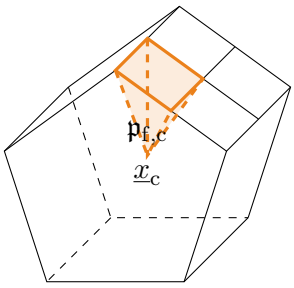


Figure 2.1 – Example of a face-based subdivision of a 3D cell with one hanging node: because of the hanging node, this pentagon-prismatic cell is considered to have 10 faces (instead of 7). A subpyramid obtained by considering a face as basis and the cell-barycenter as apex is highlighted.

A useful identity is as follows:

$$\sum_{\mathrm{f}\in\mathrm{F_c}} |\mathrm{f}|\, \underline{n}_{\mathrm{fc}} = \underline{0} \qquad \forall \mathrm{c} \in \mathrm{C}\,. \tag{2.4}$$

This identity is simply proved by observing that $\sum_{\mathrm{f}\in\mathrm{F_c}} |\mathrm{f}|\, \underline{n}_{\mathrm{fc}} = \int_{\partial\mathrm{c}} \underline{n}_{\partial\mathrm{c}} = \underline{0}$, where the first equality holds since the faces are planar.

Given a face $\mathrm{f} \in \mathrm{F}$, $\mathrm{C_f}$ is the set of the cells whose boundary contains f, that is, $\mathrm{C_f} \coloneqq \{\mathrm{c} \in \mathrm{C} \,|\, \mathrm{f} \subset \partial\mathrm{c}\}$, yielding

$$\#\mathrm{C_f} = \begin{cases} 2 & \mathrm{f} \in \mathrm{F}^i, \\ 1 & \mathrm{f} \in \mathrm{F}^b\,. \end{cases} \tag{2.5}$$

2.1.2 Mesh regularity

We are going to consider the setting of Di Pietro and Ern (2015) (see also Di Pietro and Ern (2011, Section 1.4)). The main definitions and results are recalled here for the sake of completeness. The chosen mesh framework is often considered when dealing with polyhedral meshes (see, for instance, Brezzi *et al.* (2009) and Eymard *et al.* (2010)). On the one hand, once the appropriate functional spaces are set, it allows one to recover essential properties which are classically used in the Finite Element literature such as the trace and inverse inequalities (see, for instance, Ern and Guermond (2004, Section 1.7)). On the other hand, the needed assumptions are fairly general and are often met in most of the meshes that are commonly used in practical implementations or applications.

Before stating the assumptions on the mesh regularity, some further definitions are required. Consider a mesh sequence $\mathrm{M_H} \coloneqq \{\mathrm{M_h}\}_{\mathrm{h}\in\mathrm{H}}$, where the set $\mathrm{H} \coloneqq \{\mathrm{h} \in \mathbb{R} \,|\, \mathrm{h} > 0\}$ is such that it is countable and its only accumulation point is 0. The index h indicates the size of the mesh, see Definition 2.4 above. Let us recall the classical definition of a simplex.

Definition 2.7 - Simplex. Fix a dimension $d \geq 2$. For any $d' = \{1, \dots, d\}$, given a set of $(d'+1)$ points in $\mathbb{R}^d$ $\{\underline{x}_0, \dots, \underline{x}_{d'}\}$, such that the vectors $\{\underline{x}_1 - \underline{x}_0, \dots, \underline{x}_{d'} - \underline{x}_0\}$ are linearly independent, the convex hull of the points is called a d'-dimensional simplex in $\mathbb{R}^d$ (in short, simplex if $d' = d$). ◦

Definition 2.7 leads to a triangle if $d' = 2$, and to a tetrahedron if $d' = 3$.

Definition 2.8 - Matching simplicial submesh. Fix a dimension $d \geq 2$ and take a general mesh $\mathrm{M_h} = \{\mathrm{F}, \mathrm{C}\}$ and a simplicial one $\mathfrak{S}_\mathrm{h} = \{\mathfrak{F}, \mathfrak{C}\}$ (all its faces and cells are simplexes). Then, $\mathfrak{S}_\mathrm{h}$ is said to be a *matching simplicial submesh* of $\mathrm{M_h}$ if:

(i) For any $\mathfrak{c} \in \mathfrak{C}$ with vertices $\{\underline{x}_0, \dots, \underline{x}_d\}$, for any $\mathfrak{c}' \in \mathfrak{C}$, $\mathfrak{c}' \neq \mathfrak{c}$ such that $\partial\mathfrak{c}' \cap \partial\mathfrak{c}$ is not empty, this subset is a lower-dimensional simplex formed from a subset of $\{\underline{x}_0, \dots, \underline{x}_d\}$;

(ii) For any $\mathfrak{c} \in \mathfrak{C}$, there exists only one $\mathrm{c} \in \mathrm{C}$ such that $\mathfrak{c} \subset \mathrm{c}$;

(iii) For any $\mathfrak{f} \in \mathfrak{F}$, there exists only one $\mathrm{f} \in \mathrm{F}$ such that $\mathfrak{f} \subset \mathrm{f}$. ∘

The following definition deals with classical regularity requirements of mesh sequences and will be assumed to hold for all the mesh sequences considered in this Thesis.

Definition 2.9 - Shape regularity. A mesh sequence $\mathrm{M_H}$ is said to be shape-regular if for all $\mathrm{h} \in \mathrm{H}$:

(SR1) $\mathrm{M_h}$ admits a matching simplicial submesh $\mathfrak{S}_\mathrm{h}$ which is itself shape-regular in the usual sense of Ciarlet (1978): there exists $\rho_1 > 0$, independent of $\mathrm{h} \in \mathrm{H}$, such that

$$\rho_1 \mathrm{h}_\mathfrak{c} \leq r_\mathfrak{c} \qquad \forall \mathfrak{c} \in \mathfrak{C}\,, \tag{2.6}$$

where $r_\mathfrak{c}$ is the inradius of $\mathfrak{c}$ (the diameter of the largest ball inscribed in $\mathfrak{c}$);

(SR2) There exists $\rho_2 > 0$, independent of $\mathrm{h} \in \mathrm{H}$, such that

$$\begin{aligned} \rho_2 \mathrm{h_c} \leq & \mathrm{h}_\mathfrak{c} \qquad && \forall \mathrm{c} \in \mathrm{C},\ \forall \mathfrak{c} \in \mathfrak{C} \text{ such that } \mathfrak{c} \subset \mathrm{c}\,, \\ \rho_2 \mathrm{h_f} \leq & \mathrm{h}_\mathfrak{f} \qquad && \forall \mathrm{f} \in \mathrm{F},\ \forall \mathfrak{f} \in \mathfrak{F} \text{ such that } \mathfrak{f} \subset \mathrm{f}\,. \quad \circ \end{aligned} \tag{2.7}$$

Remark 2.10 - Equivalent length scales. Notice that assumptions (SR1) and (SR2) lead to

$$\rho_1\rho_2 \mathrm{h_c} \leq \mathrm{h_f} \leq \mathrm{h_c} \quad \forall \mathrm{c} \in \mathrm{C},\ \forall \mathrm{f} \in \mathrm{F_c}\,. \quad \diamond \tag{2.8}$$

Remark 2.11 - Uniformly bounded cardinality. Consider a mesh $\mathrm{M_h} = \{\mathrm{F}, \mathrm{C}\} \in \mathrm{M}_H$ and its simplicial matching submesh $\mathfrak{S}_\mathrm{h} = \{\mathfrak{F}, \mathfrak{C}\}$. Let $\mathfrak{C}_\mathrm{c} := \{\mathfrak{c} \in \mathfrak{C}\}$ and $\mathfrak{F}_\mathrm{f} := \{\mathfrak{f} \in \mathfrak{F}\}$. Then, assuming Definition 2.9, Di Pietro and Ern (2011, Lemmata 1.40-41) prove that $\#\mathfrak{C}_\mathrm{c}$, $\#\mathfrak{F}_\mathrm{f}$ and $\#\mathrm{F_c}$ are uniformly bounded with respect to h. ◇

Remark 2.12 - Original CDO mesh requirements. Owing to Remark 2.11, the requirements for a sequence of meshes to be of class (MR) according to Bonelle (2014, Definition 5.8) (see also Bonelle and Ern (2014)), which collects the mesh regularity assumptions in the first version of CDO, are met by a shape-regular sequence in the sense of Definition 2.9. The mesh regularity assumptions considered for the CDO framework in Cantin (2016) are also met by Definition 2.9, provided that the star-shapedness (see Definition 2.6) is additionally assumed. Inequality (2.8) is recovered also from class (MR) (allowing one to replace hypothesis (SR2) with the uniform boundedness of the simplicial sub-decomposition, see Brezzi *et al.* (2009)). ◇

2.2 Functional discrete setting and degrees of freedom

We set in this section the functional spaces considered in the CDO-Fb discretization and how they translate into DoFs.

Let us set some preliminary notation. For a generic mesh entity $z = \mathrm{c}$ or $z = \mathrm{f}$, $\underline{\mathbb{P}}^q(z)$ (resp. $\underline{\underline{\mathbb{P}}}$ denotes the $\mathbb{R}^d$-valued (resp. tensor-valued) polynomials of degree q at most and defined on z. For instance, if $d = 3$, $\underline{\mathbb{P}}^1(\mathrm{c})$ collects the $\mathbb{R}^3$-valued affine functions defined on c. If the polynomials are scalar, the underline is omitted: $\mathbb{P}^1(\mathrm{c})$ collects the scalar-valued affine functions defined on c. With an abuse of notation, if now one considers a collection Z of mesh entities instead of a single mesh entity as the domain of the polynomials, $\mathbb{P}^q(\mathrm{Z})$ refers to piecewise-polynomials functions; for instance, $\mathbb{P}^1(\mathrm{F_c}) := \bigtimes_{\mathrm{f}\in\mathrm{F_c}} \mathbb{P}^1(\mathrm{f})$.

2.2.1 Degrees of freedom in a CDO-Fb discretization

The CDO-Fb setting is a lowest-order discretization, hence it considers piecewise constant variables. A discrete, vector-valued variable, such as the velocity for instance, is discretized using the following spaces:

$$\underline{\mathcal{U}}_z \coloneqq \underline{\mathbb{P}}^0(z) \equiv \mathbb{R}^d\,, \quad \text{with } z \in \mathrm{F} \text{ or } z \in \mathrm{C}\,. \tag{2.9}$$

A generic element of $\underline{\mathcal{U}}_z$ is usually reported with a lower-case underlined letter and a subscript indicating the mesh entity on which it is defined: for instance, we write $\underline{u}_\mathrm{f} \in \underline{\mathcal{U}}_\mathrm{f}$. In the CDO-Fb discretization, the velocity is hybrid, meaning that it has DoFs attached to both faces and cells. Considering a cell $\mathrm{c} \in \mathrm{C}$, the local hybrid velocity space associated with c is defined as follows:

$$\widehat{\underline{\mathcal{U}}}_\mathrm{c} \coloneqq \bigtimes_{\mathrm{f}\in\mathrm{F_c}} \underline{\mathcal{U}}_\mathrm{f} \times \underline{\mathcal{U}}_\mathrm{c} \equiv \mathbb{R}^{d(\#\mathrm{F_c}+1)}\,, \quad \forall \mathrm{c} \in \mathrm{C}\,. \tag{2.10}$$

Hybrid variables and spaces will be denoted with a hat: $\widehat{\underline{u}}_\mathrm{c} \coloneqq ((\underline{u}_\mathrm{f})_{\mathrm{f}\in\mathrm{F_c}}, \underline{u}_\mathrm{c}) \in \widehat{\underline{\mathcal{U}}}_\mathrm{c}$. When considering the global space, the face values are uniquely defined, so that the global hybrid velocity space is not simply the union of the local ones, but instead

$$\widehat{\underline{\mathcal{U}}}_\mathrm{h} \coloneqq \bigtimes_{\mathrm{f}\in\mathrm{F}} \underline{\mathcal{U}}_\mathrm{f} \times \bigtimes_{\mathrm{c}\in\mathrm{C}} \underline{\mathcal{U}}_\mathrm{c} \equiv \mathbb{R}^{d(\#\mathrm{F}+\#\mathrm{C})}\,. \tag{2.11}$$

For a generic $\widehat{\underline{v}}_\mathrm{h} \in \widehat{\underline{\mathcal{U}}}_\mathrm{h}$, we use the notation

$$\widehat{\underline{v}}_\mathrm{h} \coloneqq (\underline{v}_\mathrm{F}, \underline{v}_\mathrm{C}) = ((\underline{v}_\mathrm{f})_{\mathrm{f}\in\mathrm{F}}, (\underline{v}_\mathrm{c})_{\mathrm{c}\in\mathrm{C}})\,. \tag{2.12}$$

Remark 2.13 - Summation over face-based DoFs. Owing to the single-valuedness of the face-based DoFs and the skew-symmetry of the normal vector at an internal face with respect to the two adjacent cells, one has for all $\widehat{\underline{v}}_\mathrm{h} \in \widehat{\underline{\mathcal{U}}}_\mathrm{h}$,

$$\begin{aligned}\sum_{\mathrm{c}\in\mathrm{C}}\sum_{\mathrm{f}\in\mathrm{F_c}}\int_\mathrm{f} \underline{v}_\mathrm{f}\cdot\underline{n}_\mathrm{fc} &= \sum_{\mathrm{f}\in\mathrm{F}}\sum_{\mathrm{c}\in\mathrm{C_f}}\int_\mathrm{f} \underline{v}_\mathrm{f}\cdot\underline{n}_\mathrm{fc} = \sum_{\mathrm{f}\in\mathrm{F}^b}\int_\mathrm{f} \underline{v}_\mathrm{f}\cdot\underline{n}_\mathrm{f} + \sum_{\mathrm{f}\in\mathrm{F}^i}\sum_{\mathrm{c}\in\mathrm{C_f}}\int_\mathrm{f} \underline{v}_\mathrm{f}\cdot\underline{n}_\mathrm{fc} \\ &= \sum_{\mathrm{f}\in\mathrm{F}^b}\int_\mathrm{f} \underline{v}_\mathrm{f}\cdot\underline{n}_\mathrm{f} + \sum_{\mathrm{f}\in\mathrm{F}^i}\int_\mathrm{f} \underline{v}_\mathrm{f}\cdot(\underline{n}_\mathrm{fc}-\underline{n}_\mathrm{fc}) = \sum_{\mathrm{f}\in\mathrm{F}^b}\int_\mathrm{f} \underline{v}_\mathrm{f}\cdot\underline{n}_\mathrm{f}\,,\end{aligned} \tag{2.13}$$

where only the boundary faces are left since the contribution of the internal ones sums to zero. Using the same arguments, a similar identity is proven for a continuous function $\underline{v} \in \underline{\mathcal{C}}^0(\overline{\Omega})$:

$$\sum_{\mathrm{c}\in\mathrm{C}}\sum_{\mathrm{f}\in\mathrm{F_c}}\int_\mathrm{f} \underline{v}\cdot\underline{n}_\mathrm{fc} = \sum_{\mathrm{f}\in\mathrm{F}^b}\int_\mathrm{f} \underline{v}\cdot\underline{n}_\mathrm{f}\,. \quad \diamond \tag{2.14}$$

Concerning the CDO-Fb discrete pressure, it is attached to the cells only. Hence, following a similar notation to that for the velocity, the local and global discrete pressure spaces are

$$\mathcal{P}_\mathrm{c} \coloneqq \mathbb{P}^0(\mathrm{c}) \equiv \mathbb{R}, \quad \forall \mathrm{c} \in \mathrm{C}, \qquad \mathcal{P}_\mathrm{h} \coloneqq \bigtimes_{\mathrm{c}\in\mathrm{C}} \mathcal{P}_\mathrm{c} \equiv \mathbb{R}^{\#\mathrm{C}}\,. \tag{2.15}$$

The complete set of DoFs for velocity and pressure in a polyhedral cell is shown in Fig. 2.2.

When devising the CDO-Fb discretization of NSE, we consider for the sake of simplicity homogeneous Dirichlet boundary conditions (BCs) for the velocity and a zero mean-value constraint on the pressure. We take into consideration these two constraints directly in the (discrete) functional spaces, namely we additionally define:

$$\widehat{\underline{\mathcal{U}}}_{\mathrm{h},0} \coloneqq \left\{\widehat{\underline{v}}_\mathrm{h} \in \widehat{\underline{\mathcal{U}}}_\mathrm{h} \,|\, \underline{v}_\mathrm{f} = \underline{0}\ \forall \mathrm{f} \in \mathrm{F}^b\right\}\,, \tag{2.16}$$

$$\mathcal{P}_{\mathrm{h},*} \coloneqq \left\{q_\mathrm{h} \in \mathcal{P}_\mathrm{h} \,|\, \sum_{\mathrm{c}\in\mathrm{C}} |\mathrm{c}|\, q_\mathrm{c} = 0\right\}\,. \tag{2.17}$$

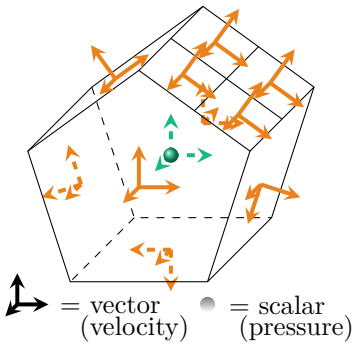


Figure 2.2 – Velocity (vectors) and pressure (ball) degrees of freedom for a CDO-Fb discretization on a cell with one hanging node. Face-based DoFs are depicted in orange, cell-based ones in green.

2.2.2 Reduction maps

Reduction maps are operators that allow one to compute discrete DoFs from continuous (or smooth enough) functions. For our framework, where the discrete spaces are composed of functions which are entity-wise constant, it is sufficient to consider L^2-orthogonal projections (that is, mean values).

Definition 2.14 - Local orthogonal projections. Let $\mathrm{c} \in \mathrm{C}$ and consider $z = \mathrm{c}$ or $z = \mathrm{f}$ with $\mathrm{f} \in \mathrm{F_c}$. Set $\mathcal{D}(\pi_z) = L^1(\mathrm{c})$ if $z = \mathrm{c}$, and $\mathcal{D}(\pi_z) = H^s(\mathrm{c})$, $s > \frac{1}{2}$, if $z = \mathrm{f} \in \mathrm{F_c}$. The orthogonal L^2-projection corresponds to the average of the function on c or the average of the trace of the function on f, and is defined as follows

$$\begin{array}{rccl} \pi_z\colon & \mathcal{D}(\pi_z) & \to & \mathbb{P}^0(z) \equiv \mathbb{R} \\ & q & \mapsto & \dfrac{1}{|z|}\displaystyle\int_z q\,. \end{array} \tag{2.18a}$$

Notice that the subscript denotes the entity onto which the reduction is made. For d-valued functions, such as the velocity, one defines similarly

$$\begin{array}{rccl} \underline{\pi}_z\colon & \mathcal{D}(\underline{\pi}_z) & \to & \underline{\mathbb{P}}^0(z) \equiv \mathbb{R}^d \\ & \underline{v} & \mapsto & \dfrac{1}{|z|}\displaystyle\int_z \underline{v}\,, \end{array} \tag{2.18b}$$

where $\mathcal{D}(\underline{\pi}_z) = \underline{L}^1(\mathrm{c})$ if $z = \mathrm{c}$, and $\mathcal{D}(\underline{\pi}_z) = \underline{H}^s(\mathrm{c})$, $s > \frac{1}{2}$, if $z = \mathrm{f} \in \mathrm{F_c}$. Recall that the underlined notation indicates a vector-valued field; thus, for instance, $\underline{L}^2(z) := [L^2(z)]^d$, and similarly for $\underline{H}^1$. Finally, when dealing with the projection onto hybrid spaces, we set

$$\begin{array}{rccl} \widehat{\underline{\pi}}_\mathrm{c}\colon & \underline{H}^s(\mathrm{c}) & \to & \widehat{\underline{\mathcal{U}}}_\mathrm{c} \\ & \underline{v} & \mapsto & \Big((\underline{\pi}_\mathrm{f}(\underline{v}))_{\mathrm{f}\in\mathrm{F_c}}\,,\, \underline{\pi}_\mathrm{c}(\underline{v})\Big)\,, \end{array} \tag{2.18c}$$

with $s > \frac{1}{2}$. ◦

Definition 2.15 - Global orthogonal projections. Definition 2.14 is extended readily to the global spaces by setting

$$\begin{aligned} &\widehat{\underline{\pi}}_\mathrm{h}(\underline{v}) := \Big((\underline{\pi}_\mathrm{f}(\underline{v}))_{\mathrm{f}\in\mathrm{F_c}}\,,\, (\underline{\pi}_\mathrm{c}(\underline{v}))_{\mathrm{c}\in\mathrm{C}}\Big) \in \widehat{\underline{\mathcal{U}}}_\mathrm{h}\,, && \forall \underline{v} \in \underline{H}^s(\Omega)\,, s > \frac{1}{2}\,, \\ &\pi_\mathrm{h}(q) := (\pi_\mathrm{c}(q))_{\mathrm{c}\in\mathrm{C}} \in \mathcal{P}_\mathrm{h}\,, && \forall q \in L^2(\Omega)\,. \end{aligned} \tag{2.19}$$

Notice that $\widehat{\underline{\pi}}_\mathrm{h}(\underline{v}) \in \widehat{\underline{\mathcal{U}}}_{\mathrm{h},0}$ if $\underline{v} \in \underline{H}^1_0(\Omega)$ and $\pi_\mathrm{h}(q) \in \mathcal{P}_{\mathrm{h},*}$ if $q \in L^2_*(\Omega)$. ◦

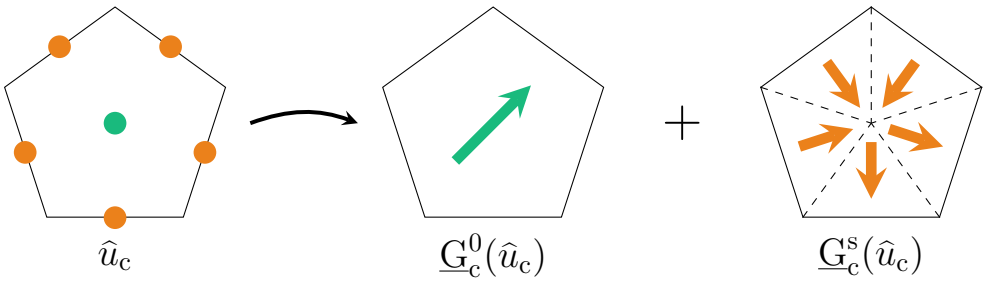


Figure 2.3 – Building stages of the gradient reconstruction operator $\underline{\mathrm{G}}_{\mathrm{c}} = \underline{\mathrm{G}}_{\mathrm{c}}^{0} + \underline{\mathrm{G}}^{\mathrm{s}}$. For the sake of clarity, we have considered a scalar-valued hybrid variable $\widehat{u}_{\mathrm{c}}$. The consistent part $\underline{\mathrm{G}}_{\mathrm{c}}^{0}$ is cell-wise constant, the stabilization part $\underline{\mathrm{G}}_{\mathrm{c}}^{\mathrm{s}}$ is piecewise constant on the cell subpartition. Face- and cell-based quantities are shown, respectively, in orange and green.

2.3 Velocity gradient reconstruction

The first operator needed in the discretization of the NSE is the reconstruction of the gradient of the velocity, used to deal with the diffusive part of the problem. This operator is based on an orthogonal consistency-stabilization split identified in Bonelle *et al.* (2015) and it has been considered for the scalar Laplacian problem in Bonelle (2014, Ch. 8). Here, we readily extend it to the vector-valued case and recall its main properties.

Definition 2.16 - Local velocity gradient reconstruction. Consider a cell $\mathrm{c} \in \mathrm{C}$ and its pyramidal partition $\mathfrak{P}_{\mathrm{c}}$. The tensor-valued velocity gradient reconstruction $\underline{\underline{\mathrm{G}}}_{\mathrm{c}}(\widehat{\underline{u}}_{\mathrm{c}})$ is defined as follows

$$\begin{array}{rccl} \underline{\underline{\mathrm{G}}}_{\mathrm{c}}: & \widehat{\mathcal{U}}_{\mathrm{c}} & \to & \underline{\underline{\mathbb{P}}}^{0}(\mathfrak{P}_{\mathrm{c}}) := [\mathbb{P}^{0}(\mathfrak{P}_{\mathrm{c}})]^{d\times d} \\ & \widehat{\underline{u}}_{\mathrm{c}} & \mapsto & \underline{\underline{\mathrm{G}}}_{\mathrm{c}}^{0}(\widehat{\underline{u}}_{\mathrm{c}}) + \underline{\underline{\mathrm{G}}}_{\mathrm{c}}^{\mathrm{s}}(\widehat{\underline{u}}_{\mathrm{c}}) \end{array} \tag{2.20a}$$

where $\underline{\underline{\mathrm{G}}}_{\mathrm{c}}^{0}: \widehat{\mathcal{U}}_{\mathrm{c}} \to \underline{\underline{\mathbb{P}}}^{0}(\mathrm{c})$ is cell-wise constant and defined as follows:

$$\underline{\underline{\mathrm{G}}}_{\mathrm{c}}^{0}(\widehat{\underline{u}}_{\mathrm{c}}) := \frac{1}{|\mathrm{c}|} \sum_{\mathrm{f}\in\mathrm{F}_{\mathrm{c}}} |\mathrm{f}| \, (\underline{u}_{\mathrm{f}} - \underline{u}_{\mathrm{c}}) \otimes \underline{n}_{\mathrm{fc}}, \tag{2.20b}$$

and $\underline{\underline{\mathrm{G}}}_{\mathrm{c}}^{\mathrm{s}}: \widehat{\mathcal{U}}_{\mathrm{c}} \to \underline{\underline{\mathbb{P}}}^{0}(\mathfrak{P}_{\mathrm{c}})$ is piecewise constant on the subpyramids: $\forall \mathrm{f} \in \mathrm{F}_{\mathrm{c}}$, we set

$$\underline{\underline{\mathrm{G}}}_{\mathrm{c}}^{\mathrm{s}}(\widehat{\underline{u}}_{\mathrm{c}})_{|\mathfrak{p}_{\mathrm{f,c}}} := \beta \frac{|\mathrm{f}|}{|\mathfrak{p}_{\mathrm{f,c}}|} \Big((\underline{u}_{\mathrm{f}} - \underline{u}_{\mathrm{c}}) - \underline{\underline{\mathrm{G}}}_{\mathrm{c}}^{0}(\widehat{\underline{u}}_{\mathrm{c}})(\underline{x}_{\mathrm{f}} - \underline{x}_{\mathrm{c}}) \Big) \otimes \underline{n}_{\mathrm{fc}} \,, \tag{2.20c}$$

with $\beta > 0$ being an arbitrary parameter. Unless stated otherwise, we will take $\beta := 1$. The principles of the gradient reconstruction are illustrated in Fig. 2.3. ◦

Definition 2.17 - Global velocity gradient reconstruction. The global version of the gradient $\underline{\underline{\mathrm{G}}}_{\mathrm{h}}(\widehat{\underline{u}}_{\mathrm{h}})$ simply collects the local instances:

$$\begin{array}{rccl} \underline{\underline{\mathrm{G}}}_{\mathrm{h}}: & \widehat{\mathcal{U}}_{\mathrm{h}} & \to & [\underline{\underline{\mathbb{P}}}^{0}(\{\mathfrak{P}_{\mathrm{c}}\}_{\mathrm{c}\in\mathrm{C}})]^{\#\mathrm{C}} \\ & \widehat{\underline{u}}_{\mathrm{h}} & \mapsto & \Big(\underline{\underline{\mathrm{G}}}_{\mathrm{c}}(\widehat{\underline{u}}_{\mathrm{c}})\Big)_{\mathrm{c}\in\mathrm{C}} \,. \end{array} \tag{2.21}$$

◦

Remark 2.18 - Comparison with similar schemes. Some remarks about the gradient reconstruction have already been made in Section 1.2.2, in particular for the gradient of scalar functions and an elliptic problem, settings which we consider again here for the sake

of simplicity. It has been proven in Bonelle (2014, Prop. 8.38) that the CDO-Fb formulation of the gradient is equivalent to the one of the HFV/SUSHI scheme developed in Eymard *et al.* (2010) for the choice $\beta := \frac{1}{\sqrt{d}}$ of the user-defined stabilization parameter. This is not surprising since it often happens that lowest-order schemes hinging on a stabilized gradient reconstruction are equivalent up to a comparable stabilization. This is the case for the Generalized Crouzeix–Raviart scheme from Di Pietro and Lemaire (2015) as well. The gradient discretization proposed in this framework can be recovered from the CDO-Fb one by choosing $\beta := 1$.

We have already pointed out in Section 1.2.2 (see also Fig. 1.5) the differences between CDO-Fb and the lowest-order HHO method (see Di Pietro *et al.* (2014) for instance for the simplest setting). In particular, the gradient reconstruction in HHO($k = 0$) is cell-wise constant and a stabilization is added to the scheme, whereas in CDO-Fb the gradient reconstruction is piecewise constant on the cell subdivision and no additional stabilization (apart from the one built-in through $\underline{\underline{\mathrm{G}}}_{\mathrm{c}}^{\mathrm{s}}$) is needed. Lemma 2.23 shows that the HHO (stabilized) bilinear form is equivalent to the one resulting from the CDO-Fb discretization up to a comparable stabilization coefficient. ◇

The design of $\underline{\underline{\mathrm{G}}}_{\mathrm{c}}^{0}$ stems from the following geometric identity, valid when the faces are planar (see, e.g. Droniou and Eymard (2006) for the proof)

$$\sum_{\mathrm{f}\in\mathrm{F_c}} |\mathrm{f}|\,(\underline{x}_{\mathrm{f}} - \underline{x}_{\mathrm{c}}) \otimes \underline{n}_{\mathrm{fc}} = |\mathrm{c}|\,\underline{\underline{\mathsf{Id}}}\,. \tag{2.22}$$

$\underline{\underline{\mathrm{G}}}_{\mathrm{c}}^{0}$ is called the consistent part of the gradient reconstruction since it is exact for affine functions. Let us prove this result.

Lemma 2.19 - Exactness of $\underline{\underline{\mathrm{G}}}_{\mathrm{c}}^{0}$ on $\underline{\mathbb{P}}^1(\mathrm{c})$. *Given the definitions* (2.20b) *of* $\underline{\underline{\mathrm{G}}}_{\mathrm{c}}^{0}$ *and* (2.18c) *of* $\underline{\widehat{\pi}}_{\mathrm{c}}$*, it holds*

$$\underline{\underline{\mathrm{G}}}_{\mathrm{c}}^{0} \circ \underline{\widehat{\pi}}_{\mathrm{c}} = \underline{\underline{\nabla}} \quad \text{on } \underline{\mathbb{P}}^1(\mathrm{c})\,, \qquad \forall \mathrm{c} \in \mathrm{C}\,. \tag{2.23}$$

Proof. Consider a generic affine function $\underline{\mathbb{P}}^1(\mathrm{c}) \ni \underline{\phi}(\underline{x}) := \underline{\underline{T}}\,\underline{x} + \underline{b}$, with $\underline{\underline{T}} \in \mathbb{R}^{d\times d}$ and $\underline{b} \in \mathbb{R}^d$. We are going to prove that $\underline{\underline{\mathrm{G}}}_{\mathrm{c}}^{0}(\underline{\widehat{\pi}}(\underline{\phi})) = \underline{\underline{\nabla}}\,\underline{\phi} = \underline{\underline{T}}$. Recall that $\underline{\phi}_z := \underline{\pi}_z(\underline{\phi}) = \frac{1}{|z|}\int_z \underline{\phi} = (\underline{\phi})(\underline{x}_z)$ for $z = \mathrm{f}, \mathrm{c}$ where the last equality is ensured by the linearity of $\underline{\phi}$ and having chosen the entity barycenter $\underline{x}_z$. Then, $\underline{\phi}_{\mathrm{f}} - \underline{\phi}_{\mathrm{c}} = \underline{\underline{T}}(\underline{x}_{\mathrm{f}} - \underline{x}_{\mathrm{c}}) = \underline{\underline{\nabla}}\,\underline{\phi}(\underline{x}_{\mathrm{f}} - \underline{x}_{\mathrm{c}})$, and plugging this identity into (2.20b) and using (2.22), one gets $\underline{\underline{\mathrm{G}}}_{\mathrm{c}}^{0}(\underline{\widehat{\pi}}_{\mathrm{c}}(\underline{\phi})) = \underline{\underline{\nabla}}\,\underline{\phi}$. □

Moving to $\underline{\underline{\mathrm{G}}}_{\mathrm{c}}^{\mathrm{s}}$, this part of the gradient reconstruction embodies a stabilization relying on a first-order Taylor expansion of the form $\underline{v}_{\mathrm{f}} \approx \underline{v}_{\mathrm{c}} + \underline{\underline{\mathrm{G}}}_{\mathrm{c}}^{0}(\underline{\widehat{v}}_{\mathrm{c}})(\underline{x}_{\mathrm{f}} - \underline{x}_{\mathrm{c}})$. Let us give two useful properties of $\underline{\underline{\mathrm{G}}}_{\mathrm{c}}^{\mathrm{s}}$.

Corollary 2.20 - $\underline{\underline{\mathrm{G}}}_{\mathrm{c}}^{0}$ on $\underline{\mathbb{P}}^1(\mathrm{c})$. $\underline{\underline{\mathrm{G}}}_{\mathrm{c}}^{\mathrm{s}}$ *vanishes for affine functions. In particular, given the definitions* (2.20c) *of* $\underline{\underline{\mathrm{G}}}_{\mathrm{c}}^{0}$ *and* (2.18c) *of* $\underline{\widehat{\pi}}_{\mathrm{c}}$*, it holds*

$$\underline{\underline{\mathrm{G}}}_{\mathrm{c}}^{\mathrm{s}}(\underline{\widehat{\pi}}_{\mathrm{c}}(\underline{\phi})) = \underline{\underline{0}}, \qquad \forall \underline{\phi} \in \underline{\mathbb{P}}^1(\mathrm{c})\,,\ \forall \mathrm{c} \in \mathrm{C}\,. \tag{2.24}$$

Proof. One proceeds as in the proof of Lemma 2.19. Considering a generic affine function $\underline{\phi} \in \underline{\mathbb{P}}^1(\mathrm{c})$, one can write $\underline{\phi}_{\mathrm{f}} - \underline{\phi}_{\mathrm{c}} = \underline{\underline{\nabla}}\,\underline{\phi}(\underline{x}_{\mathrm{f}} - \underline{x}_{\mathrm{c}}) = \underline{\underline{\mathrm{G}}}_{\mathrm{c}}^{0}(\underline{\widehat{\pi}}_{\mathrm{c}}(\underline{\phi}))(\underline{x}_{\mathrm{f}} - \underline{x}_{\mathrm{c}})$ where the last equality comes from Lemma 2.19. It readily follows that $\underline{\underline{\mathrm{G}}}_{\mathrm{c}}^{\mathrm{s}}(\underline{\widehat{\pi}}_{\mathrm{c}}(\underline{\phi}))_{|\mathfrak{p}_{\mathrm{f,c}}} = \underline{\underline{0}}$ for all $\mathrm{f} \in \mathrm{F_c}$. □

Lemma 2.21 - Average of $\underline{\underline{\mathrm{G}}}_{\mathrm{c}}$. *The following holds true:*

$$\int_{\mathrm{c}} \underline{\underline{\mathrm{G}}}_{\mathrm{c}}^{\mathrm{s}}(\underline{\widehat{v}}_{\mathrm{c}}) = \underline{\underline{0}}, \qquad \forall \underline{\widehat{v}}_{\mathrm{c}} \in \underline{\widehat{\mathcal{U}}}_{\mathrm{c}}\,,\ \forall \mathrm{c} \in \mathrm{C}\,. \tag{2.25}$$

Consequently:

(i) $\underline{\underline{\mathrm{G}}}_{\mathrm{c}}^{0}$ *and* $\underline{\underline{\mathrm{G}}}_{\mathrm{c}}^{\mathrm{s}}$ *are* L^2*-orthogonal, in the sense that*

$$\int_{\mathrm{c}} \underline{\underline{\mathrm{G}}}_{\mathrm{c}}^{0}(\widehat{\underline{v}}_{\mathrm{c}}) : \underline{\underline{\mathrm{G}}}_{\mathrm{c}}^{\mathrm{s}}(\widehat{\underline{v}}_{\mathrm{c}}) = 0, \qquad \forall \widehat{\underline{v}}_{\mathrm{c}} \in \widehat{\underline{\mathcal{U}}}_{\mathrm{c}}\,,\ \forall \mathrm{c} \in \mathrm{C}\,. \tag{2.26}$$

(ii) $\underline{\underline{\mathrm{G}}}_{\mathrm{c}}^{0}$ *is the average of* $\underline{\underline{\mathrm{G}}}_{\mathrm{c}}$*:*

$$\frac{1}{|\mathrm{c}|}\int_{\mathrm{c}} \underline{\underline{\mathrm{G}}}_{\mathrm{c}}(\widehat{\underline{v}}_{\mathrm{c}}) = \underline{\underline{\mathrm{G}}}_{\mathrm{c}}^{0}(\widehat{\underline{v}}_{\mathrm{c}}), \qquad \forall\forall \widehat{\underline{v}}_{\mathrm{c}} \in \widehat{\underline{\mathcal{U}}}_{\mathrm{c}}\,,\ \forall \mathrm{c} \in \mathrm{C}\,. \tag{2.27}$$

Proof. (2.26) is readily obtained from (2.25) by recalling that $\underline{\underline{\mathrm{G}}}_{\mathrm{c}}^{0}$ is constant on a cell and observing that, for all $\mathrm{c} \in \mathrm{C}$ and all $\widehat{\underline{v}}_{\mathrm{c}} \in \widehat{\underline{\mathcal{U}}}_{\mathrm{c}}$

$$\int_{\mathrm{c}} \underline{\underline{\mathrm{G}}}_{\mathrm{c}}^{0}(\widehat{\underline{v}}_{\mathrm{c}}) : \underline{\underline{\mathrm{G}}}_{\mathrm{c}}^{\mathrm{s}}(\widehat{\underline{v}}_{\mathrm{c}}) = \underline{\underline{\mathrm{G}}}_{\mathrm{c}}^{0}(\widehat{\underline{v}}_{\mathrm{c}}) : \left(\int_{\mathrm{c}} \underline{\underline{\mathrm{G}}}_{\mathrm{c}}^{\mathrm{s}}(\widehat{\underline{v}}_{\mathrm{c}})\right) = 0\,. \tag{2.28}$$

Similarly, owing to the definition (2.20a) of $\underline{\underline{\mathrm{G}}}_{\mathrm{c}}$, the following holds for all $\widehat{\underline{v}}_{\mathrm{c}} \in \widehat{\underline{\mathcal{U}}}_{\mathrm{c}}$ and all $\mathrm{c} \in \mathrm{C}$,

$$\int_{\mathrm{c}} \underline{\underline{\mathrm{G}}}_{\mathrm{c}}(\widehat{\underline{v}}_{\mathrm{c}}) = \int_{\mathrm{c}} \underline{\underline{\mathrm{G}}}_{\mathrm{c}}^{0}(\widehat{\underline{v}}_{\mathrm{c}}) + \underline{\underline{\mathrm{G}}}_{\mathrm{c}}^{\mathrm{s}}(\widehat{\underline{v}}_{\mathrm{c}}) = \int_{\mathrm{c}} \underline{\underline{\mathrm{G}}}_{\mathrm{c}}^{0}(\widehat{\underline{v}}_{\mathrm{c}}) = |\mathrm{c}|\, \underline{\underline{\mathrm{G}}}_{\mathrm{c}}^{0}(\widehat{\underline{v}}_{\mathrm{c}})\,, \tag{2.29}$$

which proves (2.27). Hence, it is left to prove (2.25).

$\underline{\underline{\mathrm{G}}}_{\mathrm{c}}^{\mathrm{s}}(\underline{v}_{\mathrm{c}})$ being piecewise constant on the subpyramids, one gets

$$\int_{\mathrm{c}} \underline{\underline{\mathrm{G}}}_{\mathrm{c}}^{\mathrm{s}}(\widehat{\underline{v}}_{\mathrm{c}}) = \sum_{\mathrm{f}\in\mathrm{F}_{\mathrm{c}}} \int_{\mathfrak{p}_{\mathrm{f,c}}} \underline{\underline{\mathrm{G}}}_{\mathrm{c}}^{\mathrm{s}}(\widehat{\underline{v}}_{\mathrm{c}})_{|\mathfrak{p}_{\mathrm{f,c}}} = \sum_{\mathrm{f}\in\mathrm{F}_{\mathrm{c}}} |\mathrm{f}| \left((\underline{v}_{\mathrm{f}} - \underline{v}_{\mathrm{c}}) - \underline{\underline{\mathrm{G}}}_{\mathrm{c}}^{0}(\widehat{\underline{v}}_{\mathrm{c}})(\underline{x}_{\mathrm{f}} - \underline{x}_{\mathrm{c}})\right) \otimes \underline{n}_{\mathrm{fc}}\,, \tag{2.30}$$

where the last equality is obtained using the definition (2.20c) and the fact that $\underline{\underline{\mathrm{G}}}_{\mathrm{c}}^{\mathrm{s}}$ is constant on each subpyramid. Now, (2.25) is readily obtained by recalling the definition of $\underline{\underline{\mathrm{G}}}_{\mathrm{c}}^{0}$, see (2.20b), and using (2.22). □

Proposition 2.22 - Exactness of $\underline{\underline{\mathrm{G}}}_{\mathrm{c}}$ on $\underline{\mathbb{P}}^1(\mathrm{c})$. *Given the definitions* (2.20a) *of* $\underline{\underline{\mathrm{G}}}_{\mathrm{c}}$ *and* (2.18c) *of* $\widehat{\underline{\pi}}_{\mathrm{c}}$*, it holds*

$$\underline{\underline{\mathrm{G}}}_{\mathrm{c}} \circ \widehat{\underline{\pi}}_{\mathrm{c}} = \underline{\underline{\nabla}} \quad \text{on } \underline{\mathbb{P}}^1(\mathrm{c})\,. \tag{2.31}$$

Proof. The result is readily obtained from Lemma 2.19 and Corollary 2.20. □

Let us define the following discrete norm on $\widehat{\underline{\mathcal{U}}}_{\mathrm{h},0}$:

$$\|\widehat{\underline{v}}_{\mathrm{h}}\|_{1,\mathrm{h}}^2 := \sum_{\mathrm{c}\in\mathrm{C}} \|\widehat{\underline{v}}_{\mathrm{c}}\|_{1,\mathrm{c}}^2 \qquad \|\widehat{\underline{v}}_{\mathrm{h}}\|_{1,\mathrm{c}}^2 := \sum_{\mathrm{f}\in\mathrm{F}_{\mathrm{c}}} \frac{1}{\mathrm{h}_{\mathrm{c}}} \|\underline{v}_{\mathrm{f}} - \underline{v}_{\mathrm{c}}\|_{\underline{L}^2(\mathrm{f})}^2 = \sum_{\mathrm{f}\in\mathrm{F}_{\mathrm{c}}} \frac{1}{\mathrm{h}_{\mathrm{c}}} |\mathrm{f}|\, |\underline{v}_{\mathrm{f}} - \underline{v}_{\mathrm{c}}|_2^2\,, \tag{2.32}$$

where $|\underline{x}|_2^2 := \sum_{i=1}^{d} x_i^2$ is the Euclidean norm in $\mathbb{R}^d$ and h_{c} has been introduced in Definition 2.4. The last equality at the right-hand side of (2.32) is obtained simply by observing that $(\underline{v}_{\mathrm{f}} - \underline{v}_{\mathrm{c}})$ is constant on each face.

Let us state the main result concerning the gradient reconstruction operator $\underline{\underline{\mathrm{G}}}_{\mathrm{h}}$.

Lemma 2.23 - Stability of $\underline{\underline{\mathrm{G}}}_{\mathrm{h}}$. *There exists* $\delta > 0$*, independent of* $\mathrm{h} \in \mathrm{H}$*, such that, for all* $\widehat{\underline{v}}_{\mathrm{c}} \in \widehat{\underline{\mathcal{U}}}_{\mathrm{c}}$ *and all* $\mathrm{c} \in \mathrm{C}$

$$\delta\, \|\widehat{\underline{v}}_{\mathrm{c}}\|_{1,\mathrm{c}}^2 \le \left\|\underline{\underline{\mathrm{G}}}_{\mathrm{c}}(\widehat{\underline{v}}_{\mathrm{c}})\right\|_{\underline{\underline{L}}^2(\mathrm{c})}^2 \le \delta^{-1}\, \|\widehat{\underline{v}}_{\mathrm{c}}\|_{1,\mathrm{c}}^2\,. \tag{2.33}$$

Consequently, summing over all the mesh cell

$$\delta\, \|\widehat{\underline{v}}_{\mathrm{h}}\|_{1,\mathrm{h}}^2 \le \left\|\underline{\underline{\mathrm{G}}}_{\mathrm{h}}(\widehat{\underline{v}}_{\mathrm{h}})\right\|_{\underline{\underline{L}}^2(\Omega)}^2 \le \delta^{-1}\, \|\widehat{\underline{v}}_{\mathrm{h}}\|_{1,\mathrm{h}}^2\,. \tag{2.34}$$

Proof. A similar result has been proved (for the scalar-valued case) in Eymard *et al.* (2010, Lemma 4.1) for the HFV/SUSHI framework and, up to the stabilization parameter, can be adapted to the CDO-Fb case as well. However, for the sake of completeness, we outline the proof. We shall use the notation $a \lesssim b$ to mean that $a \leq Cb$, where $C > 0$ is constant independent of h but possibly depending on the mesh regularity parameters.

We start by proving the second inequality of (2.33). Owing to the orthogonality of $\underline{\underline{\mathrm{G}}}_{\mathrm{c}}^{0}$ and $\underline{\underline{\mathrm{G}}}_{\mathrm{c}}^{\mathrm{s}}$, see (2.26), we have

$$\left\|\underline{\underline{\mathrm{G}}}_{\mathrm{c}}(\widehat{\underline{v}}_{\mathrm{c}})\right\|_{\underline{\underline{L}}^2(\mathrm{c})}^2 = \left\|\underline{\underline{\mathrm{G}}}_{\mathrm{c}}^{0}(\widehat{\underline{v}}_{\mathrm{c}}) + \underline{\underline{\mathrm{G}}}_{\mathrm{c}}^{\mathrm{s}}(\widehat{\underline{v}}_{\mathrm{c}})\right\|_{\underline{\underline{L}}^2(\mathrm{c})}^2 = \left\|\underline{\underline{\mathrm{G}}}_{\mathrm{c}}^{0}(\widehat{\underline{v}}_{\mathrm{c}})\right\|_{\underline{\underline{L}}^2(\mathrm{c})}^2 + \left\|\underline{\underline{\mathrm{G}}}_{\mathrm{c}}^{\mathrm{s}}(\widehat{\underline{v}}_{\mathrm{c}})\right\|_{\underline{\underline{L}}^2(\mathrm{c})}^2 . \tag{2.35}$$

Recalling the definition (2.20b) of $\underline{\underline{\mathrm{G}}}_{\mathrm{c}}^{0}$, one can write

$$\begin{aligned} \left\|\underline{\underline{\mathrm{G}}}_{\mathrm{c}}^{0}(\widehat{\underline{v}}_{\mathrm{c}})\right\|_{\underline{\underline{L}}^2(\mathrm{c})}^2 &\leq \frac{1}{|\mathrm{c}|}\left(\sum_{\mathrm{f}\in\mathrm{F}_{\mathrm{c}}} |\mathrm{f}|\right)\left(\sum_{\mathrm{f}\in\mathrm{F}_{\mathrm{c}}} |\mathrm{f}|\,|(\underline{v}_{\mathrm{f}} - \underline{v}_{\mathrm{c}}) \otimes \underline{n}_{\mathrm{fc}}|^2\right) \\ &\lesssim \frac{1}{\mathrm{h}_{\mathrm{c}}}\left(\sum_{\mathrm{f}\in\mathrm{F}_{\mathrm{c}}} |\mathrm{f}|\,|\underline{v}_{\mathrm{f}} - \underline{v}_{\mathrm{c}}|^2\right) = \|\widehat{\underline{v}}_{\mathrm{c}}\|_{1,\mathrm{c}}^2 , \end{aligned} \tag{2.36}$$

where we used on the first line the Cauchy–Schwarz inequality and, on the second line, the identity

$$|\underline{a} \otimes \underline{b}| = |\underline{a}|\,|\underline{b}| \qquad \forall \underline{a}, \underline{b} \in \mathbb{R}^d , \tag{2.37}$$

and geometrical inequalities which result from the mesh regularity assumptions in Definition 2.9. Let $\underline{\tilde{\mathrm{e}}}_{\mathrm{f,c}} := \underline{x}_{\mathrm{f}} - \underline{x}_{\mathrm{c}}$. Addressing now $\underline{\underline{\mathrm{G}}}_{\mathrm{c}}^{\mathrm{s}}$, we proceed similarly to the derivation of (2.36) and get

$$\left\|\underline{\underline{\mathrm{G}}}_{\mathrm{c}}^{\mathrm{s}}(\widehat{\underline{v}}_{\mathrm{c}})\right\|_{\underline{\underline{L}}^2(\mathrm{c})}^2 \leq \sum_{\mathrm{f}\in\mathrm{F}_{\mathrm{c}}} \frac{\beta^2 d\,|\mathrm{f}|}{\left|\underline{\tilde{\mathrm{e}}}_{\mathrm{f,c}}\right|}\left|(\underline{v}_{\mathrm{f}} - \underline{v}_{\mathrm{c}}) - \underline{\underline{\mathrm{G}}}_{\mathrm{c}}^{0}(\widehat{\underline{v}}_{\mathrm{c}})\underline{\tilde{\mathrm{e}}}_{\mathrm{f,c}}\right|^2 \lesssim \|\widehat{\underline{v}}_{\mathrm{c}}\|_{1,\mathrm{c}}^2 \tag{2.38}$$

where we used (2.37), the geometric identity $|\mathrm{f}|\left|\underline{\tilde{\mathrm{e}}}_{\mathrm{f,c}}\right| = d\,|\mathfrak{p}_{\mathrm{f,c}}|$ for all $\mathrm{c} \in \mathrm{C}$ and all $\mathrm{f} \in \mathrm{F}_{\mathrm{c}}$, and (2.36). Putting (2.36) and (2.38) together, one obtains the second inequality of (2.33).

Now, we address the first inequality. Owing to the definition (2.20c) of $\underline{\underline{\mathrm{G}}}_{\mathrm{c}}^{\mathrm{s}}$, one has, for all $\mathrm{f} \in \mathrm{F}_{\mathrm{c}}$,

$$(\underline{v}_{\mathrm{f}} - \underline{v}_{\mathrm{c}}) \otimes \underline{n}_{\mathrm{fc}} = \frac{1}{\beta}\underline{\underline{\mathrm{G}}}_{\mathrm{c}}^{\mathrm{s}}(\widehat{\underline{v}}_{\mathrm{c}}) + (\underline{\underline{\mathrm{G}}}_{\mathrm{c}}^{0}(\widehat{\underline{v}}_{\mathrm{c}})\underline{\tilde{\mathrm{e}}}_{\mathrm{f,c}}) \otimes \underline{n}_{\mathrm{fc}} . \tag{2.39}$$

Let us consider the module of (2.39). Using the triangle inequality, (2.37), the definition (2.8) of h_{c}, and the mesh regularity assumptions given in Definition 2.9, we obtain

$$|\underline{v}_{\mathrm{f}} - \underline{v}_{\mathrm{c}}|^2 \lesssim \mathrm{h}_{\mathrm{c}}^2\left(\left|\underline{\underline{\mathrm{G}}}_{\mathrm{c}}^{\mathrm{s}}(\widehat{\underline{v}}_{\mathrm{c}})\right|^2 + \left|\underline{\underline{\mathrm{G}}}_{\mathrm{c}}^{0}(\widehat{\underline{v}}_{\mathrm{c}})\right|^2\right) = \mathrm{h}_{\mathrm{c}}^2\left|\underline{\underline{\mathrm{G}}}_{\mathrm{c}}(\widehat{\underline{v}}_{\mathrm{c}})_{|\mathfrak{p}_{\mathrm{f,c}}}\right|^2 , \quad \forall \mathrm{f} \in \mathrm{F}_{\mathrm{c}} . \tag{2.40}$$

Now, summing (2.40) over all $\mathrm{f} \in \mathrm{F}_{\mathrm{c}}$ gives

$$\|\widehat{\underline{v}}_{\mathrm{c}}\|_{1,\mathrm{c}}^2 \lesssim \frac{1}{\mathrm{h}_{\mathrm{c}}}(\#\mathrm{F}_{\mathrm{c}}\mathrm{h}_{\mathrm{c}}^{d-1})\mathrm{h}^2\left|\underline{\underline{\mathrm{G}}}_{\mathrm{c}}(\widehat{\underline{v}}_{\mathrm{c}})_{|\mathfrak{p}_{\mathrm{f,c}}}\right|^2 \lesssim \mathrm{h}_{\mathrm{c}}^d\left|\underline{\underline{\mathrm{G}}}_{\mathrm{c}}(\widehat{\underline{v}}_{\mathrm{c}})\right|^2 \lesssim \left\|\underline{\underline{\mathrm{G}}}_{\mathrm{c}}(\widehat{\underline{v}}_{\mathrm{c}})\right\|_{\underline{\underline{L}}^2(\mathrm{c})}^2 , \tag{2.41}$$

which proves the first inequality in (2.33) and thus concludes the proof. □

Remark 2.24 - Alternative length scale. For the sake of simplicity, we used the diameter h_{c} of the cell $\mathrm{c} \in \mathrm{C}$ in $\|\cdot\|_{1,\mathrm{h}}$. Owing to the mesh regularity (see Definition 2.9), any other local equivalent length scale can be chosen. One can consider for instance the norm

$$\|\widehat{\underline{v}}_{\mathrm{h}}\|_{1,\tilde{\mathrm{e}}}^2 := \sum_{\mathrm{c}\in\mathrm{C}}\sum_{\mathrm{f}\in\mathrm{F}_{\mathrm{c}}} \frac{1}{\left|\underline{\tilde{\mathrm{e}}}_{\mathrm{f,c}}\right|}\|\underline{v}_{\mathrm{f}} - \underline{v}_{\mathrm{c}}\|_{\underline{L}^2(\mathrm{f})}^2 , \tag{2.42}$$

where $\underline{\tilde{\mathrm{e}}}_{\mathrm{f,c}} := \underline{x}_{\mathrm{f}} - \underline{x}_{\mathrm{c}}$, and $\underline{x}_z$ is the barycenter of the mesh entity (face or cell) z. This norm fits the spirit to the CDO-Fb framework of Bonelle (2014) and Bonelle and Ern (2014). ◇

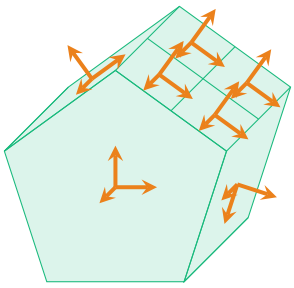


Figure 2.4 – The discrete divergence is cell-wise constant (the whole cell is colored). Only face-based DoFs (in orange) of the velocity are used (see Remark 2.27).

2.4 Velocity-pressure coupling

The divergence plays an essential role when considering the NSE. In fact, classically, in the related variational formulation, continuous or discrete, it is this operator that ensures the velocity-pressure coupling and gives the characteristic saddle-point structure to the problem.

2.4.1 Main definitions and basic properties

Definition 2.25 - Local discrete divergence. Consider a cell $\mathrm{c} \in \mathrm{C}$. The local discrete divergence operator within the CDO framework is defined as follows:

$$\begin{aligned} \mathrm{D_c}\colon \quad \widehat{\underline{\mathcal{U}}}_\mathrm{c} &\to \mathcal{P}_\mathrm{c} \\ \widehat{\underline{u}}_\mathrm{c} &\mapsto \mathrm{tr}(\underline{\underline{\mathrm{G}}}^0_\mathrm{c}(\widehat{\underline{u}}_\mathrm{c})) = \frac{1}{|\mathrm{c}|}\sum_{\mathrm{f}\in\mathrm{F_c}} |\mathrm{f}|\,(\underline{u}_\mathrm{f} - \underline{u}_\mathrm{c})\cdot\underline{n}_\mathrm{fc}\,. \quad \circ \end{aligned} \tag{2.43}$$

Definition 2.26 - Global discrete divergence. As it was the case for the velocity gradient, the global version of the divergence operator is defined as the collection of the local discrete divergences:

$$\begin{aligned} \mathrm{D_h}\colon \quad \widehat{\underline{\mathcal{U}}}_\mathrm{h} &\to \mathcal{P}_\mathrm{h} \\ \widehat{\underline{u}}_\mathrm{h} &\mapsto (\mathrm{D_c}(\widehat{\underline{u}}_\mathrm{c}))_{\mathrm{c}\in\mathrm{C}}\,. \end{aligned} \tag{2.44}$$

$\circ$

Remark 2.27 - Simplified formulation. Even if, formally, the operator $\mathrm{D_c}$ involves all of the DoFs the velocity in the cell c and its faces, it is readily seen that only the face-based ones are relevant:

$$\mathrm{D_c}(\widehat{\underline{u}}_\mathrm{c}) = \frac{1}{|\mathrm{c}|}\left(\sum_{\mathrm{f}\in\mathrm{F_c}} |\mathrm{f}|\,\underline{u}_\mathrm{f}\cdot\underline{n}_\mathrm{fc} - \underline{u}_\mathrm{c}\cdot\sum_{\mathrm{f}\in\mathrm{F_c}} |\mathrm{f}|\,\underline{n}_\mathrm{fc}\right) = \frac{1}{|\mathrm{c}|}\sum_{\mathrm{f}\in\mathrm{F_c}} |\mathrm{f}|\,\underline{u}_\mathrm{f}\cdot\underline{n}_\mathrm{fc}\,, \tag{2.45}$$

where the last equality follows from (2.4). This is the formulation that we will consider in what follows and also in our implementations. See Fig. 2.4 for an illustration of the construction of the divergence. $\diamond$

Remark 2.28 - Divergence theorem. The design of the discrete operator $\mathrm{D_c}$ can be related to the divergence theorem and this link is easily drawn from (2.45). In fact, an alternative design approach could have started from the divergence theorem $\int_\mathrm{c} \underline{\nabla}\cdot\underline{u} = \sum_{\mathrm{f}\in\mathrm{F_c}} \int_\mathrm{f} \underline{u}\cdot\underline{n}_\mathrm{fc}$ and choosing to approximate $\underline{u}$ at this stage, which would have led us to (2.45). $\diamond$

Remark 2.29 - Link to the gradient reconstruction. The local divergence operator stems from the discrete local gradient reconstruction operator similarly to what happens for the continuous differential operator, see (2.43). Notice however that only the consistent part of $\underline{\underline{\mathrm{G}}}_{\mathrm{c}}$, namely $\underline{\underline{\mathrm{G}}}_{\mathrm{c}}^{0}$, is used. However, if the stability part were to be used as well, its action will vanish as soon as the divergence is tested against a cell-wise constant function, as is the case with any discrete pressure. In fact, considering a constant test function $q_{\mathrm{c}} \in \mathbb{P}^0(\mathrm{c})$, one obtains

$$\int_{\mathrm{c}} q_{\mathrm{c}} \operatorname{tr}(\underline{\underline{\mathrm{G}}}_{\mathrm{c}}^{\mathrm{s}}(\widehat{\underline{u}}_{\mathrm{c}})) = q_{\mathrm{c}} \int_{\mathrm{c}} \operatorname{tr}(\underline{\underline{\mathrm{G}}}_{\mathrm{c}}^{\mathrm{s}}(\widehat{\underline{u}}_{\mathrm{c}})) = 0\,, \tag{2.46}$$

owing to the linearity of the trace and to (2.25). ◇

Remark 2.30 - Discrete divergence and reduction. The commutation between the divergence and the reduction by projection can be easily shown: For all $\underline{v} \in \underline{H}^1(\mathrm{c})$, we have

$$\pi_{\mathrm{c}}(\underline{\nabla} \cdot \underline{v}) = \frac{1}{|\mathrm{c}|} \int_{\mathrm{c}} \underline{\nabla} \cdot \underline{v} = \frac{1}{|\mathrm{c}|} \sum_{\mathrm{f} \in \mathrm{F_c}} \int_{\mathrm{f}} \underline{v} \cdot \underline{n}_{\mathrm{fc}} = \mathrm{D_c}(\widehat{\underline{\pi}}_{\mathrm{c}}(\underline{v}))\,, \tag{2.47}$$

owing to the divergence theorem, the hypothesis about planar faces, and finally (2.45). The relation (2.47) can be extended to the global operator $\mathrm{D_h}$ when it is tested against a function $q \in \mathbb{P}^0(\mathrm{C})$. As a matter of fact, we have for all $\underline{v} \in \underline{H}^1(\Omega)$,

$$\int_{\Omega} q \underline{\pi}_{\mathrm{h}}(\underline{\nabla} \cdot \underline{v}) = \sum_{\mathrm{c} \in \mathrm{C}} \underline{\pi}_{\mathrm{c}}(\underline{\nabla} \cdot \underline{v}) \int_{\mathrm{c}} q = \sum_{\mathrm{c} \in \mathrm{C}} \mathrm{D_c}(\widehat{\underline{\pi}}_{\mathrm{c}}(\underline{v})) \int_{\mathrm{c}} q = \int_{\Omega} \mathrm{D_h}(\widehat{\underline{\pi}}_{\mathrm{h}}(\underline{v})) q\,. \tag{2.48}$$

This proves that

$$\pi_{\mathrm{h}}(\underline{\nabla} \cdot \underline{v}) = \mathrm{D_h}(\widehat{\underline{\pi}}_{\mathrm{h}}(\underline{v}))\,, \qquad \forall \underline{v} \in \underline{H}^1(\Omega)\,. \quad \diamond \tag{2.49}$$

Remark 2.31 - $\mathrm{D_c}$ and the original CDO divergence of Bonelle (2014) and Bonelle and Ern (2014). The discrete velocity divergence $\mathrm{D_c}$ may be bridged to the original CDO divergence operator $\mathsf{DIV_c}$ from Bonelle and Ern (2014) and Bonelle (2014). Given a face-defined function $\phi_{\mathrm{F_c}} = (\phi_{\mathrm{f}})_{\mathrm{f} \in \mathrm{F_c}} \in \mathbb{P}^0(\mathrm{F_c})$, the definition of $\mathsf{DIV_c}$ is

$$\mathsf{DIV_c}: \quad \mathcal{F}_{\mathrm{c}} := \mathbb{P}^0(\mathrm{F_c}) \to \mathbb{P}^0(\mathrm{c}), \qquad \mathsf{DIV_c}(\phi_{\mathrm{F_c}}) := \sum_{\mathrm{f} \in \mathrm{F_c}} \iota_{\mathrm{f,c}} \phi_{\mathrm{f}}\,, \tag{2.50}$$

and the reduction for a generic function $\underline{\phi} \in \underline{H}^s(\Omega)$, $s > \frac{1}{2}$, is

$$\mathsf{R}_{\mathcal{F}_{\mathrm{c}}}: \quad \underline{H}^s(\Omega) \to \mathcal{F}_{\mathrm{c}}\,, \qquad \mathsf{R}_{\mathcal{F}_{\mathrm{c}}}(\underline{\phi})_{|\mathrm{f}} = \int_{\mathrm{f}} \underline{\phi} \cdot \underline{n}_{\mathrm{f}}\,, \qquad \forall\, \mathrm{f} \in \mathrm{F_c}\,. \tag{2.51}$$

Then we have

$$\mathsf{DIV_c}(\mathsf{R}_{\mathcal{F}_{\mathrm{c}}}(\underline{\phi})) = \sum_{\mathrm{f} \in \mathrm{F_c}} \int_{\mathrm{f}} (\iota_{\mathrm{f,c}} \underline{n}_{\mathrm{f}}) \cdot \underline{\phi}_{\mathrm{f}} = |\mathrm{c}|\, \mathrm{D_c}(\widehat{\underline{\pi}}_{\mathrm{c}}(\underline{\phi}))\,, \tag{2.52}$$

where we used the fact that the faces are planar and that $\underline{n}_{\mathrm{fc}} = \iota_{\mathrm{f,c}} \underline{n}_{\mathrm{f}}$. ◇

Remark 2.32 - Comparison with other schemes. The same divergence operator as in (2.43) is found in the HMM framework of Droniou *et al.* (2015). It was already shown in Bonelle (2014) and Bonelle and Ern (2014) how the CDO and the HMM gradient operator are equivalent up to a stabilization, whose action, however, vanishes when considering the divergence. Similar operators may be found in other frameworks to which the CDO framework has already been linked. An early version of the MFD (Beirão da Veiga *et al.*, 2009b), which later became part of the HMM, relies on an operator which considers only the velocity fluxes. Moreover, the divergence operator proposed for the HHO method in Di Pietro *et al.* (2016) is equivalent to $\mathrm{D_c}$ when specified for the lowest-order case, $k = 0$. ◇

2.4.2 Inf-sup condition

The inf-sup condition (see Proposition 1.3) plays a paramount role in the variational formulation of the NSE. Recall that this condition means that there exists $\beta > 0$ such that

$$\inf_{q\in L^2_*(\Omega)} \sup_{\underline{v}\in \underline{H}^1_0(\Omega)} \frac{b(\underline{v}, q)}{\|q\|_{L^2_*(\Omega)} \|\underline{v}\|_{\underline{H}^1_0(\Omega)}} \geq \beta\,, \tag{2.53}$$

where for all $\underline{v} \in \underline{H}^1(\Omega)$ and all $q \in L^2(\Omega)$, $b(\underline{v}, q) := -\int_\Omega q\underline{\nabla}\cdot\underline{v}$. We prove in this section that an analogous inequality holds for the CDO discrete divergence $\mathrm{D_h}$ and the discrete functional spaces $\widehat{\underline{\mathcal{U}}}_{\mathrm{h},0}$ and $\mathcal{P}_{\mathrm{h},*}$ defined in (2.16) and (2.17). Recall that $\widehat{\underline{\mathcal{U}}}_{\mathrm{h},0}$ is equipped with the following norm defined in (2.32):

$$\|\widehat{\underline{v}}_\mathrm{h}\|^2_{1,\mathrm{h}} := \sum_{\mathrm{c}\in\mathrm{C}} \sum_{\mathrm{f}\in\mathrm{F_c}} \frac{1}{\mathrm{h_c}} |\mathrm{f}|\, |\underline{v}_\mathrm{f} - \underline{v}_\mathrm{c}|^2_2\,, \tag{2.54}$$

whereas, a norm with which to equip $\mathcal{P}_{\mathrm{h},*}$ is

$$\|q_\mathrm{h}\|^2_\mathrm{h} := \sum_{\mathrm{c}\in\mathrm{C}} |\mathrm{c}|\, q_\mathrm{c}^2 = \|q_\mathrm{h}\|^2_{L^2(\Omega)}\,. \tag{2.55}$$

Moreover, we define the discrete bilinear form $b_\mathrm{h}\colon \widehat{\underline{\mathcal{U}}}_{\mathrm{h},0} \times \mathcal{P}_{\mathrm{h},*} \to \mathbb{R}$ associated with the velocity-pressure coupling as follows:

$$b_\mathrm{h}(\widehat{\underline{v}}_\mathrm{h}, q_\mathrm{h}) := \sum_{\mathrm{c}\in\mathrm{C}} b_\mathrm{c}(\widehat{\underline{v}}_\mathrm{c}, q_\mathrm{c})\,, \qquad b_\mathrm{c}(\widehat{\underline{v}}_\mathrm{c}, q_\mathrm{c}) := -\int_\mathrm{c} \mathrm{D_c}(\widehat{\underline{v}}_\mathrm{c}) q_\mathrm{c} = -\,|\mathrm{c}|\,\mathrm{D_c}(\widehat{\underline{v}}_\mathrm{c}) q_\mathrm{c}\,. \tag{2.56}$$

The bilinear form $b_\mathrm{h}(\cdot,\cdot)$ is the discrete counterpart of the continuous bilinear form $b(\cdot,\cdot)$ used in the inf-sup condition (2.53).

Lemma 2.33 - CDO-Fb inf-sup condition. *There exists $\beta_* > 0$ such that, for all* $\mathrm{h} \in \mathrm{H}$,

$$\inf_{q_\mathrm{h}\in\mathcal{P}_{\mathrm{h},*}} \sup_{\widehat{\underline{v}}_\mathrm{h}\in\widehat{\underline{\mathcal{U}}}_{\mathrm{h},0}} \frac{b_\mathrm{h}(\widehat{\underline{v}}_\mathrm{h}, q_\mathrm{h})}{\|q_\mathrm{h}\|_\mathrm{h}\, \|\widehat{\underline{v}}_\mathrm{h}\|_{1,\mathrm{h}}} \geq \beta_*\,. \tag{2.57}$$

Proof. A proof of (2.57) for the HHO framework can be found in Di Pietro *et al.* (2016) and it can adapted to the CDO-Fb case by setting $k = 0$ for the polynomial degree in HHO. For the sake of completeness, we outline the proof. The main idea is to start from the continuous version of the inf-sup condition (2.53) and choose a suitable projection (also called Fortin operator) to recover it at the discrete level.

Since $\mathcal{P}_{\mathrm{h},*} \subset L^2_*(\Omega)$ (cf. (2.17)), using (2.53), we infer that for all $q_\mathrm{h} \in \mathcal{P}_{\mathrm{h},*}$, and all $\mathrm{h} \in \mathrm{H}$, there exists $\underline{v}_{q_\mathrm{h}} \in \underline{H}^1_0(\Omega)$ satisfying

$$\underline{\nabla}\cdot\underline{v}_{q_\mathrm{h}} = -q_\mathrm{h} \quad \text{and} \quad \beta \left\|\underline{\underline{\nabla}}\,\underline{v}_{q_\mathrm{h}}\right\|_{\underline{\underline{L}}^2(\Omega)} \leq \|q_\mathrm{h}\|_{L^2(\Omega)} = \|q_\mathrm{h}\|_\mathrm{h}\,. \tag{2.58}$$

The projection of $\underline{v}_{q_\mathrm{h}}$ onto the discrete space $\widehat{\underline{\mathcal{U}}}_{\mathrm{h},0}$ is a good candidate to satisfy (2.53). Let us set $\widehat{\underline{v}}_{q_\mathrm{h}} := \Big((\underline{v}_\mathrm{f} := \underline{\pi}_\mathrm{f}(\underline{v}_{q_\mathrm{h}}))_{\mathrm{f}\in\mathrm{F}}, (\underline{v}_\mathrm{c} := \underline{\pi}_\mathrm{c}(\underline{v}_{q_\mathrm{h}}))_{\mathrm{c}\in\mathrm{C}}\Big)$. Notice that $\widehat{\underline{v}}_{q_\mathrm{h}} \in \widehat{\underline{\mathcal{U}}}_{\mathrm{h},0}$ since $\underline{v}_{q_\mathrm{h}} \in \underline{H}^1_0(\Omega)$. One has

$$b_\mathrm{h}(\widehat{\underline{v}}_{q_\mathrm{h}}, q_\mathrm{h}) = -\sum_{\mathrm{c}\in\mathrm{C}} |\mathrm{c}|\,\mathrm{D_c}(\widehat{\underline{v}}_{q_\mathrm{h},\mathrm{c}}) q_\mathrm{c} = -\sum_{\mathrm{c}\in\mathrm{C}} \int_\mathrm{c} q_\mathrm{c}\underline{\nabla}\cdot\underline{v}_{q_\mathrm{h}} = -\int_\Omega \underline{\nabla}\cdot\underline{v}_{q_\mathrm{h}} q_\mathrm{h} = \|q_\mathrm{h}\|^2_\mathrm{h}\,, \tag{2.59}$$

where we used the definition (2.56) of $b_\mathrm{h}(\cdot,\cdot)$, the commutation between the discrete divergence and the projection (see Remark 2.30), and the first identity in (2.58).

The next step is to investigate the discrete norm of $\underline{\widehat{v}}_{q_\mathrm{h}}$. We have

$$\begin{aligned}
\left\|\underline{\widehat{v}}_{q_\mathrm{h}}\right\|_{1,\mathrm{h}}^2 &= \sum_{\mathrm{c}\in\mathrm{C}}\sum_{\mathrm{f}\in\mathrm{F_c}} \frac{1}{\mathrm{h_c}} \left\|\pi_\mathrm{f}(\underline{v}_{q_\mathrm{h}}) - \pi_\mathrm{c}(\underline{v}_{q_\mathrm{h}})\right\|_{\underline{L}^2(\mathrm{f})}^2 \\
&= \sum_{\mathrm{c}\in\mathrm{C}}\sum_{\mathrm{f}\in\mathrm{F_c}} \frac{1}{\mathrm{h_c}} \left\|\underline{\pi}_\mathrm{f}\left(\underline{v}_{q_\mathrm{h}} - \underline{\pi}_\mathrm{c}(\underline{v}_{q_\mathrm{h}})\right)\right\|_{\underline{L}^2(\mathrm{f})}^2 \\
&\le \sum_{\mathrm{c}\in\mathrm{C}}\sum_{\mathrm{f}\in\mathrm{F_c}} \frac{1}{\mathrm{h_c}} \left\|\underline{v}_{q_\mathrm{h}} - \underline{\pi}_\mathrm{c}(\underline{v}_{q_\mathrm{h}})\right\|_{\underline{L}^2(\mathrm{f})}^2 \\
&\le \sum_{\mathrm{c}\in\mathrm{C}} C_1 \left\|\underline{\underline{\nabla}}\,\underline{v}_{q_\mathrm{h}}\right\|_{\underline{L}^2(\mathrm{c})}^2 = C_1 \left\|\underline{\underline{\nabla}}\,\underline{v}_{q_\mathrm{h}}\right\|_{\underline{L}^2(\Omega)}^2 \\
&\le C_1\beta^{-2}\left\|q_\mathrm{h}\right\|_\mathrm{h}^2 ,
\end{aligned} \tag{2.60}$$

where we used in the first line the fact that the projection over a cell is constant over a face and the linearity of the projection, the $L^2(\mathrm{f})$-stability of the projection $\underline{\pi}_\mathrm{f}$ at line two, the multiplicative trace inequality from Di Pietro and Ern (2011, Lemma 1.49) combined with the local Poincaré inequality at line three, and finally the bound in (2.58) at line four. The constant C_1 does not depend on $\mathrm{h}\in\mathrm{H}$.

Now putting (2.59) and (2.60) together, we infer that

$$\begin{aligned}
\left\|q_\mathrm{h}\right\|_\mathrm{h}^2 = b_\mathrm{h}(\underline{\widehat{v}}_{q_\mathrm{h}}, q_\mathrm{h}) &= \frac{b_\mathrm{h}(\underline{\widehat{v}}_{q_\mathrm{h}}, q_\mathrm{h})}{\left\|\underline{\widehat{v}}_{q_\mathrm{h}}\right\|_{1,\mathrm{h}}} \left\|\underline{\widehat{v}}_{q_\mathrm{h}}\right\|_{1,\mathrm{h}} \\
&\le \left(\sup_{\underline{\widehat{v}}_\mathrm{h}\in\underline{\widehat{\mathcal{U}}}_{\mathrm{h},0}} \frac{b_\mathrm{h}(\underline{\widehat{v}}_\mathrm{h}, q_\mathrm{h})}{\left\|\underline{\widehat{v}}_\mathrm{h}\right\|_{1,\mathrm{h}}}\right) C_1^{\frac{1}{2}}\beta^{-1}\left\|q_\mathrm{h}\right\|_\mathrm{h} .
\end{aligned} \tag{2.61}$$

The sought relation is obtained by dividing both sides by $\left\|q_\mathrm{h}\right\|_\mathrm{h}$ and setting $\beta_* := \beta C_1^{-\frac{1}{2}}$. □

2.5 Scalar-valued advection and vector-valued convection

The scalar-valued advection operator (see Section 2.5.1) is at the core of the development of the operator which will be used to deal with the convection term in the NSE (see Section 2.5.2). That is why the two operators are discussed in the same section.

2.5.1 Scalar-valued advection

Consider a Lipschitz-continuous advective field $\underline{\beta}$ with bounded divergence, i.e.,

$$\underline{\beta} \in \underline{\mathrm{Lip}}(\overline{\Omega}) := \mathrm{Lip}(\overline{\Omega};\mathbb{R}^d) \quad \text{and} \quad \underline{\nabla}\cdot\underline{\beta} \in L^\infty(\Omega) . \tag{2.62}$$

Let $\partial\Omega^\pm := \left\{\underline{x}\in\partial\Omega \,|\, \underline{\beta}\cdot\underline{n}_{\partial\Omega} \gtrless 0\right\}$, and consider the following hybrid functional subspaces, which are the scalar counterpart of (2.10) and (2.11):

$$\begin{aligned}
\widehat{\mathcal{S}}_\mathrm{c} &:= \bigtimes_{\mathrm{f}\in\mathrm{F_c}} \mathcal{S}_\mathrm{f} \times \mathcal{S}_\mathrm{c} \equiv \mathbb{R}^{\#\mathrm{F_c}+1} \qquad \forall \mathrm{c}\in\mathrm{C} , \\
\widehat{\mathcal{S}}_\mathrm{h} &:= \bigtimes_{\mathrm{f}\in\mathrm{F}} \mathcal{S}_\mathrm{f} \times \bigtimes_{\mathrm{c}\in\mathrm{C}} \mathcal{S}_\mathrm{c} \equiv \mathbb{R}^{\#\mathrm{F}+\#\mathrm{C}} .
\end{aligned} \tag{2.63}$$

We introduce below the CDO operator needed to discretize the continuous form related to the advection term, namely, $t^\mathrm{s}(\underline{\beta}; s, r) := \int_\Omega(\underline{\beta}\cdot\underline{\nabla}\, s)r + \int_{\partial\Omega}(\underline{\beta}\cdot\underline{n}_{\partial\Omega})^- sr$, with s, r smooth enough. Recall that $t^\mathrm{s}(\cdot;\cdot,\cdot)$ is considered to formulate the PDE $-\underline{\beta}\underline{\nabla}\, s = f$ in Ω and $s = g$ on $\partial\Omega^-$.

Definition 2.34 - Scalar-valued advection. The scalar-valued CDO bilinear form discretizing the advection operator on a given mesh cell $\mathrm{c} \in \mathrm{C}$ is defined as follows:

$$\begin{aligned} t_{\mathrm{c}}^{\mathrm{s}}(\underline{\beta};\cdot,\cdot)\colon \quad \widehat{\mathcal{S}}_{\mathrm{c}} \times \widehat{\mathcal{S}}_{\mathrm{c}} &\to \mathbb{R} \\ (\widehat{s}_{\mathrm{c}}, \widehat{r}_{\mathrm{c}}) &\mapsto \sum_{\mathrm{f}\in\mathrm{F}_{\mathrm{c}}} \int_{\mathrm{f}} (\underline{\beta}\cdot\underline{n}_{\mathrm{fc}})(s_{\mathrm{f}} - s_{\mathrm{c}}) r_{\mathrm{c}} \\ &\quad + \frac{1}{2} \sum_{\mathrm{f}\in\mathrm{F}_{\mathrm{c}}} \int_{\mathrm{f}} (\underline{\beta}\cdot\underline{n}_{\mathrm{fc}})(s_{\mathrm{f}} - s_{\mathrm{c}})(r_{\mathrm{f}} - r_{\mathrm{c}})\,. \end{aligned} \tag{2.64}$$

The global bilinear form is obtained as usual by summing over all the mesh cells with the addition of two terms whose role will be explained in Remark 2.35:

$$\begin{aligned} t_{\mathrm{h}}^{\mathrm{s}}(\underline{\beta};\cdot,\cdot)\colon \quad \widehat{\mathcal{S}}_{\mathrm{h}} \times \widehat{\mathcal{S}}_{\mathrm{h}} &\to \mathbb{R} \\ (\widehat{s}_{\mathrm{h}}, \widehat{r}_{\mathrm{h}}) &\mapsto \sum_{\mathrm{c}\in\mathrm{C}} t_{\mathrm{c}}^{\mathrm{s}}(\underline{\beta};\widehat{s}_{\mathrm{h}},\widehat{r}_{\mathrm{h}}) + t_{\mathrm{h}}^{\mathrm{s},\partial}(\underline{\beta};\widehat{s}_{\mathrm{h}},\widehat{r}_{\mathrm{h}}) + t_{\mathrm{h}}^{\mathrm{s},\mathrm{u}}(\underline{\beta};\widehat{s}_{\mathrm{h}},\widehat{r}_{\mathrm{h}})\,, \end{aligned} \tag{2.65}$$

with

$$t_{\mathrm{h}}^{\mathrm{s},\partial}(\underline{\beta};\widehat{s}_{\mathrm{h}},\widehat{r}_{\mathrm{h}}) := \sum_{\mathrm{f}\in\mathrm{F}^b} \int_{\mathrm{f}} (\underline{\beta}\cdot\underline{n}_{\mathrm{fc}})^{-} s_{\mathrm{f}} r_{\mathrm{f}}\,, \tag{2.66}$$

$$t_{\mathrm{h}}^{\mathrm{s},\mathrm{u}}(\underline{\beta};\widehat{s}_{\mathrm{h}},\widehat{r}_{\mathrm{h}}) := \frac{1}{2}\Xi^{\mathrm{upw}} \sum_{\mathrm{f}\in\mathrm{F}^i} \sum_{\mathrm{c}\in\mathrm{C}_{\mathrm{f}}} \int_{\mathrm{f}} |\beta\cdot\underline{n}_{\mathrm{fc}}|\,(s_{\mathrm{f}} - s_{\mathrm{c}})(r_{\mathrm{f}} - r_{\mathrm{c}})\,, \tag{2.67}$$

and $\Xi^{\mathrm{upw}} \in \{0,1\}$ is a user-defined parameter, where, for any $x \in \mathbb{R}$, $(x)^{\pm} := \frac{1}{2}(|x| \pm x)$. ∘

Remark 2.35 - Roles of the terms. Let us clarify the roles of the different terms in Eqs. (2.64) to (2.67). The first line of (2.64) discretizes the volumetric part of the usual continuous variational formulation. In fact, an approximation of the gradient of s on a face may be noticed: $s_{\mathrm{f}} - s_{\mathrm{c}}$. The second term is needed to recover a positivity property in a consistent way in the same spirit of dG methods (see Di Pietro and Ern (2011, Section 2.2)), cf. Remark 2.39 below. Moreover, the second term plays the same role as the Temam's trick (see Temam (1977)) in the context of vector-valued convection, see also Remark 2.46. Let us now move to (2.66): $t_{\mathrm{h}}^{\mathrm{s},\partial}(\underline{\beta};\cdot,\cdot)$ embodies the weak enforcement of the boundary conditions at the inlet $\partial\Omega^{-}$, as is usually needed when considering a problem with an advection term but no diffusion. If diffusion is present, this term is still useful to temper oscillations that may be caused by boundary outflow layers (Bazilevs and Hughes, 2007; Schieweck, 2008). Finally, concerning (2.67), if the need arises, one can choose to add a stabilization by upwinding using a technique introduced in the discontinuous Galerkin framework by Brezzi *et al.* (2004): it just suffices to set $\Xi^{\mathrm{upw}} := 1$ in (2.67). Otherwise, using $\Xi^{\mathrm{upw}} := 0$ amounts to considering a scheme with the so-called centered fluxes. ◇

Remark 2.36 - Comparison with similar schemes. The bilinear form $t_{\mathrm{h}}^{\mathrm{s}}(\underline{\beta};\cdot,\cdot)$ is inspired from the HHO framework (Di Pietro *et al.*, 2015) in the lowest-order case $k := 0$. However, the present discretization has been applied directly to the continuous formulation without performing an integration by parts. ◇

Remark 2.37 - Simplifications. Notice that the advection velocity $\underline{\beta}$ appears only in face integrals. Since all the terms but those related to $\underline{\beta}$ are constant on a face, the only quantities left to compute are the fluxes of $\underline{\beta}$ at the faces: $\int_{\mathrm{f}} \underline{\beta}\cdot\underline{n}_{\mathrm{fc}}$. Since we are assuming planar faces, only the knowledge of $\underline{\beta}_{\mathrm{f}} := \underline{\pi}_{\mathrm{f}}(\underline{\beta})$ is necessary since it is readily shown that $\int_{\mathrm{f}} \underline{\beta}\cdot\underline{n}_{\mathrm{fc}} = |\mathrm{f}|\,\underline{\beta}_{\mathrm{f}}\cdot\underline{n}_{\mathrm{fc}}$. Suppose now that the partition F^b of the boundary $\partial\Omega$ is such that there exist no $\mathrm{f} \in \mathrm{F}^b$ such that $\mathrm{f} \cap \partial\Omega^{-} \neq \emptyset$ and $\mathrm{f} \cap \partial\Omega^{+} \neq \emptyset$ at the same time. Then, a similar observation holds for $(\underline{\beta}\cdot\underline{n}_{\mathrm{fc}})^{-}$. In conclusion, only the normal part of the advection

field at faces is needed to build (2.65) and all its terms. Thus, two advection fields differing only in the tangential part will carry the same contribution to the discrete problem since their normal components are identical. ◇

Remark 2.38 - Symmetry. Let us reorganize the addends in (2.64). We have

$$t_{\mathrm{c}}^{\mathrm{s}}(\underline{\beta};\widehat{s}_{\mathrm{c}},\widehat{r}_{\mathrm{c}}) = \frac{1}{2}\sum_{\mathrm{f}\in\mathrm{F}_{\mathrm{c}}}\int_{\mathrm{f}}(\underline{\beta}\cdot\underline{n}_{\mathrm{fc}})(\overbrace{s_{\mathrm{f}}r_{\mathrm{f}}-s_{\mathrm{c}}r_{\mathrm{c}}}^{\text{symmetric}}+\overbrace{s_{\mathrm{f}}r_{\mathrm{c}}-s_{\mathrm{c}}r_{\mathrm{f}}}^{\text{skew-symmetric}}), \tag{2.68}$$

where we took care of separating symmetric and skew-symmetric terms. Notice that the boundary contribution (2.66) and the upwind contribution (2.67) are symmetric, hence the only skew-symmetric part of $t_{\mathrm{h}}^{\mathrm{s}}(\underline{\beta};\cdot,\cdot)$ comes from $\sum_{\mathrm{c}\in\mathrm{C}} t_{\mathrm{c}}^{\mathrm{s}}(\underline{\beta};\cdot,\cdot)$. ◇

Remark 2.39 - Discrete integration by parts and positivity. Consider the following Integration by Parts (IBP) result:

$$\int_{\Omega}\Big((\underline{\beta}\cdot\underline{\nabla}\,s)r+(\underline{\beta}\cdot\underline{\nabla}\,r)s\Big) = -\int_{\Omega}\underline{\nabla}\cdot\underline{\beta}\,sr+\int_{\partial\Omega}(\underline{\beta}\cdot\underline{n}_{\partial\Omega})sr\,, \tag{2.69}$$

valid for smooth functions s, $r\in\mathcal{C}^1(\overline{\Omega})$. Let us prove that a discrete counterpart is satisfied by $t_{\mathrm{c}}^{\mathrm{s}}(\underline{\beta};\cdot,\cdot)$. We have

$$\begin{aligned}\sum_{\mathrm{c}\in\mathrm{C}}\Big(t_{\mathrm{c}}^{\mathrm{s}}(\underline{\beta};\widehat{s}_{\mathrm{c}},\widehat{r}_{\mathrm{c}})+t_{\mathrm{c}}^{\mathrm{s}}(\underline{\beta};\widehat{r}_{\mathrm{c}},\widehat{s}_{\mathrm{c}})\Big) &= \sum_{\mathrm{c}\in\mathrm{C}}\sum_{\mathrm{f}\in\mathrm{F}_{\mathrm{c}}}\frac{1}{2}\int_{\mathrm{f}}(\underline{\beta}\cdot\underline{n}_{\mathrm{fc}})(s_{\mathrm{f}}-s_{\mathrm{c}})(r_{\mathrm{f}}+r_{\mathrm{c}})\\ &\quad+\sum_{\mathrm{c}\in\mathrm{C}}\sum_{\mathrm{f}\in\mathrm{F}_{\mathrm{c}}}\frac{1}{2}\int_{\mathrm{f}}(\underline{\beta}\cdot\underline{n}_{\mathrm{fc}})(s_{\mathrm{f}}+s_{\mathrm{c}})(r_{\mathrm{f}}-r_{\mathrm{c}})\\ &= -\sum_{\mathrm{c}\in\mathrm{C}}s_{\mathrm{c}}r_{\mathrm{c}}\sum_{\mathrm{f}\in\mathrm{F}_{\mathrm{c}}}\int_{\mathrm{f}}(\underline{\beta}\cdot\underline{n}_{\mathrm{fc}})+\sum_{\mathrm{c}\in\mathrm{C}}\sum_{\mathrm{f}\in\mathrm{F}_{\mathrm{c}}}\int_{\mathrm{f}}(\underline{\beta}\cdot\underline{n}_{\mathrm{fc}})s_{\mathrm{f}}r_{\mathrm{f}}\,,\end{aligned} \tag{2.70}$$

where the contribution of the skew-symmetric part of $t_{\mathrm{c}}^{\mathrm{s}}(\underline{\beta};\cdot,\cdot)$ vanishes. Observe that, owing to Remark 2.13, when summing over the mesh cells, the contribution of the internal faces to the second term vanishes as well. Finally, using the divergence theorem, one can write

$$\sum_{\mathrm{c}\in\mathrm{C}}\Big(t_{\mathrm{c}}^{\mathrm{s}}(\underline{\beta};\widehat{s}_{\mathrm{c}},\widehat{r}_{\mathrm{c}})+t_{\mathrm{c}}^{\mathrm{s}}(\underline{\beta};\widehat{r}_{\mathrm{c}},\widehat{s}_{\mathrm{c}})\Big) = -\sum_{\mathrm{c}\in\mathrm{C}}\int_{\mathrm{c}}(\underline{\nabla}\cdot\underline{\beta})s_{\mathrm{c}}r_{\mathrm{c}}+\sum_{\mathrm{f}\in\mathrm{F}^b}\int_{\mathrm{f}}(\underline{\beta}\cdot\underline{n}_{\mathrm{f}})s_{\mathrm{f}}r_{\mathrm{f}}\,, \tag{2.71}$$

which is the sought discrete version of (2.69).

Consider now $t_{\mathrm{h}}^{\mathrm{s}}(\underline{\beta};\cdot,\cdot)$ as defined in (2.65). Choose $\widehat{r}_{\mathrm{h}}=\widehat{s}_{\mathrm{h}}$ and use (2.71). Proceeding in a similar fashion as it was done for (2.71), one can write

$$\begin{aligned}t_{\mathrm{h}}^{\mathrm{s}}(\underline{\beta};\widehat{s}_{\mathrm{h}},\widehat{s}_{\mathrm{h}}) = &-\frac{1}{2}\sum_{\mathrm{c}\in\mathrm{C}}\int_{\mathrm{c}}(\underline{\nabla}\cdot\underline{\beta})s_{\mathrm{c}}^2+\frac{1}{2}\sum_{\mathrm{f}\in\mathrm{F}^b}\int_{\mathrm{f}}\left|\underline{\beta}\cdot\underline{n}_{\mathrm{f}}\right|s_{\mathrm{f}}^2\\ &+\Xi^{\mathrm{upw}}\frac{1}{2}\sum_{\mathrm{f}\in\mathrm{F}^i}\sum_{\mathrm{c}\in\mathrm{C}_{\mathrm{f}}}\int_{\mathrm{f}}\left|\underline{\beta}\cdot\underline{n}_{\mathrm{f}}\right|(s_{\mathrm{f}}-s_{\mathrm{c}})^2\,.\end{aligned} \tag{2.72}$$

Hence, if $\underline{\nabla}\cdot\underline{\beta}\geq$ a.e. in Ω,

$$t_{\mathrm{h}}^{\mathrm{s}}(\underline{\beta};\widehat{s}_{\mathrm{h}},\widehat{s}_{\mathrm{h}})\geq 0\,,\quad\forall\widehat{s}_{\mathrm{h}}\in\widehat{\mathcal{S}}_{\mathrm{h}}\,. \tag{2.73}$$

Notice that (2.73) is the discrete counterpart of a positivity results for the classical advection form $t^{\mathrm{s}}(\underline{\beta};s,r):=\int_{\Omega}(\underline{\beta}\cdot\underline{\nabla}\,s)r-\int_{\partial\Omega^-}(\underline{\beta}\cdot\underline{n}_{\partial\Omega})sr$, see, for instance, Di Pietro and Ern (2011, Lemma 2.8). ◇

Remark 2.40 - Conservative formulation. Hinging on the well-known differential identity $\underline{\nabla} \cdot (\underline{\beta} s) = \underline{\beta} \cdot \underline{\nabla} s + \underline{\nabla} \cdot \underline{\beta}\, s$, one can modify $t_{\mathrm{h}}^{\mathrm{s}}(\underline{\beta}; \cdot, \cdot)$ to obtain a conservative version of the advection operator. In particular, $t_{\mathrm{c}}^{\mathrm{s}}(\underline{\beta}; \cdot, \cdot)$ is modified as follows:

$$\begin{aligned} t_{\mathrm{c}}^{\mathrm{s,csv}}(\underline{\beta}; \widehat{s}_{\mathrm{c}}, \widehat{r}_{\mathrm{c}}) &\coloneqq t_{\mathrm{c}}^{\mathrm{s}}(\underline{\beta}_{\mathrm{c}}; \widehat{s}_{\mathrm{c}}, \widehat{r}_{\mathrm{c}}) + \int_{\mathrm{c}} (\underline{\nabla} \cdot \underline{\beta}) s_{\mathrm{c}} r_{\mathrm{c}} \\ &= \frac{1}{2} \sum_{\mathrm{f} \in \mathrm{F_c}} \int_{\mathrm{f}} (\underline{\beta} \cdot \underline{n}_{\mathrm{fc}})(s_{\mathrm{f}} - s_{\mathrm{c}})(r_{\mathrm{f}} - r_{\mathrm{c}}) + \sum_{\mathrm{f} \in \mathrm{F_c}} \int_{\mathrm{f}} (\underline{\beta} \cdot \underline{n}_{\mathrm{fc}}) s_{\mathrm{f}} r_{\mathrm{c}} \\ &= \frac{1}{2} \sum_{\mathrm{f} \in \mathrm{F_c}} \int_{\mathrm{f}} (\underline{\beta} \cdot \underline{n}_{\mathrm{fc}})(s_{\mathrm{f}} r_{\mathrm{f}} + s_{\mathrm{c}} r_{\mathrm{c}} + s_{\mathrm{f}} r_{\mathrm{c}} - s_{\mathrm{c}} r_{\mathrm{f}}) \,, \end{aligned} \tag{2.74}$$

whereas $t_{\mathrm{h}}^{\mathrm{s},\partial}(\underline{\beta}; \cdot, \cdot)$ and $t_{\mathrm{h}}^{\mathrm{s,u}}(\underline{\beta}; \cdot, \cdot)$ do not change. ◇

Let us state and prove the main result to be used in the error analysis if the bilinear form $t_{\mathrm{h}}^{\mathrm{s}}(\underline{\beta}; \cdot, \cdot)$ were to be used to approximate an advection problem. This result consists of bounding the consistency error and stating that it exhibits a decay rate as $\mathcal{O}\left(\mathrm{h}^{\frac{1}{2}}\right)$. We only detail the bound on the consistency error to highlight the properties of the scheme. Set

$$\tau_0 \coloneqq \max \left(\left\| \underline{\nabla} \cdot \underline{\beta} \right\|_{L^\infty(\Omega)} , \frac{\left\| \underline{\beta} \right\|_{\underline{L}^\infty(\Omega)}}{\mathrm{h}_\Omega} \right) , \tag{2.75}$$

where h_Ω is the diameter of the domain, defined similarly to (2.8). The length scale h_Ω is introduced to make the definition of τ_0 dimensionally consistent, so that τ_0 scales as the reciprocal of a time scale. Define also the following norm based on (2.72):

$$\|\widehat{r}_{\mathrm{h}}\|_{\underline{\beta}}^2 \coloneqq \sum_{\mathrm{c} \in \mathrm{C}} \tau_0 \left|\mathrm{c}\right| r_{\mathrm{c}}^2 + \sum_{\mathrm{f} \in \mathrm{F}^b} \int_{\mathrm{f}} \left| \underline{\beta} \cdot \underline{n}_{\mathrm{f}} \right| r_{\mathrm{f}}^2 + \sum_{\mathrm{c} \in \mathrm{C}} \sum_{\mathrm{f} \in \mathrm{F_c}} \int_{\mathrm{f}} \left| \underline{\beta} \cdot \underline{n}_{\mathrm{f}} \right| (r_{\mathrm{f}} - r_{\mathrm{c}})^2 \,. \tag{2.76}$$

Proposition 2.41 - Bound on consistency error. *For all $s \in H^1(\Omega)$, define the consistency error as the following linear functional: For all $\widehat{r}_{\mathrm{h}} \in \widehat{\mathcal{S}}_{\mathrm{h}}$,*

$$\mathcal{E}_{t^{\mathrm{s}}}(\widehat{r}_{\mathrm{h}}) \coloneqq \sum_{\mathrm{c} \in \mathrm{C}} \int_{\mathrm{c}} (\underline{\beta} \cdot \underline{\nabla} s) r_{\mathrm{c}} + \sum_{\mathrm{f} \in \mathrm{F}^b} \int_{\mathrm{f}} (\underline{\beta} \cdot \underline{n}_{\mathrm{f}})^- s r_{\mathrm{f}} - t_{\mathrm{h}}^{\mathrm{s}}(\underline{\beta}; \widehat{\pi}_{\mathrm{h}}(s), \widehat{r}_{\mathrm{h}}) \,. \tag{2.77}$$

Then, assuming the advection field $\underline{\beta}$ satisfies (2.62)*, and that the considered mesh sequence is shape-regular as in Definition 2.9, there exists a constant C, independent of* h*, but possibly depending on the mesh regularity parameters, such that*

$$|\mathcal{E}_{t^{\mathrm{s}}}(\widehat{r}_{\mathrm{h}})| \lesssim \max \left((\tau_0 \mathrm{h})^{\frac{1}{2}}, \left| \underline{\beta} \right|_{\underline{L}^\infty(\Omega)}^{\frac{1}{2}} \right) \mathrm{h}^{\frac{1}{2}} \left\| \underline{\nabla} s \right\|_{\underline{L}^2(\Omega)} \|\widehat{r}_{\mathrm{h}}\|_{\underline{\beta}} \,. \tag{2.78}$$

Proof. We shall use the notation $a \lesssim b$ to mean that $a \leq Cb$ whenever the constant C satisfies the dependency statement made in Proposition 2.41. The technique of the proof is inspired by Ern and Guermond (2006a), see also Di Pietro *et al.* (2015) and Di Pietro and Ern (2011). Applying the usual hybrid notation, let $\widehat{s}_{\mathrm{h}} \coloneqq \widehat{\pi}_{\mathrm{h}}(s)$, i.e., we set $s_{\mathrm{c}} \coloneqq \pi_{\mathrm{c}}(s)$ for all $\mathrm{c} \in \mathrm{C}$ and $s_{\mathrm{f}} \coloneqq \pi_{\mathrm{f}}(s)$ for all $\mathrm{f} \in \mathrm{F}$. Let us rewrite the first term of $\mathcal{E}_{t^{\mathrm{s}}}(\widehat{r}_{\mathrm{h}})$:

$$\begin{aligned} \sum_{\mathrm{c} \in \mathrm{C}} \int_{\mathrm{c}} (\underline{\beta} \cdot \underline{\nabla} s) r_{\mathrm{c}} &= -\sum_{\mathrm{c} \in \mathrm{C}} \int_{\mathrm{c}} (\underline{\nabla} \cdot \underline{\beta}) s r_{\mathrm{c}} + \sum_{\mathrm{c} \in \mathrm{C}} \sum_{\mathrm{f} \in \mathrm{F_c}} \int_{\mathrm{f}} (\underline{\beta} \cdot \underline{n}_{\mathrm{fc}}) s r_{\mathrm{c}} \\ &= -\sum_{\mathrm{c} \in \mathrm{C}} \int_{\mathrm{c}} (\underline{\nabla} \cdot \underline{\beta}) s r_{\mathrm{c}} \\ &\quad + \sum_{\mathrm{c} \in \mathrm{C}} \sum_{\mathrm{f} \in \mathrm{F_c}} \int_{\mathrm{f}} (\underline{\beta} \cdot \underline{n}_{\mathrm{fc}}) s (r_{\mathrm{c}} - r_{\mathrm{f}}) + \sum_{\mathrm{f} \in \mathrm{F}^b} \int_{\mathrm{f}} (\underline{\beta} \cdot \underline{n}_{\mathrm{f}}) s r_{\mathrm{f}} \,, \end{aligned} \tag{2.79}$$

where the first equality is recovered by applying the IBP result (2.69) and observing that, since $r_{\mathrm{c}} \in \mathbb{P}^0(\mathrm{c})$, the term $\int_{\mathrm{c}}(\underline{\beta} \cdot \underline{\nabla} r_{\mathrm{c}})s$ vanishes, and the second equality follows from the identity established in Remark 2.13 and adding/subtracting r_{f}. Similarly, proceeding as in Remark 2.39, one can write

$$\begin{aligned}\sum_{\mathrm{c}\in\mathrm{C}}\sum_{\mathrm{f}\in\mathrm{F_c}}\int_{\mathrm{f}}(\underline{\beta}\cdot\underline{n}_{\mathrm{fc}})(s_{\mathrm{f}}-s_{\mathrm{c}})r_{\mathrm{c}} = &-\sum_{\mathrm{c}\in\mathrm{C}}\int_{\mathrm{c}}(\underline{\nabla}\cdot\underline{\beta})s_{\mathrm{c}}r_{\mathrm{c}} + \sum_{\mathrm{c}\in\mathrm{C}}\sum_{\mathrm{f}\in\mathrm{F_c}}\int_{\mathrm{f}}(\underline{\beta}\cdot\underline{n}_{\mathrm{fc}})s_{\mathrm{f}}(r_{\mathrm{c}}-r_{\mathrm{f}})\\ &+\sum_{\mathrm{f}\in\mathrm{F}^b}\int_{\mathrm{f}}(\underline{\beta}\cdot\underline{n}_{\mathrm{f}})s_{\mathrm{f}}r_{\mathrm{f}}\,.\end{aligned} \tag{2.80}$$

Plugging (2.79) and (2.80) in (2.77) and noticing that $(x)^- + x = (x)^+$ for all $x \in \mathbb{R}$, one obtains:

$$\begin{aligned}\mathcal{E}_{t^{\mathrm{s}}}(\widehat{r}_{\mathrm{h}}) = &-\sum_{\mathrm{c}\in\mathrm{C}}\int_{\mathrm{c}}(\underline{\nabla}\cdot\underline{\beta})(s-s_{\mathrm{c}})r_{\mathrm{c}} + \sum_{\mathrm{f}\in\mathrm{F}^b}\int_{\mathrm{f}}(\underline{\beta}\cdot\underline{n}_{\mathrm{f}})^+(s-s_{\mathrm{f}})r_{\mathrm{f}}\\ &-\sum_{\mathrm{c}\in\mathrm{C}}\sum_{\mathrm{f}\in\mathrm{F_c}}\int_{\mathrm{f}}(\underline{\beta}\cdot\underline{n}_{\mathrm{fc}})(s-s_{\mathrm{f}})(r_{\mathrm{f}}-r_{\mathrm{c}}) - \sum_{\mathrm{c}\in\mathrm{C}}\sum_{\mathrm{f}\in\mathrm{F_c}}\int_{\mathrm{c}}\frac{1}{2}(\underline{\beta}\cdot\underline{n}_{\mathrm{fc}})(s_{\mathrm{f}}-s_{\mathrm{c}})(r_{\mathrm{f}}-r_{\mathrm{c}})\\ &-t_{\mathrm{h}}^{\mathrm{s,u}}(\underline{\beta};\widehat{s}_{\mathrm{h}},\widehat{r}_{\mathrm{h}})\,.\end{aligned} \tag{2.81}$$

Consider now the absolute value of $\mathcal{E}_{t^{\mathrm{s}}}(\widehat{r}_{\mathrm{h}})$. Let us bound each term on the right-hand side of (2.81). For the first one, we have

$$\begin{aligned}\left|-\sum_{\mathrm{c}\in\mathrm{C}}\int_{\mathrm{c}}(\underline{\nabla}\cdot\underline{\beta})(s-s_{\mathrm{c}})r_{\mathrm{c}}\right| &\le \left\|\underline{\nabla}\cdot\underline{\beta}\right\|_{L^\infty(\Omega)}\left(\sum_{\mathrm{c}\in\mathrm{C}}|s-s_{\mathrm{c}}|^2_{L^2(\mathrm{c})}\right)^{\frac{1}{2}}\left(\sum_{\mathrm{c}\in\mathrm{C}}\int_{\mathrm{c}}r_{\mathrm{c}}^2\right)^{\frac{1}{2}}\\ &\lesssim \tau_0\mathrm{h}\,\|\underline{\nabla}\,s\|_{\underline{L}^2(\Omega)}\left(\sum_{\mathrm{c}\in\mathrm{C}}\int_{\mathrm{c}}r_{\mathrm{c}}^2\right)^{\frac{1}{2}}\\ &= \tau_0^{\frac{1}{2}}\mathrm{h}\,\|\underline{\nabla}\,s\|_{\underline{L}^2(\Omega)}\left(\sum_{\mathrm{c}\in\mathrm{C}}\int_{\mathrm{c}}\tau_0 r_{\mathrm{c}}^2\right)^{\frac{1}{2}}\\ &\le \tau_0^{\frac{1}{2}}\mathrm{h}\,\|\underline{\nabla}\,s\|_{\underline{L}^2(\Omega)}\,\|\widehat{r}_{\mathrm{h}}\|_{\underline{\beta}}\,,\end{aligned} \tag{2.82}$$

where we used the Cauchy-Schwarz inequality, the definition (2.75) of τ_0, and the local Poincaré inequality in order to bound $\|s - s_{\mathrm{c}}\|_{L^2(\Omega)}$, see for instance Ern and Guermond (2004, Proposition 1.134) or Di Pietro and Ern (2011, Lemma 1.59). Moving on, it is readily proven that, for all $\mathrm{c} \in \mathrm{C}$ and all $\mathrm{f} \in \mathrm{F_c}$,

$$\|s_{\mathrm{f}} - s_{\mathrm{c}}\|_{L^2(\mathrm{f})} = \|\pi_{\mathrm{f}}(s - s_{\mathrm{c}})\|_{L^2(\mathrm{f})} \le \|s - s_{\mathrm{c}}\|_{L^2(\mathrm{f})} \lesssim \mathrm{h_c}^{\frac{1}{2}}\,\|\underline{\nabla}\,s\|_{\underline{L}^2(\mathrm{c})}\,, \tag{2.83}$$

where one uses the linearity, the stability of π_{f}, and the approximation property from Di Pietro and Ern (2011, Lemma 1.59) of the orthogonal projection. Similarly, using the triangle inequality leads to

$$\|s - s_{\mathrm{f}}\|_{L^2(\mathrm{f})} \le \|s - s_{\mathrm{c}}\|_{L^2(\mathrm{f})} + \|s_{\mathrm{c}} - s_{\mathrm{f}}\|_{L^2(\mathrm{f})} \lesssim \mathrm{h_c}^{\frac{1}{2}}\,\|\underline{\nabla}\,s\|_{\underline{L}^2(\mathrm{c})}\,. \tag{2.84}$$

Using the fact that the number of faces of a cell is uniformly bounded, these last two results may be easily extended to $\|\cdot\|^2_{L^2(\partial\mathrm{c})} := \sum_{\mathrm{f}\in\mathrm{F_c}}\|\cdot\|^2_{L^2(\mathrm{f})}$ The second term on the right-hand

side of (2.81) is then treated similarly to (2.82) and using (2.84). We get

$$\begin{aligned}\left\|\sum_{\mathrm{f}\in\mathrm{F}^b}\int_{\mathrm{f}}(\underline{\beta}\cdot\underline{n}_{\mathrm{fc}})^+(s-s_{\mathrm{f}})r_{\mathrm{f}}\right\| &\le \left\|\underline{\beta}\right\|^{\frac12}_{\underline{L}^\infty(\Omega)}\left(\sum_{\mathrm{f}\in\mathrm{F}^b}\|s-s_{\mathrm{f}}\|^2_{L^2(\mathrm{f})}\right)^{\frac12}\left(\sum_{\mathrm{f}\in\mathrm{F}^b}\int_{\mathrm{f}}\left|\underline{\beta}\cdot\underline{n}_{\mathrm{f}}\right|r_{\mathrm{f}}^2\right)^{\frac12}\\ &\lesssim \left|\underline{\beta}\right|^{\frac12}_{\underline{L}^\infty(\Omega)}\mathrm{h}^{\frac12}\,\|\underline{\nabla}\,s\|_{\underline{L}^2(\Omega)}\left(\sum_{\mathrm{f}\in\mathrm{F}^b}\int_{\mathrm{f}}\left|\underline{\beta}\cdot\underline{n}_{\mathrm{f}}\right|r_{\mathrm{f}}^2\right)^{\frac12}\\ &\le \tau_0^{\frac12}\mathrm{h}_\Omega^{\frac12}\mathrm{h}^{\frac12}\,\|\underline{\nabla}\,s\|_{\underline{L}^2(\Omega)}\,\|\widehat{r}_{\mathrm{h}}\|_{\underline{\beta}}\,.\end{aligned}\tag{2.85}$$

Owing to (2.83) and (2.84), the last three terms of (2.81) are bounded in the same way. Taking, for instance, the third term, we obtain

$$\begin{aligned}&\left|\sum_{\mathrm{c}\in\mathrm{C}}\sum_{\mathrm{f}\in\mathrm{F}_{\mathrm{c}}}\int_{\mathrm{f}}(\underline{\beta}\cdot\underline{n}_{\mathrm{fc}})(s-s_{\mathrm{f}})(r_{\mathrm{f}}-r_{\mathrm{c}})\right|\\ &\qquad\lesssim \left\|\underline{\beta}\right\|^{\frac12}_{\underline{L}^\infty(\Omega)}\left(\sum_{\mathrm{c}\in\mathrm{C}}\sum_{\mathrm{f}\in\mathrm{F}_{\mathrm{c}}}\|s-s_{\mathrm{f}}\|^2_{L^2(\mathrm{f})}\right)^{\frac12}\left(\sum_{\mathrm{c}\in\mathrm{C}}\sum_{\mathrm{f}\in\mathrm{F}_{\mathrm{c}}}\int_{\mathrm{f}}\left|\underline{\beta}\cdot\underline{n}_{\mathrm{f}}\right|r_{\mathrm{f}}^2\right)^{\frac12}\\ &\qquad\lesssim \tau_0^{\frac12}\mathrm{h}_\Omega^{\frac12}\mathrm{h}^{\frac12}\,\|\underline{\nabla}\,s\|_{\underline{L}^2(\Omega)}\,\|\widehat{r}_{\mathrm{h}}\|_{\underline{\beta}}\,,\end{aligned}\tag{2.86}$$

where the last inequality is obtained using (2.84) and then summing over the mesh cells. Similar results hold for the last two terms of (2.81). Putting (2.82), (2.85), and (2.86) together and using the definition (2.76) of $\|\cdot\|_{\underline{\beta}}$ proves (2.78). □

2.5.2 Vector-valued convection

The CDO-Fb discretization of the scalar-valued advection operator described in Section 2.5.1 is now extended to act on $\mathbb{R}^d$-valued fields in order to deal with the convection operator encountered in the NSE. Thus our aim is to provide a discretization of the following well-known trilinear form:

$$t(\underline{w};\underline{u},\underline{v}) := \int_\Omega\Big((\underline{w}\cdot\underline{\nabla})\underline{u}\Big)\cdot\underline{v} = \int_\Omega \underline{v}^T\underline{\nabla}\,\underline{u}\,\underline{w} = \int_\Omega\sum_{i,j=1}^d w_i\frac{\partial u_j}{\partial x_i}v_j\,,\tag{2.87}$$

where the convective field is now denoted by $\underline{w}$ instead of $\underline{\beta}$.

Let us start by giving the vector-valued counterpart of $t_{\mathrm{h}}^{\mathrm{s}}(\underline{\beta};\cdot,\cdot)$ defined in (2.65). Since in the discrete NSE, the convection field is discrete as well, $\underline{w}$ has to be replaced by a hybrid variable, that we denote by $\widehat{\underline{w}}_{\mathrm{h}}\in\widehat{\underline{\mathcal{U}}}_{\mathrm{h}}$. Owing to Remark 2.37, only the face-based DoFs of $\widehat{\underline{w}}_{\mathrm{h}}$ are actually relevant to the formulation of the discrete trilinear form.

Definition 2.42 - Convection - Discrete trilinear form. The vector-valued trilinear forms, counterparts of Eqs. (2.64) to (2.67), are:

$$\begin{array}{rccl} t_{\mathrm{h}}(\cdot;\cdot,\cdot)\colon & \widehat{\underline{\mathcal{U}}}_{\mathrm{h}}\times\widehat{\underline{\mathcal{U}}}_{\mathrm{h}}\times\widehat{\underline{\mathcal{U}}}_{\mathrm{h}} & \to & \mathbb{R}\\ & (\widehat{\underline{w}}_{\mathrm{h}},\widehat{\underline{u}}_{\mathrm{h}},\widehat{\underline{v}}_{\mathrm{h}}) & \mapsto & \displaystyle\sum_{\mathrm{c}\in\mathrm{C}} t_{\mathrm{c}}(\widehat{\underline{w}}_{\mathrm{c}};\widehat{\underline{u}}_{\mathrm{c}},\widehat{\underline{v}}_{\mathrm{c}}) + t_{\mathrm{h}}^{\partial}(\widehat{\underline{w}}_{\mathrm{h}};\widehat{\underline{u}}_{\mathrm{h}},\widehat{\underline{v}}_{\mathrm{h}})\\ & & & +t_{\mathrm{h}}^{\mathrm{u}}(\widehat{\underline{w}}_{\mathrm{h}};\widehat{\underline{u}}_{\mathrm{h}},\widehat{\underline{v}}_{\mathrm{h}})\end{array}\tag{2.88}$$

with

$$t_{\mathrm{c}}(\widehat{\underline{w}}_{\mathrm{c}};\widehat{\underline{u}}_{\mathrm{c}},\widehat{\underline{v}}_{\mathrm{c}}) \coloneqq \sum_{\mathrm{f}\in\mathrm{F_c}} |\mathrm{f}|\,(\underline{w}_{\mathrm{f}}\cdot\underline{n}_{\mathrm{fc}})(\underline{u}_{\mathrm{f}}-\underline{u}_{\mathrm{c}})\cdot\underline{v}_{\mathrm{c}} + \frac{1}{2}\sum_{\mathrm{f}\in\mathrm{F_c}} |\mathrm{f}|\,(\underline{w}_{\mathrm{f}}\cdot\underline{n}_{\mathrm{fc}})(\underline{u}_{\mathrm{f}}-\underline{u}_{\mathrm{c}})\cdot(\underline{v}_{\mathrm{f}}-\underline{v}_{\mathrm{c}})\,, \tag{2.89}$$

$$t^{\partial}_{\mathrm{h}}(\widehat{\underline{w}}_{\mathrm{h}};\widehat{\underline{u}}_{\mathrm{h}},\widehat{\underline{v}}_{\mathrm{h}}) \coloneqq \sum_{\mathrm{f}\in\mathrm{F}^b} |\mathrm{f}|\,(\underline{w}_{\mathrm{f}}\cdot\underline{n}_{\mathrm{fc}})^{-}\underline{u}_{\mathrm{f}}\cdot\underline{v}_{\mathrm{f}}\,, \tag{2.90}$$

$$t^{\mathrm{u}}_{\mathrm{h}}(\widehat{\underline{w}}_{\mathrm{h}};\widehat{\underline{u}}_{\mathrm{h}},\widehat{\underline{v}}_{\mathrm{h}}) \coloneqq \frac{1}{2}\Xi^{\mathrm{upw}}\sum_{\mathrm{f}\in\mathrm{F}^i}\sum_{\mathrm{c}\in\mathrm{C_f}} |\mathrm{f}|\,|\underline{w}_{\mathrm{f}}\cdot\underline{n}_{\mathrm{fc}}|\,(\underline{u}_{\mathrm{f}}-\underline{u}_{\mathrm{c}})\cdot(\underline{v}_{\mathrm{f}}-\underline{v}_{\mathrm{c}})\,. \tag{2.91}$$

Here, $\Xi^{\mathrm{upw}} \in \{0,1\}$ has the same meaning as in (2.65), namely $\Xi^{\mathrm{upw}} \coloneqq 1$ activates a stabilization by upwinding. ○

Remark 2.43 - Roles of the terms. The comments made in Remark 2.35 on $t^{\mathrm{s}}_{\mathrm{h}}(\underline{\beta};\cdot,\cdot)$ can be readily adapted to the vector-valued case. ◇

Remark 2.44 - Symmetry. Rearranging the terms in (2.88), we have

$$\begin{aligned} t_{\mathrm{h}}(\widehat{\underline{w}}_{\mathrm{h}};\widehat{\underline{u}}_{\mathrm{h}},\widehat{\underline{v}}_{\mathrm{h}}) = \frac{1}{2}\sum_{\mathrm{c}\in\mathrm{C}}\sum_{\mathrm{f}\in\mathrm{F_c}} |\mathrm{f}|\,(\underline{w}_{\mathrm{f}}\cdot\underline{n}_{\mathrm{fc}})(\overbrace{\underline{u}_{\mathrm{f}}\cdot\underline{v}_{\mathrm{f}}-\underline{u}_{\mathrm{c}}\cdot\underline{v}_{\mathrm{c}}}^{\text{symmetric}}+\overbrace{\underline{u}_{\mathrm{f}}\cdot\underline{v}_{\mathrm{c}}-\underline{u}_{\mathrm{c}}\cdot\underline{v}_{\mathrm{f}}}^{\text{skew-symmetric}}) \\ + \underbrace{t^{\partial}_{\mathrm{h}}(\widehat{\underline{w}}_{\mathrm{h}};\widehat{\underline{u}}_{\mathrm{h}},\widehat{\underline{v}}_{\mathrm{h}})}_{\text{symmetric}} + \underbrace{t^{\mathrm{u}}_{\mathrm{h}}(\widehat{\underline{w}}_{\mathrm{h}};\widehat{\underline{u}}_{\mathrm{h}},\widehat{\underline{v}}_{\mathrm{h}})}_{\text{symmetric}}\,. \end{aligned} \tag{2.92}$$

As in Remark 2.38, the only skew-symmetric part of $t_{\mathrm{h}}(\cdot;\cdot,\cdot)$ results from $t_{\mathrm{c}}(\cdot;\cdot,\cdot)$. ◇

Lemma 2.45 - Positivity and skew-symmetry of $t_{\mathrm{h}}(\cdot;\cdot,\cdot)$. *Fix a discretely divergence-free field $\widehat{\underline{w}}_{\mathrm{h}} \in \widehat{\underline{\mathcal{U}}}_{\mathrm{h}}$, i.e.,*

$$\mathrm{D_c}(\widehat{\underline{w}}_{\mathrm{c}}) = 0 \qquad \forall \mathrm{c}\in\mathrm{C}\,. \tag{2.93}$$

Then:

i) For all $\widehat{\underline{u}}_{\mathrm{h}} \in \widehat{\underline{\mathcal{U}}}_{\mathrm{h}}$,

$$t_{\mathrm{h}}(\widehat{\underline{w}}_{\mathrm{h}};\widehat{\underline{u}}_{\mathrm{h}},\widehat{\underline{u}}_{\mathrm{h}}) \geq 0\,. \tag{2.94}$$

ii) Suppose additionally that the normal component of $\widehat{\underline{w}}_{\mathrm{h}}$ is null at the boundary:

$$\underline{w}_{\mathrm{f}}\cdot\underline{n}_{\mathrm{f}} = 0 \qquad \forall \mathrm{f}\in\mathrm{F}^b\,. \tag{2.95}$$

Then, whenever $\Xi^{\mathrm{upw}} = 0$, $t_{\mathrm{h}}(\widehat{\underline{w}}_{\mathrm{h}};\cdot,\cdot)$ is skew-symmetric, i.e.

$$t_{\mathrm{h}}(\widehat{\underline{w}}_{\mathrm{h}};\widehat{\underline{u}}_{\mathrm{h}},\widehat{\underline{u}}_{\mathrm{h}}) = 0\,, \quad \forall \widehat{\underline{u}}_{\mathrm{h}} \in \widehat{\underline{\mathcal{U}}}_{\mathrm{h}}\,. \tag{2.96}$$

Proof. Let us start by addressing *i)*. Set $\widehat{\underline{v}}_{\mathrm{h}} \coloneqq \widehat{\underline{u}}_{\mathrm{h}}$ in Definition 2.42 and let us consider each term separately. The stabilization by upwinding, defined in (2.91), yields

$$t^{\mathrm{u}}_{\mathrm{h}}(\underline{w}_{\mathrm{h}};\widehat{\underline{u}}_{\mathrm{h}},\widehat{\underline{u}}_{\mathrm{h}}) = \frac{1}{2}\Xi^{\mathrm{upw}}\sum_{\mathrm{f}\in\mathrm{F}^i}\sum_{\mathrm{c}\in\mathrm{C_f}} |\mathrm{f}|\,|\underline{w}_{\mathrm{f}}\cdot\underline{n}_{\mathrm{fc}}|\,|\underline{u}_{\mathrm{f}}-\underline{u}_{\mathrm{c}}|_2^2 \geq 0\,, \tag{2.97}$$

which holds for $\Xi^{\mathrm{upw}} \in \{0,\,1\}$. Moving to the boundary term defined in (2.90), it is readily seen that

$$t^{\partial}_{\mathrm{h}}(\widehat{\underline{w}}_{\mathrm{h}};\widehat{\underline{u}}_{\mathrm{h}},\widehat{\underline{u}}_{\mathrm{h}}) = \sum_{\mathrm{f}\in\mathrm{F}^b} |\mathrm{f}|\,(\underline{w}_{\mathrm{f}}\cdot\underline{n}_{\mathrm{fc}})^{-}\,|\underline{u}_{\mathrm{f}}|_2^2\,. \tag{2.98}$$

Now, considering (2.89) and dropping the skew-symmetric terms (see (2.92)), one has

$$\sum_{\mathrm{c}\in\mathrm{C}} t_{\mathrm{c}}(\widehat{\underline{w}}_{\mathrm{c}};\widehat{\underline{u}}_{\mathrm{c}},\widehat{\underline{u}}_{\mathrm{c}}) = \frac{1}{2}\sum_{\mathrm{c}\in\mathrm{C}}\sum_{\mathrm{f}\in\mathrm{F}} |\mathrm{f}|\,(\underline{w}_{\mathrm{f}}\cdot\underline{n}_{\mathrm{fc}})(|\underline{u}_{\mathrm{f}}|_2^2 - |\underline{u}_{\mathrm{c}}|_2^2)\,. \tag{2.99}$$

Consider now the second term on right-hand side, the one dealing with the cell-based DoFs. With simple manipulations and owing to the definition of the discrete divergence in (2.43), one gets

$$\sum_{\mathrm{c}\in\mathrm{C}}\sum_{\mathrm{f}\in\mathrm{F}} |\mathrm{f}|\,(\underline{w}_{\mathrm{f}}\cdot\underline{n}_{\mathrm{fc}})\,|\underline{u}_{\mathrm{c}}|_2^2 = \sum_{\mathrm{c}\in\mathrm{C}} |\underline{u}_{\mathrm{c}}|_2^2 \sum_{\mathrm{f}\in\mathrm{F}} |\mathrm{f}|\,(\underline{w}_{\mathrm{f}}\cdot\underline{n}_{\mathrm{fc}}) = \sum_{\mathrm{c}\in\mathrm{C}} |\mathrm{c}|\,|\underline{u}_{\mathrm{c}}|_2^2\,\mathrm{D}_{\mathrm{c}}(\widehat{\underline{w}}_{\mathrm{c}}) = 0\,, \tag{2.100}$$

where the last equality is obtained owing to the assumption (2.93). Summing (2.98) and (2.99) and using (2.100), one has

$$\sum_{\mathrm{c}\in\mathrm{C}} t_{\mathrm{c}}(\widehat{\underline{w}}_{\mathrm{c}};\widehat{\underline{u}}_{\mathrm{c}},\widehat{\underline{u}}_{\mathrm{c}}) + t_{\mathrm{h}}^{\partial}(\widehat{\underline{w}}_{\mathrm{h}};\widehat{\underline{u}}_{\mathrm{h}},\widehat{\underline{u}}_{\mathrm{h}}) = \sum_{\mathrm{f}\in\mathrm{F}^b} |\mathrm{f}|\,(\underline{w}_{\mathrm{f}}\cdot\underline{n}_{\mathrm{fc}})^+\,|\underline{u}_{\mathrm{f}}|_2^2 \geq 0\,, \tag{2.101}$$

since, for all $x\in\mathbb{R}$, $(x)^- + \frac{1}{2}|x| = (x)^+ \geq 0$. Hence, combining (2.97) and (2.101) yields (2.94), which concludes the first part of the proof.

Let us now prove *ii)*. For this part, we have supposed $\Xi^{\mathrm{upw}} = 0$, hence we drop $t_{\mathrm{h}}^{\mathrm{u}}(\cdot;\cdot,\cdot)$. From (2.95), one deduces that $t_{\mathrm{h}}^{\partial}(\widehat{\underline{w}}_{\mathrm{h}};\widehat{\underline{u}}_{\mathrm{h}},\widehat{\underline{v}}_{\mathrm{h}}) = 0$. Thus, only $t_{\mathrm{c}}(\widehat{\underline{w}}_{\mathrm{h}};\widehat{\underline{u}}_{\mathrm{h}},\widehat{\underline{v}}_{\mathrm{h}})$ is left to analyze. Owing to the first line in (2.92), one may write

$$\begin{aligned} t_{\mathrm{h}}(\widehat{\underline{w}}_{\mathrm{h}};\widehat{\underline{u}}_{\mathrm{h}},\widehat{\underline{v}}_{\mathrm{h}}) = {} & \frac{1}{2}\sum_{\mathrm{c}\in\mathrm{C}}\sum_{\mathrm{f}\in\mathrm{F}_{\mathrm{c}}} |\mathrm{f}|\,(\underline{w}_{\mathrm{f}}\cdot\underline{n}_{\mathrm{fc}})(\underline{u}_{\mathrm{f}}\cdot\underline{v}_{\mathrm{c}} - \underline{u}_{\mathrm{c}}\cdot\underline{v}_{\mathrm{f}}) \\ & + \underbrace{\frac{1}{2}\sum_{\mathrm{c}\in\mathrm{C}}\sum_{\mathrm{f}\in\mathrm{F}_{\mathrm{c}}} |\mathrm{f}|\,(\underline{w}_{\mathrm{f}}\cdot\underline{n}_{\mathrm{fc}})(\underline{u}_{\mathrm{f}}\cdot\underline{v}_{\mathrm{f}})}_{\mathcal{I}_1} - \underbrace{\frac{1}{2}\sum_{\mathrm{c}\in\mathrm{C}}\sum_{\mathrm{f}\in\mathrm{F}_{\mathrm{c}}} |\mathrm{f}|\,(\underline{w}_{\mathrm{f}}\cdot\underline{n}_{\mathrm{fc}})(\underline{u}_{\mathrm{c}}\cdot\underline{v}_{\mathrm{c}})}_{\mathcal{I}_2}\,. \end{aligned} \tag{2.102}$$

The term $\mathcal{I}_2$ is dealt with similarly to (2.100), hence one has

$$\mathcal{I}_2 = \sum_{\mathrm{c}\in\mathrm{C}} \mathrm{D}_{\mathrm{c}}(\widehat{\underline{w}}_{\mathrm{c}})\underline{u}_{\mathrm{c}}\cdot\underline{v}_{\mathrm{c}} = 0\,. \tag{2.103}$$

Let us address $\mathcal{I}_1$. Invert the cell- and face-summations and separate internal and boundary faces:

$$\mathcal{I}_1 = \sum_{\mathrm{f}\in\mathrm{F}^b} |\mathrm{f}|\,(\underline{w}_{\mathrm{f}}\cdot\underline{n}_{\mathrm{f}})(\underline{u}_{\mathrm{f}}\cdot\underline{v}_{\mathrm{f}}) + \sum_{\mathrm{f}\in\mathrm{F}^i}\sum_{\mathrm{c}\in\mathrm{C}_{\mathrm{f}}} |\mathrm{f}|\,(\underline{w}_{\mathrm{f}}\cdot\underline{n}_{\mathrm{fc}})(\underline{u}_{\mathrm{f}}\cdot\underline{v}_{\mathrm{f}}) = 0\,, \tag{2.104}$$

where we concluded using the hypothesis (2.95) for the first term and the usual properties on the summation on internal faces, see Remark 2.13, for the second term. Plugging (2.103) and (2.104) into (2.102) one gets

$$t_{\mathrm{h}}(\widehat{\underline{w}}_{\mathrm{h}};\widehat{\underline{u}}_{\mathrm{h}},\widehat{\underline{v}}_{\mathrm{h}}) = \frac{1}{2}\sum_{\mathrm{c}\in\mathrm{C}}\sum_{\mathrm{f}\in\mathrm{F}_{\mathrm{c}}} |\mathrm{f}|\,(\underline{w}_{\mathrm{f}}\cdot\underline{n}_{\mathrm{fc}})(\underline{u}_{\mathrm{f}}\cdot\underline{v}_{\mathrm{c}} - \underline{u}_{\mathrm{c}}\cdot\underline{v}_{\mathrm{f}})\,. \tag{2.105}$$

This proves that $t_{\mathrm{h}}(\widehat{\underline{w}}_{\mathrm{h}};\cdot,\cdot)$ is skew-symmetric whenever the assumptions (2.93) and (2.95) are satisfied. □

Remark 2.46 - Dissipativity. Property (2.94) is crucial in the context of the NSE since it establishes the dissipativity of the discrete problem, and therefore the possibility of deriving a priori estimates for the discrete NSE. Other schemes (based, e.g., on continuous finite elements or discontinuous Galerkin methods) may not guarantee this dissipativity property. In this context, a popular trick based on Temam (1977) is to add a consistent term in the scheme so that the discrete trilinear form becomes skew-symmetric (or dissipative). Lemma 2.45 shows that the CDO trilinear form $t_{\mathrm{h}}(\cdot;\cdot,\cdot)$ has the Temam's trick built-in (recall the positivity term mentioned in Remark 2.35). ◇

Remark 2.47 - Alternative convection formulations. Other trilinear formulations can be derived by applying differential identities to the operator $(\underline{u} \cdot \underline{\nabla})\underline{u}$. The reader is referred, for instance, to Charnyi *et al.* (2017) which gives an overview of several of these formulations with a particular focus on their conservation properties. The development of these alternative formulations within the CDO-Fb framework is left as future work. ◇

Remark 2.48 - Discrete integration by parts. Following Remark 2.39, let us investigate the global contribution of $t_{\mathrm{c}}(\cdot;\cdot,\cdot)$ (we drop the upwinding and the boundary term in (2.88)):

$$\begin{aligned}\sum_{\mathrm{c}\in\mathrm{C}}\Big(t_{\mathrm{c}}(\widehat{\underline{w}}_{\mathrm{c}};\widehat{\underline{u}}_{\mathrm{c}},\widehat{\underline{v}}_{\mathrm{c}})+t_{\mathrm{c}}(\widehat{\underline{w}}_{\mathrm{c}};\widehat{\underline{v}}_{\mathrm{c}},\widehat{\underline{u}}_{\mathrm{c}})\Big) &= \sum_{\mathrm{c}\in\mathrm{C}}\sum_{\mathrm{f}\in\mathrm{F}_{\mathrm{c}}}|\mathrm{f}|\,(\underline{w}_{\mathrm{f}}\cdot\underline{n}_{\mathrm{fc}})(\underline{u}_{\mathrm{f}}\cdot\underline{v}_{\mathrm{f}}-\underline{u}_{\mathrm{c}}\cdot\underline{v}_{\mathrm{c}})\\ &= -\sum_{\mathrm{c}\in\mathrm{C}}\mathrm{D}_{\mathrm{c}}(\widehat{\underline{w}}_{\mathrm{h}})\underline{u}_{\mathrm{c}}\cdot\underline{v}_{\mathrm{c}}+\sum_{\mathrm{f}\in\mathrm{F}^b}|\mathrm{f}|\,(\underline{w}_{\mathrm{f}}\cdot\underline{n}_{\mathrm{f}})\underline{u}_{\mathrm{f}}\cdot\underline{v}_{\mathrm{f}}\,,\end{aligned}\tag{2.106}$$

where one discards the skew-symmetric terms and follows the steps of Lemma 2.45. Notice that (2.106) is the discrete counterpart of

$$\int_\Omega\bigg(\Big((\underline{w}\cdot\underline{\nabla})\underline{u}\Big)\cdot\underline{v}+\Big((\underline{w}\cdot\underline{\nabla})\underline{v}\Big)\cdot\underline{u}\bigg)=-\int_\Omega(\underline{\nabla}\cdot\underline{w})(\underline{u}\cdot\underline{w})+\int_{\partial\Omega}(\underline{w}\cdot\underline{n}_{\partial\Omega})(\underline{u}\cdot\underline{v})\,.\quad\diamond\tag{2.107}$$

Remark 2.49 - Comparison with similar schemes. As it was mentioned in Remark 2.36, the advection bilinear form $t_{\mathrm{h}}^{\mathrm{s}}(\underline{\beta};\cdot,\cdot)$ was inspired by the HHO framework (see in particular Di Pietro *et al.* (2015)). As matter of fact, the final trilinear form for the NSE is the same for CDO-Fb and HHO($k\coloneqq 0$). Set $\Xi^{\mathrm{upw}}\coloneqq 0$, and compare the definition of $t_{\mathrm{h}}(\cdot;\cdot,\cdot)$ in Eqs. (2.88) to (2.91) and the form provided in Botti *et al.* (2019, Remark 9) (notice that the gradients of cell-based functions vanish in HHO($k\coloneqq 0$) since one considers cell-wise constant functions).

In the HMM framework, two different convection operators have been proposed in Droniou and Eymard (2017). In the first version, initially proposed and analyzed in Droniou and Eymard (2009), the face-based DoFs of the convection field are considered and they are tested with the jump (respectively the mean) at the faces of the cell-based DoFs of the velocity (resp. test function). In doing so, the cell-based DoFs of the velocity become coupled and static condensation is no longer possible. In the second version of the discrete convection operator, the authors consider an additional local problem defined on the face-based DoFs of the velocity and fluxes at the faces of the barycentric subdivision of the cell (e.g. the faces shared by the subpyramids, see Fig. 2.1). The resulting convection operator does not use cell-based DoFs, neither for the convection field, nor for the velocity, nor for the test function: a static condensation process can be thus performed as in the present CDO-Fb scheme. ◇

Remark 2.50 - Convection limit-conformity. We show here that the discrete trilinear form $t_{\mathrm{h}}(\cdot;\cdot,\cdot)$ satisfies the requirement of *convection limit-conformity* as stated in Eymard *et al.* (2018, Definition 2.10). For simplicity, we assume that no upwind stabilization is considered, $\Xi^{\mathrm{upw}}\coloneqq 0$, and that the convection field is zero at the boundary, $\underline{w}_{\mathrm{f}}=\underline{0}$ for all $\mathrm{f}\in\mathrm{F}^b$. Owing to these assumptions, we rewrite the discrete trilinear form as follows: for all $\widehat{\underline{w}}_{\mathrm{h}}$, $\widehat{\underline{u}}_{\mathrm{h}}$, $\widehat{\underline{v}}_{\mathrm{h}}\in\widehat{\underline{\mathcal{U}}}_{\mathrm{h},0}$

$$\begin{aligned}t_{\mathrm{h}}(\widehat{\underline{w}}_{\mathrm{h}};\widehat{\underline{u}}_{\mathrm{h}},\widehat{\underline{v}}_{\mathrm{h}}) &= \frac{1}{2}\sum_{\mathrm{c}\in\mathrm{C}}\sum_{\mathrm{f}\in\mathrm{F}_{\mathrm{c}}}(\underline{w}_{\mathrm{f}}\cdot\underline{n}_{\mathrm{fc}})(\underline{u}_{\mathrm{f}}-\underline{u}_{\mathrm{c}})\cdot(\underline{v}_{\mathrm{f}}+\underline{v}_{\mathrm{c}})\\ &= \frac{1}{2}\sum_{\mathrm{c}\in\mathrm{C}}\int_{\mathrm{c}}\Big(\big(\widetilde{\underline{\mathrm{G}}}_{\mathrm{c}}(\widehat{\underline{u}}_{\mathrm{c}})\underline{\Pi}_{\mathrm{c}}^{\mathrm{F}}(\widehat{\underline{w}}_{\mathrm{c}})\big)\cdot\underline{\Pi}_{\mathrm{c}}^{\mathrm{C}}(\widehat{\underline{v}}_{\mathrm{c}})-\big(\widetilde{\underline{\mathrm{G}}}_{\mathrm{c}}(\widehat{\underline{v}}_{\mathrm{c}})\underline{\Pi}_{\mathrm{c}}^{\mathrm{F}}(\widehat{\underline{w}}_{\mathrm{c}})\big)\cdot\underline{\Pi}_{\mathrm{c}}^{\mathrm{C}}(\widehat{\underline{u}}_{\mathrm{c}})\\ &\qquad+\mathrm{D}_{\mathrm{c}}(\widehat{\underline{w}}_{\mathrm{c}})\underline{\Pi}_{\mathrm{c}}^{\mathrm{C}}(\widehat{\underline{u}}_{\mathrm{c}})\cdot\underline{\Pi}_{\mathrm{c}}^{\mathrm{C}}(\widehat{\underline{v}}_{\mathrm{c}})\Big)\,,\end{aligned}\tag{2.108}$$

where

$$\begin{aligned}
&\underline{\Pi}_{\mathrm{c}}^{\mathrm{C}}(\widehat{\underline{v}}_{\mathrm{c}}) := \underline{v}_{\mathrm{c}}\,, && \forall \mathrm{c} \in \mathrm{C}\,,\\
&\underline{\Pi}_{\mathrm{c}}^{\mathrm{F}}(\widehat{\underline{v}}_{\mathrm{c}})_{|\mathfrak{p}_{\mathrm{f,c}}} := \underline{v}_{\mathrm{f}}\,, && \forall \mathrm{c} \in \mathrm{C},\ \forall \mathrm{f} \in \mathrm{F}_{\mathrm{c}}\,,\\
&\widetilde{\underline{\underline{\mathrm{G}}}}_{\mathrm{c}}|_{\mathfrak{p}_{\mathrm{f,c}}} := \frac{|\mathrm{f}|}{|\mathfrak{p}_{\mathrm{f,c}}|}(\underline{v}_{\mathrm{f}} - \underline{v}_{\mathrm{c}}) \otimes \underline{n}_{\mathrm{fc}}\,, && \forall \mathrm{c} \in \mathrm{C},\ \forall \mathrm{f} \in \mathrm{F}_{\mathrm{c}}\,,
\end{aligned} \tag{2.109}$$

and we define the global counterparts of these operators such that for all $\mathrm{c} \in \mathrm{C}$,

$$\underline{\Pi}_{\mathrm{h}}^{\mathrm{C}}(\widehat{\underline{v}}_{\mathrm{h}})_{|\mathrm{c}} := \underline{\Pi}_{\mathrm{c}}^{\mathrm{C}}(\widehat{\underline{v}}_{\mathrm{c}})\,, \qquad \underline{\Pi}_{\mathrm{h}}^{\mathrm{F}}(\widehat{\underline{v}}_{\mathrm{h}})_{|\mathrm{c}} := \underline{\Pi}_{\mathrm{c}}^{\mathrm{F}}(\widehat{\underline{v}}_{\mathrm{c}})\,, \qquad \widetilde{\underline{\underline{\mathrm{G}}}}_{\mathrm{h}}(\widehat{\underline{v}}_{\mathrm{h}})_{|\mathrm{c}} := \widetilde{\underline{\underline{\mathrm{G}}}}_{\mathrm{c}}(\widehat{\underline{v}}_{\mathrm{c}})\,. \tag{2.110}$$

We study now some properties of these new operators. We proceed as in Lemma 2.23, using in particular (2.39)-(2.40), to obtain

$$\left\|\widetilde{\underline{\underline{\mathrm{G}}}}_{\mathrm{h}}(\widehat{\underline{v}}_{\mathrm{h}})\right\|_{\underline{\underline{L}}^2(\Omega)} \lesssim \left\|\underline{\underline{\mathrm{G}}}_{\mathrm{h}}(\widehat{\underline{v}}_{\mathrm{h}})\right\|_{\underline{\underline{L}}^2(\Omega)}\,. \tag{2.111}$$

Let us assume that $d = 3$ and let $p \in [2, 6]$ (if $d = 2$ we can take $p \in [2, \infty)$). Let $[\![\underline{v}_{\mathrm{C}}]\!]_{\mathrm{f}}$ be the jump of $\underline{v}_{\mathrm{C}}$ across the face f. We have

$$\begin{aligned}
\left\|\underline{\Pi}_{\mathrm{h}}^{\mathrm{C}}(\widehat{\underline{v}}_{\mathrm{h}})\right\|_{\underline{L}^p(\Omega)}^p &= \sum_{\mathrm{c}\in\mathrm{C}} |\mathrm{c}|\,|\underline{v}_{\mathrm{c}}|^p \lesssim \left(\sum_{\mathrm{f}\in\mathrm{F}} \frac{|\mathrm{f}|}{\mathrm{h}_{\mathrm{c}}}\,|[\![\underline{v}_{\mathrm{C}}]\!]_{\mathrm{f}} \pm \underline{v}_{\mathrm{f}}|_2^2\right)^{\frac{p}{2}}\\
&\lesssim \left(\sum_{\mathrm{c}\in\mathrm{C}}\sum_{\mathrm{f}\in\mathrm{F}_{\mathrm{c}}} \frac{|\mathrm{f}|}{\mathrm{h}_{\mathrm{c}}}\,|\underline{v}_{\mathrm{f}} - \underline{v}_{\mathrm{c}}|_2^2\right)^{\frac{p}{2}}\\
&= \|\widehat{\underline{v}}_{\mathrm{h}}\|_{1,\mathrm{h}}^p \lesssim \left\|\underline{\underline{\mathrm{G}}}_{\mathrm{h}}(\widehat{\underline{v}}_{\mathrm{h}})\right\|_{\underline{\underline{L}}^2(\Omega)}^p\,,
\end{aligned} \tag{2.112}$$

where we used a discrete Sobolev inequality in the first line (see, for instance, Eymard *et al.* (2010)) and Lemma 2.23 in the third line. Similarly, for $\underline{\Pi}_{\mathrm{h}}^{\mathrm{F}}$, we obtain:

$$\begin{aligned}
\left\|\underline{\Pi}_{\mathrm{h}}^{\mathrm{F}}(\widehat{\underline{v}}_{\mathrm{h}})\right\|_{\underline{L}^p(\Omega)}^p &= \sum_{\mathrm{c}\in\mathrm{C}}\sum_{\mathrm{f}\in\mathrm{F}_{\mathrm{c}}} |\mathfrak{p}_{\mathrm{f,c}}|\,|\underline{v}_{\mathrm{f}}|^p \lesssim \left(\sum_{\mathrm{c}\in\mathrm{C}}\sum_{\mathrm{f}'\in\mathrm{F}_{\mathfrak{P}_{\mathrm{c}}}} \frac{|\mathrm{f}'|}{\mathrm{h}_{\mathrm{c}}}\,\|[\![\underline{v}_{\mathrm{F}}]\!]_{\mathrm{f}'} \pm \underline{v}_{\mathrm{c}}\|_2^2\right)^{\frac{p}{2}}\\
&\lesssim \left(\sum_{\mathrm{c}\in\mathrm{C}}\sum_{\mathrm{f}\in\mathrm{F}_{\mathrm{c}}} \frac{|\mathrm{f}|}{\mathrm{h}_{\mathrm{c}}}\,|\underline{v}_{\mathrm{f}} - \underline{v}_{\mathrm{c}}|_2^2\right)^{\frac{p}{2}}\\
&= \|\widehat{\underline{v}}_{\mathrm{h}}\|_{1,\mathrm{h}}^p \lesssim \left\|\underline{\underline{\mathrm{G}}}_{\mathrm{h}}(\widehat{\underline{v}}_{\mathrm{h}})\right\|_{\underline{\underline{L}}^2(\Omega)}^p\,,
\end{aligned} \tag{2.113}$$

where $\mathrm{F}_{\mathfrak{P}_{\mathrm{c}}}$ collects all the internal faces of $\mathfrak{P}_{\mathrm{c}}$, the pyramidal subdivision of c, and where we used again a discrete Sobolev inequality and Lemma 2.23. Equations (2.112) and (2.113) are called *p-coercivity* in Eymard *et al.* (2018, Definition A.1) and here we need this property for $p > 4$. We can now proceed as in the proof of Eymard *et al.* (2018, Lemma A.2) and obtain in particular

$$\lim_{\mathrm{h}\to 0} t_{\mathrm{h}}(\widehat{\underline{w}}_{\mathrm{h}}; \widehat{\underline{u}}_{\mathrm{h}}, \widehat{\underline{v}}_{\mathrm{h}}) = \frac{1}{2}\big(t(\underline{w}; \underline{u}, \underline{v}) - t(\underline{w}; \underline{v}, \underline{u})\big)\,. \tag{2.114}$$

A similar result holds for the HHO convection form as well and it has been defined *sequentially consistency* and proven in Di Pietro and Krell (2018, Prop. 6). ◇

2.6 Source term

The last brick that we will need to build the CDO-Fb discrete NSE is how to deal with a possible source term in the momentum balance equation. The aim is to approximate the linear form at the right-hand side of the classical variational formulation:

$$l(\underline{v}) \coloneqq \int_\Omega \underline{f} \cdot \underline{v} \,. \tag{2.115}$$

The key point is to define a lifting (or reconstruction) operator, denoted by $\underline{\mathtt{L}}_{\mathrm{h}}$, that takes a generic discrete hybrid element of $\widehat{\underline{\mathcal{U}}}_{\mathrm{h},0}$ and gives a continuous (or smooth enough) function which can be tested against the body force, so that one can approximate (2.115) as follows:

$$l(\underline{v}) = \int_\Omega \underline{f} \cdot \underline{v} \approx \int_\Omega \underline{f} \cdot \underline{\mathtt{L}}_{\mathrm{h}}(\widehat{\underline{v}}_{\mathrm{h}}) =: l(\underline{\mathtt{L}}_{\mathrm{h}}(\widehat{\underline{v}}_{\mathrm{h}}))\,. \tag{2.116}$$

The straightforward choice is to extract only the cell-based values of the velocity, that is, such that $\underline{\mathtt{L}}_{\mathrm{h}}(\widehat{\underline{v}}_{\mathrm{h}})_{|\mathrm{c}} \coloneqq \underline{v}_{\mathrm{c}}$ for all $\mathrm{c} \in \mathrm{C}$ and all $\widehat{\underline{v}}_{\mathrm{h}} \in \widehat{\underline{\mathcal{U}}}_{\mathrm{h},0}$. Then, starting from the right-hand side of (2.116), one obtains:

$$\underline{\mathtt{L}}_{\mathrm{h}}(\widehat{\underline{v}}_{\mathrm{h}}) \coloneqq \underline{v}_{\mathrm{C}}\,, \qquad l(\underline{\mathtt{L}}_{\mathrm{h}}(\widehat{\underline{v}}_{\mathrm{h}})) = \int_\Omega \underline{f} \cdot \widehat{\underline{v}}_{\mathrm{C}} = \sum_{\mathrm{c}\in\mathrm{C}} \int_{\mathrm{c}} \underline{f} \cdot \underline{v}_{\mathrm{c}} = \sum_{\mathrm{c}\in\mathrm{C}} \underline{v}_{\mathrm{c}} \cdot \left(\int_{\mathrm{c}} \underline{f}\right)\,. \tag{2.117}$$

Equation (2.117) is the discrete formulation of the source term that we will employ in this Thesis. Hence, in practice, one only needs to integrate the body force in each mesh cell.

Remark 2.51 - Alternative choices. Other choices are possible and may involve a different definition of $\underline{\mathtt{L}}_{\mathrm{h}}$ or a modification of the linear form (2.115). One could take advantage of $\underline{\underline{\mathrm{G}}}_{\mathrm{h}}$ and use it in a Taylor expansion: For all $\mathrm{c} \in \mathrm{C}$ and all $\mathrm{f} \in \mathrm{F}_{\mathrm{c}}$, we can set

$$\underline{\mathtt{L}}_{\mathrm{h}}(\widehat{\underline{v}}_{\mathrm{h}})(\underline{x}) \coloneqq \underline{v}_{\mathrm{c}} + \underline{\underline{\mathrm{G}}}_{\mathrm{c}}(\widehat{\underline{v}}_{\mathrm{c}})_{|\mathfrak{p}_{\mathrm{f},\mathrm{c}}}(\underline{x} - \underline{x}_{\mathrm{c}}) \qquad \forall \underline{x} \in \mathfrak{p}_{\mathrm{f},\mathrm{c}}\,. \tag{2.118}$$

This choice is reminiscent of the reconstruction operators commonly used in the HHO framework in the lowest-order case $k \coloneqq 0$, see, for instance, Di Pietro *et al.* (2014, Eq. (15)).

Furthermore, the linear form (2.115) may accommodated incorporate information about the forcing term. For instance, suppose that there exists $\phi \in H^1(\Omega)$ such that $\underline{f} = \underline{\nabla}\,\phi$, and that the potential ϕ is known. Then, integrating by parts, we have at continuous level

$$\int_\Omega \underline{f} \cdot \underline{v} = \sum_{\mathrm{f}\in\mathrm{F}^b} \int_{\mathrm{f}} \phi(\underline{v} \cdot \underline{n}_{\mathrm{f}}) - \sum_{\mathrm{c}\in\mathrm{C}} \int_{\mathrm{c}} \phi(\underline{\nabla} \cdot \underline{v})\,. \tag{2.119}$$

Now, plugging a CDO-Fb discretization in (2.119), one may approximate the right-hand side as follows:

$$\int_\Omega \underline{f} \cdot \underline{v} \approx \sum_{\mathrm{f}\in\mathrm{F}^b} (\underline{v}_{\mathrm{f}} \cdot \underline{n}_{\mathrm{f}}) \left(\int_{\mathrm{f}} \phi\right) - \sum_{\mathrm{c}\in\mathrm{C}} \mathrm{D}_{\mathrm{c}}(\widehat{\underline{v}}_{\mathrm{c}}) \left(\int_{\mathrm{c}} \phi\right)\,. \tag{2.120}$$

The main difference between (2.117) and the two alternatives proposed in this remark is that in (2.117) only cell-based DoFs of $\widehat{\underline{v}}_{\mathrm{h}}$ are relevant, whereas both the face- and the cell-based DoFs are employed in (2.118) and (2.119). ◇

Chapter 3

The steady Navier–Stokes equations

Contents

3.1 Stokes equations with face-based CDO 62
3.1.1 Continuous formulation 62
3.1.2 CDO formulation 62
3.1.3 Algebraic viewpoint 64
3.2 Navier–Stokes equations with face-based CDO 68
3.2.1 Continuous formulation 68
3.2.2 CDO formulation 69
3.2.3 Algebraic viewpoint 70
3.3 Preliminary numerical setting 71
3.3.1 Meshes 71
3.3.2 Error norms and quadrature rules 72
3.3.3 Implementation 75
3.4 Numerical results: Stokes equations 75
3.4.1 2D Bercovier–Engelman 76
3.4.2 3D modified Taylor–Green Vortex 77
3.5 Numerical results: Navier–Stokes equations 80
3.5.1 2D Burggraf flow 81
3.5.2 3D Modified Taylor–Green Vortex 82
3.5.3 2D lid-driven cavity 84

In this chapter, the CDO-Fb operators introduced in Chapter 2 are used to discretize the Stokes and Navier–Stokes equations (NSE). Only steady problems are considered in this chapter. Hence, the target problem is: Find $(\underline{u}, p) \in \underline{H}^1_0(\Omega) \times L^2_*(\Omega)$, such that

$$\begin{cases} -\nu \underline{\Delta}\, \underline{u} + \xi^{\mathrm{NS}} (\underline{u} \cdot \underline{\nabla}\,) \underline{u} + \underline{\nabla}\, p = \underline{f} & \text{in } \Omega\,, \\ \underline{\nabla} \cdot \underline{u} = 0 & \text{in } \Omega\,, \\ \underline{u} = \underline{u}_\partial & \text{on } \partial\Omega\,, \end{cases} \tag{3.1}$$

where $\nu > 0$ is the viscosity, and ξ^{NS} is a dummy parameter which lets us recover the Stokes, $\xi^{\mathrm{NS}} := 0$, or the NSE, $\xi^{\mathrm{NS}} := 1$. For the sake of simplicity, we will mostly deal with homogeneous Dirichlet Boundary Conditions (BCs), $\underline{u}_\partial := \underline{0}$ on $\partial\Omega$, but other types

of BCs can be considered. In the first two sections of this chapter we apply the CDO-Fb framework to devise discrete versions of the Stokes and NSE. Then we present the setting for our numerical experiments. Finally, in the last two sections of this chapter, numerical computations are presented in order to verify the soundness of the design and of the implementation of the CDO-Fb schemes. The results presented in this chapter have been partly presented in Bonelle, Ern, and Milani (2020).

3.1 Stokes equations with face-based CDO

In this section, the classical variational formulation of the Stokes equations is translated into the CDO-Fb framework. Some properties of the resulting discrete problem are studied, and an algebraic and more implementation-oriented vision is given as well.

3.1.1 Continuous formulation

The classical (continuous) variational formulation, already briefly introduced in Section 1.3.2, involves the functional spaces

$$\underline{H}_0^1(\Omega) := \{\underline{v} \in \underline{H}^1(\Omega) \,|\, \underline{v}_{|\partial\Omega} = \underline{0}\}\,, \tag{3.2a}$$

$$L_*^2(\Omega) := \{q \in L^2(\Omega) \,|\, \int_\Omega q = 0\}\,. \tag{3.2b}$$

The variational formulation then reads: Find $(\underline{u}, p) \in \underline{H}_0^1(\Omega) \times L_*^2(\Omega)$ such that

$$\begin{cases} \nu a(\underline{u}, \underline{v}) + b(\underline{v}, p) = l(\underline{v}) & \forall \underline{v} \in \underline{H}_0^1(\Omega)\,, \\ b(\underline{u}, q) = 0 & \forall q \in L_*^2(\Omega)\,, \end{cases} \tag{3.3}$$

where

$$\begin{gathered} a(\underline{u}, \underline{v}) := \int_\Omega \underline{\underline{\nabla}}\, \underline{u} : \underline{\underline{\nabla}}\, \underline{v}\,, \qquad b(\underline{u}, q) := -\int_\Omega q \underline{\nabla} \cdot \underline{u}\,, \\ l(\underline{v}) := \int_\Omega \underline{f} \cdot \underline{v}\,. \end{gathered} \tag{3.4}$$

Remark 3.1 - Pressure test space. In (3.3), it is possible to consider the larger pressure test space $L^2(\Omega)$, since it is readily seen that

$$-\, b(\underline{u}, 1) = \int_\Omega \underline{\nabla} \cdot \underline{u} = \int_{\partial\Omega} \underline{u} \cdot \underline{n}_{\partial\Omega} = 0\,, \tag{3.5}$$

since $\underline{u} \in \underline{H}_0^1(\Omega)$. Seeking $p \in L_*^2(\Omega)$ classically allows one to uniquely define the pressure. ◇

3.1.2 CDO formulation

Let us introduce appropriate discrete functional spaces for the above problem. Recall that

$$\widehat{\underline{\mathcal{U}}}_{\mathrm{h}} := \bigtimes_{\mathrm{f}\in\mathrm{F}} \underline{\mathcal{U}}_{\mathrm{f}} \times \bigtimes_{\mathrm{c}\in\mathrm{C}} \underline{\mathcal{U}}_{\mathrm{c}}\,, \tag{3.6}$$

$$\mathcal{P}_{\mathrm{h}} := \bigtimes_{\mathrm{c}\in\mathrm{C}} \mathcal{P}_{\mathrm{c}}\,. \tag{3.7}$$

The related spaces where the homogeneous Dirichlet boundary conditions over the whole boundary $\partial\Omega$ for the velocity and the zero mean-value constraint on the pressure have been

enforced as follows:

$$\widehat{\underline{\mathcal{U}}}_{\mathrm{h},0} \coloneqq \left\{ \widehat{\underline{v}}_{\mathrm{h}} \in \widehat{\underline{\mathcal{U}}}_{\mathrm{h}} \,|\, \underline{v}_{\mathrm{f}} = \underline{0} \ \forall \mathrm{f} \in \mathrm{F}^b \right\} , \tag{3.8}$$

$$\mathcal{P}_{\mathrm{h},*} \coloneqq \left\{ q_{\mathrm{h}} \in \mathcal{P}_{\mathrm{h}} \,|\, \sum_{\mathrm{c}\in\mathrm{C}} |\mathrm{c}|\, q_{\mathrm{c}} = 0 \right\} . \tag{3.9}$$

Recall that these spaces are equipped with the norms $\|\cdot\|_{1,\mathrm{h}}$ and $\|\cdot\|_{\mathrm{h}}$ defined (2.32) in and (2.55), respectively.

The approach to build the discrete bilinear forms consists in considering the local ones for any cell $\mathrm{c} \in \mathrm{C}$, and then summing them together. The problem hence reads: Find $(\widehat{\underline{u}}_{\mathrm{h}},\, p_{\mathrm{h}}) \in \widehat{\underline{\mathcal{U}}}_{\mathrm{h},0} \times \mathcal{P}_{\mathrm{h},*}$ such that:

$$\begin{cases} \nu a_{\mathrm{h}}(\widehat{\underline{u}}_{\mathrm{h}}, \widehat{\underline{v}}_{\mathrm{h}}) + b_{\mathrm{h}}(\widehat{\underline{v}}_{\mathrm{h}}, p_{\mathrm{h}}) = l(\underline{v}_{\mathrm{C}}) & \forall \widehat{\underline{v}}_{\mathrm{h}} \in \widehat{\underline{\mathcal{U}}}_{\mathrm{h},0} , \\ b_{\mathrm{h}}(\widehat{\underline{u}}_{\mathrm{h}}, q_{\mathrm{h}}) = 0 & \forall q_{\mathrm{h}} \in \mathcal{P}_{\mathrm{h},*} , \end{cases} \tag{3.10}$$

with (recall (2.56) and (2.117))

$$\begin{aligned} & a_{\mathrm{h}}(\widehat{\underline{u}}_{\mathrm{h}}, \widehat{\underline{v}}_{\mathrm{h}}) \coloneqq \sum_{\mathrm{c}\in\mathrm{C}} a_{\mathrm{c}}(\widehat{\underline{u}}_{\mathrm{c}}, \widehat{\underline{v}}_{\mathrm{c}}) , && a_{\mathrm{c}}(\widehat{\underline{u}}_{\mathrm{c}}, \widehat{\underline{v}}_{\mathrm{c}}) \coloneqq \int_{\mathrm{c}} \underline{\underline{\mathrm{G}}}_{\mathrm{c}}(\widehat{\underline{u}}_{\mathrm{c}}) : \underline{\underline{\mathrm{G}}}_{\mathrm{c}}(\widehat{\underline{v}}_{\mathrm{c}}) , \\ & b_{\mathrm{h}}(\widehat{\underline{v}}_{\mathrm{h}}, p_{\mathrm{h}}) \coloneqq \sum_{\mathrm{c}\in\mathrm{C}} b_{\mathrm{c}}(\widehat{\underline{v}}_{\mathrm{c}}, p_{\mathrm{c}}) , && b_{\mathrm{c}}(\widehat{\underline{v}}_{\mathrm{c}}, p_{\mathrm{c}}) \coloneqq -\int_{\mathrm{c}} \mathrm{D}_{\mathrm{c}}(\widehat{\underline{v}}_{\mathrm{c}}) q_{\mathrm{c}} = -\,|\mathrm{c}|\, \mathrm{D}_{\mathrm{c}}(\widehat{\underline{v}}_{\mathrm{c}}) q_{\mathrm{c}} , \\ & l(\underline{v}_{\mathrm{C}}) \coloneqq \sum_{\mathrm{c}\in\mathrm{C}} \int_{\mathrm{c}} \underline{f} \cdot \underline{v}_{\mathrm{c}} . \end{aligned} \tag{3.11}$$

Remark 3.2 - Pressure test space. Similarly to what has been pointed out in Remark 3.1, the whole discrete space $\mathcal{P}_{\mathrm{h}}$ can be used as the pressure test space, since

$$-\, b_{\mathrm{h}}(\widehat{\underline{u}}_{\mathrm{c}}, 1) = \sum_{\mathrm{c}\in\mathrm{C}} |\mathrm{c}|\, \mathrm{D}_{\mathrm{c}}(\widehat{\underline{u}}_{\mathrm{h}}) = \sum_{\mathrm{f}\in\mathrm{F}^i} \sum_{\mathrm{c}\in\mathrm{C}_{\mathrm{f}}} |\mathrm{f}|\, \underline{u}_{\mathrm{f}} \cdot \underline{n}_{\mathrm{fc}} + \sum_{\mathrm{f}\in\mathrm{F}^b} |\mathrm{f}|\, \underline{u}_{\mathrm{f}} \cdot \underline{n}_{\mathrm{f}} = \sum_{\mathrm{f}\in\mathrm{F}^b} |\mathrm{f}|\, \underline{u}_{\mathrm{f}} \cdot \underline{n}_{\mathrm{f}} = 0 , \tag{3.12}$$

where we used the definition (3.11) of $b_{\mathrm{h}}(\cdot,\cdot)$ and the definition (2.43) of D_{c}, exchanged the order of summation between cells and faces, used the single-valuedness of $\underline{u}_{\mathrm{f}}$ and the skew-symmetry of $\underline{n}_{\mathrm{fc}}$ on two adjacent cells (cf. Remark 2.13), and finally used the boundary conditions. ◇

It has been proved in Lemma 2.33, that $\widehat{\underline{\mathcal{U}}}_{\mathrm{h},0}$, $\mathcal{P}_{\mathrm{h},*}$, and $b_{\mathrm{h}}(\cdot,\cdot)$ are such that there exists $\beta_* > 0$ verifying, for all $\mathrm{h} \in \mathrm{H}$,

$$\inf_{q_{\mathrm{h}}\in\mathcal{P}_{\mathrm{h},*}} \sup_{\widehat{\underline{v}}_{\mathrm{h}}\in\widehat{\underline{\mathcal{U}}}_{\mathrm{h},0}} \frac{b_{\mathrm{h}}(\widehat{\underline{v}}_{\mathrm{h}}, q_{\mathrm{h}})}{\|q_{\mathrm{h}}\|_{\mathrm{h}}\, \|\widehat{\underline{v}}_{\mathrm{h}}\|_{1,\mathrm{h}}} \geq \beta_* . \tag{3.13}$$

This property in turn ensures that the problem (3.10) is well-posed.

Exploiting the HHO error analysis in the lowest-order (i.e., choosing $k \coloneqq 0$ as the polynomial degree), see Di Pietro *et al.* (2016, Thm. 7), we can state the following convergence result:

Lemma 3.3 - Error estimate. *Let* $(\underline{u},\, p) \in \underline{H}^1_0(\Omega) \times L^2_*(\Omega)$ *and* $(\widehat{\underline{u}}_{\mathrm{h}},\, p_{\mathrm{h}}) \in \widehat{\underline{\mathcal{U}}}_{\mathrm{h},0} \times \mathcal{P}_{\mathrm{h},*}$ *be the solutions to* (3.3) *and* (3.10)*, respectively. Suppose additionally that* $\underline{u} \in \underline{H}^2(\Omega)$ *and* $p \in H^1(\Omega)$*. Then the following holds true:*

$$\|\widehat{\underline{\pi}}_{\mathrm{h}}(\underline{u}) - \widehat{\underline{u}}_{\mathrm{h}}\|_{1,\mathrm{h}} \lesssim \mathrm{h} \left(\|\underline{u}\|_{\underline{H}^2(\Omega)} + \frac{1}{\nu} \|p\|_{H^1(\Omega)} \right) , \tag{3.14a}$$

$$\|\pi_{\mathrm{h}}(p) - p_{\mathrm{h}}\|_{L^2(\Omega)} \lesssim \mathrm{h} \left(\nu \|\underline{u}\|_{\underline{H}^2(\Omega)} + \|p\|_{H^1(\Omega)} \right) . \tag{3.14b}$$

Moreover, if the assumption on full elliptic regularity pickup holds true, then

$$\|\underline{\pi}_{\mathrm{c}}(\underline{u}) - \underline{u}_{\mathrm{c}}\|_{\underline{L}^2(\Omega)} \lesssim \mathrm{h}^2 \left(\|\underline{u}\|_{\underline{H}^2(\Omega)} + \frac{1}{\nu} \|p\|_{H^1(\Omega)} \right) . \tag{3.15}$$

Remark 3.4 - Pressure-robustness. The error estimates from Lemma 3.3 show that the present CDO-Fb discretization is not pressure-robust. This means that the velocity error will suffer from pollution coming from the pressure error as soon as the pressure cannot be represented exactly in the discrete space $\mathcal{P}_{\mathrm{h},*}$. Because of the factor $\frac{1}{\nu}$ on the right-hand side of (3.14a), this issue becomes problematic as $\nu \to 0^+$, i.e. in the context of the NSE, as the Reynolds number becomes large. A way to ensure the pressure-robustness of the scheme is to define an appropriate lifting operator $\underline{\mathtt{L}}_{\mathrm{h}}$ to use with test functions when dealing with the source term, i.e., one replaces $l(\underline{v}_{\mathrm{C}})$ by $l(\underline{\mathtt{L}}_{\mathrm{h}}(\widehat{\underline{v}}_{\mathrm{h}}))$ in (3.10). In particular, one should use $\underline{\mathtt{L}}_{\mathrm{h}} : \widehat{\underline{\mathcal{U}}}_{\mathrm{h}} \to H(div, \Omega)$ such that, for all $\widehat{\underline{v}}_{\mathrm{h}} \in \widehat{\underline{\mathcal{U}}}_{\mathrm{h},0}$ verifying $\mathrm{D}_{\mathrm{h}}(\widehat{\underline{v}}_{\mathrm{h}}) = 0$, then $\underline{\nabla} \cdot \underline{\mathtt{L}}_{\mathrm{h}}(\widehat{\underline{v}}_{\mathrm{h}}) = 0$. This new definition ensures that, whenever there exists $\phi \in H^1(\Omega)$ such that $\underline{f} = \underline{\nabla}\,\phi$, then $\int_\Omega \underline{f} \cdot \underline{\mathtt{L}}_{\mathrm{h}}(\widehat{\underline{v}}_{\mathrm{h}}) = 0$ if $\mathrm{D}_{\mathrm{h}}(\widehat{\underline{v}}_{\mathrm{h}}) = 0$. Further details on how to build such a reconstruction operator using Raviart–Thomas finite elements on simplicial meshes can be found in Section 3.4 of Di Pietro *et al.* (2016) to which the reader is referred to, see also Thm. 4 therein for the resulting convergence result and Remarks 5 and 11 therein for the differences with other schemes from the literature. The idea of modifying the test functions on the right-hand side of the momentum balance equation to ensure pressure-robustness has been developed in Linke (2014), see also John *et al.* (2017) and Lederer *et al.* (2017). ◇

Remark 3.5 - Other BCs. In the presentation of the problem, both in the continuous and discrete settings, homogeneous Dirichlet BCs are considered for the sake of simplicity, and they are taken into account directly in the functional spaces (cf. (3.2a)). However, other BCs can be considered. The test cases presented in this Thesis mainly deal with non-homogeneous Dirichlet BCs. Other BCs can be considered, for instance, Neumann, outlet, Robin, slip (see also Section 3.3.1), or periodicity. ◇

Remark 3.6 - Dirichlet BCs treatment. In practice, the boundary DoFs related to Dirichlet BCs are left in the system. Several ways are available to deal with them, most of them are briefly introduced in Ern and Guermond (2004, Ch. 8.4): an algebraic manipulation virtually eliminating the related DoFs, and the Nitsche's boundary-penalty method (Nitsche, 1971; Freund and Stenberg, 1995; Juntunen and Stenberg, 2009; Burman, 2012). We mostly use the algebraic manipulation: it is a very straightforward procedure, with no additional arbitrary parameter, and, even if the symmetry of the matrix may be impacted, iterative solvers maintain good performances (see, e.g. Ern and Guermond (2004, Prop. 8.18)). The procedure is detailed later in this section, see for instance (3.27)-(3.29). ◇

3.1.3 Algebraic viewpoint

We are going to give here the algebraic structure of the CDO-Fb formulation (3.10), that is, how the discrete problem (3.10) is recast into matrices. For the sake of simplicity, we are going to deal with a local problem concerning only a given mesh cell $\mathrm{c} \in \mathrm{C}$. This will allow us to have a better view on how the operators act on the different DoFs. We use the notation $\mathbf{0}_d$ for the null vector in $\mathbb{R}^d$

Setting

The algebraic vector $\mathbf{U}_\mathrm{c} \in \mathbb{R}^{d(\#\mathrm{F_c}+1)+1}$ gathers face- and cell-based velocity DoFs as well as the pressure DoF, all related to the cell c:

$$\mathbf{U}_\mathrm{c} = [\overbrace{\mathbf{u}_{\mathrm{f}_1^i}^T \ \dots \ \vdots \ \mathbf{u}_{\mathrm{f}_1^b}^T \ \dots}^{d\#\mathrm{F_c}} \ | \ \overbrace{\mathbf{u}_\mathrm{c}^T}^{d} \ \| \ \overbrace{p_\mathrm{c}}^{1} \,]^T . \tag{3.16}$$

Every vector $\mathbf{u}_z \in \mathbb{R}^d$ in (3.16) contains the velocity DoFs associated with the generic mesh entity z, z denoting a face $\mathrm{f} \in \mathrm{F_c}$ or the cell $\mathrm{c} \in \mathrm{C}$. We have arbitrarily chosen an order: the pressure DoF comes last, just after the cell-based velocity ones; among the face-based ones, those related to internal faces (denoted by f_n^i) precede those associated with the boundary faces (f_n^b), if any. The vertical bars in (3.16) are just a graphical expedient to stress the different types of DoFs: dashed lines $\vdots$ separate internal and boundary faces, single lines $|$ faces and cell DoFs, double lines $\|$ velocity and pressure DoFs. The overbraces tell us the size of each subarray. The vector of the right-hand size of the system, $\mathbf{F}_\mathrm{c}$, is similarly structured.

Diffusive contribution

Let us start by investigating the diffusion term. The matrix $\mathbf{G}_\mathrm{c} \in \mathbb{R}^{\widetilde{d}_\mathrm{G} \times \widetilde{d}_\mathrm{G}}$, with $\widetilde{d}_\mathrm{G} = d(\#\mathrm{F_c}+1)$, corresponding to $\int_\mathrm{c} \underline{\underline{\mathrm{G}}}_\mathrm{c}(\widehat{\underline{u}}_\mathrm{c}) : \underline{\underline{\mathrm{G}}}_\mathrm{c}(\widehat{\underline{v}}_\mathrm{c})$ has the following structure

$$\mathbf{G}_\mathrm{c} = \underbrace{\left[\begin{array}{ccc:ccc|c} \mathbf{G}_{\mathrm{f}_1^i\mathrm{f}_1^i} & \dots & & \mathbf{G}_{\mathrm{f}_1^i\mathrm{f}_1^b} & \dots & & \mathbf{G}_{\mathrm{f}_1^i\mathrm{c}} \\ \vdots & \ddots & & \vdots & \ddots & & \vdots \\ \hdashline \mathbf{G}_{\mathrm{f}_1^b\mathrm{f}_1^i} & \dots & & \mathbf{G}_{\mathrm{f}_1^b\mathrm{f}_1^b} & \dots & & \mathbf{G}_{\mathrm{f}_1^b\mathrm{c}} \\ \vdots & \ddots & & \vdots & \ddots & & \vdots \\ \hline \mathbf{G}_{\mathrm{c}\mathrm{f}_1^i} & \dots & & \mathbf{G}_{\mathrm{c}\mathrm{f}_1^b} & \dots & & \mathbf{G}_{\mathrm{cc}} \end{array}\right]}_{d\#\mathrm{F_c}} \begin{array}{l} \left.\vphantom{\begin{array}{c}\mathbf{G}\\ \vdots\\ \mathbf{G}\\ \vdots\end{array}}\right\} d\#\mathrm{F_c} \\ \left.\vphantom{\mathbf{G}}\right\} d \end{array} \tag{3.17}$$

The gradient acts on all the face- and the cell-based velocity DoFs, but it never couples two different Cartesian velocity components. Hence each internal matrix $\mathbf{G}_{n,m} \in \mathbb{R}^{d \times d}$ is diagonal: $\mathbf{G}_{n,m} = G_{n,m}\mathbf{I}_{dd}$, with $\mathbf{I}_{dd}$ being the identity matrix of dimension $d \times d$ and $G_{n,m} \in \mathbb{R}$, $n, m \in \{\mathrm{f}_1, \dots, \mathrm{f}_{\#\mathrm{F_c}}, \mathrm{c}\}$, is the entry that one computes for the scalar-valued version of the Laplace operator (Bonelle, 2014, Section 8.3).

Velocity divergence operator

Consider the divergence operator $\mathrm{D_c}$, defined in (2.43), which couples velocity and pressure DoFs, and is used both to discretize the incompressibility constraint and the pressure gradient:

$$\mathbf{D}_\mathrm{c} = \overbrace{\left[\mathbf{d}_{\mathrm{f}_1^i}^T \ \dots \ \vdots \ \mathbf{d}_{\mathrm{f}_1^b}^T \ \dots\right.}^{d\#\mathrm{F_c}} \ | \ \overbrace{\left.\mathbf{0}_d^T\right]}^{d} , \tag{3.18}$$

where $\mathbf{d}_\mathrm{f} := |\mathrm{f}|\, \underline{n}_\mathrm{fc}$ for all $\mathrm{f} \in \mathrm{F_c}$. One has $\mathbf{D}_\mathrm{c} \in \mathbb{R}^{1 \times (d(\#\mathrm{F_c}+1))}$. Notice that, as outlined in Remark 2.27, the cell-based velocity DoFs are not relevant in the divergence operator.

Source term

Recalling (2.117), the right-hand side vector $\mathbf{F}_{\mathrm{c}}$ has contributions corresponding to the velocity cell-based DoFs:

$$\mathbf{S}_{\mathrm{c}} = \left[\begin{array}{ccc|c||c} \mathbf{0}_d^T & \dots & \mathbf{0}_d^T & \dots & \mathbf{s}_{\mathrm{c}}^T & 0 \end{array}\right]^T , \tag{3.19}$$

where $\mathbf{s}_{\mathrm{c}} := \int_{\mathrm{c}} \underline{f}$.

Final system

One can now consider the system related to the mesh cell $\mathrm{c} \in \mathrm{C}$: it is obtained by simply combining the matrices $\mathbf{G}_{\mathrm{c}}$, $\mathbf{D}_{\mathrm{c}}$, and $\mathbf{S}_{\mathrm{c}}$. It takes the form

$$\left[\begin{array}{c||c} \mathbf{A}_{\mathrm{c}} & \mathbf{B}_{\mathrm{c}}^T \\ \hline\hline \mathbf{B}_{\mathrm{c}} & 0 \end{array}\right] \mathbf{U}_{\mathrm{c}} = \mathbf{F}_{\mathrm{c}} , \tag{3.20}$$

where, in order to make the saddle-point structure clearer, we defined $\mathbf{B}_{\mathrm{c}} := -\mathbf{D}_{\mathrm{c}}$ for the coupling (we consider here the negative divergence in the mass balance), and used a generic matrix $\mathbf{A}_{\mathrm{c}}$ for the velocity-velocity block. For the Stokes problem, we have

$$\mathbf{A}_{\mathrm{c}} := \nu \mathbf{G}_{\mathrm{c}} , \tag{3.21}$$

since the only contribution is the diffusive one, and similarly $\mathbf{F}_{\mathrm{c}} := \mathbf{S}_{\mathrm{c}}$. This general writing will be handy later when dealing with the convection term and the mass one (coming from the time derivative).

Each block of the local system (3.20) is dispatched into the global one and assembled (by simple summation) according to the global ordering of the DoFs. Assuming that the order presented above (face-based velocity, cell-based velocity, and then pressure DoFs) is kept in the global system as well, one may write

$$\left[\begin{array}{c|c||c} \mathbf{A}_{\mathrm{FF}} & \mathbf{A}_{\mathrm{FC}} & \mathbf{B}_{\mathrm{CF}}^T \\ \hline \mathbf{A}_{\mathrm{CF}} & \mathbf{A}_{\mathrm{CC}} & \mathbf{0}_{\mathrm{CC}} \\ \hline\hline \mathbf{B}_{\mathrm{CF}} & \mathbf{0}_{\mathrm{CC}} & \mathbf{0}_{\mathrm{CC}} \end{array}\right] \left[\begin{array}{c} \mathbf{u}_{\mathrm{F}} \\ \hline \mathbf{u}_{\mathrm{C}} \\ \hline\hline \mathbf{p}_{\mathrm{C}} \end{array}\right] = \left[\begin{array}{c} \mathbf{0}_{\mathrm{F}} \\ \hline \mathbf{f}_{\mathrm{C}} \\ \hline\hline \mathbf{0}_{\mathrm{C}} \end{array}\right] . \tag{3.22}$$

Here, we have set

$$\begin{aligned} \mathbf{u}_{\mathrm{F}} &= \left[\mathbf{u}_{\mathrm{f}_1}^T \dots \mathbf{u}_{\mathrm{f}_n}^T \dots \mathbf{u}_{\mathrm{f}_{\#\mathrm{F}}}^T\right]^T \in \mathbb{R}^{d\#\mathrm{F}} , & \mathbf{u}_{\mathrm{C}} &= \left[\mathbf{u}_{\mathrm{c}_1}^T \dots \mathbf{u}_{\mathrm{c}_n}^T \dots \mathbf{u}_{\mathrm{c}_{\#\mathrm{C}}}^T\right]^T \in \mathbb{R}^{d\#\mathrm{C}} , \\ \mathbf{p}_{\mathrm{C}} &= \left[p_{\mathrm{c}_1} \dots p_{\mathrm{c}_n} \dots p_{\mathrm{c}_{\#\mathrm{C}}}\right]^T \in \mathbb{R}^{\#\mathrm{C}} , & \mathbf{f}_{\mathrm{C}} &= \left[\mathbf{f}_{\mathrm{c}_1}^T \dots \mathbf{f}_{\mathrm{c}_n}^T \dots \mathbf{f}_{\mathrm{c}_{\#\mathrm{C}}}^T\right]^T \in \mathbb{R}^{d\#\mathrm{C}} , \end{aligned} \tag{3.23}$$

and the global matrices are assembled from the local ones in such a way that

$$\begin{aligned} \mathbf{A}_{\mathrm{FF}} &\in \mathbb{R}^{(d\#\mathrm{F})\times(d\#\mathrm{F})} , & \mathbf{A}_{\mathrm{CC}} &\in \mathbb{R}^{(d\#\mathrm{C})\times(d\#\mathrm{C})} , \\ \mathbf{A}_{\mathrm{FC}} &\in \mathbb{R}^{(d\#\mathrm{F})\times(d\#\mathrm{C})} , & \mathbf{B}_{\mathrm{CF}} &\in \mathbb{R}^{(\#\mathrm{C})\times(d\#\mathrm{C})} . \end{aligned} \tag{3.24}$$

Since the Stokes problem is symmetric, at least in its velocity-velocity block, the submatrices $\mathbf{A}_{\mathrm{FF}}$ and $\mathbf{A}_{\mathrm{CC}}$ are symmetric, and one also has $\mathbf{A}_{\mathrm{FC}} = \mathbf{A}_{\mathrm{CF}}^T$.

Elimination of the cell-based unknowns

One can notice that the cell-based velocity DoFs of two distinct cells are not coupled directly together and they do not contribute to the velocity-pressure coupling (cf. Remark 2.27). Therefore, they can be eliminated before the assembly stage and then recovered as a post-processing. This operation hinges on the notion of Schur complement and is often called *static condensation* in the literature.

In the static condensation procedure, one expresses $\mathbf{u}_{\mathrm{c}}$ in terms of $\mathbf{u}_{\mathrm{f}}$ in (3.22), obtaining

$$\left[\begin{array}{c|c||c} \mathbf{A}_{\mathrm{FF}} - \mathbf{A}_{\mathrm{FC}}\mathbf{A}_{\mathrm{CC}}^{-1}\mathbf{A}_{\mathrm{CF}} & \mathbf{0}_{\mathrm{FC}} & \mathbf{B}_{\mathrm{CF}}^{T} \\ \hline \mathbf{A}_{\mathrm{CC}}^{-1}\mathbf{A}_{\mathrm{CF}} & \mathbf{I}_{\mathrm{CC}} & \mathbf{0}_{\mathrm{CC}} \\ \hline\hline \mathbf{B}_{\mathrm{CF}} & \mathbf{0}_{\mathrm{CC}} & \mathbf{0}_{\mathrm{CC}} \end{array}\right] \left[\begin{array}{c} \mathbf{u}_{\mathrm{F}} \\ \hline \mathbf{u}_{\mathrm{C}} \\ \hline\hline \mathbf{p}_{\mathrm{C}} \end{array}\right] = \left[\begin{array}{c} -\mathbf{A}_{\mathrm{FC}}\mathbf{A}_{\mathrm{CC}}^{-1}\mathbf{f}_{\mathrm{C}} \\ \hline \mathbf{A}_{\mathrm{CC}}^{-1}\mathbf{f}_{\mathrm{C}} \\ \hline\hline \mathbf{0}_{\mathrm{C}} \end{array}\right]. \tag{3.25}$$

Notice that $\mathbf{A}_{\mathrm{CC}}$ is diagonal. Indeed, its nonzero terms are those associated with the diffusion, and they do not couple the cells nor the Cartesian components of the velocity. Hence, the inverse of $\mathbf{A}_{\mathrm{CC}}$ is very simple to compute in practice. Moreover, in order to effectively reduce the size of the global system and potentially speed up the performance, one does not assemble the second line of (3.25): one just needs to store $\mathbf{A}_{\mathrm{CC}}^{-1}\mathbf{A}_{\mathrm{CF}} \in \mathbb{R}^{d\#\mathrm{C}\times d\#\mathrm{F}}$ and $\mathbf{A}_{\mathrm{CC}}^{-1}\mathbf{f}_{\mathrm{C}} \in \mathbb{R}^{d\#\mathrm{C}}$ and then one computes $\mathbf{u}_{\mathrm{c}}$ in a post-processing stage. The final global system then involves only face-based velocity DoFs and pressure DoFs:

$$\left[\begin{array}{c||c} \mathbf{A}_{\mathrm{FF}} - \mathbf{A}_{\mathrm{FC}}\mathbf{A}_{\mathrm{CC}}^{-1}\mathbf{A}_{\mathrm{CF}} & \mathbf{B}_{\mathrm{CF}}^{T} \\ \hline\hline \mathbf{B}_{\mathrm{CF}} & \mathbf{0}_{\mathrm{CC}} \end{array}\right] \left[\begin{array}{c} \mathbf{u}_{\mathrm{F}} \\ \hline\hline \mathbf{p}_{\mathrm{C}} \end{array}\right] = \left[\begin{array}{c} -\mathbf{A}_{\mathrm{FC}}\mathbf{A}_{\mathrm{CC}}^{-1}\mathbf{f}_{\mathrm{C}} \\ \hline\hline \mathbf{0}_{\mathrm{C}} \end{array}\right]. \tag{3.26}$$

The size will then be $(d\#\mathrm{F} + \#\mathrm{C}) \times (d\#\mathrm{F} + \#\mathrm{C})$.

It is worth mentioning that the static condensation procedure may be considered at the local level, before the assembly stage.

BCs: algebraic treatment

We conclude this section by describing the algebraic treatment of the boundary conditions. We recall the procedure outlined in Ern and Guermond (2004, Section 8.4.3). At this stage, the cell-based velocity DoFs have been eliminated, the global matrix thus contains only face-based velocity DoFs and the pressure DoFs. We rewrite (3.26) as follows:

$$\left[\begin{array}{c|c||c} \widetilde{\mathbf{A}}_{\mathrm{F}^i\mathrm{F}^i} & \widetilde{\mathbf{A}}_{\mathrm{F}^i\mathrm{F}^b} & \mathbf{B}_{\mathrm{CF}^i}^{T} \\ \widetilde{\mathbf{A}}_{\mathrm{F}^b\mathrm{F}^i} & \widetilde{\mathbf{A}}_{\mathrm{F}^b\mathrm{F}^b} & \mathbf{B}_{\mathrm{CF}^b}^{T} \\ \hline\hline \mathbf{B}_{\mathrm{CF}^i} & \mathbf{B}_{\mathrm{CF}^b} & \mathbf{0}_{\mathrm{CC}} \end{array}\right] \left[\begin{array}{c} \mathbf{u}_{\mathrm{F}^i} \\ \mathbf{u}_{\mathrm{F}^b} \\ \hline\hline \mathbf{p}_{\mathrm{C}} \end{array}\right] = \left[\begin{array}{c} \widetilde{\mathbf{f}}_{\mathrm{F}^i} \\ \widetilde{\mathbf{f}}_{\mathrm{F}^b} \\ \hline\hline \mathbf{0}_{\mathrm{C}} \end{array}\right] \tag{3.27}$$

where this time, we separated internal and boundary faces, i.e., we decomposed the matrices $\widetilde{\mathbf{A}}_{\mathrm{FF}} := \mathbf{A}_{\mathrm{FF}} - \mathbf{A}_{\mathrm{FC}}\mathbf{A}_{\mathrm{CC}}^{-1}\mathbf{A}_{\mathrm{CF}}$ and $\mathbf{B}_{\mathrm{CF}}$ and the vector $\widetilde{\mathbf{f}}_{\mathrm{F}} := -\mathbf{A}_{\mathrm{FC}}\mathbf{A}_{\mathrm{CC}}^{-1}\mathbf{f}_{\mathrm{C}}$ into two subblocks.

Suppose that Dirichlet BCs, possibly inhomogeneous, are enforced on the whole boundary as $\underline{u}_{|\partial\Omega} = \underline{u}_{\partial}$, so that we want to enforce at the discrete level

$$\mathbf{u}_{\mathrm{F}^b} = \mathbf{u}_{\mathrm{F}^\partial} := [\underline{\pi}_{\mathrm{f}_1^b}(\underline{u}_\partial) \dots \underline{\pi}_{\mathrm{f}_{\#\mathrm{F}^b}^b}(\underline{u}_\partial)]^T. \tag{3.28}$$

In order to take into account this piece of information while keeping the boundary DoFs inside the system, one modifies the second line of (3.27) in order to have an identity block on the diagonal, zeros elsewhere on that line, and one uses the projection of the Dirichlet data on the right-hand side. On the other lines of the system, the blocks related to boundary

faces are set to zero and the right-hand side is modified by using $\mathbf{u}_{\mathrm{F}^\partial}$. The final system is as follows:

$$\left[\begin{array}{c|c||c} \widetilde{\mathbf{A}}_{\mathrm{F}^i\mathrm{F}^i} & \mathbf{0}_{\mathrm{F}^i\mathrm{F}^b} & \mathbf{B}_{\mathrm{CF}^i}^T \\ \hline \mathbf{0}_{\mathrm{F}^b\mathrm{F}^i} & \mathbf{I}_{\mathrm{F}^b\mathrm{F}^b} & \mathbf{0}_{\mathrm{F}^b\mathrm{C}} \\ \hline\hline \mathbf{B}_{\mathrm{CF}^i} & \mathbf{0}_{\mathrm{CF}^b} & \mathbf{0}_{\mathrm{CC}} \end{array}\right] \left[\begin{array}{c} \mathbf{u}_{\mathrm{F}^i} \\ \hline \mathbf{u}_{\mathrm{F}^b} \\ \hline\hline \mathbf{p}_{\mathrm{C}} \end{array}\right] = \left[\begin{array}{c} \widetilde{\mathbf{f}}_{\mathrm{F}^i} - \widetilde{\mathbf{A}}_{\mathrm{F}^i\mathrm{F}^b}\mathbf{u}_{\mathrm{F}^\partial} \\ \hline \mathbf{u}_{\mathrm{F}^\partial} \\ \hline\hline -\mathbf{B}_{\mathrm{Cf}^b}\mathbf{u}_{\mathrm{F}^\partial} \end{array}\right]. \tag{3.29}$$

The above procedure can be performed locally before the assembly of the global system, and this is indeed preferable since it can result in better performance and memory savings. For the sake of simplicity, we have dealt only with the situation where Dirichlet boundary conditions are enforced over the whole boundary. However, the procedure may be performed with straightforward modifications in the case where only a subset of the boundary is of Dirichlet type.

Linear solvers for the Stokes equations

We have presented in Section 1.3.4 some techniques to deal with saddle-point problems by means of iterative methods. To start with, we usually consider the augmented system (see Hestenes (1969), Glowinski and Le Tallec (1989), and Benzi *et al.* (2005, 2011) for details) of (3.22):

$$\begin{gathered} \begin{bmatrix} \mathbf{A}_\lambda & \mathbf{B}^T \\ \mathbf{B} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{p} \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \mathbf{0} \end{bmatrix}, \\ \mathbf{A}_\lambda := \mathbf{A} + \lambda \mathbf{B}^T\mathbf{B}, \end{gathered} \tag{3.30}$$

where $\lambda \geq 0$ is the user-defined augmentation parameter. The formulation (3.30) is still not efficiently exploitable with iterative solvers, but it will be treated by considering a Golub-Kahan Bidiagonalization (GKB) (see Arioli (2013)) or an Uzawa algorithm (see Arrow *et al.* (1958)). The latter leads us to the following procedure (see (1.16)): Given an initial guess $\mathbf{p}^0$, iterate on $k \geq 1$ until convergence:

$$\begin{aligned} &\mathbf{A}_\lambda \mathbf{u}^k + \mathbf{B}^T\mathbf{p}^{k-1} = \mathbf{f}, \\ &\mathbf{p}^k = \mathbf{p}^{k-1} - \lambda \mathbf{B}\mathbf{u}^k. \end{aligned} \tag{3.31}$$

If most of the times iterative solvers will be our first choice to solve the first line of (3.31), we still sometimes employ direct solvers, in particular an LU factorization, if the size of the system allows it.

3.2 Navier–Stokes equations with face-based CDO

Similarly to what has been done in Section 3.1 for the Stokes equations, the problem obtained by applying a CDO-Fb discretization to the NSE (take $\xi^{\mathrm{NS}} := 1$ in (3.1)) is discussed in this section.

3.2.1 Continuous formulation

Assuming homogeneous Dirichlet BCs, the classical variational formulation of the NSE reads: Find $(\underline{u}, p) \in \underline{H}_0^1(\Omega) \times L_*^2(\Omega)$ such that

$$\begin{cases} \nu a(\underline{u}, \underline{v}) + t(\underline{u}; \underline{u}, \underline{v}) + b(\underline{v}, p) = l(\underline{v}) & \forall \underline{v} \in \underline{H}_0^1(\Omega), \\ b(\underline{u}, q) = 0 & \forall q \in L_*^2(\Omega), \end{cases} \tag{3.32}$$

where the definitions of the functional spaces are in (3.2), those of the bilinear forms $a(\cdot,\cdot)$ and $b(\cdot,\cdot)$ and the linear form $l(\cdot)$ in (3.4), and the one of $t(\cdot;\cdot,\cdot)$ in (2.87) (i.e., $t(\underline{w}; \underline{u}, \underline{v}) := \int_\Omega \underline{v}^T(\underline{\underline{\nabla}}\,\underline{u})\underline{w}$).

3.2.2 CDO formulation

Let us recall that the discrete velocity space, $\widehat{\underline{\mathcal{U}}}_{\mathrm{h},0}$, is defined in (3.8) and the discrete pressure space, $\mathcal{P}_{\mathrm{h},*}$, is defined in (3.9). The CDO-Fb formulation of the NSE then reads: Find $(\widehat{\underline{u}}_{\mathrm{h}},\, p_{\mathrm{h}}) \in \widehat{\underline{\mathcal{U}}}_{\mathrm{h},0} \times \mathcal{P}_{\mathrm{h},*}$ such that:

$$\begin{cases} \nu a_{\mathrm{h}}(\widehat{\underline{u}}_{\mathrm{h}}, \widehat{\underline{v}}_{\mathrm{h}}) + t_{\mathrm{h}}(\widehat{\underline{u}}_{\mathrm{h}}; \widehat{\underline{u}}_{\mathrm{h}}, \widehat{\underline{v}}_{\mathrm{h}}) + b_{\mathrm{h}}(\widehat{\underline{v}}_{\mathrm{h}}, p_{\mathrm{h}}) = l(\underline{v}_{\mathrm{C}}) & \forall \widehat{\underline{v}}_{\mathrm{h}} \in \widehat{\underline{\mathcal{U}}}_{\mathrm{h},0}, \\ b_{\mathrm{h}}(\widehat{\underline{u}}_{\mathrm{h}}, q_{\mathrm{h}}) = 0 & \forall q_{\mathrm{h}} \in \mathcal{P}_{\mathrm{h},*}\,. \end{cases} \tag{3.33}$$

With respect to the discrete Stokes equation (3.10), one remarks the addition of the discrete convection trilinear form $t_{\mathrm{h}}(\cdot;\cdot,\cdot)$ defined in (2.88), that is,

$$\begin{aligned} t_{\mathrm{h}}(\widehat{\underline{w}}_{\mathrm{h}}; \widehat{\underline{u}}_{\mathrm{h}}, \widehat{\underline{v}}_{\mathrm{h}}) = \frac{1}{2} \sum_{\mathrm{c}\in\mathrm{C}} \sum_{\mathrm{f}\in\mathrm{F}_{\mathrm{c}}} |\mathrm{f}|\, (\underline{w}_{\mathrm{f}} \cdot \underline{n}_{\mathrm{fc}})(\underline{u}_{\mathrm{f}} - \underline{u}_{\mathrm{c}}) \cdot (\underline{v}_{\mathrm{f}} + \underline{v}_{\mathrm{c}}) \\ + \sum_{\mathrm{f}\in\mathrm{F}^b} |\mathrm{f}|\, (\underline{w}_{\mathrm{f}} \cdot \underline{n}_{\mathrm{f}})^{-} \underline{u}_{\mathrm{f}} \cdot \underline{v}_{\mathrm{f}} \\ + \Xi^{\mathrm{upw}} \frac{1}{2} \sum_{\mathrm{f}\in\mathrm{F}^i} \sum_{\mathrm{c}\in\mathrm{C}_{\mathrm{f}}} |\mathrm{f}|\, |\underline{w}_{\mathrm{f}} \cdot \underline{n}_{\mathrm{fc}}|\, (\underline{u}_{\mathrm{f}} - \underline{u}_{\mathrm{c}}) \cdot (\underline{v}_{\mathrm{f}} - \underline{v}_{\mathrm{c}})\,. \end{aligned} \tag{3.34}$$

The bilinear forms $a_{\mathrm{h}}(\cdot,\cdot)$ and $b_{\mathrm{h}}(\cdot,\cdot)$, and the linear one $l(\cdot)$ have already been defined in (3.11).

As announced in Section 1.3.4, a Picard algorithm is used to deal with the nonlinearity of the NSE. Applying it to (3.33), one obtains: Given an initial guess $\underline{u}_{\mathrm{h}}^0 \in \widehat{\underline{\mathcal{U}}}_{\mathrm{h},0}$, iterate on $k \geq 1$ until convergence: Find $(\widehat{\underline{u}}_{\mathrm{h}}^k,\, p_{\mathrm{h}}^k) \in \widehat{\underline{\mathcal{U}}}_{\mathrm{h},0} \times \mathcal{P}_{\mathrm{h},*}$ such that:

$$\begin{cases} \nu a_{\mathrm{h}}(\widehat{\underline{u}}_{\mathrm{h}}^k, \widehat{\underline{v}}_{\mathrm{h}}) + t_{\mathrm{h}}(\widehat{\underline{u}}_{\mathrm{h}}^{k-1}; \widehat{\underline{u}}_{\mathrm{h}}^k, \widehat{\underline{v}}_{\mathrm{h}}) + b_{\mathrm{h}}(\widehat{\underline{v}}_{\mathrm{h}}, p_{\mathrm{h}}^k) = l(\underline{v}_{\mathrm{C}}) & \forall \widehat{\underline{v}}_{\mathrm{h}} \in \widehat{\underline{\mathcal{U}}}_{\mathrm{h},0}\,, \\ b_{\mathrm{h}}(\widehat{\underline{u}}_{\mathrm{h}}^k, q_{\mathrm{h}}) = 0 & \forall q_{\mathrm{h}} \in \mathcal{P}_{\mathrm{h},*}\,. \end{cases} \tag{3.35}$$

We have yet to define the discrete residual of the Picard iterations. As stated in Section 1.3.4, we consider an approximation of the L^2-norm of the increment of the velocity normalized by the norm of a reference velocity, in general evaluated using the previous step solution. Although using a cell-based norm (for instance, adapting (2.55) to the case of vector-valued functions) is a possible choice, we prefer to use a (semi)norm based on face-based DoFs, namely:

$$\|\widehat{\underline{v}}_{\mathrm{h}}\|_{\mathrm{F}}^2 \coloneqq \sum_{\mathrm{c}\in\mathrm{C}} \sum_{\mathrm{f}\in\mathrm{F}_{\mathrm{c}}} |\mathfrak{p}_{\mathrm{f,c}}|\, |\underline{v}_{\mathrm{f}}|_2^2\,. \tag{3.36}$$

To motivate the choice of $\|\cdot\|_{\mathrm{F}}$, recall that owing to the static condensation (see Section 3.1.3 and (3.25) in particular), these DoFs of the velocity are eliminated from the final CDO system. For better performance, it is thus preferable to avoid the recovery of the cell-based DoFs during the Picard iterations. Moreover, we are addressing the convection term and our discrete operator uses face-based DoFs only. The (semi)norm defined in (3.36) is thus an attractive choice. The final stopping criterion then reads:

$$\frac{\left\|\underline{u}^k - \underline{u}^{k-1}\right\|_{\mathrm{F}}}{\|\underline{u}^{k-1}\|_{\mathrm{F}}} < \varepsilon^{\mathrm{P}}, \tag{3.37}$$

where ε^{P} is a user-defined tolerance. Finally, a maximum number of iterations, K, is also set, so that the algorithm stops and fails whenever $k > K$.

Remark 3.7 - Stopping criterion. Other types of quantities, relative or not, and other types of norms might be chosen for the stopping criterion. The normalization by using the

norm of the previous-step solution might appear inconvenient, since it changes at each step. However, as the Picard algorithm is expected to be convergent, the quantity $\|\underline{u}^{k-1}\|_{L^2(\Omega)}$ should stabilize after an initial transitory phase. Other normalizations might be chosen, for instance based on the initial iterate, if a non-null one is available. The latter strategy might come in handy with unsteady problems, especially when one can use the solution from the previous time step as initial guess and when the solution is not supposed to vary a lot, due to a quasi-steady regime or a small time step value. Another important aspect of the stopping criterion is the considered norm. In (3.37) we considered only the face-based L^2-norm of the velocity. One may argue that a more appropriate norm for the velocity would be an H^1-like norm. Moreover, one may want to take into account the pressure as well. ◇

Remark 3.8 - Initial guess. When considering a steady problem, one usually does not have any approximation or knowledge of the solution, hence a null initial guess is considered: $\underline{u}^0 \equiv \underline{0}$. This also means that the first Picard iteration actually amounts to solving a Stokes problem. Alternatively, the solution of the same (steady) problem but at a different (lower) Reynolds number can be chosen. When unsteady problems are considered, once the time discretization has been deployed, the previous-step solution provides a suitable initial guess: At each time step t^n, one initializes the Picard algorithm with $\underline{u}^{n,0} := \underline{u}^{n-1}$. ◇

3.2.3 Algebraic viewpoint

The only contribution to the system (3.33) whose algebraic structure has not yet been investigated is the one resulting from the discrete trilinear form. $t(\cdot;\cdot,\cdot)$. As it was the case for the gradient reconstruction, if the convection field is known (condition met in practice if one uses Picard iterations, see (3.35)), let us say $\underline{\widehat{w}}_{\mathrm{h}}$, the convection operator becomes linear and can be represented by local matrices in each mesh cell c. Each local matrix, denoted below by $\mathbf{T}_{\mathrm{c}}(\underline{\widehat{w}}_{\mathrm{c}})$, consists of diagonal submatrices since the linearized convection operator does not couple the Cartesian components of the velocity. This leads to the following structure:

$$\mathbf{T}_{\mathrm{c}}(\underline{\widehat{w}}_{\mathrm{c}}) = \left[\begin{array}{ccc:ccc|c} \mathbf{T}_{\mathrm{f}_1^i\mathrm{f}_1^i} & \mathbf{0}_{dd} & & \mathbf{0}_{dd} & \cdots & & \mathbf{T}_{\mathrm{f}_1^i\mathrm{c}} \\ \vdots & \ddots & & \vdots & \ddots & & \vdots \\ \hdashline \mathbf{0}_{dd} & \cdots & & \mathbf{T}_{\mathrm{f}_1^b\mathrm{f}_1^b} & \mathbf{0}_{dd} & & \mathbf{T}_{\mathrm{f}_1^b\mathrm{c}} \\ \vdots & \ddots & & \vdots & \ddots & & \vdots \\ \hline \mathbf{T}_{\mathrm{c}\mathrm{f}_1^i} & \cdots & & \mathbf{T}_{\mathrm{c}\mathrm{f}_1^b} & \cdots & & \mathbf{T}_{\mathrm{cc}} \end{array}\right], \tag{3.38}$$

where, for the sake of readability, we have dropped the dependency on $\underline{\widehat{w}}_{\mathrm{c}}$ in the submatrices. One readily finds that

$$\begin{gathered} \mathbf{T}_{\mathrm{ff}} = \frac{1}{2}\,|\mathrm{f}|\,(\underline{w}_{\mathrm{f}}\cdot\underline{n}_{\mathrm{fc}})\mathbf{I}_{dd}\,, \qquad \mathbf{T}_{\mathrm{cc}} = -\frac{1}{2}\sum_{\mathrm{c}\in\mathrm{C}}|\mathrm{f}|\,(\underline{w}_{\mathrm{f}}\cdot\underline{n}_{\mathrm{fc}}) = -\frac{1}{2}\mathrm{D}_{\mathrm{c}}(\underline{\widehat{w}}_{\mathrm{c}})\mathbf{I}_{dd}\,, \\ \mathbf{T}_{\mathrm{cf}} = -\mathbf{T}_{\mathrm{fc}} = \frac{1}{2}\,|\mathrm{f}|\,(\underline{w}_{\mathrm{f}}\cdot\underline{n}_{\mathrm{fc}})\mathbf{I}_{dd}\,. \end{gathered} \tag{3.39}$$

Modifications might be needed in order to take into account the additional terms of $t_{\mathrm{h}}(\cdot;\cdot,\cdot)$ coming from (2.90) and (2.91). This leads to

$$\begin{aligned} \mathbf{T}_{\mathrm{f}^i\mathrm{f}^i} &= \frac{1}{2}\left|\mathrm{f}^i\right|\Big((\underline{w}_{\mathrm{f}^i}\cdot\underline{n}_{\mathrm{f}^i\mathrm{c}}) + \Xi^{\mathrm{upw}}\,|\underline{w}_{\mathrm{f}^i}\cdot\underline{n}_{\mathrm{f}^i\mathrm{c}}|\Big)\mathbf{I}_{dd} && \forall \mathrm{f}^i\in\mathrm{F}_{\mathrm{c}}\cap\mathrm{F}^i\,, \\ \mathbf{T}_{\mathrm{f}^b\mathrm{f}^b} &= \left|\mathrm{f}^b\right|\Big(\frac{1}{2}(\underline{w}_{\mathrm{f}^b}\cdot\underline{n}_{\mathrm{f}^b}) + (\underline{w}_{\mathrm{f}^b}\cdot\underline{n}_{\mathrm{f}^b})^-\Big)\mathbf{I}_{dd} = \frac{1}{2}\left|\mathrm{f}^b\right|\,|\underline{w}_{\mathrm{f}^b}\cdot\underline{n}_{\mathrm{f}^b}|\,\mathbf{I}_{dd} && \forall \mathrm{f}^b\in\mathrm{F}_{\mathrm{c}}\cap\mathrm{F}^b\,. \end{aligned} \tag{3.40}$$

Recall now the local system (3.22), i.e.

$$\left[\begin{array}{c\|c} \mathbf{A}_{\mathrm{c}} & \mathbf{B}_{\mathrm{c}}^T \\ \hline \mathbf{B}_{\mathrm{c}} & 0 \end{array}\right]\mathbf{U}_{\mathrm{c}} = \mathbf{F}_{\mathrm{c}}\,, \tag{3.41}$$

Now, in the context of the NSE, one has $\mathbf{A}_{\mathrm{c}} := \nu\mathbf{G}_{\mathrm{c}} + \mathbf{T}_{\mathrm{c}}(\underline{\widehat{u}}_{\mathrm{c}})$ if (3.33) is considered, and $\mathbf{A}_{\mathrm{c}} := \nu\mathbf{G}_{\mathrm{c}} + \mathbf{T}_{\mathrm{c}}(\underline{\widehat{u}}_{\mathrm{c}}^{k-1})$ if (3.35) is considered.

Remark 3.9 - Simplifications. Assuming that $\mathrm{D}_{\mathrm{c}}(\underline{\widehat{w}}_{\mathrm{c}}) = 0$ for all $\mathrm{c} \in \mathrm{C}$, one simply has $\mathbf{T}_{\mathrm{cc}} = \mathbf{0}$. This is a condition often met since, for instance, the advection field $\underline{\widehat{w}}_{\mathrm{h}}$ considered in the Picard procedure is discretely incompressible. Moreover, since face-based quantities are single-valued at internal faces, the first term of the internal face block of (3.40) disappears after assembly. Hence, one ends up with just assembling $\mathbf{T}_{\mathrm{f}^i\mathrm{f}^i} = \frac{1}{2}\left|\mathrm{f}^i\right| \Xi^{\mathrm{upw}} \left|\underline{w}_{\mathrm{f}^i} \cdot \underline{n}_{\mathrm{f}^i\mathrm{c}}\right| \mathbf{I}_{dd}$. Notice that $\mathbf{T}_{\mathrm{f}^i\mathrm{f}^i} = \mathbf{0}_{dd}$ if $\Xi^{\mathrm{upw}} = 0$. Finally, in the context of Lemma 2.45 and the corresponding assumptions on $\underline{\widehat{w}}_{\mathrm{h}}$, the structure of $\mathbf{T}_{\mathrm{c}}$ can be further simplified, hence obtaining a skew-symmetric matrix. ◇

3.3 Preliminary numerical setting

In this section, we introduce the setting for the numerical experiments that will be carried out in this Thesis. In particular, a presentation of the meshes, the error norms, and the practical implementation is given.

3.3.1 Meshes

Both two- and three-dimensional test cases will be considered in the numerical experiments.

2D computations in *Code_Saturne*

Only 3D computations are allowed in *Code_Saturne*. Hence, a workaround is necessary to run 2D test cases. Our approach is to consider a one-layer 3D mesh made of prisms, namely a 2D lattice extruded in the z-direction. The depth is not discretized, so that only one cell is introduced in the z-direction. An example of such a mesh is shown in Fig. 3.1. The BCs need to be adapted, too. The Dirichlet BCs related to the original 2D problem are easily extended to the higher dimensional case by considering a Dirichlet datum which is independent of z. The 3D problem is closed by considering a set of BCs on the artificial boundary planes parallel to the xy-plane. In particular, the normal velocity component (the z-component) is set to 0, whereas the tangential part is left free by means of a homogeneous Neumann condition. Altogether, this boils down to a slip BC:

$$\underline{u} \cdot \underline{n}_{\pm z} = 0, \qquad \underline{t}_{\pm z} - (\underline{t}_{\pm z} \cdot \underline{n}_{\pm z})\underline{n}_{\pm z} = \underline{0} \qquad \text{on } \partial\Omega_{\pm xy} \tag{3.42}$$

where $\partial\Omega_{\pm xy}$ are the artificial planes, $\underline{n}_{\pm z}$ their normal vectors, $\underline{t}_{\pm z} := \underline{\underline{\sigma}}\underline{n}_{\pm z}$ and $\underline{\underline{\sigma}} := \nu\underline{\underline{\nabla}}\,\underline{u} - p\underline{\underline{\mathsf{Id}}}$. These BCs are dealt with by a Nitsche technique (Nitsche, 1971): one can find an example of this technique applied to the slip BC and Stokes problem in Freund and Stenberg (1995).

Polyhedra and cells with hanging nodes

It has been mentioned that the CDO-Fb scheme can handle polyhedral meshes. This feature comes in handy when one has to deal with hanging nodes. In fact, one splits the face on which the hanging node lies into several coplanar ones (this pre-processing operation increases the number of faces of the cell). An example taken from a locally refined Cartesian mesh is given in Fig. 3.2. The square on the left has one face split by a hanging node. The cell is then considered as a pentagon (with two coplanar faces). The advantage of this procedure is its simplicity. Once the mesh has been pre-processed, the CDO-Fb scheme does not need any special treatment to handle hanging nodes.

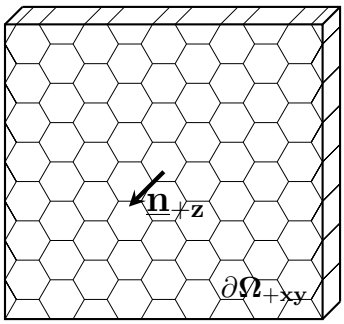


Figure 3.1 – Example of 2D mesh extruded in order to be compatible with *Code_Saturne* computations. No refinement is considered in the (virtual) z-direction.

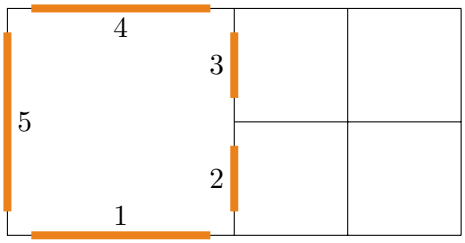


Figure 3.2 – Example of a cell with a hanging node: the hanging node splits the face into two coplanar faces, so that the square cell becomes a pentagon with its five faces marked in orange.

Commonly used meshes

Several mesh sequences have been considered in 2D, see Fig. 3.3, and in 3D, see Fig. 3.4, aiming at representing a large panel of possible configurations. Standard meshes such as regular Cartesian (Figs. 3.3a and 3.4a) or simplicial (Figs. 3.3c and 3.4d) are included. Moreover, in order to investigate the performances of the CDO-Fb method on more general grids, tests are run on purely polyhedral meshes (Figs. 3.3d, 3.4b and 3.4f), on meshes with hanging nodes (usually resulting from local refinements as in Figs. 3.3b and 3.4b), and even on distorted (Kershaw) meshes which satisfy very poor regularity assumptions (Fig. 3.4c). Most of these meshes are part or have been built from those proposed in the benchmark session of the FVCA conference (see for instance Fořt *et al.* (2011)).

3.3.2 Error norms and quadrature rules

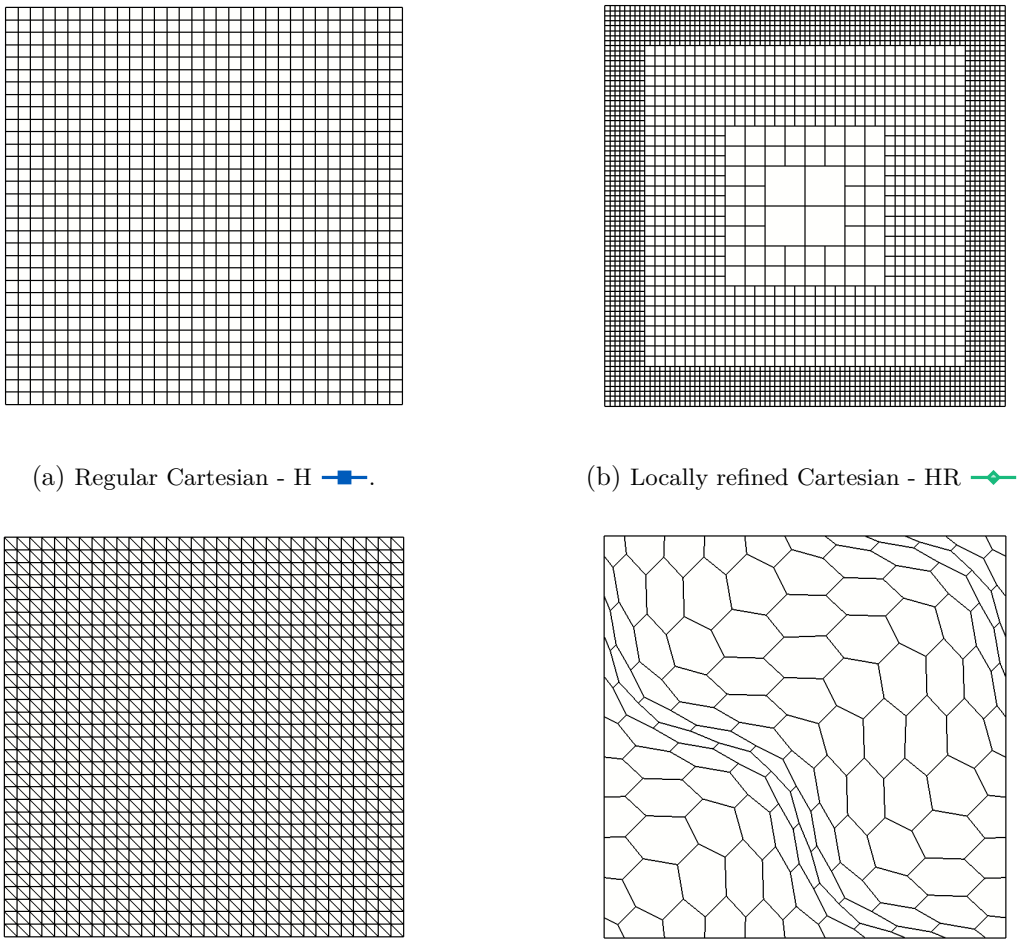
Let us dwell for a moment on the discrete error norms that will be used for the analysis of the test cases. First, given a generic discrete cell-based function $g_{\mathrm{h}} := (g_{\mathrm{c}})_{\mathrm{c}\in\mathrm{C}} \in \mathbb{R}^{l\#\mathrm{C}}$, where, at this stage, the dimension l is left unspecified, we define a discrete cell-based L^2-like norm as follows:

$$\|g_{\mathrm{h}}\|_{\mathrm{C}}^2 := \sum_{\mathrm{c}\in\mathrm{C}} |\mathrm{c}|\, |g_{\mathrm{c}}|_2^2 \ , \tag{3.43}$$

where $|\cdot|_2$ is the Euclidean norm in $\mathbb{R}^l$. This is similar to the definition of the pressure norm $\|\cdot\|_{\mathrm{h}}$, see (2.55). With an abuse of notation, we will use the same symbol for hybrid variables as well:

$$\|\widehat{g}_{\mathrm{h}}\|_{\mathrm{C}}^2 := \sum_{\mathrm{c}\in\mathrm{C}} |\mathrm{c}|\, |g_{\mathrm{c}}|_2^2 \qquad \forall \widehat{g}_{\mathrm{h}} := ((g_{\mathrm{f}})_{\mathrm{f}\in\mathrm{F}},\, (g_{\mathrm{c}})_{\mathrm{c}\in\mathrm{C}}) \ , \tag{3.44}$$

meaning that the face-based DoFs are not considered in this norm, so that we are actually defining a seminorm. Let now $p \in L^2_*(\Omega)$ be the exact pressure. We define $\mathrm{E}_{\mathrm{h}}(p) \in \mathcal{P}_{\mathrm{h}}$ as



(a) Regular Cartesian - H.

(b) Locally refined Cartesian - HR.

(c) Triangles - T.

(d) Polygons - PrG.

Figure 3.3 – Examples of 2D meshes with their marker used in the plots.

the difference between the projection of the exact pressure and the discrete pressure:

$$\mathrm{E}_{\mathrm{h}}(p) \coloneqq (\pi_{\mathrm{c}}(p) - p_{\mathrm{c}})_{\mathrm{c}\in\mathrm{C}} \ . \tag{3.45}$$

Similarly, let $\underline{u} \in \underline{H}^1(\Omega)$ denote the exact velocity field. The velocity error is defined as

$$\widehat{\underline{\mathrm{E}}}_{\mathrm{h}}(\underline{u}) \coloneqq \left((\underline{\pi}_{\mathrm{f}}(\underline{u}) - \underline{u}_{\mathrm{f}})_{\mathrm{f}\in\mathrm{F}}\,,\ (\underline{\pi}_{\mathrm{c}}(\underline{u}) - \underline{u}_{\mathrm{c}})_{\mathrm{c}\in\mathrm{C}}\right) \ . \tag{3.46}$$

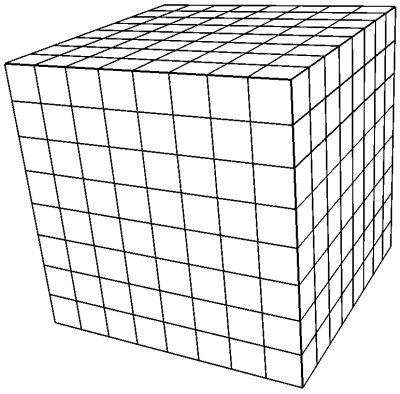
We will report the pressure and velocity L^2-like errors measured as

$$\|\mathrm{E}_{\mathrm{h}}(p)\|_{\mathrm{C}} \quad \text{and} \quad \left\|\widehat{\underline{\mathrm{E}}}_{\mathrm{h}}(\underline{u})\right\|_{\mathrm{C}} \ . \tag{3.47}$$

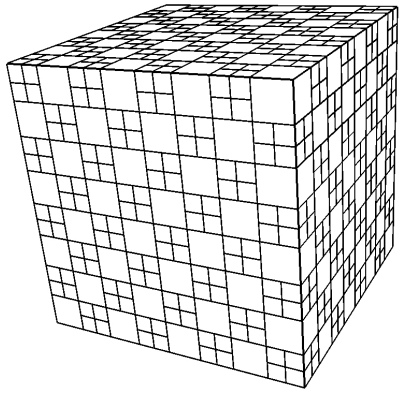
We will also be interested in tracking the error on the discrete velocity gradient $\underline{\underline{\mathrm{G}}}_{\mathrm{h}}$, defined in (2.20a), since it involves both face- and cell-based velocity DoFs. Since $\underline{\underline{\mathrm{G}}}_{\mathrm{h}}$ is piecewise constant on subpyramids of every mesh cell, the H^1-like velocity error is defined as follows:

$$\left\|\underline{\underline{\mathrm{G}}}_{\mathrm{h}}(\widehat{\underline{\mathrm{E}}}_{\mathrm{h}}(\underline{u}))\right\|_{\mathrm{C}}^2 \coloneqq \sum_{\mathrm{c}\in\mathrm{C}}\sum_{\mathrm{f}\in\mathrm{F}} |\mathfrak{p}_{\mathrm{f,c}}| \left|\underline{\underline{\mathrm{G}}}_{\mathrm{c}}(\widehat{\underline{\mathrm{E}}}_{\mathrm{c}}(\underline{u}))_{|\mathfrak{p}_{\mathrm{f,c}}}\right|_2^2 \ , \tag{3.48}$$

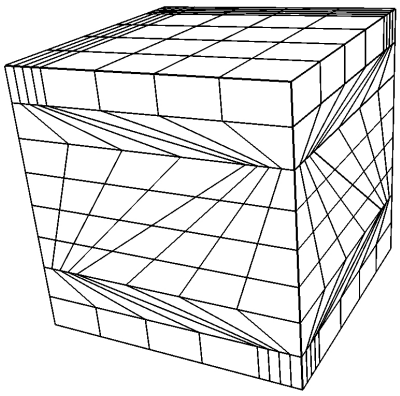
with $\widehat{\underline{\mathrm{E}}}_{\mathrm{c}}(\underline{u})$ defined in (3.46).



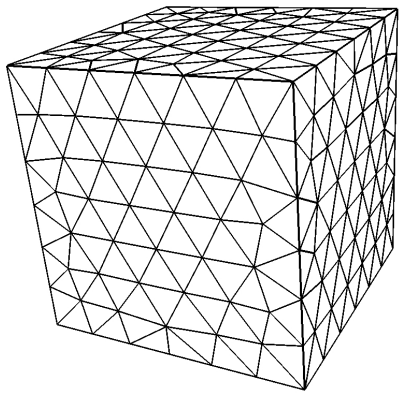
(a) Cartesian - H.



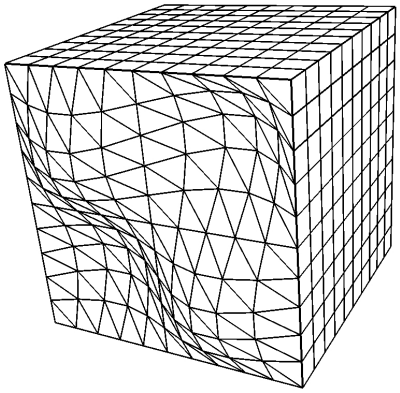
(b) CheckerBoard - CB.



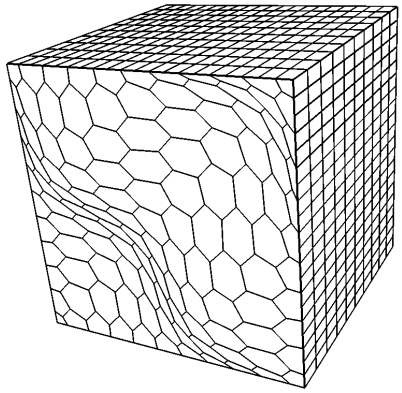
(c) Kershaw - Ker.



(d) Tetrahedra - T.



(e) Prisms - Triangular bases - PrT.



(f) Prisms - Polygonal bases - PrG.

Figure 3.4 – Examples of 3D meshes with their marker used in the plots.

Since second-order convergence rates in space are expected for the L^2-like (semi)norms, it is important to use accurate enough quadrature rules to evaluate the projection of the reference solution. We will proceed with a subdivision of each mesh element into simplices, obtained for $d = 3$ by considering the barycenter of the element, the barycenter of a face $\mathrm{f} \in \mathrm{F_c}$, and the two vertices of one of the edges forming the boundary of f. We can then invoke in each simplex appropriate quadrature rules of fourth- or fifth-order Ern and Guermond (2004, Section 8.1). This should be enough to avoid that the quadrature errors affect the evaluation of the approximation error of the discrete solution. Recall that we are assuming that the mesh cells are star-shaped with respect to their barycenter. This is the case for all the meshes in Figs. 3.3 and 3.4. The subdivision is avoided if the element itself is a simplex.

No transformation into reference elements is applied: so far, in our computations, this choice has never led to inaccurate results and, in fact, with the only exception of the Kershaw series, the meshes that we usually consider (Figs. 3.3 and 3.4) lead to a subdivision with uniformly bounded regularity parameters. In fact, the need for integrals in CDO computation arises only in evaluating body forces or BCs.

3.3.3 Implementation

The developments of the CDO schemes presented in this Thesis have been added to the related module available in *Code_Saturne* (Archambeau *et al.*, 2004).[1] We give in this section some details on the implementation. All the results that will be shown in this and the following chapters have been obtained using the *Code_Saturne* implementation and, unless stated otherwise, performed on a Intel i7 laptop with 32GB RAM.

In all the test cases presented below, the following values was set for the gradient stabilization parameter (see (2.20c)) $\beta := 1$.

Most of the solvers discussed in Section 1.3.4 are available in *Code_Saturne*. It is worth mentioning that the Augmented-Lagrangian–Uzawa and the Golub–Kahan Bidiagonalization algorithms which deal with saddle-point problems are natively available in *Code_Saturne* within a parallelized framework. In our numerical experiments, depending on the nature of the linear system to solve, we will use a preconditioned Conjugate-Gradient. As preconditioners, we will mainly use a Jacobi or an Algebraic Multi-Grid. Moreover, an interface to the external libraries `MUMPS` (Amestoy *et al.*, 2001) and `PETSc` (Balay *et al.*, 1997) allows us to have access to `LU` and `LDLT` direct solvers, as well as alternative algorithms to the above-mentioned linear solvers and preconditioners.

Remark 3.10 - Pressure average. In the present implementation, the constraint on the average of the pressure is not taken into account in the building stage of the problem (see also Remark 3.2) but rather treated as a post-processing after the resolution of the discrete system. ◇

3.4 Numerical results: Stokes equations

In this section and in Section 3.5, several two- and three-dimensional test cases are presented aiming at verifying numerically the properties of the CDO-Fb scheme discussed in Sections 3.1 and 3.2 and the soundness of our implementation.

The current section deals with the steady Stokes problem. Two cases taken from the benchmark considered in Cancès and Omnes (2017) are presented: the Bercovier–Engelman flow in 2D and a Taylor-Green Vortex (TGV) flow adapted to 3D. Both problems have an analytical reference solution which allows us to measure the spatial orders of convergence.

[1] `https://github.com/code-saturne/code_saturne`

3.4.1 2D Bercovier–Engelman

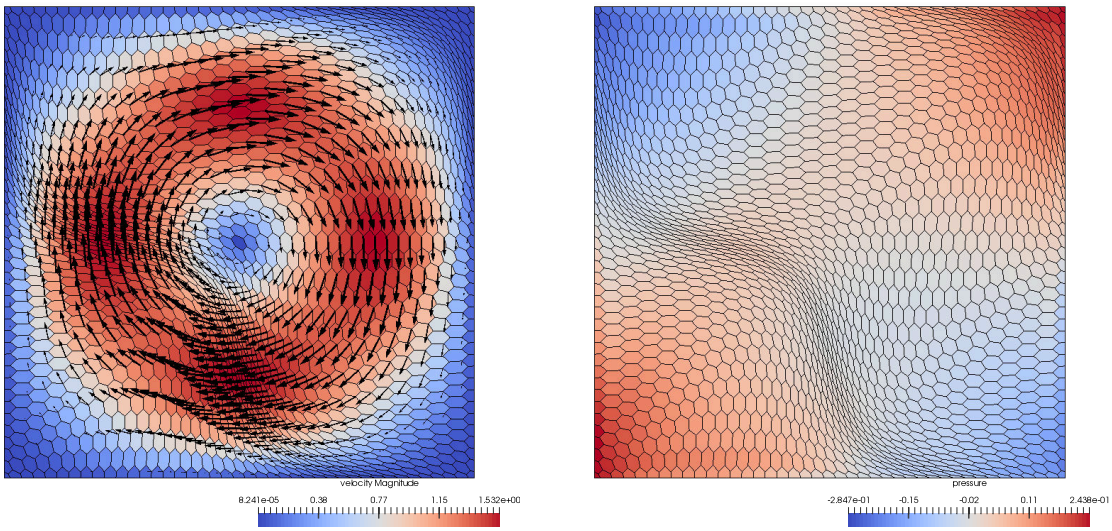
The Bercovier–Engelman test case (Bercovier and Engelman, 1979) is based on a 2D analytic solution to the Stokes problem. It has been proposed in the benchmark Cancès and Omnes (2017, test case 2.1). It consists of a polynomial solution illustrated in Fig. 3.5 on a PrG mesh (see Fig. 3.3d). The details are

$$\begin{cases} \underline{u}_{\mathrm{BE}}(x,y) := [\overline{u}(x,y), -\overline{u}(y,x)]^T\,, \\ p_{\mathrm{BE}}(x,y) := \left(x-\frac{1}{2}\right)\left(y-\frac{1}{2}\right)\,, \\ \quad \overline{u}(x,y) := -256x^2(x-1)^2y(y-1)(2y-1)\,, \end{cases} \tag{3.49}$$

on $\Omega := [0,\,1]^2$, and the viscosity is set to $\nu := 1$. Homogeneous Dirichlet BCs are considered and the source term is computed from (3.49), giving

$$\begin{cases} \underline{f}_{\mathrm{BE}}(x,y) := [\overline{f}(x,y) + \left(y-\frac{1}{2}\right), -\overline{f}(y,x) + \left(x-\frac{1}{2}\right)]^T\,, \\ \quad \overline{f}(x,y) := 256\left(6x^2(x-1)^2(2y-1) + 2y(y-1)(2y-1)(6x^2-12x+2)\right)\,. \end{cases} \tag{3.50}$$

A direct sparse solver from MUMPS has been used for this test case.



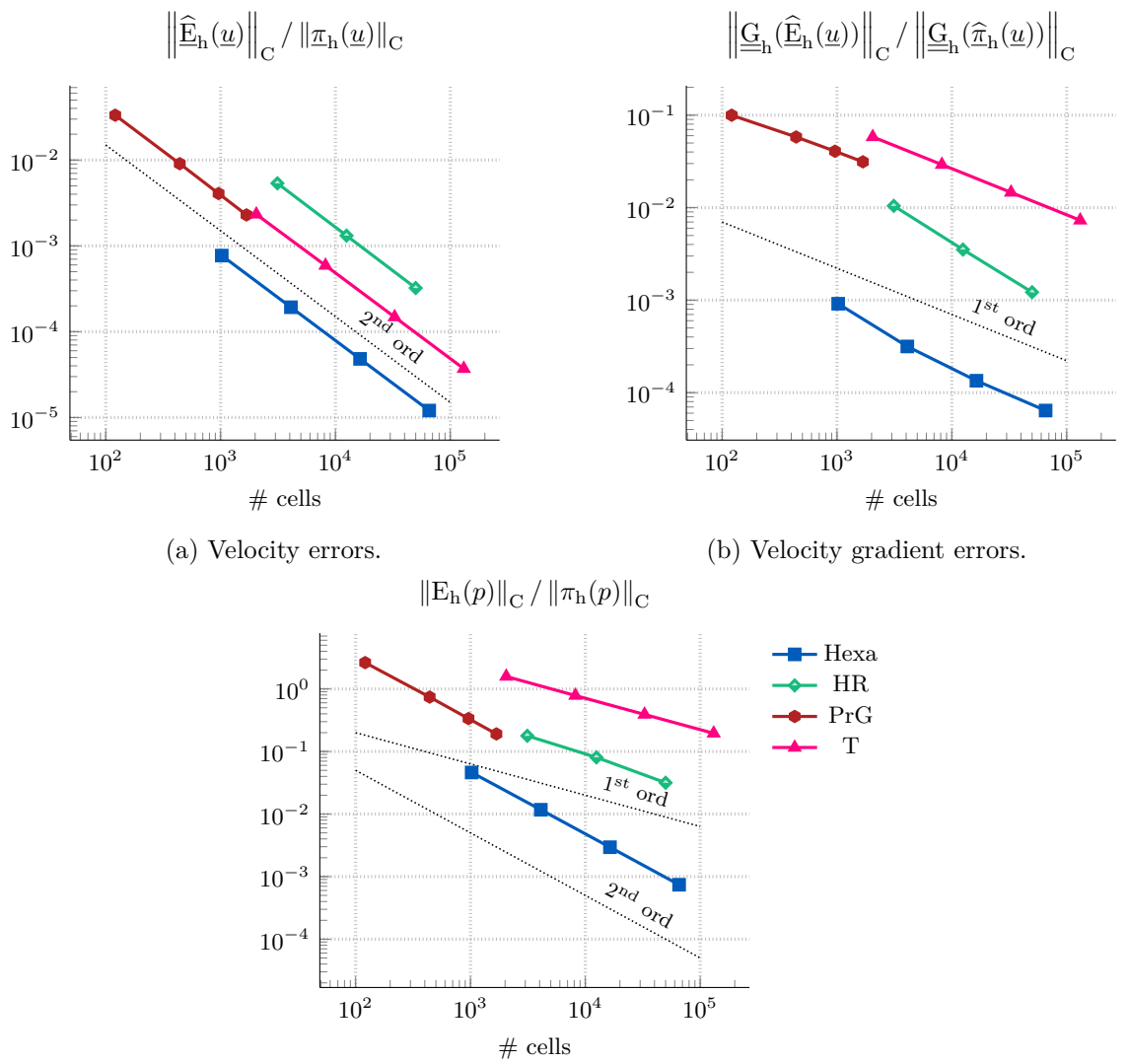
(a) Velocity: directions and magnitude.

(b) Pressure.

Figure 3.5 – Bercovier–Engelman test case (3.49), reference solution. Finest mesh of the PrG sequence (see Fig. 3.3d).

The results are presented in Fig. 3.6. The expected orders of convergence are recovered: second-order for the discrete L^2-norm of the velocity error and first-order for the velocity gradient error and the pressure L^2-error. Optimal orders are obtained also for polygonal meshes, such as the refined Cartesian HR and those with distorted cells, e.g. PrG. Moreover, the pressure errors show sometimes an order of convergence higher than expected: the ones observed on Cartesian (—■—) and polygonal (—●—) meshes are quite close to second order.

Remark 3.11 - Comparisons. By taking advantage of the benchmark proposed in Cancès and Omnes (2017, Case 2.2), the results of CDO can be compared to those of other participants. In particular, we consider three contributions of lowest-order method: a DDFV method with cell-based velocity proposed by Delcourte and Omnes (2017), a DDFV method



(a) Velocity errors.

(b) Velocity gradient errors.

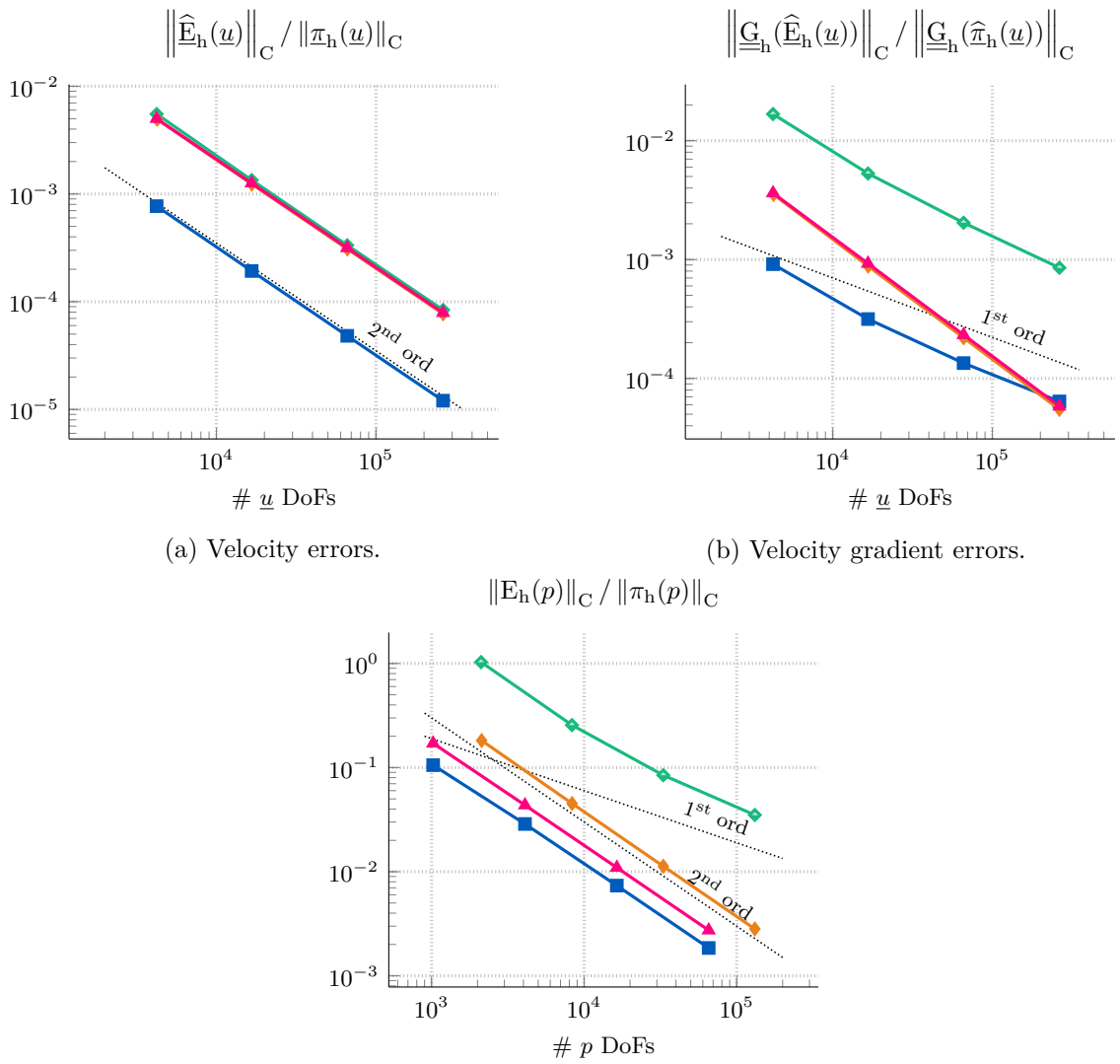
(c) Pressure errors.

Figure 3.6 – 2D Stokes Bercovier–Engelman test case (3.49) - Spatial convergence.

with cell-based pressure by Boyer *et al.* (2017), and a Finite Difference-Volume (FDV) method by Angeli *et al.* (2017). The results for Cartesian meshes are shown in Fig. 3.7. The dimensions of the discrete functional velocity and pressure spaces of each method are considered for a fair comparison. The CDO schemes provide satisfactory results when compared to the other methods, especially for the L^2-norm of the velocity. The details of the results obtained on the finest mesh of the series are given in Table 3.1, which allows us to compare the differences in the dimension of the different discrete settings. ◇

3.4.2 3D modified Taylor–Green Vortex

For this second test case, a 3D solution to the Stokes problem has been chosen. It is drawn from the benchmark considered in Cancès and Omnes (2017, test case 2.2) and it is a steady 3D adaptation to the Stokes problem of the well-known Taylor–Green Vortex (Taylor and Green, 1937). The 3D counterpart is a sinusoidal solution shown in Fig. 3.8. The velocity



(a) Velocity errors.

(b) Velocity gradient errors.

(c) Pressure errors.

Figure 3.7 – 2D Stokes Bercovier–Engelman test case (3.49) - Comparison with FVCA VII participants. Methods: CDO (Bonelle *et al.*, 2020), DDFV-$\underline{u}$ (Boyer *et al.*, 2017), DDFV-p (Delcourte and Omnes, 2017), (Angeli *et al.*, 2017).

Table 3.1 – 2D Stokes Bercovier–Engelman test case (3.49) - Comparison with FVCA VII participants - Details from Fig. 3.7 for finest mesh, Cartesian with 256^2 cells. Methods: CDO (Bonelle *et al.*, 2020), DDFV-$\underline{u}$ (Boyer *et al.*, 2017), DDFV-p (Delcourte and Omnes, 2017), (Angeli *et al.*, 2017).

Method	# $\underline{u}$ DoFs	# p DoFs	$\underline{u}$ L^2 error	$\underline{u}$ H^1 error	p L^2 error
CDO	263168	65536	$1.21e{-5}$	$6.41e{-5}$	$1.85e{-3}$
DDFV-$\underline{u}$	263170	131584	$8.27e{-5}$	$5.89e{-4}$	$4.08e{-2}$
DDFV-p	263200	131601	$7.72e{-5}$	$5.49e{-5}$	$2.82e{-3}$
FDV	263168	65536	$7.89e{-5}$	$5.83e{-5}$	$2.74e{-3}$

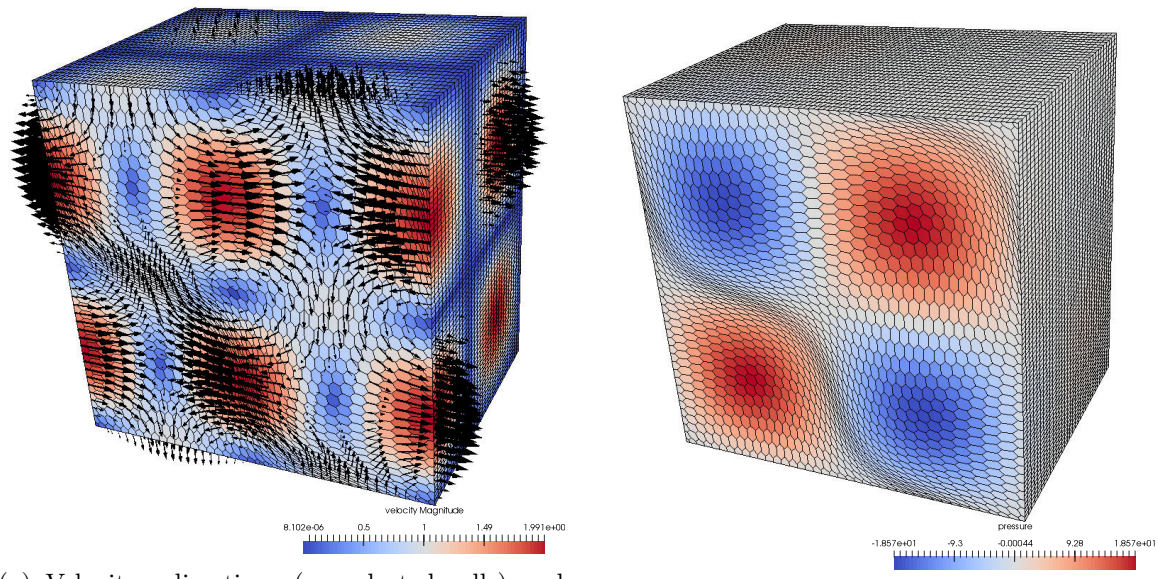
field is divergence-free, but a non-zero source term is needed in the momentum equation:

$$\begin{cases} \underline{u}_{3\mathrm{TGV}}(x,y,z) := \begin{bmatrix} -2\cos(2\pi x)\sin(2\pi y)\sin(2\pi z) \\ \sin(2\pi x)\cos(2\pi y)\sin(2\pi z) \\ \sin(2\pi x)\sin(2\pi y)\cos(2\pi z) \end{bmatrix}, \\ p_{3\mathrm{TGV}}(x,y,z) := -6\pi\sin(2\pi x)\sin(2\pi y)\sin(2\pi z), \end{cases} \tag{3.51}$$

on $\Omega := [0,\,1]^3$, the viscosity is set to $\nu := 1$, and finally the source term is

$$\underline{f}_{3\mathrm{TGV}}(x,y,z) := [-36\pi^2\cos(2\pi x)\sin(2\pi y)\sin(2\pi z),\,0,\,0]^T\,. \tag{3.52}$$

Non-homogeneous Dirichlet BCs corresponding to (3.51) are enforced.



(a) Velocity: directions (on selected cells) and magnitude.

(b) Pressure.

Figure 3.8 – 3D Taylor–Green Vortex test case (3.51). Domain sliced at $x = 0.75$. Finest mesh of the PrG sequence (see Fig. 3.4f).

Remark 3.12 - GKB procedure and linear solver. For this test case, a Golub–Kahan Bidiagonalization (GKB) has been considered to tackle the saddle-point problem. The tolerance of the procedure was set to 10^{-10} on the absolute residual of the momentum and mass equations. A Conjugate Gradient (CG) iterative solver was used at each step of the GKB procedure with the tolerance of the CG set to 10^{-5}. An in-house K-cycle Algebraic Multi-Grid preconditioner Notay (2010) was considered. Notice that the tolerance of the internal solver is higher than the one of the global procedure, but this latter works in incremental mode, so that the aforementioned choice appears to be coherent. Preliminary tests have been run to confirm this intuition and they are shown in Table 3.2. The considered mesh is the finest grid from the Cartesian sequence, which is supposed to lead to the most accurate result among all the considered sequences and refinements. No significant difference is observed on the error under analysis, hence confirming that the tolerances are not too loose. Notice however that somewhat less tight tolerances could be considered as well. ◇

The results for the 3D TGV case are shown in Fig. 3.9. As before, the expected orders of convergence for the velocity (second order), its gradient (first order) and the pressure (first order) are recovered, and the pressure errors are sometimes better than expected, especially for the Cartesian and Prismatic polygonal meshes.

Table 3.2 – L^2- and H^1-error of the velocity and L^2-error of the pressure obtained on a Cartesian mesh composed of 64^3 cells with a GKB procedure with no augmentation parameter, a CG linear solver and an AMG preconditioner. The tolerances of the iterative procedures (ϵ_{GKB} and ϵ_{CG}) vary. In the first columns, ϵ_{GKB} is fixed and ϵ_{CG} varies, then vice versa.

	$\epsilon_{GKB} = 10^{-10}$, ϵ_{CG} varies			ϵ_{GKB} varies, $\epsilon_{\mathrm{CG}} = 10^{-5}$		
	10^{-4}	10^{-5}	10^{-8}	10^{-6}	10^{-10}	10^{-12}
$\frac{\left\lVert \widehat{\underline{\mathrm{E}}}(\underline{u})_{\mathrm{h}} \right\rVert_{\mathrm{C}}}{\left\lVert \underline{\pi}_{\mathrm{h}}(\underline{u}) \right\rVert_{\mathrm{C}}}$	$1.84e{-3}$	$1.80e{-3}$	$1.80e{-3}$	$1.84e{-3}$	$1.80e{-3}$	$1.80e{-3}$
$\frac{\left\lVert \underline{\underline{\mathrm{G}}}_{\mathrm{h}}(\widehat{\underline{\mathrm{E}}}_{\mathrm{h}}(\underline{u})) \right\rVert_{\mathrm{C}}}{\left\lVert \underline{\underline{\mathrm{G}}}_{\mathrm{h}}(\widehat{\underline{\pi}}_{\mathrm{h}}(\underline{u})) \right\rVert_{\mathrm{C}}}$	$2.01e{-3}$	$2.01e{-3}$	$2.01e{-3}$	$2.01e{-3}$	$2.01e{-3}$	$2.01e{-3}$
$\frac{\lVert \mathrm{E}_{\mathrm{h}}(p) \rVert_{\mathrm{C}}}{\lVert \pi_{\mathrm{h}}(p) \rVert_{\mathrm{C}}}$	$2.50e{-3}$	$2.50e{-3}$	$2.50e{-3}$	$2.50e{-3}$	$2.50e{-3}$	$2.50e{-3}$

Table 3.3 – 3D Stokes Taylor–Green Vortex test case (3.51). Comparisons of CDO (Bonelle *et al.*, 2020) and FDV (Angeli *et al.*, 2017). Cartesian meshes.

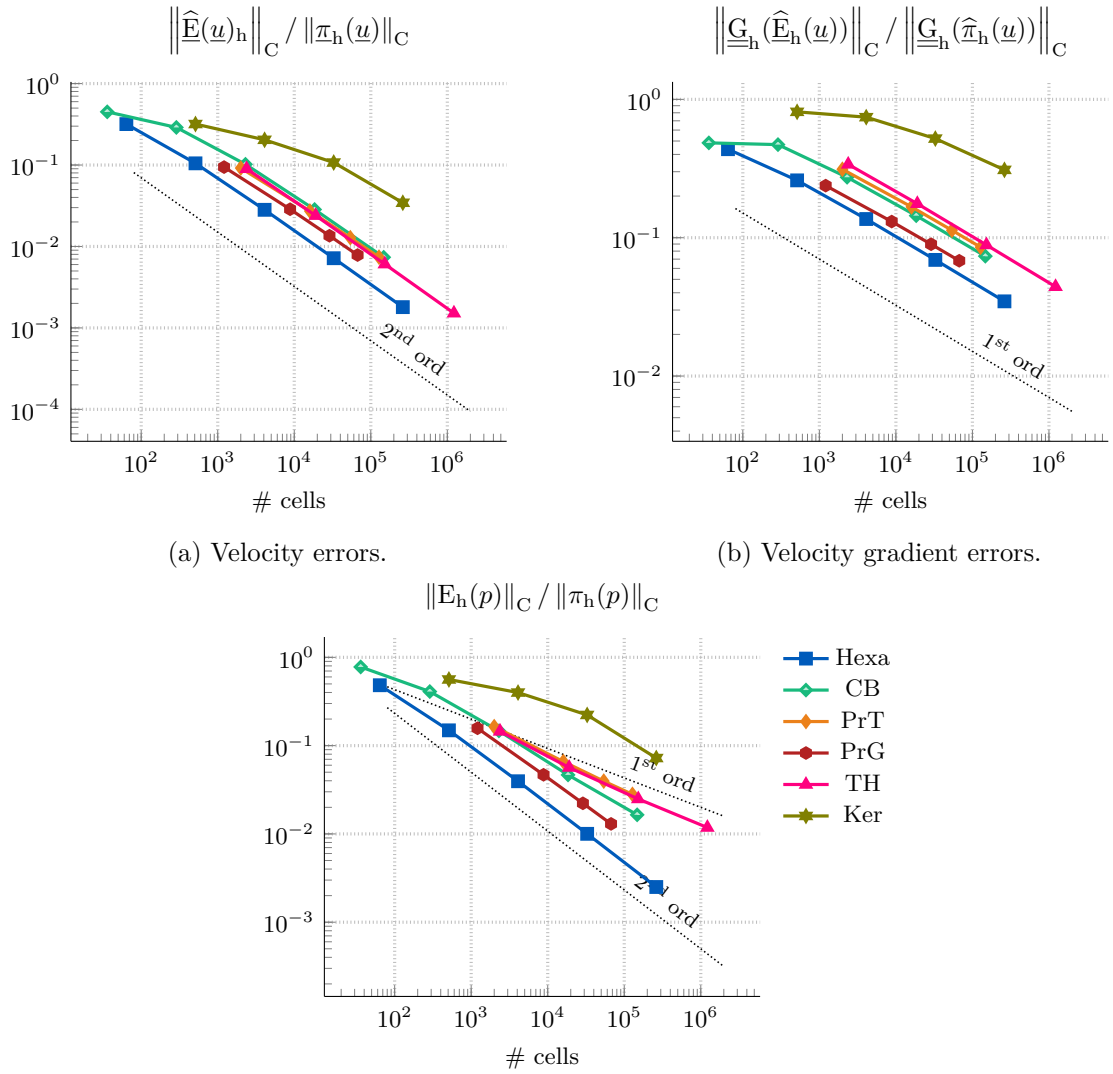
Mesh	# $\underline{u}$ DoFs	# p DoFs	$\underline{u}$ L^2 error	ord	$\underline{u}$ H^1 error	ord	p L^2 error	ord
				CDO				
H4	720	64	$3.18e{-1}$	–	$4.36e{-1}$	–	$4.83e{-1}$	–
H8	5184	512	$1.05e{-1}$	1.69	$2.60e{-1}$	0.79	$1.49e{-1}$	1.70
H16	39168	4096	$2.82e{-2}$	1.95	$1.36e{-1}$	0.96	$3.95e{-2}$	1.92
H32	304128	32768	$7.18e{-3}$	2.00	$6.91e{-2}$	1.00	$1.00e{-2}$	1.98
				FDV (Angeli *et al.*, 2017)				
H4	720	64	$3.43e{-1}$	–	$1.25e{+0}$	–	$2.21e{-1}$	–
H8	5184	512	$9.98e{-2}$	2.03	$5.48e{-1}$	1.25	$5.53e{-2}$	1.99
H16	39168	4096	$2.26e{-2}$	2.05	$1.92e{-1}$	1.56	$1.40e{-2}$	1.98
H32	304128	32768	$5.66e{-3}$	2.03	$7.64e{-2}$	1.35	$3.53e{-3}$	1.99

Remark 3.13 - Irregularity in the convergence rates. A plateau in the convergence is observed for the CB (—◇—) and Ker (—✱—) mesh sequences. Concerning the CB meshes, it happens at the coarsest meshes and it might be explained by the fact that for those meshes the ratio of boundary faces to internal faces is quite high: the related DoFs are consequently exactly imposed by considering the BCs, so that the accuracy in this case is facilitated. Concerning the Kershaw mesh family, it should be said that it has rather poor regularity properties. Nonetheless, the regularity slightly improves in the most refined meshes, which is reflected by a slight improvement of the convergence rates on the finer meshes. ◇

Remark 3.14 - Comparison. We move as in Remark 3.11 and compare the results obtained with CDO-FB with those proposed by Angeli *et al.* (2017) with a FDV scheme for the FVCA VIII benchmark. The detailed data is given in Table 3.3. Differently from the 2D case (see Remark 3.11 and Fig. 3.7), CDO is slightly less accurate than FDV, but the two schemes lead to very similar results. ◇

3.5 Numerical results: Navier–Stokes equations

Numerical experiments are now going to be carried out on the CDO-Fb Navier–Stokes problem. As it has been done with the Stokes one (Section 3.4), both 2D and 3D test cases are considered. First, two test cases with analytical solution (the Burggraf flow and an



(a) Velocity errors.

(b) Velocity gradient errors.

(c) Pressure errors.

Figure 3.9 – 3D Stokes Taylor–Green Vortex test case (3.51) - Spatial convergence.

adapted 3D TGV) will allow us to verify the orders of convergence, and then the classical lid-driven cavity test case will be addressed in a 2D setting.

3.5.1 2D Burggraf flow

The Burggraf flow is a 2D analytical polynomial solution to the NSE (Burggraf, 1966):

$$\begin{cases} \underline{u}_{\mathrm{BRG}}(x,y) := \left[8f(x)g'(y),\ -8f'(x)g(y)\right]^T, \\ p_{\mathrm{BRG}}(x,y) := 8\,\dfrac{1}{Re}\left[F(x)g'''(y) + f'(x)g'(y)\right] + 64F_2(x)\left[g(y)g''(y) - [g'(y)]^2\right]. \end{cases} \tag{3.53}$$

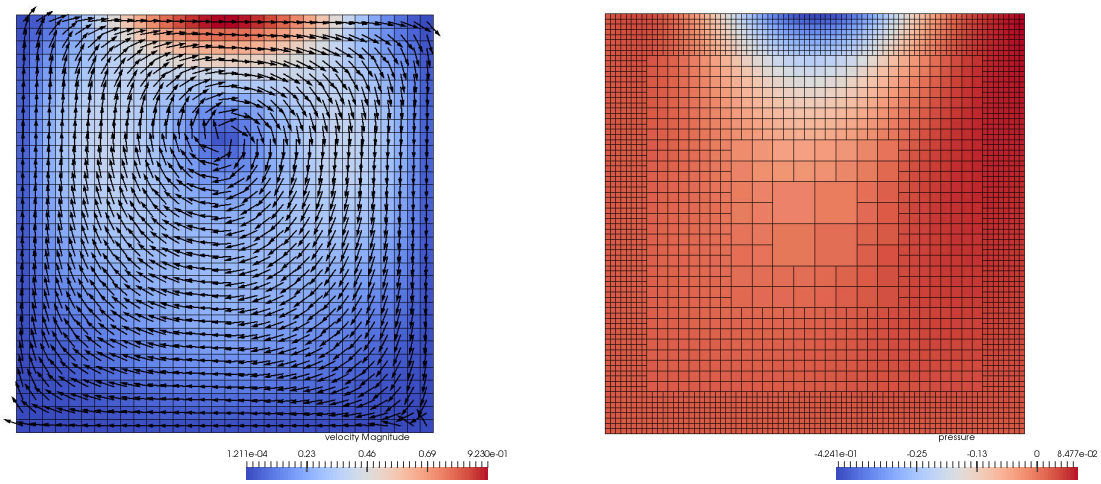
Notice that the pressure has an average different than zero, so that the zero average condition will be enforced before comparing exact pressure to the discrete pressure resulting from the CDO-Fb scheme. The body force is

$$\underline{f}_{\mathrm{BRG}}(x,y) := \left[0,\ \frac{8}{Re}[24F(x) + 2f'(x)g''(y) + f'''(x)g(y)] + 64[F_2(x)G_1(y) - g(y)g'(y)F_1(x)]\right]^T, \tag{3.54}$$

with

$$\begin{cases} f(x) := x^2\,(x-1)^2\,, & g(y) := y^2\,(y-1)\,(y+1)\,, \\ F(x) := \displaystyle\int_0^x f(\widetilde{x})\mathrm{d}\widetilde{x}\,, & G(y) := \displaystyle\int_0^y g(\widetilde{y})\mathrm{d}\widetilde{y}\,, \\ F_1(x) := f(x)f''(x) - [f'(x)]^2\,, & G_1(y) := g(y)g'''(y) - g'(y)g''(y)\,, \\ F_2(x) := \displaystyle\int_0^x f(\widetilde{x})f'(\widetilde{x})\mathrm{d}\widetilde{x} = [f(x)]^2/2\,. \end{cases} \tag{3.55}$$

The domain is $\Omega := [0,\,1]^2$ and $Re := 100$. Considering a reference length $L := 1$ and velocity $U := 1$, one finds $\nu = \frac{1}{Re} = 0.01$.



(a) Velocity: directions and magnitude, coarsest Certesian mesh.

(b) Pressure, coarsest locally refined Cartesian mesh.

Figure 3.10 – Burggraf test case (3.53).

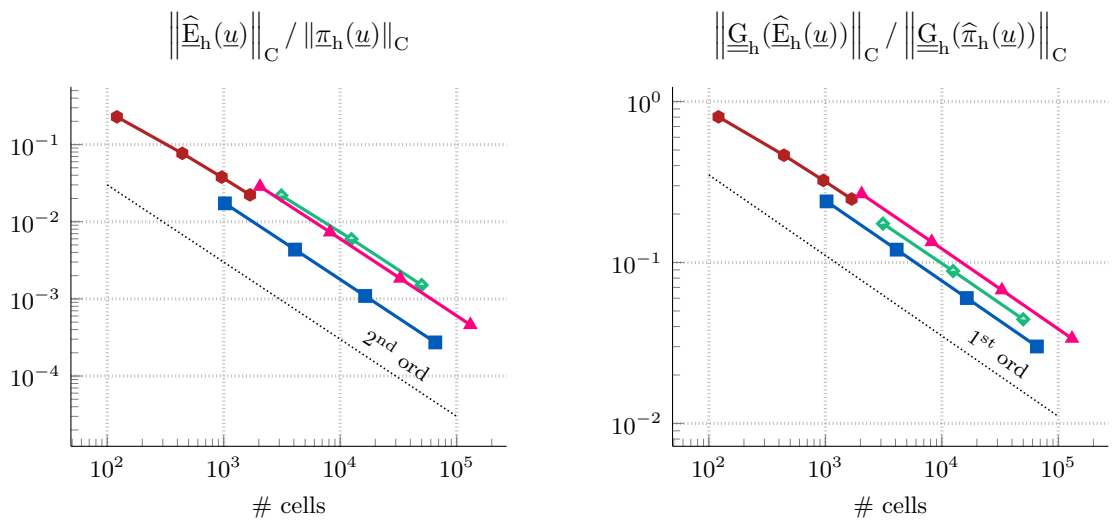
No upwinding stabilization has been considered, i.e. $\Xi^{\mathrm{upw}} := 0$, and a direct LU sparse solver from MUMPS has been used to deal with the linear systems. The mesh sequences in Fig. 3.3 have been used. The Picard tolerance was set to $\varepsilon^{\mathrm{P}} := 10^{-6}$ and about 15 iterations were needed.

The results, displayed in Fig. 3.11, reflect what has already been observed in the Stokes case. The expected orders of convergence have been recovered and higher rates are observed for the convergence of the error pressure on certain mesh sequences.

3.5.2 3D Modified Taylor–Green Vortex

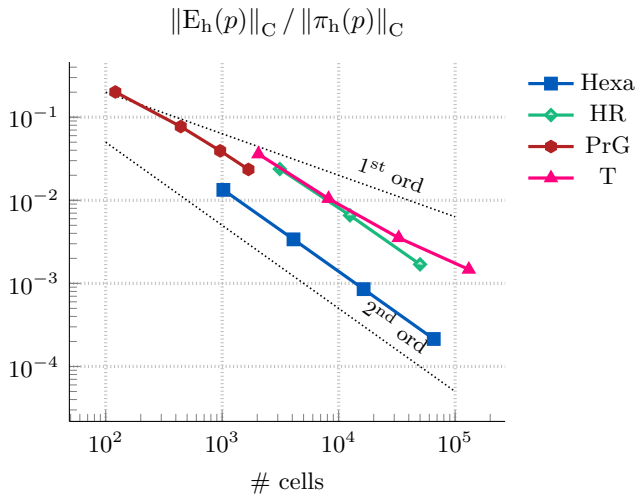
We consider again the 3D TGV test case addressed in Section 3.4. We tested $\nu = 1$ and $\nu = 0.1$, leading to $Re = 1$ and $Re = 10$. The source term, however, needs to be adapted in order to take into account the convection term. This leads to

$$\begin{cases} \underline{f}_{\mathrm{3TGV,NS}}(x,y,z) := -\,12\pi^2 \begin{bmatrix} (1+2\nu)\cos(2\pi x)\sin(2\pi y)\sin(2\pi z) \\ (1-\nu)\sin(2\pi x)\cos(2\pi y)\sin(2\pi z) \\ (1-\nu)\sin(2\pi x)\sin(2\pi y)\cos(2\pi z) \end{bmatrix} \\ \qquad\qquad -\,\dfrac{\pi}{2}\begin{bmatrix} -2\sin(2\pi x)\,(\cos(2\pi y)+\cos(2\pi z)-2) \\ \sin(2\pi y)\,(\cos(2\pi x)-2\cos(2\pi z)+1) \\ \sin(2\pi z)\,(\cos(2\pi x)-2\cos(2\pi y)+1) \end{bmatrix}. \end{cases} \tag{3.56}$$



(a) Velocity errors.

(b) Velocity gradient errors.



(c) Pressure errors.

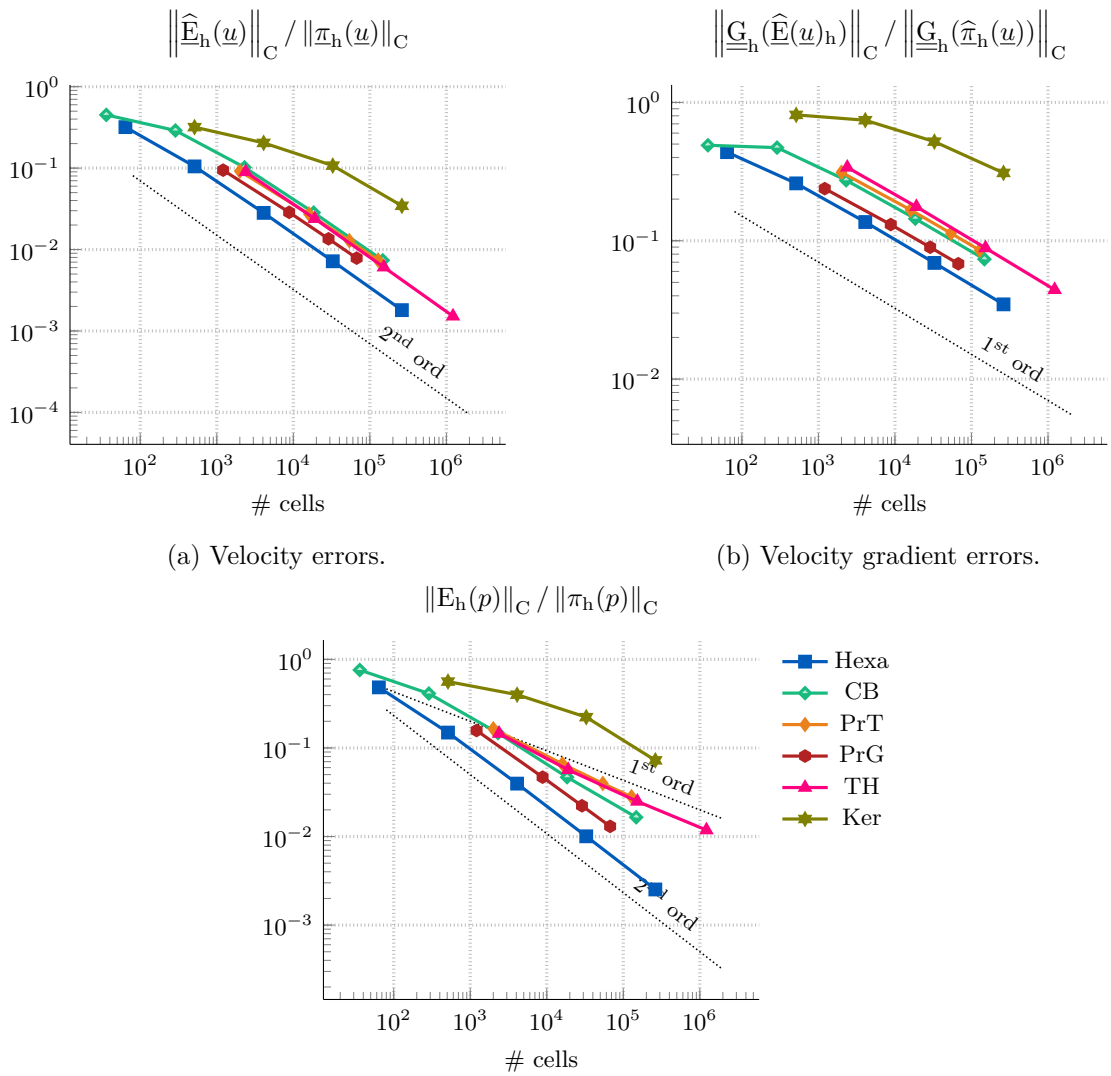
Figure 3.11 – 2D Navier–Stokes Burggraf flow (3.53), $Re = 100$ - Spatial convergence.

The mesh sequences illustrated in Fig. 3.4 have been considered. An Augmented Lagrangian–Uzawa method has been employed to solve the saddle-point problems with an augmentation parameter $\lambda = 100$. The stopping criterion for the ALU method is based on the maximum between the relative increment of the solution at the current (solver) iteration and the global norm of the divergence of the solution. The internal linear systems obtained at each iteration of the ALU method have been solved using a LU direct solver from MUMPS. Concerning the nonlinear solver, typically, 5 iterations of the Picard algorithm were needed to reach convergence to the requested tolerance, $\varepsilon^{\mathrm{P}} := 10^{-6}$. No upwind stabilization has been considered, i.e. $\Xi^{\mathrm{upw}} := 0$.

The results are shown in Fig. 3.12 for $Re = 1$, and in Fig. 3.13 for $Re = 10$. As it was the case for the previous experiments, good results have been observed for all the three errors under consideration (the velocity, its gradient and the pressure), and higher than expected rates have been observed for the pressure with the regular Cartesian mesh sequence and the prismatic mesh sequence with polygonal basis. Some irregularities are observed, but the comments made in Remark 3.13 apply here too.

When comparing the results for the two Reynolds numbers, one can notice that the

pressure errors do not vary too much. On the contrary, both the L^2- and H^1-like discrete velocity errors increase when moving from $Re = 1$ to $Re = 10$: this is a consequence of the lack of pressure-robustness of the scheme, i.e. the velocity errors are not viscosity-independent.



(a) Velocity errors.

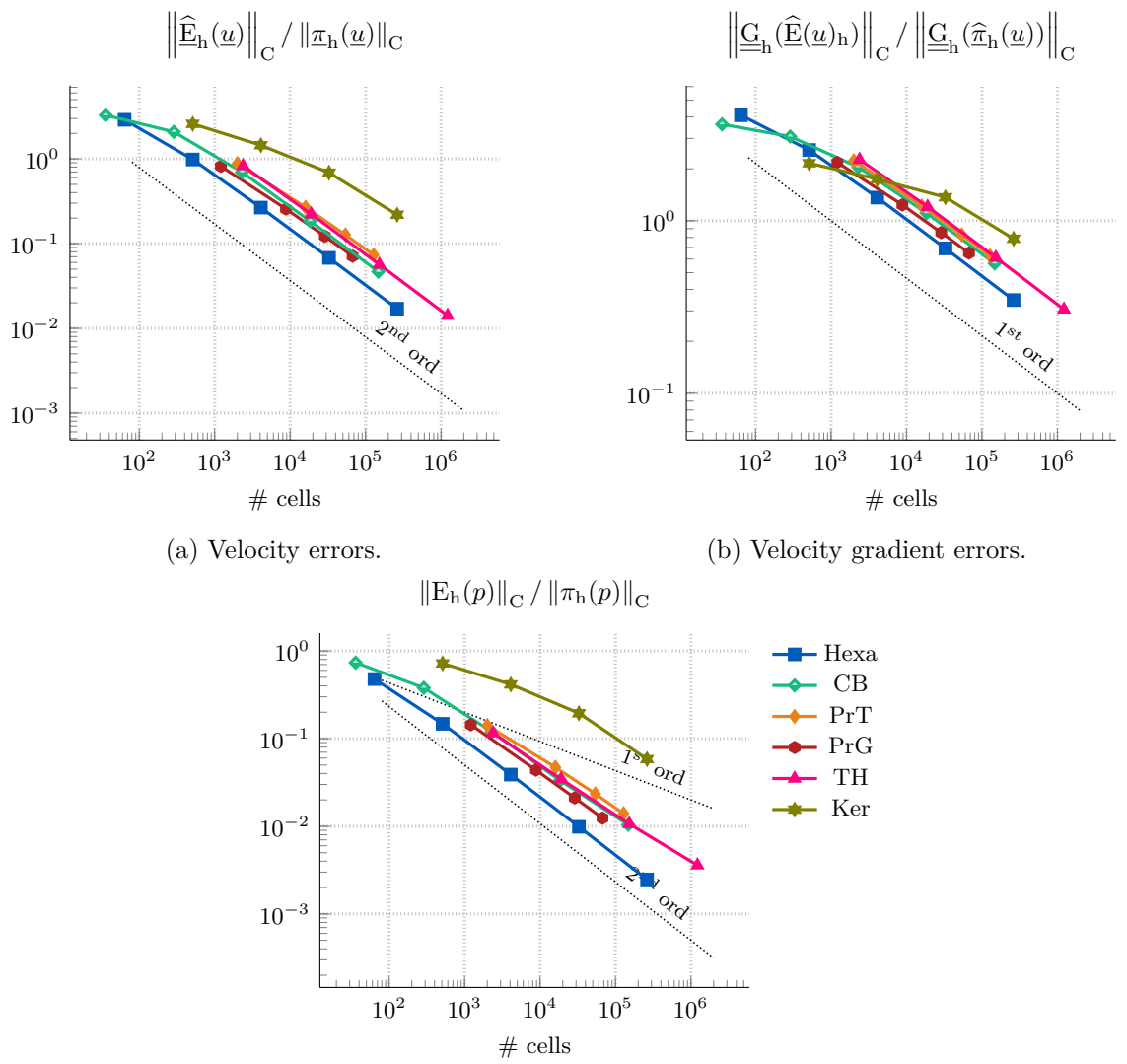
(b) Velocity gradient errors.

(c) Pressure errors.

Figure 3.12 – 3D Navier–Stokes Taylor–Green Vortex test case (3.51), $Re = 1$ - Spatial convergence.

3.5.3 2D lid-driven cavity

The lid-driven cavity problem is a well-known validation test case. The fluid is contained in a domain whose walls except the upper one are fixed. The upper wall has uniform velocity stirring the fluid, leading to different configurations of vortices depending on the velocity of the wall and the viscosity of the fluid. The flow is stationary up to a certain value of the Reynolds number. It is accepted in the litterature that a 2D stationary solution is observed up to $Re = 8000$ (Cazemier *et al.*, 1998; Poliashenko and Aidun, 1995; Bruneau and Saad, 2006). No analytical solution is available but the literature is rich with references, among others Ghia *et al.* (1982), Botella and Peyret (1998), and Erturk *et al.* (2005). Even though



(a) Velocity errors.

(b) Velocity gradient errors.

(c) Pressure errors.

Figure 3.13 – 3D Navier–Stokes Taylor–Green Vortex test case (3.51), $Re = 10$ - Spatial convergence.

3D experiments of this sort are available (see among others Albensoeder and Kuhlmann (2005) and Botti *et al.* (2019)), we focus on the more common two-dimensional setting to temper the computational burden.

The domain is $\Omega := [0, 1]^2$. All the boundary conditions are of Dirichlet type. The velocity of the moving wall is set to $\underline{u}_{\mathrm{mw}} := U\underline{e}_x$ with $U := 1$, the other walls are fixed. The configuration is depicted in Fig. 3.14. The viscosity may vary in order to let us test different Reynolds numbers given by $Re = \frac{UL}{\nu} = \frac{1}{\nu}$, since, given the dimension of the computational domain, one takes $L := 1$.

We are going to study how the CDO-Fb solutions compare with the results of the literature by looking at the velocity, the vorticity, and the pressure. The computations are performed mainly on pseudo-2D Cartesian meshes (see Section 3.3.1 about how one mimics 2D meshes in *Code_Saturne*), some tests with other meshes will be run to verify the method on less regular meshes. Given the reduced size of the problem (for instance, the most refined Cartesian mesh that we considered has 511×511 cells), a direct LU solver has been used to deal with the linear systems. No upwind stabilization has been considered,

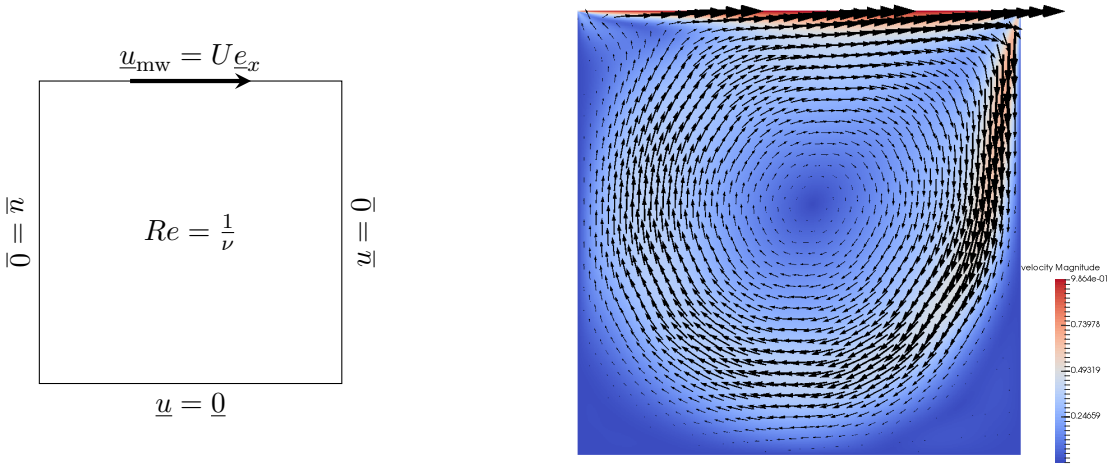


Figure 3.14 – Lid-driven cavity. Left: problem setup. Right: velocity (magnitude and directions on selected cells) obtained for $Re = 1000$ on a 511×511 Cartesian mesh with the CDO-Fb discretization.

i.e. $\Xi^{\mathrm{upw}} := 0$. The tolerance for the Picard algorithm was set to $\varepsilon^{\mathrm{P}} := 10^{-7}$.

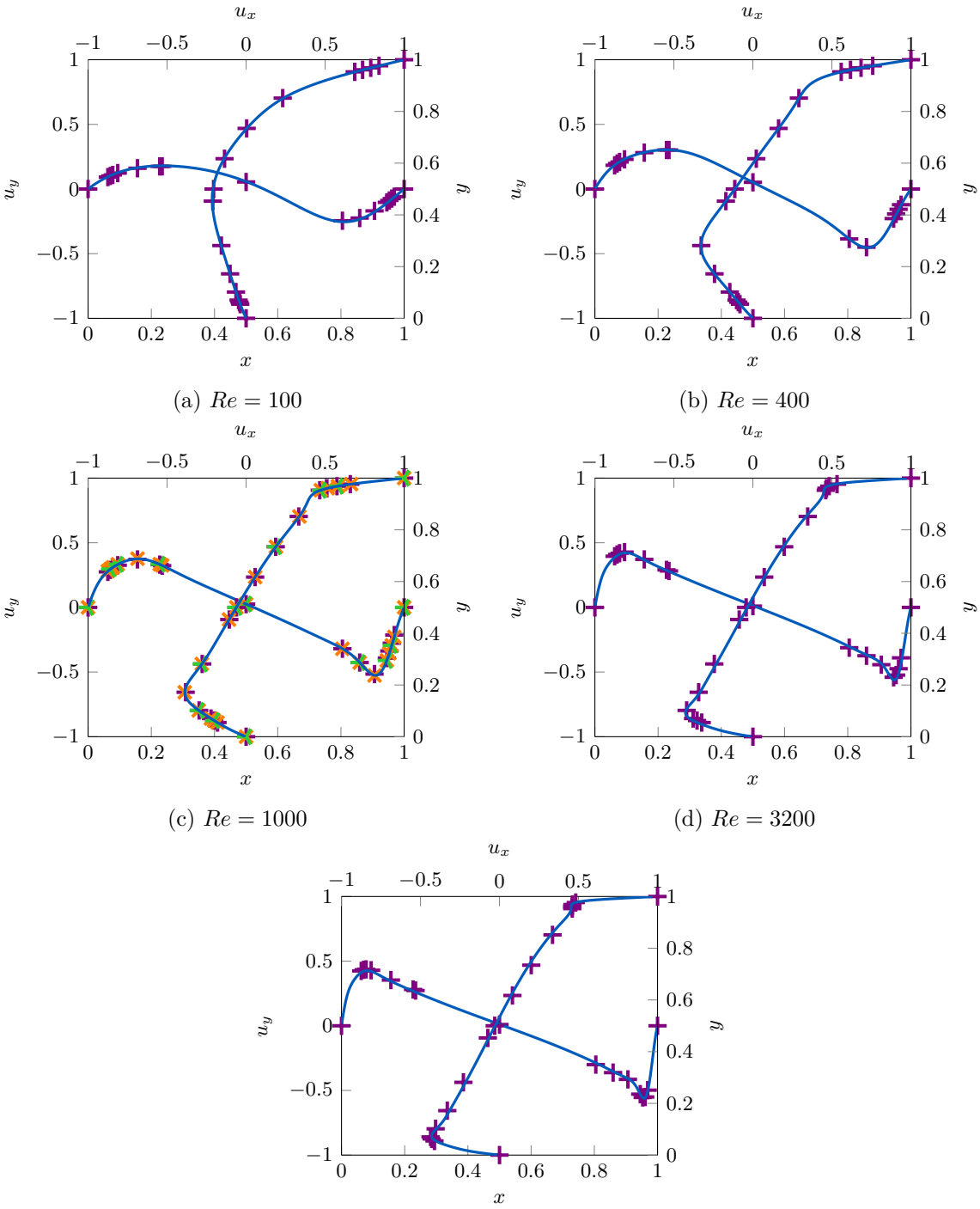
The computations have been carried out for several Reynolds numbers, namely $Re \in \{100, 400, 1000, 3200, 5000\}$. In Fig. 3.15 we report the vertical and horizontal velocity profiles at the symmetry axes of the domain ($x = \frac{1}{2}$ and $y = \frac{1}{2}$) obtained on a 511×511 Cartesian grid: the cell-based values of the velocity are shown. Results taken from Ghia *et al.* (1982) (+) and, when available, Botella and Peyret (1998) (×) and Bruneau and Saad (2006) (-<) are printed in order to have a comparison. At this discretization level, the results obtained with CDO-Fb are basically superimposed to the results from the considered references.

The mesh used in Fig. 3.15, Cartesian with 511×511 cells, is indeed quite refined, hence the good results observed might not be so surprising. We next fix the Reynolds number to $Re = 1000$, and run some computations on different levels of discretization. The results are shown in Fig. 3.16. Excluding the coarsest one (-·-·- 63×63), hence starting from the 127×127 Cartesian mesh, the CDO results turn out to be fairly accurate with respect to the literature, and no significant gain is observed between the two most refined meshes, namely 255×255 - - - and 511×511 ——.

Remark 3.15 - Singularities. No specific treatment has been considered to treat the singularities of the solution to the lid-driven cavity test case that appear at the corners of the moving wall of the domain. ◇

Remark 3.16 - Nonlinearity treatment. As expected, the number of iterations of the Picard algorithm needed to achieve the prescribed tolerance, $\varepsilon^{\mathrm{P}} := 10^{-7}$, depends on the Reynolds number. For moderate values, $Re \leq 1000$, less than 30 iterations are needed, and this number does not seem to depend much on the discretization level. For instance, all the computations shown in Fig. 3.16 need the same number of iterations. This number grows rapidly with the Reynolds number, as shown in Fig. 3.17. For instance, up to 171 iterations are needed for $Re = 5000$. Moreover, recalling that the computations were started with a null initial guess (hence, by solving a Stokes problem), some gains might have been achieved if, for instance, the solution obtained for a lower Reynolds number was fed to the algorithm as initial guess. ◇

Besides the velocity profiles, other quantities may be compared to the references from the literature. In Fig. 3.18 we report the vorticity and the pressure profiles for different



(a) $Re = 100$

(b) $Re = 400$

(c) $Re = 1000$

(d) $Re = 3200$

(e) $Re = 5000$

Figure 3.15 – Lid-driven cavity: horizontal and vertical velocity profiles (cell-based DoFs) on the symmetry axes for different Reynolds numbers. —: CDO; +: Ghia *et al.* (1982), ×: Botella and Peyret (1998), ⊀: Bruneau and Saad (2006). Cartesian mesh 511×511.

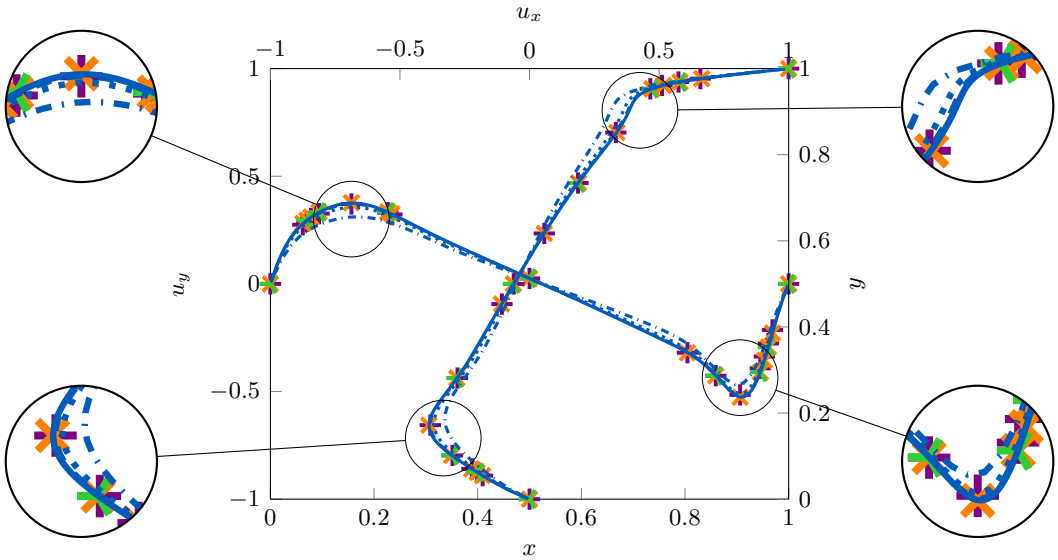


Figure 3.16 – Lid-driven cavity: horizontal and vertical velocity profiles (cell-based DoFs) on the symmetry axes. $Re = 1000$. References: +: Ghia *et al.* (1982), ×: Botella and Peyret (1998), ⊀: Botella and Peyret (1998). CDO-Fb with Cartesian meshes: 511×511 ——, 255×255 - - -, 127×127 ·····, 63×63 -·-·-.

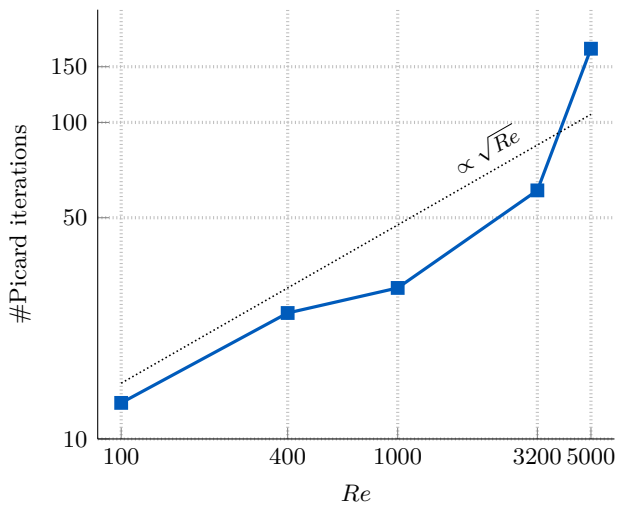


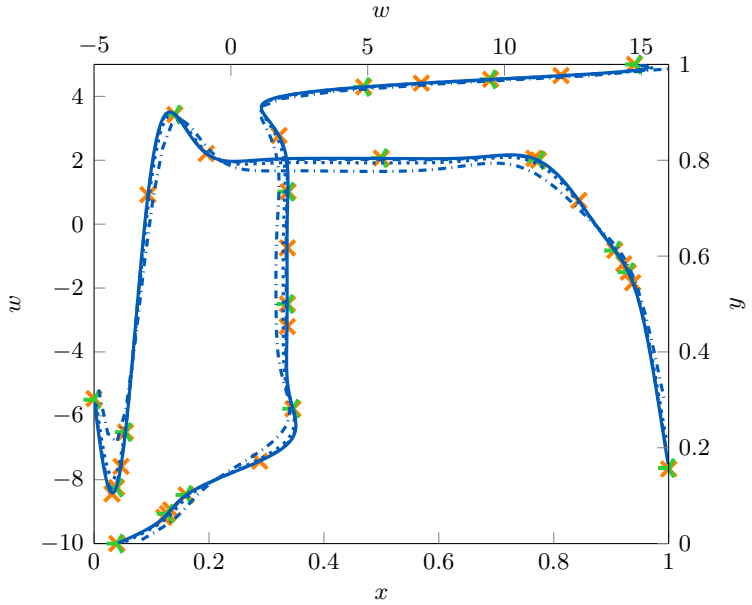
Figure 3.17 – Lid-driven cavity: iterations needed by the Picard algorithm to reach the prescribed tolerance, $\varepsilon^{\mathrm{P}} := 10^{-7}$, at several Reynolds numbers on a 511×511 Cartesian mesh.

mesh refinements on Cartesian meshes, and compare them to the data provided in Botella and Peyret (1998) for $Re = 1000$. To be consistent with Botella and Peyret (1998), we set $U = -1$ and the zero-average constraint on the pressure is replaced by setting $p\left(\frac{1}{2}, \frac{1}{2}\right) = 0$. Moreover, the discrete vorticity ω_{c} is defined to be cell-wise constant and is computed using the extra-diagonal entries of the consistent discrete gradient $\underline{\underline{\mathrm{G}}}^0_{\mathrm{c}}$ (see (2.20b)) as follows:

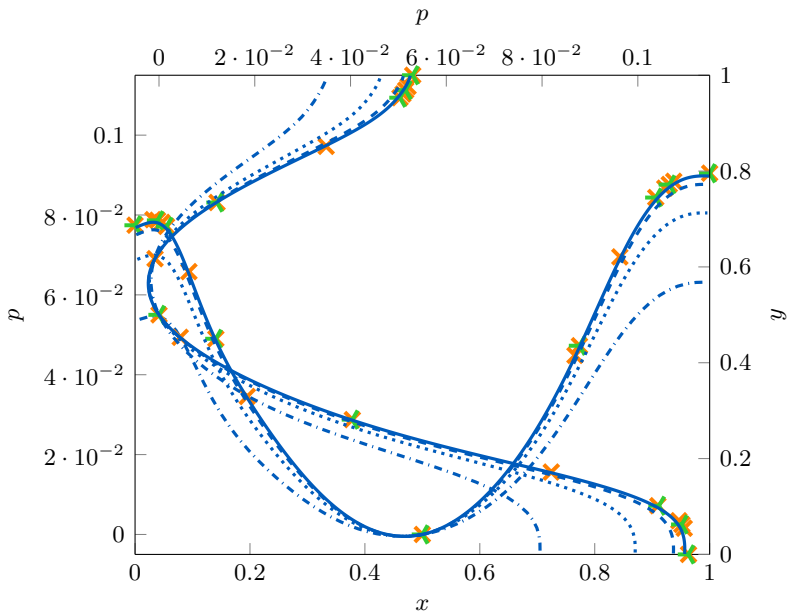
$$\omega_{\mathrm{c}} := [\underline{\underline{\mathrm{G}}}^0_{\mathrm{c}}(\widehat{\underline{u}}_{\mathrm{c}})]_{yx} - [\underline{\underline{\mathrm{G}}}^0_{\mathrm{c}}(\widehat{\underline{u}}_{\mathrm{c}})]_{xy} \qquad \forall \mathrm{c} \in \mathrm{C}\,. \tag{3.57}$$

Since we are dealing with a two-dimensional test case, the vorticity is a scalar. The vorticity results in Fig. 3.18 reflect those for the velocity: the discrete vortices are close to the reference data, even on fairly coarse meshes. The effect of the mesh refinement is more visible in the pressure results, where the values at the boundary are sensibly underestimated on coarse meshes.

Finally, generic meshes are considered to test the CDO scheme on non-Cartesian grids. The meshes at hand are the triangular one, such as the one tagged as T (cf. Fig. 3.3c), and one taken from the Locally Refined Cartesian series (cf. Fig. 3.3b). Both meshes have a number of cells similar to the 255×255 Cartesian mesh, which will be used for reference. The results for $Re = 1000$ are reported in Fig. 3.19. Both non-Cartesian meshes lead to a slightly lower accuracy with respect to the Cartesian mesh, especially around the extrema of the profiles. All in all, the overall results are quite satisfactory and the accuracy remains quite good on polyhedral meshes.

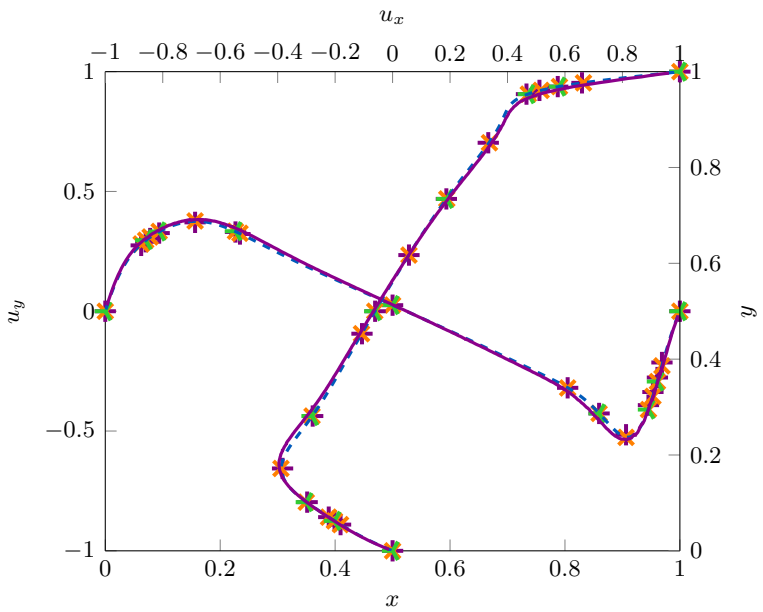


(a) Vorticity.

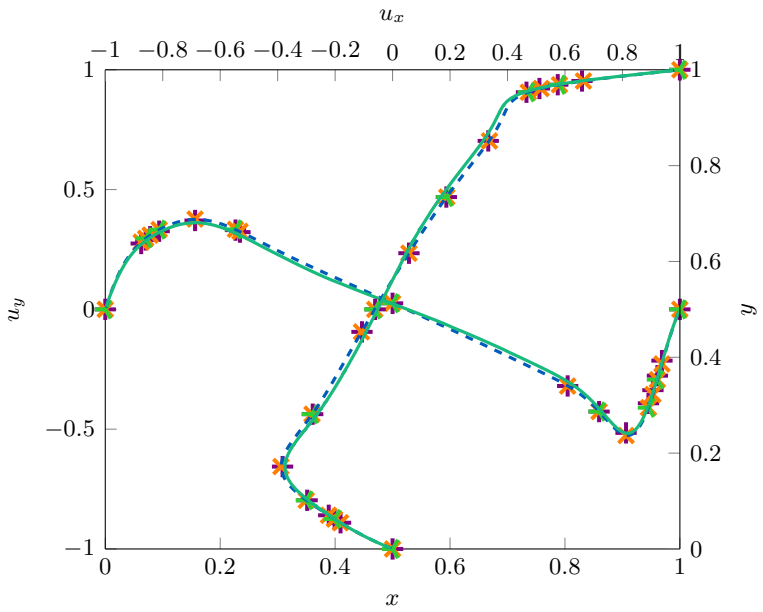


(b) Pressure.

Figure 3.18 – Lid-driven cavity (here $U = -1$): horizontal and vertical vorticity (above) and pressure (below) profiles on the symmetry axes. The discrete vorticity is defined in (3.57), and the pressure is normalized to have $p\left(\frac{1}{2}, \frac{1}{2}\right) = 0$. $Re = 1000$. References: ×: Botella and Peyret (1998), ⊀: Bruneau and Saad (2006). CDO-Fb with Cartesian meshes: 511×511 ——, 255×255 - - -, 127×127 ·····, 63×63 -·-·-.



(a) Triangular mesh - T.



(b) Refined Cartesian mesh - HR.

Figure 3.19 – Lid-driven cavity: horizontal and vertical velocity profiles (cell-based DoFs) on the symmetry axes. $Re = 1000$. References: +: Ghia *et al.* (1982), ×: Botella and Peyret (1998), ⊀: Bruneau and Saad (2006). CDO-Fb meshes: Cartesian: - - -, Triangular: ——, Refined Cartesian: ——.

Chapter 4

First-order time-stepping techniques for the Navier–Stokes equations

Contents

4.1 Preliminary notions . . . 94
4.1.1 Continuous setting . . . 94
4.1.2 Time-discrete setting . . . 95
4.1.3 Fully discrete setting . . . 95
4.2 Velocity-pressure couplings and time-stepping techniques . . . 96
4.2.1 Monolithic approach and the saddle-point problem . . . 97
4.2.2 The Artificial Compressibility technique . . . 98
4.3 Convection treatments . . . 101
4.3.1 Implicit convection: Picard algorithm . . . 101
4.3.2 Linearized convection . . . 103
4.3.3 Explicit convection . . . 104
4.4 Numerical results: Stokes equations . . . 105
4.4.1 Convergence in time . . . 107
4.4.2 Efficiency results . . . 108
4.4.3 Polyhedral meshes . . . 110
4.5 Numerical results: Navier–Stokes equations . . . 113
4.5.1 Convergence in time . . . 114
4.5.2 Convection treatments and dissipativity . . . 116
4.5.3 Stability results with an explicit convection . . . 116
4.6 Detailed results . . . 122
4.6.1 Stokes equations . . . 122
4.6.2 Navier–Stokes equations . . . 126

After having dealt with steady problems, we now move on to unsteady ones. The target problem is then

$$\begin{cases} \dfrac{\partial \underline{u}}{\partial t} - \nu \underline{\Delta}\, \underline{u} + \xi^{\mathrm{NS}} (\underline{u} \cdot \underline{\nabla}\,) \underline{u} + \underline{\nabla}\, p = \underline{f} & \text{in } \Omega \times (0,T)\,, \\ \underline{\nabla} \cdot \underline{u} = 0 & \text{in } \Omega \times (0,T)\,, \\ \underline{u} = \underline{u}_\partial & \text{on } \partial\Omega \times (0,T)\,, \\ \underline{u}_{|t=0} = \underline{u}_0 & \text{on } \Omega\,, \end{cases} \tag{4.1}$$

with $T > 0$ denoting the finite observation time of the flow. Recall that the parameter ξ^{NS} allows us to switch from the Stokes ($\xi^{\mathrm{NS}} := 0$) to the Navier–Stokes equations (NSE, $\xi^{\mathrm{NS}} := 1$). To be compatible with the divergence constraint, the (non)homogeneous Dirichlet datum must satisfy $\int_{\partial\Omega} \underline{u}_\partial \cdot \underline{n}_{\partial\Omega} = 0$ for any $t \in (0, T)$. The Initial Condition (IC), $\underline{u}_0$, is a given function and we assume that it satisfies

$$\underline{\nabla} \cdot \underline{u}_0 = 0 \text{ in } \Omega \quad \text{and} \quad \underline{u}_0 = \underline{u}_\partial \text{ on } \partial\Omega\,. \tag{4.2}$$

These conditions mean that the initial velocity field is compatible. For simplicity we assume that $\underline{u}_\partial = \underline{0}$ in the presentation of the schemes.

This chapter opens with an overview on the two velocity-pressure coupling techniques that we use to address the NSE, the monolithic approach and the Artificial Compressibility (AC) method. A second part deals with various ways of handling the convection term in the time-dependent case. Finally, numerical results are presented to support the theory and to identify the best numerical strategies in practice.

4.1 Preliminary notions

In this section various settings are introduced, moving from the continuous to the fully (time and space) discrete one.

4.1.1 Continuous setting

Let us briefly recall some classical concepts of functional analysis for parabolic PDEs. We will follow Ern and Guermond (2004, Ch. 6). For a more thorough and in-depth discussion specific to the NSE, the reader may also refer to Temam (1977, Ch. III).

Given a function $q(\underline{x}, t)$ defined on the space-time cylinder $\Omega \times (0, T)$ and taking values in $\mathbb{R}^l$, $l \geq 1$, we may look at it as a function of t only and with values in a Banach space V composed of functions from Ω to $\mathbb{R}^l$ and equipped with a norm denoted by $\|\cdot\|_V$. Thus, the elements of V only depend on the space variable $\underline{x}$. In other words, we associate with the function $q(\underline{x}, t)$ the function

$$q\colon (0, T) \ni t \mapsto q(t) \equiv q(\cdot, t) \in V\,, \tag{4.3}$$

and we keep the same notation for simplicity.

Let $\mathcal{C}^\ell([0, T]; V)$ be the Banach space of ℓ-continuously differentiable functions with values in V. We assume that there is a smooth enough solution to the unsteady NSE (4.1) such that

$$\underline{u} \in \mathcal{C}^0([0, T]; \underline{H}^1_0(\Omega)) \cap \mathcal{C}^1([0, T]; \underline{L}^2(\Omega))\,, \quad p \in \mathcal{C}^0([0, T]; L^2_*(\Omega))\,, \tag{4.4}$$

and we assume that $\underline{f} \in \mathcal{C}^0([0, T]; \underline{L}^2(\Omega))$.

A more generic setting for the unsteady NSE is to consider a weak solution in a suitable Bochner–Hilbert space. One critical issue is then the regularity in time of the velocity time-derivative and of the pressure.

Remark 4.1 - Initial pressure. In general, the initial pressure is not a datum for the unsteady NSE. Here we restrict ourselves to smooth solutions so that we can assume that the initial pressure is well-defined. ◇

4.1.2 Time-discrete setting

In order to discretize the reference time interval $[0, T]$, we consider an increasing sequence of time nodes $(t^n)_{n=0,\dots,N}$, with $t^0 = 0$, $t^N \geq T$ and N being a natural positive number. For the sake of simplicity, we assume the time step to be constant, so that, once a time step Δt is chosen, we can write $t^n := n\Delta t$, $n = 0, \dots, N$, $N := \lceil \frac{T}{\Delta t} \rceil$ and $[0,\, T] \subseteq \bigcup_{n=0}^{N-1} [t^n, t^{n+1}]$. In general, Δt is chosen so that there is indeed an integer N such that $N\Delta t = T$.

In this chapter, we consider a simple first-order time-discretization scheme so that the focus is on the velocity-pressure coupling and on how to deal with the convection and the resulting nonlinearity. We thus consider a simple backward Euler time discretization: For $n = 1, \dots, N$, given $\underline{u}^{n-1} \in \underline{H}_0^1(\Omega)$ from the previous time step or the initial condition, find $(\underline{u}^n, p^n) \in \underline{H}_0^1(\Omega) \times L_*^2(\Omega)$ solving

$$\begin{cases} \dfrac{\underline{u}^n - \underline{u}^{n-1}}{\Delta t} - \nu \underline{\Delta}\, \underline{u}^n + \xi^{\mathrm{NS}} (\underline{u}^n \cdot \underline{\nabla}\,) \underline{u}^n + \underline{\nabla}\, p^n = \underline{f}^n := \underline{f}(t^n)\,, \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \underline{\nabla} \cdot \underline{u}^n = 0\,. \end{cases} \tag{4.5}$$

The variational formulation associated with (4.5) is: Find $(\underline{u}^n, p^n) \in \underline{H}_0^1(\Omega) \times L_*^2(\Omega)$ solving

$$\begin{cases} \dfrac{1}{\Delta t} m(\underline{u}^n - \underline{u}^{n-1}, \underline{v}) + \nu a(\underline{u}^n, \underline{v}) + \xi^{\mathrm{NS}} t(\underline{u}^n; \underline{u}^n, \underline{v}) + b(\underline{v}, p^n) = l^n(\underline{v})\,, \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad b(\underline{u}^n, q) = 0\,, \end{cases} \tag{4.6}$$

for all $\underline{v} \in \underline{H}_0^1(\Omega)$ and all $q \in L_*^2(\Omega)$. Recall that

$$a(\underline{w}, \underline{v}) := \int_\Omega \underline{\underline{\nabla}}\, \underline{w} : \underline{\underline{\nabla}}\, \underline{v}\,, \qquad b(\underline{w}, q) := -\int_\Omega q \underline{\nabla} \cdot \underline{w}\,, \tag{4.7a}$$

$$t(\underline{w}; \underline{u}, \underline{v}) := \int_\Omega \Big((\underline{w} \cdot \underline{\nabla}\,) \underline{u} \Big) \cdot \underline{v}\,, \tag{4.7b}$$

and define

$$m(\underline{w}, \underline{v}) := \int_\Omega \underline{w} \cdot \underline{v}\,, \qquad l^n(\underline{v}) := \int_\Omega \underline{f}^n \cdot \underline{v}\,. \tag{4.7c}$$

At this stage, we are considering an implicit treatment of the convection term and a strong velocity-pressure coupling. Alternative strategies are presented later in this chapter.

After a time discretization has been introduced, as in (4.5), the space time function $q \in \mathcal{C}^0([0, T]; V)$ is approximated by $q_N := \{q^n\}_{n=0,\dots,N} \in [V]^{N+1}$ where, for each n, $q^n \approx q(\cdot, t^n) \in V$.

4.1.3 Fully discrete setting

The spatial discretization of (4.6) is now considered. Recall that in the face-based CDO (CDO-Fb) formulation, the velocity $\underline{u}^n$ is approximated as $\widehat{\underline{u}}_{\mathrm{h}}^n \in \widehat{\underline{\mathcal{U}}}_{\mathrm{h},0}$ and the pressure p^n as $p_{\mathrm{h}}^n \in \mathcal{P}_{\mathrm{h},*}$. We then set $\widehat{\underline{u}}_{\mathrm{h},N} := (\widehat{\underline{u}}_{\mathrm{h}}^n)_{n=0,\dots,N} \in [\widehat{\underline{\mathcal{U}}}_{\mathrm{h},0}]^{N+1}$ and $p_{\mathrm{h},N} := (p_{\mathrm{h}}^n)_{n=0,\dots,N} \in [\mathcal{P}_{\mathrm{h},*}]^{N+1}$. For the sake of brevity, whenever no confusion arises between the steady and unsteady problems, the subscript N is dropped: e.g. we write $\widehat{\underline{u}}_{\mathrm{h}} := (\widehat{\underline{u}}_{\mathrm{h}}^n)_{n=0,\dots,N}$.

Similarly to how the source term is treated (see Section 2.6 and (3.19)), the part of the problem resulting from the time-discretization, $\frac{\underline{u}^n - \underline{u}^{n-1}}{\Delta t}$ in (4.5), only involves cell-based DoFs. This is natural in hybrid methods where the equations associated with the cell-based test functions express the balance of a conservation property (e.g. the velocity momentum) in each mesh cell, whereas the equations associated with the face-based test functions express the equilibrium of the fluxes across a given face. Recall that for a given function $\widehat{\underline{v}}_{\mathrm{h}} \in \widehat{\underline{\mathcal{U}}}_{\mathrm{h},0}$, we write $\widehat{\underline{v}}_{\mathrm{h}} = (\underline{v}_{\mathrm{C}}, \underline{v}_{\mathrm{F}})$ to identify the cell- and face-based components of $\widehat{\underline{v}}_{\mathrm{h}}$. Then, (4.6)

translates as follows in the CDO-Fb framework: For $n = 1, \dots, N$, given $\underline{u}_{\mathrm{h}}^{n-1} \in \widehat{\underline{\mathcal{U}}}_{\mathrm{h},0}$ from the previous time step or the initial condition, find $(\widehat{\underline{u}}_{\mathrm{h}}^{n}, p_{\mathrm{h}}^{n}) \in \widehat{\underline{\mathcal{U}}}_{\mathrm{h},0} \times \mathcal{P}_{\mathrm{h},*}$ solving

$$\begin{cases} \dfrac{1}{\Delta t} m(\underline{u}_{\mathrm{C}}^{n} - \underline{u}_{\mathrm{C}}^{n-1}, \underline{v}_{\mathrm{C}}) + \nu a_{\mathrm{h}}(\widehat{\underline{u}}_{\mathrm{h}}^{n}, \widehat{\underline{v}}_{\mathrm{h}}) + \xi^{\mathrm{NS}} t_{\mathrm{h}}(\widehat{\underline{u}}_{\mathrm{h}}^{n}; \widehat{\underline{u}}_{\mathrm{h}}^{n}, \widehat{\underline{v}}_{\mathrm{h}}) + b_{\mathrm{h}}(\widehat{\underline{v}}_{\mathrm{h}}, p_{\mathrm{h}}^{n}) = l^{n}(\underline{v}_{\mathrm{C}}) \,, \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad b_{\mathrm{h}}(\widehat{\underline{u}}_{\mathrm{h}}^{n}, q_{\mathrm{h}}) = 0 \,, \end{cases} \tag{4.8}$$

for all $\widehat{\underline{v}}_{\mathrm{h}} \in \widehat{\underline{\mathcal{U}}}_{\mathrm{h},0}$ and all $q_{\mathrm{h}} \in \mathcal{P}_{\mathrm{h},*}$. Notice that

$$m(\underline{w}_{\mathrm{C}}, \underline{v}_{\mathrm{C}}) := \sum_{\mathrm{c} \in \mathrm{C}} m_{\mathrm{c}}(\underline{w}_{\mathrm{c}}, \underline{v}_{\mathrm{c}}) \,, \qquad m_{\mathrm{c}}(\underline{w}_{\mathrm{c}}, \underline{v}_{\mathrm{c}}) := \int_{\mathrm{c}} \underline{w}_{\mathrm{c}} \cdot \underline{v}_{\mathrm{c}} = |\mathrm{c}| \, \underline{w}_{\mathrm{c}} \cdot \underline{v}_{\mathrm{c}} \,. \tag{4.9}$$

Remark 4.2 - Right-hand side of (4.8)**.** The discussion made in Section 2.6 about how to deal with the source term in the CDO-Fb context, represented by $l^{n}(\underline{v}_{\mathrm{C}})$ in (4.8), is also relevant (after appropriate adaptations) to the unsteady case. In particular, it is possible to consider the two alternatives proposed in Remark 2.51 involving a velocity reconstruction for the test functions. ◇

Time-related mass matrix

From the algebraic point of view, the bilinear term $m(\underline{u}_{\mathrm{C}}^{n}, \underline{v}_{\mathrm{C}})$ is taken into account locally in every cell $\mathrm{c} \in \mathrm{C}$ as a mass matrix lumped on the cell-based DoFs

$$\mathbf{M}_{\mathrm{c}} := \underbrace{\left[\begin{array}{c|c} \mathbf{0}_{\mathrm{F_c F_c}} & \mathbf{0}_{\mathrm{F_c}d} \\ \hline \mathbf{0}_{d\mathrm{F_c}} & |\mathrm{c}| \, \mathbf{I}_{dd} \end{array}\right]}_{d\#\mathrm{F_c} \qquad d} \begin{array}{l} \}d\#\mathrm{F_c} \\ \}d \end{array} \,. \tag{4.10}$$

Similarly, the local contribution of $m(\underline{u}_{\mathrm{C}}^{n-1}, \underline{v}_{\mathrm{C}})$ to the right-hand side reads:

$$\mathbf{R}_{\mathrm{c}}^{n-1} = \left[\begin{array}{c|c||c} \mathbf{0}_{\mathrm{F_c}}^{T} & (\mathbf{m}_{\mathrm{c}})^{T} & 0 \end{array}\right]^{T} \,, \tag{4.11}$$

where $\mathbf{m}_{\mathrm{c}}^{n-1} = |\mathrm{c}| \, \underline{u}_{\mathrm{c}}^{n-1}$.

Recall the local linear system presented in Section 3.1.3, see especially (3.20):

$$\left[\begin{array}{c||c} \mathbf{A}_{\mathrm{c}} & \mathbf{B}_{\mathrm{c}}^{T} \\ \hline\hline \mathbf{B}_{\mathrm{c}} & 0 \end{array}\right] \mathbf{U}_{\mathrm{c}} = \mathbf{F}_{\mathrm{c}} \,. \tag{4.12}$$

For the unsteady NSE, one has

$$\begin{aligned} \mathbf{A}_{\mathrm{c}} &:= \frac{1}{\Delta t} \mathbf{M}_{\mathrm{c}} + \nu \mathbf{G}_{\mathrm{c}} + \xi^{\mathrm{NS}} \mathbf{T}_{\mathrm{c}}(\widehat{\underline{u}}_{\mathrm{c}}^{n}) \,, \\ \mathbf{B}_{\mathrm{c}} &:= \mathbf{D}_{\mathrm{c}} \,, \\ \mathbf{F}_{\mathrm{c}}^{n} &:= \mathbf{S}_{\mathrm{c}}^{n} + \frac{1}{\Delta t} \mathbf{R}_{\mathrm{c}}^{n-1} \,. \end{aligned} \tag{4.13}$$

4.2 Velocity-pressure couplings and time-stepping techniques

In this section, we detail two coupling techniques used in time-stepping schemes for the NSE: the monolithic approach and the AC method. They have already been briefly introduced in

Section 1.3.5. Here, we give more details. Recall that the monolithic approach aims at the maximum accuracy of the solution, whereas the AC technique focuses on the efficiency.

Since the velocity-pressure coupling is a consequence of the incompressibility constraint, it is present both in the Stokes and NSE. Hence, in order to keep the discussion as simple as possible, we focus on the Stokes problem in the rest of this section, i.e. we set $\xi^{\mathrm{NS}} := 0$ in (4.8).

4.2.1 Monolithic approach and the saddle-point problem

We will use the term *monolithic approach* to refer to the classical formulation of the Stokes or NSE which leads to a saddle-point system. As it was pointed out in Section 1.3.5, adapted algorithms and considerable numerical effort might be needed to solve this type of problem. On the other hand, one obtains the best possible solution that the chosen time-scheme and the spatial discretization provide, since no further approximations are considered. In particular, the velocity field is discretely divergence-free.

The discrete CDO system given in (4.8) with $\xi^{\mathrm{NS}} := 0$ becomes: For $n = 1, \dots, N$, given $\underline{\widehat{u}}_{\mathrm{h}}^{n-1} \in \underline{\widehat{\mathcal{U}}}_{\mathrm{h},0}$, find $(\underline{\widehat{u}}_{\mathrm{h}}^{n}, p_{\mathrm{h}}^{n}) \in \underline{\widehat{\mathcal{U}}}_{\mathrm{h},0} \times \mathcal{P}_{\mathrm{h},*}$ such that

$$\begin{cases} \dfrac{1}{\Delta t} m(\underline{u}_{\mathrm{C}}^{n} - \underline{u}_{\mathrm{C}}^{n-1}, \underline{v}_{\mathrm{C}}) + \nu a_{\mathrm{h}}(\underline{\widehat{u}}_{\mathrm{h}}^{n}, \underline{\widehat{v}}_{\mathrm{h}}) + b_{\mathrm{h}}(\underline{\widehat{v}}_{\mathrm{h}}, p_{\mathrm{h}}^{n}) = l^{n}(\underline{v}_{\mathrm{C}}) \,, & \text{(4.14a)} \\ b_{\mathrm{h}}(\underline{\widehat{u}}_{\mathrm{h}}^{n}, q_{\mathrm{h}}) = 0 \,, & \text{(4.14b)} \end{cases}$$

for all $\underline{\widehat{v}}_{\mathrm{h}} \in \underline{\widehat{\mathcal{U}}}_{\mathrm{h},0}$ and all $q_{\mathrm{h}} \in \mathcal{P}_{\mathrm{h},*}$. Similarly the algebraic realization presented in Eqs. (4.10) to (4.13) becomes:

$$\begin{gathered} \left[\begin{array}{c||c} \mathbf{A}_{\mathrm{c}} & \mathbf{B}_{\mathrm{c}}^{T} \\ \hline\hline \mathbf{B}_{\mathrm{c}} & 0 \end{array}\right] \mathbf{U}_{\mathrm{c}}^{n} = \mathbf{F}_{\mathrm{c}}^{n} \,, \\ \mathbf{A}_{\mathrm{c}} := \frac{1}{\Delta t}\mathbf{M}_{\mathrm{c}} + \nu \mathbf{G}_{\mathrm{c}} \,, \quad \mathbf{B}_{\mathrm{c}} := \mathbf{D}_{\mathrm{c}} \,, \quad \mathbf{F}_{\mathrm{c}}^{n} := \mathbf{S}_{\mathrm{c}}^{n} + \frac{1}{\Delta t}\mathbf{R}_{\mathrm{c}}^{n-1} \,. \end{gathered} \tag{4.15}$$

Energy balance

Let us investigate the energy balance of the CDO-Fb Stokes problem with a monolithic approach and an Implicit Euler time discretization. Given a discrete velocity field $\underline{\widehat{v}}_{\mathrm{h}} \in \underline{\widehat{\mathcal{U}}}_{\mathrm{h}}$, we define the corresponding (cell-based) kinetic energy as

$$\mathcal{E}_{\mathrm{kin,h}}(\underline{\widehat{v}}_{\mathrm{h}}) := \frac{1}{2} \|\underline{v}_{\mathrm{C}}\|^{2}_{\underline{L}^{2}(\Omega)} \,, \qquad \|\underline{v}_{\mathrm{C}}\|^{2}_{\underline{L}^{2}(\Omega)} = \|\underline{\widehat{v}}_{\mathrm{h}}\|^{2}_{\mathrm{C}} = \sum_{\mathrm{c}\in\mathrm{C}} |\mathrm{c}|\, |\underline{v}_{\mathrm{c}}|_{2}^{2} \,. \tag{4.16}$$

It is clear that $\mathcal{E}_{\mathrm{kin,h}}(\underline{\widehat{v}}_{\mathrm{h}}) \geq 0$ for all $\underline{\widehat{v}}_{\mathrm{h}} \in \underline{\widehat{\mathcal{U}}}_{\mathrm{h}}$.

Lemma 4.3 - Energy balance for (4.14). *Assume* $\underline{f} := \underline{0}$. *Let* $n = 1, \dots, N$ *and let* $(\underline{\widehat{u}}_{\mathrm{h}}^{n}, p_{\mathrm{h}}^{n})$ *solve* (4.14). *The following holds true:*

$$\mathcal{E}_{\mathrm{kin,h}}(\underline{\widehat{u}}_{\mathrm{h}}^{n}) - \mathcal{E}_{\mathrm{kin,h}}(\underline{\widehat{u}}_{\mathrm{h}}^{n-1}) + \underbrace{\mathcal{E}_{\mathrm{kin,h}}(\underline{\widehat{u}}_{\mathrm{h}}^{n} - \underline{\widehat{u}}_{\mathrm{h}}^{n-1})}_{\geq 0} + \underbrace{\nu \Delta t \left\| \underline{\underline{\mathrm{G}}}_{\mathrm{h}}(\underline{\widehat{u}}_{\mathrm{h}}^{n}) \right\|^{2}_{\underline{\underline{L}}^{2}(\Omega)}}_{\geq 0} = 0 \,. \tag{4.17}$$

The two rightmost terms in (4.17) being nonnegative, the scheme (4.14) is dissipative, i.e. we have

$$\mathcal{E}_{\mathrm{kin,h}}(\underline{\widehat{u}}_{\mathrm{h}}^{n}) \leq \mathcal{E}_{\mathrm{kin,h}}(\underline{\widehat{u}}_{\mathrm{h}}^{n-1}) \,. \tag{4.18}$$

Proof. Consider (4.14a) (recall $\underline{f} := \underline{0}$), multiply it by Δt and set $\widehat{\underline{v}}_{\mathrm{h}} := \widehat{\underline{u}}_{\mathrm{h}}^n$. The incompressibility constraint (4.14b) enforced by the monolithic approach gives $b_{\mathrm{h}}(\widehat{\underline{u}}_{\mathrm{h}}^n, p_{\mathrm{h}}^n) = 0$. Concerning the diffusion contribution, using the definition of $a_{\mathrm{h}}(\cdot,\cdot)$, (3.11), it is readily seen that

$$\nu a_{\mathrm{h}}(\widehat{\underline{u}}_{\mathrm{h}}^n, \widehat{\underline{u}}_{\mathrm{h}}^n) = \nu \left\| \underline{\underline{\mathrm{G}}}_{\mathrm{h}}(\widehat{\underline{u}}_{\mathrm{h}}^n) \right\|^2_{\underline{\underline{L}}^2(\Omega)} . \tag{4.19}$$

It is now left to investigate the mass-related term $m(\cdot,\cdot)$. Owing to the definition (4.9) and trivial manipulations, one has

$$m(\underline{u}_{\mathrm{C}}^n - \underline{u}_{\mathrm{C}}^{n-1}, \underline{u}_{\mathrm{C}}^n) = \sum_{\mathrm{c}\in\mathrm{C}} \int_{\mathrm{c}} (\underline{u}_{\mathrm{c}}^n - \underline{u}_{\mathrm{c}}^{n-1}) \cdot \underline{u}_{\mathrm{c}}^n = \left(\underline{u}_{\mathrm{C}}^n - \underline{u}_{\mathrm{C}}^{n-1}, \underline{u}_{\mathrm{C}}^n\right)_{\underline{L}^2(\Omega)} . \tag{4.20}$$

The following identity is readily verified

$$\pm\, 2\, (\underline{v}, \underline{w})_{\underline{L}^2(\Omega)} = \|\underline{v} \pm \underline{w}\|^2_{\underline{L}^2(\Omega)} - \|\underline{v}\|^2_{\underline{L}^2(\Omega)} - \|\underline{w}\|^2_{\underline{L}^2(\Omega)} . \tag{4.21}$$

Plugging (4.21) in (4.20), one has

$$\begin{aligned} m(\underline{u}_{\mathrm{C}}^n - \underline{u}_{\mathrm{C}}^{n-1}, \underline{u}_{\mathrm{C}}^n) &= \frac{1}{2} \|\underline{u}_{\mathrm{C}}^n\|^2_{\underline{L}^2(\Omega)} - \frac{1}{2} \left\|\underline{u}_{\mathrm{C}}^{n-1}\right\|^2_{\underline{L}^2(\Omega)} + \frac{1}{2} \left\|\underline{u}_{\mathrm{C}}^n - \underline{u}_{\mathrm{C}}^{n-1}\right\|^2_{\underline{L}^2(\Omega)} \\ &= \mathcal{E}_{\mathrm{kin,h}}(\widehat{\underline{u}}_{\mathrm{h}}^n) - \mathcal{E}_{\mathrm{kin,h}}(\widehat{\underline{u}}_{\mathrm{h}}^{n-1}) + \mathcal{E}_{\mathrm{kin,h}}(\widehat{\underline{u}}_{\mathrm{h}}^n - \widehat{\underline{u}}_{\mathrm{h}}^{n-1}) , \end{aligned} \tag{4.22}$$

where the last equality follows from the definition (4.16) of $\mathcal{E}_{\mathrm{kin,h}}(\cdot)$. One has thus recovered the three leftmost terms in (4.17), and the proof is complete. □

4.2.2 The Artificial Compressibility technique

Let us recall here the main features of the Artificial Compressibility technique and discuss how it is adapted to the CDO-Fb setting.

Consider the Stokes equations discretized only in time, similarly to what has been done in (4.5). The key point of the AC method is that the velocity-pressure coupling in the momentum equation is broken by considering an additional grad-div term and by approximating the pressure with the previous step solution. The pressure is subsequently updated at the end of the time step. When applied to (4.5) with $\xi^{\mathrm{NS}} := 0$, this procedure leads to

$$\begin{cases} \dfrac{\underline{u}^n - \underline{u}^{n-1}}{\Delta t} - \nu \underline{\Delta}\, \underline{u}^n + \underline{\nabla}\, p^n = \underline{f}^n , \\ p^n - p^{n-1} + \nu\eta \underline{\nabla} \cdot \underline{u}^n = 0 , \end{cases} \tag{4.23}$$

Where η is a user-dependent nondimensional parameter. Rearranging (4.23) leads to

$$\frac{\underline{u}^n - \underline{u}^{n-1}}{\Delta t} - \nu(\underline{\Delta}\, \underline{u}^n + \eta \underline{\nabla}\, \underline{\nabla} \cdot \underline{u}^n) = \underline{f}^n - \underline{\nabla}\, p^{n-1} , \tag{4.24a}$$

$$p^n = p^{n-1} - \nu\eta \underline{\nabla} \cdot \underline{u}^n . \tag{4.24b}$$

Let us have a look at the role of the additional grad-div operator in the momentum equation. Applying an integration by parts and taking into account the boundary conditions yields for any smooth functions $\underline{v}$ and $\underline{w}$

$$-\int_\Omega (\underline{\nabla}\, \underline{\nabla} \cdot \underline{w}) \cdot \underline{v} = \int_\Omega \underline{\nabla} \cdot \underline{w}\, \underline{\nabla} \cdot \underline{v} + \int_{\partial\Omega} \underline{\nabla} \cdot \underline{w} (\underline{v} \cdot \underline{n}_{\partial\Omega}) = \int_\Omega \underline{\nabla} \cdot \underline{w}\, \underline{\nabla} \cdot \underline{v} . \tag{4.25}$$

Hence the grad-div term leads to a dissipative contribution to the energy balance.

Let us apply the AC technique to the CDO-Fb discrete problem (4.8) with $\xi^{\mathrm{NS}} := 0$. The grad-div term is discretized by using the CDO-Fb divergence operator (given Definition 2.25), and which we recall here for convenience

$$\begin{aligned} d_{\mathrm{h}}(\widehat{\underline{w}}_{\mathrm{h}}, \widehat{\underline{v}}_{\mathrm{h}}) &:= \sum_{\mathrm{c}\in\mathrm{C}} d_{\mathrm{c}}(\widehat{\underline{w}}_{\mathrm{c}}, \widehat{\underline{v}}_{\mathrm{c}}) \,, \\ d_{\mathrm{c}}(\widehat{\underline{w}}_{\mathrm{c}}, \widehat{\underline{v}}_{\mathrm{c}}) &:= \int_{\mathrm{c}} \mathrm{D}_{\mathrm{c}}(\widehat{\underline{w}}_{\mathrm{c}})\mathrm{D}_{\mathrm{c}}(\widehat{\underline{v}}_{\mathrm{c}}) = \frac{1}{|\mathrm{c}|}\left(\sum_{\mathrm{f}\in\mathrm{F}_{\mathrm{c}}} |\mathrm{f}|\, \underline{n}_{\mathrm{fc}} \cdot \underline{w}_{\mathrm{f}}\right)\left(\sum_{\mathrm{f}\in\mathrm{F}_{\mathrm{c}}} |\mathrm{f}|\, \underline{n}_{\mathrm{fc}} \cdot \underline{v}_{\mathrm{f}}\right) . \end{aligned} \tag{4.26}$$

The AC version of (4.8) thus reads: For $n = 1, \ldots, N$, given $\widehat{\underline{u}}_{\mathrm{h}}^{n-1} \in \widehat{\underline{\mathcal{U}}}_{\mathrm{h},0}$, find $\widehat{\underline{u}}_{\mathrm{h}}^{n} \in \widehat{\underline{\mathcal{U}}}_{\mathrm{h},0}$ such that

$$\frac{1}{\Delta t} m(\underline{u}_{\mathrm{C}}^{n} - \underline{u}_{\mathrm{C}}^{n-1}, \underline{v}_{\mathrm{C}}) + \nu\left(a_{\mathrm{h}}(\widehat{\underline{u}}_{\mathrm{h}}^{n}, \widehat{\underline{v}}_{\mathrm{h}}) + \eta d_{\mathrm{h}}(\widehat{\underline{u}}_{\mathrm{h}}^{n}, \widehat{\underline{v}}_{\mathrm{h}})\right) = l^{n}(\underline{v}_{\mathrm{C}}) - b_{\mathrm{h}}(\widehat{\underline{v}}_{\mathrm{h}}, p_{\mathrm{h}}^{n-1}) \,, \tag{4.27a}$$

for all $\widehat{\underline{v}}_{\mathrm{h}} \in \widehat{\underline{\mathcal{U}}}_{\mathrm{h},0}$, and then update the pressure as follows:

$$p_{\mathrm{h}}^{n} = p_{\mathrm{h}}^{n-1} - \nu\eta \mathrm{D}_{\mathrm{h}}(\widehat{\underline{u}}_{\mathrm{h}}^{n}) \,. \tag{4.27b}$$

See Remark 4.5 below on the pressure initialization.

Remark 4.4 - Pressure mass matrix. In order to derive (4.27b), one considers the mass matrix associated with the spatial discretization chosen for the pressure. In the CDO-Fb discretization, the pressure is cell-wise constant and the resulting mass-matrix is then diagonal with entries equal to the volumes of the cells. Similarly, the divergence, which is at the right-hand side of (4.27b), is cell-wise constant as well. Hence, it is trivial to eliminate the mass matrix and this yields (4.27b). ◇

From an algebraic standpoint, the grad-div term in (4.27a) translates into the following matrix:

$$\mathbf{L}_{\mathrm{c}} = \left[\begin{array}{ccc|ccc|c} \mathbf{L}_{\mathrm{f}_1^i\mathrm{f}_1^i} & \cdots & & \mathbf{L}_{\mathrm{f}_1^i\mathrm{f}_1^b} & \cdots & & \mathbf{0}_{dd} \\ \vdots & \ddots & & \vdots & \ddots & & \vdots \\ \hline \mathbf{L}_{\mathrm{f}_1^b\mathrm{f}_1^i} & \cdots & & \mathbf{L}_{\mathrm{f}_1^b\mathrm{f}_1^b} & \cdots & & \mathbf{0}_{dd} \\ \vdots & \ddots & & \vdots & \ddots & & \vdots \\ \hline \mathbf{0}_{dd} & \cdots & & \mathbf{0}_{dd} & \cdots & & \mathbf{0}_{dd} \end{array}\right] , \tag{4.28}$$

where $\mathbf{L}_{\mathrm{f}_l\mathrm{f}_m} = \frac{|\mathrm{f}_l||\mathrm{f}_m|}{|\mathrm{c}|} \underline{n}_{\mathrm{f}_l} \otimes \underline{n}_{\mathrm{f}_m}$ for all $\mathrm{f}_l, \mathrm{f}_m \in \mathrm{F}_{\mathrm{c}}$. The structure of the matrix is dense in the face-face block and all the faces of a cell may be coupled together, whereas the matrix has null blocks in the cell-related part, since the cell DoFs are not involved in the computation of the divergence (see (2.45)). Moreover, differently from the gradient operator, the submatrices forming $\mathbf{L}_{\mathrm{c}}$ are not a priori diagonal and couple the different Cartesian components of the velocity. This is also the case for the diagonal blocks $\mathbf{L}_{\mathrm{f},\mathrm{f}}$.

The local problem resulting from the AC technique is substantially different from (4.15): in fact, the saddle-point structure is traded against a symmetric positive-definite matrix, and the size of the system to be solved is reduced since the pressure DoFs are no longer part of the unknowns. The AC technique combined with the CDO-Fb discretization thus leads to the following (local) two-step procedure. First, the linear system is assembled from the following local contributions: For every cell $\mathrm{c} \in \mathrm{C}$,

$$\mathbf{A}_{\mathrm{c}} \left[\begin{array}{c} \mathbf{u}_{\mathrm{f}}^{n} \\ \hline \mathbf{u}_{\mathrm{c}}^{n} \end{array}\right] = [-\mathbf{D}_{\mathrm{c}}^{T} p_{\mathrm{c}}^{n-1} \,|\, \mathbf{f}_{\mathrm{c}}]^{T} \,, \tag{4.29}$$

with

$$\begin{aligned} \mathbf{A}_{\mathrm{c}} &:= \frac{1}{\Delta t}\mathbf{M}_{\mathrm{c}} + \nu\left(\mathbf{G}_{\mathrm{c}} + \eta\mathbf{L}_{\mathrm{c}}\right) , \\ \mathbf{f}_{\mathrm{c}}^{n} &:= \mathbf{s}_{\mathrm{c}}^{n} + \frac{1}{\Delta t}\mathbf{m}_{\mathrm{c}}^{n} . \end{aligned} \tag{4.30}$$

The size of the linear system, after the elimination of the cell-based DoFs and the assembly is just $d\#\mathrm{F}$. Then, one performs the pressure update: For every $\mathrm{c} \in \mathrm{C}$,

$$p_{\mathrm{c}}^{n} = p_{\mathrm{c}}^{n-1} - \nu\eta\mathbf{D}_{\mathrm{c}}\left[\frac{\mathbf{u}_{\mathrm{f}}^{n}}{\mathbf{u}_{\mathrm{c}}^{n}}\right] , \tag{4.31}$$

Remark 4.5 - Pressure initialization. System (4.24) needs an additional IC on the pressure, p_0. If not available, it can be computed as suggested in Guermond and Minev (2015) using the velocity IC and the forcing term at $t = 0$, provided enough smoothness is available. Indeed, considering the momentum equation of the Stokes problem, $\xi^{\mathrm{NS}} := 0$, at $t = 0$ and taking its divergence, one obtains after simplifications

$$\begin{cases} \Delta p_0 = \underline{\nabla} \cdot \underline{f}_{|t=0} & \text{on } \Omega , \\ \underline{\nabla} p_0 \cdot \underline{n}_{\partial\Omega} = (\underline{f}_{|t=0} + \nu\underline{\Delta}\,\underline{u}_0) \cdot \underline{n}_{\partial\Omega} & \text{on } \partial\Omega , \end{cases} \tag{4.32}$$

where a Neumann boundary condition has been considered to close the Poisson problem on the initial pressure. However, in our numerical experiments, the initial pressure is always known analytically, so that we are going to explicitly use it. ◇

Energy balance

Differently from Section 4.2.1, we are going to analyze the dissipativity of the AC method by considering an energy which depends on the velocity and on the pressure. Specifically, given $\underline{\widehat{v}}_{\mathrm{h}} \in \underline{\widehat{\mathcal{U}}}_{\mathrm{h},0}$, $q_{\mathrm{h}} \in \mathcal{P}_{\mathrm{h},*}$ and $\Delta t, \eta, \nu > 0$, we define

$$\mathcal{E}_{\mathrm{AC,h}}(\underline{\widehat{v}}_{\mathrm{h}}, q_{\mathrm{h}}) := \mathcal{E}_{\mathrm{kin,h}}(\underline{\widehat{v}}_{\mathrm{h}}) + \frac{\Delta t}{2\nu\eta}\left\|q_{\mathrm{h}}\right\|_{L^2(\Omega)}^2 . \tag{4.33}$$

One has $\mathcal{E}_{\mathrm{AC,h}}(\underline{\widehat{v}}_{\mathrm{h}}, q_{\mathrm{h}}) \geq 0$ for all $\underline{\widehat{v}}_{\mathrm{h}} \in \underline{\widehat{\mathcal{U}}}_{\mathrm{h},0}$ and all $q_{\mathrm{h}} \in \mathcal{P}_{\mathrm{h},*}$.

Lemma 4.6 - Energy balance for (4.27). *Assume $\underline{f} := \underline{0}$. Let $n = 1, \ldots, N$ and let $(\underline{\widehat{u}}_{\mathrm{h}}^{n}, p_{\mathrm{h}}^{n})$ solve* (4.27)*. The following holds true:*

$$\mathcal{E}_{\mathrm{AC,h}}(\underline{\widehat{u}}_{\mathrm{h}}^{n}, p_{\mathrm{h}}^{n}) - \mathcal{E}_{\mathrm{AC,h}}(\underline{\widehat{u}}_{\mathrm{h}}^{n-1}, p_{\mathrm{h}}^{n-1}) + \underbrace{\mathcal{E}_{\mathrm{AC,h}}(\underline{\widehat{u}}_{\mathrm{h}}^{n} - \underline{\widehat{u}}_{\mathrm{h}}^{n-1}, p_{\mathrm{h}}^{n} - p_{\mathrm{h}}^{n-1})}_{\geq 0} + \underbrace{\nu\Delta t\left\|\underline{\underline{\mathrm{G}}}_{\mathrm{h}}(\underline{\widehat{u}}_{\mathrm{h}}^{n})\right\|_{\underline{\underline{L}}^2(\Omega)}^2}_{\geq 0} = 0 . \tag{4.34}$$

As for the monolithic scheme, the diffusion contribution $\Delta t\left\|\underline{\underline{\mathrm{G}}}_{\mathrm{h}}(\underline{\widehat{u}}_{\mathrm{h}}^{n})\right\|_{\underline{\underline{L}}^2(\Omega)}^2$ and the cross-term $\mathcal{E}_{\mathrm{AC,h}}(\underline{\widehat{u}}_{\mathrm{h}}^{n} - \underline{\widehat{u}}_{\mathrm{h}}^{n-1}, p_{\mathrm{h}}^{n} - p_{\mathrm{h}}^{n-1})$ are nonnegative, so that the AC scheme is dissipative as well.

Proof. We follow Ern and Guermond (2020, Lemma 75.2) to which the reader is referred also for a more general case ($\underline{f} \neq \underline{0}$ and a function g^n on the right-hand side of the second line of (4.23)). Instead of addressing directly (4.27), we start from (4.23) with $\underline{f} = \underline{0}$, and after simple manipulations, we obtain

$$\begin{cases} \dfrac{1}{\Delta t}m(\underline{u}_{\mathrm{C}}^{n} - \underline{u}_{\mathrm{C}}^{n-1}, \underline{v}_{\mathrm{C}}) + \nu a_{\mathrm{h}}(\underline{\widehat{u}}_{\mathrm{h}}^{n}, \underline{\widehat{v}}_{\mathrm{h}}) + b_{\mathrm{h}}(\underline{\widehat{v}}_{\mathrm{h}}, p_{\mathrm{h}}^{n}) = 0 , & \text{(4.35a)} \\ \left(p_{\mathrm{h}}^{n} - p_{\mathrm{h}}^{n-1}, q_{\mathrm{h}}\right)_{L^2(\Omega)} - \nu\eta b_{\mathrm{h}}(\underline{\widehat{u}}_{\mathrm{h}}^{n}, q_{\mathrm{h}}) = 0 , & \text{(4.35b)} \end{cases}$$

for all $\widehat{\underline{v}}_{\mathrm{h}} \in \widehat{\underline{\mathcal{U}}}_{\mathrm{h},0}$ and all $q_{\mathrm{h}} \in \mathcal{P}_{\mathrm{h},*}$. Now, we take $\widehat{\underline{v}}_{\mathrm{h}} := \widehat{\underline{u}}_{\mathrm{h}}^n$ and $q_{\mathrm{h}} := p_{\mathrm{h}}^n$, multiply (4.35a) by Δt and add the result to (4.35b) multiplied by $\frac{\Delta t}{\nu\eta}$: the terms involving the bilinear form $b_{\mathrm{h}}(\cdot,\cdot)$ sum up to zero. This gives

$$\left(\underline{u}_{\mathrm{C}}^n - \underline{u}_{\mathrm{C}}^{n-1}, \underline{u}_{\mathrm{C}}^n\right)_{\underline{L}^2(\Omega)} + \frac{\Delta t}{\nu\eta}\left(p_{\mathrm{h}}^n - p_{\mathrm{h}}^{n-1}, p_{\mathrm{h}}^n\right)_{L^2(\Omega)} + \nu\Delta t \left\|\underline{\underline{\mathrm{G}}}_{\mathrm{h}}(\widehat{\underline{u}}_{\mathrm{h}}^n)\right\|^2_{\underline{\underline{L}}^2(\Omega)} = 0\,. \tag{4.36}$$

Now, using the identity (4.21) to deal with the first two terms of (4.36) and recalling the definition (4.33) of $\mathcal{E}_{\mathrm{AC,h}}$, one recovers (4.34). □

4.3 Convection treatments

In this section, we deal with three techniques of increasing numerical simplicity to handle the nonlinearity induced by the convection term in the NSE.

4.3.1 Implicit convection: Picard algorithm

The idea here is to solve directly the nonlinearity. In order to do so, one usually takes advantage of fixed-point procedures, which are iterative procedures. We have given a brief presentation of some classes of such methods in Section 1.3.4, where we also presented the Picard iterations. Here, we briefly recast the material in the CDO-Fb setting.

Picard iterations and monolithic approach

The algorithm is readily adapted to the monolithic CDO-Fb formulation (see (4.14)) as follows: For $n = 1, \ldots, N$, iterate on $k \geq 1$ until convergence: find $(\widehat{\underline{u}}_{\mathrm{h}}^{n,k}, p_{\mathrm{h}}^{n,k}) \in \widehat{\underline{\mathcal{U}}}_{\mathrm{h},0} \times \mathcal{P}_{\mathrm{h},*}$ such that

$$\begin{cases} \dfrac{1}{\Delta t} m(\underline{u}_{\mathrm{C}}^{n,k} - \underline{u}_{\mathrm{C}}^{n-1,\infty}, \underline{v}_{\mathrm{C}}) + \nu a_{\mathrm{h}}(\widehat{\underline{u}}_{\mathrm{h}}^{n,k}, \widehat{\underline{v}}_{\mathrm{h}}) + t_{\mathrm{h}}(\widehat{\underline{u}}_{\mathrm{h}}^{n,k-1}; \widehat{\underline{u}}_{\mathrm{h}}^{n,k}, \widehat{\underline{v}}_{\mathrm{h}}) + b_{\mathrm{h}}(\widehat{\underline{v}}_{\mathrm{h}}, p_{\mathrm{h}}^{n,k}) = l^n(\underline{v}_{\mathrm{C}})\,, \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad b_{\mathrm{h}}(\widehat{\underline{u}}_{\mathrm{h}}^{n,k}, q_{\mathrm{h}}) = 0\,. \end{cases} \tag{4.37}$$

for all $\widehat{\underline{v}}_{\mathrm{h}} \in \widehat{\underline{\mathcal{U}}}_{\mathrm{h},0}$ and all $q_{\mathrm{h}} \in \mathcal{P}_{\mathrm{h},*}$. Here $\widehat{\underline{u}}_{\mathrm{h}}^{n-1,\infty}$ denotes the solution given by the Picard algorithm at the time step $n-1$. The time-stepping is initialized with an initial condition $\widehat{\underline{u}}_{\mathrm{h}}^{0,\infty} := \widehat{\underline{\pi}}_{\mathrm{h}}(\underline{u}_0)$. Moreover, at each time step, the Picard iterations have to be initialized: a suitable choice is to take the solution at the previous time step, i.e.,

$$\underline{u}_{\mathrm{h}}^{n,0} := \underline{u}_{\mathrm{h}}^{n-1,\infty}. \tag{4.38}$$

Remark 4.7 - Picard iterations and Oseen problem. At each Picard iteration of a given time step n, one has to solve an Oseen-like problem. ◇

Energy balance

Lemma 4.8 - Energy balance for (4.37). *Assume $\underline{f} := \underline{0}$. Let $n = 1, \ldots, N$ and let $(\widehat{\underline{u}}_{\mathrm{h}}^n, p_{\mathrm{h}}^n)$ solve (4.37). The following holds true:*

$$\begin{aligned} &\mathcal{E}_{\mathrm{kin,h}}(\widehat{\underline{u}}_{\mathrm{h}}^{n,k}) - \mathcal{E}_{\mathrm{kin,h}}(\widehat{\underline{u}}_{\mathrm{h}}^{n-1,\infty}) \\ &\quad + \underbrace{\mathcal{E}_{\mathrm{kin,h}}(\widehat{\underline{u}}_{\mathrm{h}}^{n,k} - \widehat{\underline{u}}_{\mathrm{h}}^{n-1,\infty})}_{\geq 0} + \underbrace{\nu\Delta t \left\|\underline{\underline{\mathrm{G}}}_{\mathrm{h}}(\widehat{\underline{u}}_{\mathrm{h}}^{n,k})\right\|^2_{\underline{\underline{L}}^2(\Omega)}}_{\geq 0} + \underbrace{t_{\mathrm{h}}(\widehat{\underline{u}}_{\mathrm{h}}^{n,k-1}; \widehat{\underline{u}}_{\mathrm{h}}^{n,k}, \widehat{\underline{u}}_{\mathrm{h}}^{n,k})}_{\geq 0} = 0\,. \end{aligned} \tag{4.39}$$

Proof. The results is readily proved by proceeding as in the proof of Lemma 4.3. The sign of the contribution of the convection term is recovered from Lemma 2.45 about the positivity of the operator $t_{\mathrm{h}}(\cdot;\cdot,\cdot)$ (see (2.94) in particular). Notice that the hypothesis on the convective field being discretely divergence-free requested in Lemma 2.45 is satisfied by $\underline{\widehat{u}}_{\mathrm{h}}^{n,k-1}$ because it has been computed by solving an Oseen-like problem with the monolithic approach. □

Lemma 4.8 can be readily extended to the limit $k \to \infty$, thus to the solution of the nonlinear system. On the other hand, since Lemma 4.8 holds for all $k \geq 1$, it means that one can stop the algorithm at any iteration k and the dissipativity of the time-scheme is always ensured.

Remark 4.9 - Skew-symmetry. The contribution of the convection term may vanish under additional assumptions, namely, if no upwind stabilization is considered in the discrete convection operator (that is $\Xi^{\mathrm{upw}} := 0$, see the definition in (2.88) and, in particular, (2.91)). As a matter of fact, it has been pointed out in Lemma 2.45 (see result *ii)* in particular) that $t_{\mathrm{h}}(\underline{\widehat{w}}_{\mathrm{h}};\cdot,\cdot)$ is skew-symmetric whenever (Sk1) $\mathrm{D}_{\mathrm{c}}(\underline{\widehat{w}}_{\mathrm{c}}) = 0$, for all $\mathrm{c} \in \mathrm{C}$, (Sk2) $\underline{w}_{\mathrm{f}} \cdot \underline{n}_{\mathrm{f}} = 0$ for all $\mathrm{f} \in \mathrm{F}^b$, and (Sk3) no upwind stabilization is considered. In (4.37), we consider $\underline{\widehat{w}}_{\mathrm{h}} := \underline{\widehat{u}}_{\mathrm{h}}^{n,k-1}$. Hypothesis (Sk1) is then satisfied since the monolithic approach is used at each iteration of the Picard algorithm. Hypothesis (Sk2) is satisfied as well, since, for all k, one seeks $\underline{\widehat{u}}_{\mathrm{h}}^{n,k}$ in $\underline{\widehat{\mathcal{U}}}_{\mathrm{h},0}$. Finally, hypothesis (Sk3) is assumed. ◇

Picard iterations and Artificial Compressibility

A Picard algorithm, or indeed any nonlinear procedure involving iterating at each time step, would somehow go against the driving principles of the AC. In fact, the method has been developed on the basis of avoiding cumbersome strategies to address the velocity-pressure coupling and accepting the related error: one would hence accept also an approximation of the nonlinearity. However, when investigating all the possible strategies, we are also going to consider the AC method with Picard iterations. The system then reads: For $n = 1, \ldots, N$, iterate on $k \geq 1$ until convergence: find $\underline{u}_{\mathrm{h}}^{n,k} \in \underline{\widehat{\mathcal{U}}}_{\mathrm{h},0}$ such that

$$\frac{1}{\Delta t} m(\underline{u}_{\mathrm{C}}^{n,k} - \underline{u}_{\mathrm{C}}^{n-1,\infty}, \underline{v}_{\mathrm{C}}) + \nu \left(a_{\mathrm{h}}(\underline{\widehat{u}}_{\mathrm{h}}^{n,k}, \underline{\widehat{v}}_{\mathrm{h}}) + \eta d_{\mathrm{h}}(\underline{\widehat{u}}_{\mathrm{h}}^{n,k}, \underline{\widehat{v}}_{\mathrm{h}}) \right) + t_{\mathrm{h}}(\underline{\widehat{u}}_{\mathrm{h}}^{n,k-1}; \underline{\widehat{u}}_{\mathrm{h}}^{n,k}, \underline{\widehat{v}}_{\mathrm{h}}) = l^n(\underline{v}_{\mathrm{C}}) - b_{\mathrm{h}}(\underline{\widehat{v}}_{\mathrm{h}}, p_{\mathrm{h}}^{n-1,\infty}) \,, \quad (4.40\mathrm{a})$$

for all $\underline{\widehat{v}}_{\mathrm{h}} \in \underline{\widehat{\mathcal{U}}}_{\mathrm{h},0}$, and then set

$$p_{\mathrm{h}}^{n} = p_{\mathrm{h}}^{n-1} - \nu\eta \mathrm{D}_{\mathrm{h}}(\underline{\widehat{u}}_{\mathrm{h}}^{n,\infty}) \,. \quad (4.40\mathrm{b})$$

Remark 4.10 - Alternative formulation. An alternative formulation of (4.40) can be obtained by taking into account the pressure in the Picard iterations as well. Namely, instead of fixing $p_{\mathrm{h}}^{n-1,\infty}$ and updating the pressure only after the velocity iterations have converged, one can update the pressure at each iteration and use it in the next one. This procedure leads to: For $n = 1, \ldots, N$, iterate on $k \geq 1$ until convergence: find $\underline{u}_{\mathrm{h}}^{n,k} \in \underline{\widehat{\mathcal{U}}}_{\mathrm{h},0}$ such that

$$\frac{1}{\Delta t} m(\underline{u}_{\mathrm{C}}^{n,k} - \underline{u}_{\mathrm{C}}^{n-1,\infty}, \underline{v}_{\mathrm{C}}) + \nu \left(a_{\mathrm{h}}(\underline{\widehat{u}}_{\mathrm{h}}^{n,k}, \underline{\widehat{v}}_{\mathrm{h}}) + \eta d_{\mathrm{h}}(\underline{\widehat{u}}_{\mathrm{h}}^{n,k}, \underline{\widehat{v}}_{\mathrm{h}}) \right) + t_{\mathrm{h}}(\underline{\widehat{u}}_{\mathrm{h}}^{n,k-1}; \underline{\widehat{u}}_{\mathrm{h}}^{n,k}, \underline{\widehat{v}}_{\mathrm{h}}) = l^n(\underline{v}_{\mathrm{C}}) - b_{\mathrm{h}}(\underline{\widehat{v}}_{\mathrm{h}}, p_{\mathrm{h}}^{n,k-1}) \,, \quad (4.41\mathrm{a})$$

for all $\underline{\widehat{v}}_{\mathrm{h}} \in \underline{\widehat{\mathcal{U}}}_{\mathrm{h},0}$, and then set

$$p_{\mathrm{h}}^{n,k} = p_{\mathrm{h}}^{n,k-1} - \nu\eta \mathrm{D}_{\mathrm{h}}(\underline{\widehat{u}}_{\mathrm{h}}^{n,k}) \,. \quad (4.41\mathrm{b})$$

The previous-time-step solution can be used as initial guess for the pressure: $p^{n,0} := p^{n-1,\infty}$. Interestingly, the system in this configuration can be bridged to the monolithic approach. Indeed, consider the semi-discretized version of (4.41) where the spatial discretization has been discarded for simplicity. Now, sum the two equations to obtain

$$\begin{cases} \dfrac{\underline{u}^{n,k} - \underline{u}^{n-1,\infty}}{\Delta t} + (\underline{u}^{n,k-1} \cdot \underline{\nabla})\underline{u}^{n,k} - \nu \underline{\Delta}\, \underline{u}^{n,k} + \underline{\nabla}\, p^{n,k} = \underline{f}^n \,, \\ \dfrac{1}{\nu\eta}(p^{n,k} - p^{n,k-1}) + \underline{\nabla} \cdot \underline{u}^{n,k} = 0 \,. \end{cases} \tag{4.42}$$

Letting $(\underline{u}^{n,\infty}, p^{n,\infty})$ be the limit for $k \to \infty$ of the solution of (4.42) (provided that such a limit exists), then it solves

$$\begin{cases} \dfrac{\underline{u}^{n,\infty} - \underline{u}^{n-1,\infty}}{\Delta t} + (\underline{u}^{n,\infty} \cdot \underline{\nabla})\underline{u}^{n,\infty} - \nu \underline{\Delta}\, \underline{u}^{n,\infty} + \underline{\nabla}\, p^{n,\infty} = \underline{f}^n \,, \\ \underline{\nabla} \cdot \underline{u}^{n,\infty} = 0 \,, \end{cases} \tag{4.43}$$

where we remark that the incompressibility constraint is now exactly satisfied. ◇

Energy balance

Differently from the monolithic case, we cannot recover a clear-cut result as in Lemma 4.8 or in Lemma 4.6 regarding the energy balance. In particular, one cannot conclude on the sign of the contribution of the convection term $t_{\mathrm{h}}(\cdot;\cdot,\cdot)$ because the incompressibility constraint of the convective field is not satisfied exactly in the AC case.

4.3.2 Linearized convection

The main disadvantage of the Picard iterations is that several linear systems have to be solved at each time step. A possible way to avoid iterating while still keeping the convection in the matrix is to linearize the convection operator by using the velocity at the previous time step.

Linearized convection and monolithic approach

When the linearization is applied in addition to the monolithic approach, the CDO problem reads: For $n = 1, \dots, N$, find $(\underline{u}_{\mathrm{h}}^n, p_{\mathrm{h}}^n) \in \widehat{\underline{\mathcal{U}}}_{\mathrm{h},0} \times \mathcal{P}_{\mathrm{h},*}$ solving

$$\begin{cases} \dfrac{1}{\Delta t} m(\underline{u}_{\mathrm{C}}^n - \underline{u}_{\mathrm{C}}^{n-1}, \underline{v}_{\mathrm{C}}) + \nu a_{\mathrm{h}}(\widehat{\underline{u}}_{\mathrm{h}}^n, \widehat{\underline{v}}_{\mathrm{h}}) + t_{\mathrm{h}}(\widehat{\underline{u}}_{\mathrm{h}}^{n-1}; \widehat{\underline{u}}_{\mathrm{h}}^n, \widehat{\underline{v}}_{\mathrm{h}}) + b_{\mathrm{h}}(\widehat{\underline{v}}_{\mathrm{h}}, p_{\mathrm{h}}^n) = l^n(\underline{v}_{\mathrm{C}}) \,, \\ b_{\mathrm{h}}(\widehat{\underline{u}}_{\mathrm{h}}^n, q_{\mathrm{h}}) = 0 \,, \end{cases} \tag{4.44}$$

for all $\widehat{\underline{v}}_{\mathrm{h}} \in \widehat{\underline{\mathcal{U}}}_{\mathrm{h},0}$ and all $q_{\mathrm{h}} \in \mathcal{P}_{\mathrm{h},*}$. Solving (4.44) is clearly equivalent to stop the Picard iterations (4.37) after the first iteration if (4.38) is chosen as initialization step. Moreover, (4.44) consists in solving a single Oseen-like problem.

Energy balance

A result equivalent to Lemma 4.8 can be formulated in this case.

Lemma 4.11 - Energy balance for (4.44). *Assume* $\underline{f} := \underline{0}$. *Let* $n = 1, \dots, N$ *and let* $(\widehat{\underline{u}}_{\mathrm{h}}^n, p_{\mathrm{h}}^n)$ *solve* (4.44). *The following holds true:*

$$\mathcal{E}_{\mathrm{kin,h}}(\widehat{\underline{u}}_{\mathrm{h}}^n) - \mathcal{E}_{\mathrm{kin,h}}(\widehat{\underline{u}}_{\mathrm{h}}^{n-1}) + \underbrace{\mathcal{E}_{\mathrm{kin,h}}(\widehat{\underline{u}}_{\mathrm{h}}^n - \widehat{\underline{u}}_{\mathrm{h}}^{n-1})}_{\geq 0} + \underbrace{\nu \Delta t \left\| \underline{\underline{\mathrm{G}}}_{\mathrm{h}}(\widehat{\underline{u}}_{\mathrm{h}}^n) \right\|^2_{\underline{\underline{L}}^2(\Omega)}}_{\geq 0} + \underbrace{t_{\mathrm{h}}(\widehat{\underline{u}}_{\mathrm{h}}^{n-1}; \widehat{\underline{u}}_{\mathrm{h}}^n, \widehat{\underline{u}}_{\mathrm{h}}^n)}_{\geq 0} = 0 \,. \tag{4.45}$$

Proof. One observes that $\widehat{\underline{u}}_{\mathrm{h}}^{n-1}$ is discretely divergence-free. Hence, one proceeds as in the proof of Lemma 4.8 and (4.45) is readily proved. □

Remark 4.12 - Skew-symmetry. The arguments made in Remark 4.9 can be adapted to the convective field $\widehat{\underline{u}}_{\mathrm{h}}^{n-1}$. One can thus conclude that the contribution of the convection term in (4.45) vanishes whenever no upwind stabilization is considered. ◇

Linearized convection and Artificial Compressibility

The problem obtained when the AC technique and a linearized convection term are used reads: For $n = 1, \ldots, N$, find $(\underline{u}_{\mathrm{h}}^n, p_{\mathrm{h}}^n) \in \widehat{\underline{\mathcal{U}}}_{\mathrm{h},0} \times \mathcal{P}_{\mathrm{h},*}$ such that

$$\frac{1}{\Delta t} m(\underline{u}_{\mathrm{C}}^n - \underline{u}_{\mathrm{C}}^{n-1}, \underline{v}_{\mathrm{C}}) + \nu \left(a_{\mathrm{h}}(\widehat{\underline{u}}_{\mathrm{h}}^n, \widehat{\underline{v}}_{\mathrm{h}}^n) + \eta d_{\mathrm{h}}(\widehat{\underline{u}}_{\mathrm{h}}^n, \widehat{\underline{v}}_{\mathrm{h}})\right) + t_{\mathrm{h}}(\widehat{\underline{u}}_{\mathrm{h}}^{n-1}; \widehat{\underline{u}}_{\mathrm{h}}^n, \widehat{\underline{v}}_{\mathrm{h}}) = l^n(\underline{v}_{\mathrm{C}}) - b_{\mathrm{h}}(\widehat{\underline{v}}_{\mathrm{h}}, p_{\mathrm{h}}^{n-1}), \quad (4.46a)$$

for all $\widehat{\underline{v}}_{\mathrm{h}} \in \widehat{\underline{\mathcal{U}}}_{\mathrm{h},0}$, and then set

$$p_{\mathrm{h}}^n = p_{\mathrm{h}}^{n-1} - \nu\eta \mathrm{D}_{\mathrm{h}}(\widehat{\underline{u}}_{\mathrm{h}}^n). \quad (4.46b)$$

Energy balance

The arguments given in Section 4.3.1 on the AC case with Picard iterations apply to the linearized convection, i.e. no definite conclusion can be drawn about the energy balance associated with (4.46).

4.3.3 Explicit convection

With the linearization of the convection field, one has reduced the number of linear systems to solve per time step to just one. A further gain in efficiency results from using a fully explicit convection term. Indeed, this yields a symmetric linear system, so that more efficient (iterative) solvers can be used than in the nonsymmetric case.

Explicit convection and monolithic approach

The CDO-Fb problem obtained after considering an explicit convection and a monolithic approach is as follows: For $n = 1, \ldots, N$, find $(\underline{u}_{\mathrm{h}}^n, p_{\mathrm{h}}^n) \in \widehat{\underline{\mathcal{U}}}_{\mathrm{h},0} \times \mathcal{P}_{\mathrm{h},*}$ solving

$$\begin{cases} \frac{1}{\Delta t} m(\underline{u}_{\mathrm{C}}^n - \underline{u}_{\mathrm{C}}^{n-1}, \underline{v}_{\mathrm{C}}) + \nu a_{\mathrm{h}}(\widehat{\underline{u}}_{\mathrm{h}}^n, \widehat{\underline{v}}_{\mathrm{h}}) + b_{\mathrm{h}}(\widehat{\underline{v}}_{\mathrm{h}}, p_{\mathrm{h}}^n) = l^n(\underline{v}_{\mathrm{C}}) - t_{\mathrm{h}}(\widehat{\underline{u}}_{\mathrm{h}}^{n-1}; \widehat{\underline{u}}_{\mathrm{h}}^{n-1}, \widehat{\underline{v}}_{\mathrm{h}}), \\ b_{\mathrm{h}}(\widehat{\underline{u}}_{\mathrm{h}}^n, q_{\mathrm{h}}) = 0, \end{cases} \quad (4.47)$$

for all $\widehat{\underline{v}}_{\mathrm{h}} \in \widehat{\underline{\mathcal{U}}}_{\mathrm{h},0}$ and all $q_{\mathrm{h}} \in \mathcal{P}_{\mathrm{h},*}$. The convection term has been moved to the right-hand side since it is a known quantity. Notice that solving (4.47) is equivalent to solving one time step of an unsteady Stokes problem.

Energy balance

When investigating the kinetic energy and testing (4.47) with $\widehat{\underline{v}}_{\mathrm{h}} = \widehat{\underline{u}}_{\mathrm{h}}^n$, one is confronted with the term $t_{\mathrm{h}}(\widehat{\underline{u}}_{\mathrm{h}}^{n-1}; \widehat{\underline{u}}_{\mathrm{h}}^{n-1}, \widehat{\underline{u}}_{\mathrm{h}}^n)$. The term related to the discrete divergence of $\widehat{\underline{u}}_{\mathrm{h}}^{n-1}$ (see the derivation of (2.100)) vanishes since we are considering a strong velocity-pressure coupling.

However, no definite sign can be recovered for $t_{\mathrm{h}}(\widehat{\underline{u}}_{\mathrm{h}}^{n-1};\widehat{\underline{u}}_{\mathrm{h}}^{n-1},\widehat{\underline{u}}_{\mathrm{h}}^{n})$. Notice that even the upwiding term is not dissipative since

$$t_{\mathrm{h}}^{\mathrm{u}}(\widehat{\underline{u}}_{\mathrm{h}}^{n-1};\widehat{\underline{u}}_{\mathrm{h}}^{n-1},\widehat{\underline{u}}_{\mathrm{h}}^{n}) = \frac{1}{2}\sum_{\mathrm{c}\in\mathrm{C}}\sum_{\mathrm{f}\in\mathrm{F_c}\cap\mathrm{F}^i} |\mathrm{f}| \left|\widehat{\underline{u}}_{\mathrm{h}}^{n-1}\cdot\underline{n}_{\mathrm{fc}}\right| (\widehat{\underline{u}}_{\mathrm{f}}^{n-1}-\widehat{\underline{u}}_{\mathrm{c}}^{n-1})\cdot(\widehat{\underline{u}}_{\mathrm{f}}^{n}-\widehat{\underline{u}}_{\mathrm{c}}^{n})\,. \tag{4.48}$$

Explicit convection and Artificial Compressibility

Similarly, considering the AC technique and an explicit convection term leads to the following procedure: For $n=1,\dots,N$, find $(\underline{u}_{\mathrm{h}}^{n},p_{\mathrm{h}}^{n})\in\widehat{\underline{\mathcal{U}}}_{\mathrm{h},0}\times\mathcal{P}_{\mathrm{h},*}$ such that

$$\begin{aligned}\frac{1}{\Delta t}m(\underline{u}_{\mathrm{C}}^{n}-\underline{u}_{\mathrm{C}}^{n-1},\underline{v}_{\mathrm{C}})+\nu\left(a_{\mathrm{h}}(\widehat{\underline{u}}_{\mathrm{h}}^{n},\widehat{\underline{v}}_{\mathrm{h}})+\nu\eta d_{\mathrm{h}}(\widehat{\underline{u}}_{\mathrm{h}}^{n},\widehat{\underline{v}}_{\mathrm{h}})\right)&\\ = l^{n}(\underline{v}_{\mathrm{C}})-b_{\mathrm{h}}(\widehat{\underline{v}}_{\mathrm{h}},p_{\mathrm{h}}^{n-1})-t_{\mathrm{h}}(\widehat{\underline{u}}_{\mathrm{h}}^{n-1};\widehat{\underline{u}}_{\mathrm{h}}^{n-1},\widehat{\underline{v}}_{\mathrm{h}})\,,&\end{aligned} \tag{4.49a}$$

for all $\widehat{\underline{v}}_{\mathrm{h}}\in\widehat{\underline{\mathcal{U}}}_{\mathrm{h},0}$, and then set

$$p_{\mathrm{h}}^{n}=p_{\mathrm{h}}^{n-1}-\nu\eta\mathrm{D}_{\mathrm{h}}(\widehat{\underline{u}}_{\mathrm{h}}^{n})\,. \tag{4.49b}$$

Among the three strategies which have been presented above, this latter one appears to be the most coherent one with the AC principles and indeed it is the one used in Guermond and Minev (2015).

Energy balance

As it was the case for the other convection treatments with the AC method, the dissipative nature of the energy balance resulting from (4.49) cannot be determined a priori.

4.4 Numerical results: Stokes equations

We start by considering the unsteady Stokes problem. The focus is on the time-stepping procedure and the coupling (hence avoiding the influence of the nonlinearity due to the convection operator of the NSE). The first goal of these tests is to verify that the proposed time-stepping algorithms recover the expected order of convergence. The second goal is to compare the accuracy and the efficiency of the AC approach described in Section 4.2.2 to the classical monolithic approach described in Section 4.2.1.

We choose again the 3D Taylor–Green Vortex solution (3.51) and we add a time dependency by modulating its amplitude:

$$\begin{cases}\underline{u}_{\mathrm{UTGV}}(x,y,z):=\alpha(t)\underline{u}_{\mathrm{3TGV}}(x,y,z)\,,\\ p_{\mathrm{UTGV}}(x,y,z):=\alpha(t)p_{\mathrm{3TGV}}(x,y,z)\,,\\ \alpha(t):=\sin(1.7\,\pi t+\frac{\pi}{5})\,,\\ \underline{u}_{\mathrm{3TGV}}(x,y,z):=\begin{bmatrix}-2\cos(2\pi x)\sin(2\pi y)\sin(2\pi z)\\ \sin(2\pi x)\cos(2\pi y)\sin(2\pi z)\\ \sin(2\pi x)\sin(2\pi y)\cos(2\pi z)\end{bmatrix}\,,\\ p_{\mathrm{3TGV}}(x,y,z):=-6\pi\sin(2\pi x)\sin(2\pi y)\sin(2\pi z)\,.\end{cases} \tag{4.50}$$

The viscosity is set to $\nu:=1$, whereas the source term is adapted as follows:

$$\begin{cases}\underline{f}_{\mathrm{UTGV}}(x,y,z):=\underline{f}_{\mathrm{3TGV}}(x,y,z)+1.7\,\pi\cos(1.7\,\pi t+\frac{\pi}{5})\underline{u}_{\mathrm{3TGV}}(x,y,z)\,,\\ \underline{f}_{\mathrm{3TGV}}(x,y,z):=[-36\pi^2\cos(2\pi x)\sin(2\pi y)\sin(2\pi z),\ 0,\ 0]^T\,.\end{cases} \tag{4.51}$$

We fix a time limit $T := 2$ and run the computations for several values of the time step: $\Delta t = \frac{1}{2}, \frac{1}{4}, \ldots, \frac{1}{128}$.

Since we are analyzing the time errors, we use very fine spatial meshes, so that the spatial error should be negligible. In particular, a Cartesian mesh of the unit cube (cf. Fig. 3.4a) with each edge of the domain divided into 256 segments has been considered. This leads to more than 16M cells and a size of the final linear system in the saddle-point problem of more than 151M unknowns. The significant size of such a mesh precluded the usage of a direct solver and also demanded more powerful resources. The computations have thus been run on EDF cluster GAIA[1] on up to 525 cores. Even if a hybrid parallelization based on `MPI+OMP` would have been possible, we chose to use all the cores at our disposal on the `MPI` side. Such costly computations will also come in handy when comparing, for instance, two linear solvers or couplings: the differences in their performances will be magnified and hence clearly visible.

We shall work first with Cartesian meshes in Sections 4.4.1 and 4.4.2 in order to explore different combinations of preconditioners and solvers, and finally the most promising ones will be tested on polyhedral meshes in Section 4.4.3.

Let us give some details on the linear solvers. When considering the monolithic approach (MONO), the GKB(γ) procedure is used. As highlighted in Section 1.3.5, this algorithm shares some features with the ALU method, in particular it needs an arbitrary augmentation parameter, γ. Since this parameter may affect the conditioning of the resulting linear systems, it is something whose effect we are interested in measuring. A similar remark applies to the AC technique and its arbitrary parameter η. Moreover, since we are addressing the Stokes problem, and since the systems are decoupled (by means of the GKB(γ) or AC(η) procedures), they are symmetric, so that we can use a Conjugate Gradient algorithm as linear solver. We may choose between two preconditioners, a Jacobi preconditioner or an in-house K-cycle Algebraic Multi-Grid (AMG) preconditioner inspired by the work of Notay. In the monolithic computations, the threshold for the GKB algorithm is 10^{-10} on the absolute residual of the momentum and mass equations, whereas the threshold of the internal CG iterative solver is 10^{-5} on the relative norm of the velocity increment. The choice for these tolerances is discussed in Remark 4.13.

Since we are now dealing with unsteady problems, discrete time errors should be introduced. Consider a space-time function

$$g(t, \underline{x}) \in \mathcal{C}^0 \left([0, T]; [L^2(\Omega)]^l \right) , \tag{4.52}$$

and a series of distinct time nodes $(t^n)_{n=0,\ldots,N}$. The dimension l is left generic, so that it may identify scalar- or vector-valued functions. With an abuse of notation, dropping the explicit time dependency, we denote by $g_{\mathrm{h}} := (g_{\mathrm{h}}^n)_{n=0,\ldots,N} \in [\mathbb{R}^l]^N$ the fully discrete approximation of $g(t, \underline{x})$, where g_{h}^n approximates $g(t^n, \cdot)$. Owing to the notation introduced in Section 4.1.2 and relying on the discrete norms defined in Section 3.3.2, one can compute a space-time L^2-like error as follows:

$$\|g_{\mathrm{h}}\|_{\ell^2,\mathrm{C}}^2 := \sum_{n=1}^{N} \Delta t \, \|g_{\mathrm{h}}^n\|_{\mathrm{C}}^2 = \sum_{n=1}^{N} \Delta t \left(\sum_{\mathrm{c} \in \mathrm{C}} |\mathrm{c}| \, |g_{\mathrm{c}}^n|_2^2 \right) , \tag{4.53}$$

where we used the definition of the discrete L^2-like spatial norm, see (3.43). For a hybrid variable, a similar definition is used by relying, this time, on (3.44):

$$\|\widehat{g}_{\mathrm{h}}\|_{\ell^2,\mathrm{C}}^2 := \sum_{n=1}^{N} \Delta t \, \|\widehat{g}_{\mathrm{h}}^n\|_{\mathrm{C}}^2 = \sum_{n=1}^{N} \Delta t \left(\sum_{\mathrm{c} \in \mathrm{C}} |\mathrm{c}| \, |g_{\mathrm{c}}^n|_2^2 \right) . \tag{4.54}$$

[1] 243rd of the TOP500 list, November 2020

Furthermore, we define the pressure error

$$\mathrm{E_h}(p) := \left((\pi_{\mathrm{c}}(p(t^n,\cdot)) - p_{\mathrm{c}}^n)_{\mathrm{c}\in\mathrm{C}}\right)_{n=1,\dots,N} , \tag{4.55}$$

(we dropped the dependency on time in the error definition for simplicity) and the L^2-like velocity error

$$\underline{\widehat{\mathrm{E}}}_{\mathrm{h}}(\underline{u}) := \left((\underline{\widehat{\pi}}_{\mathrm{c}}(\underline{u}(t^n,\cdot)) - \underline{\widehat{u}}_{\mathrm{c}}^n)_{\mathrm{c}\in\mathrm{C}}\right)_{n=1,\dots,N} . \tag{4.56}$$

The norms of interest will be:

$$\left\|\underline{\widehat{\mathrm{E}}}_{\mathrm{h}}(\underline{u})\right\|_{\ell^2,\mathrm{C}} , \qquad \|\mathrm{E_h}(p)\|_{\ell^2,\mathrm{C}} , \qquad \text{and} \qquad \left\|\underline{\underline{\mathrm{G}}}_{\mathrm{h}}(\underline{\widehat{\mathrm{E}}}_{\mathrm{h}}(\underline{u}))\right\|_{\ell^2,\mathrm{C}} , \tag{4.57}$$

where $\left\|\underline{\underline{\mathrm{G}}}_{\mathrm{h}}(\underline{\widehat{\mathrm{E}}}_{\mathrm{h}}(\underline{u}))\right\|_{\ell^2,\mathrm{C}}$ is based on the discrete H^1-like (semi)norm of the velocity (see (3.48))

$$\left\|\underline{\underline{\mathrm{G}}}_{\mathrm{h}}(\underline{\widehat{\mathrm{E}}}_{\mathrm{h}}(\underline{u}))\right\|_{\ell^2,\mathrm{C}}^2 := \sum_{n=1}^{N} \Delta t \left\|\underline{\underline{\mathrm{G}}}_{\mathrm{h}}(\underline{\widehat{\mathrm{E}}}_{\mathrm{h}}^n(\underline{u}))\right\|_{\mathrm{C}}^2 = \sum_{n=1}^{N} \Delta t \sum_{\mathrm{c}\in\mathrm{C}} \sum_{\mathrm{f}\in\mathrm{F}} |\mathfrak{p}_{\mathrm{f,c}}| \left|\underline{\underline{\mathrm{G}}}_{\mathrm{c}}(\underline{\widehat{\mathrm{E}}}_{\mathrm{c}}^n(\underline{u}))_{|\mathfrak{p}_{\mathrm{f,c}}}\right|_2^2 . \tag{4.58}$$

Remark 4.13 - GKB procedure and linear solver. As in the steady case (see also Remark 3.12), preliminary tests concerning the configuration of the GKB procedure have been run in order to ensure that the numerical setting was adapted to the problem. We used $\Delta t = \frac{T}{32} = 6.25e{-}2$. Firstly, we fix the tolerance for the stopping criterion of the GKB procedure, $\epsilon_{\mathrm{GKB}} := 10^{-10}$, and let the tolerance of the CG solver, ϵ_{CG} vary. Then, ϵ_{GKB} varies while we set $\epsilon_{\mathrm{CG}} := 10^{-5}$. The results are shown in Table 4.1. No significant difference is observed between the considered tolerances for CG. This confirms that the tolerances requested for the linear solvers do not impact the discretization errors (temporal and spatial errors are combined in this case) and corroborates our choices. Notice that less stringent tolerances could be considered to save computational time, but we preferred to stay "on the safe side". ◇

Table 4.1 – Space-time errors for the velocity and the pressure on a Cartesian mesh composed of 256^3 cells and $\Delta t = \frac{T}{32} = 6.25\,10^{-2}$. The linear systems resulting from the monolithic approach are solved using a GKB(0) procedure, a CG iterative solver and an AMG preconditioner. The tolerances of the iterative procedures (ϵ_{GKB} and ϵ_{CG}) vary. In the first column, ϵ_{GKB} is fixed and ϵ_{CG} varies, and vice versa in the second column.

	$\epsilon_{GKB} := 10^{-10}$, ϵ_{CG} varies			ϵ_{GKB} varies, $\epsilon_{\mathrm{CG}} := 10^{-5}$		
	10^{-4}	10^{-5}	10^{-6}	10^{-8}	10^{-10}	10^{-12}
$\left\|\underline{\widehat{\mathrm{E}}}_{\mathrm{h}}(\underline{u})\right\|_{\ell^2,\mathrm{C}}$	$1.77e{-}3$	$1.77e{-}3$	$1.77e{-}3$	$1.77e{-}3$	$1.77e{-}3$	$1.77e{-}3$
$\left\|\underline{\underline{\mathrm{G}}}_{\mathrm{h}}(\underline{\widehat{\mathrm{E}}}_{\mathrm{h}}(\underline{u}))\right\|_{\ell^2,\mathrm{C}}$	$1.43e{-}2$	$1.43e{-}2$	$1.43e{-}2$	$1.43e{-}2$	$1.43e{-}2$	$1.43e{-}2$
$\|\mathrm{E_h}(p)\|_{\ell^2,\mathrm{C}}$	$1.20e{-}3$	$1.20e{-}3$	$1.20e{-}3$	$1.20e{-}3$	$1.20e{-}3$	$1.20e{-}3$

4.4.1 Convergence in time

Let us start by checking the accuracy in time of the considered schemes. Three values are tested for the parameter η in the AC method: $\eta \in \{1, 10, 100\}$. The results with AC(η) method should approach those of the monolithic approach as η gets larger.

The results for the space-time velocity and pressure errors are presented in Fig. 4.1 (complete details for selected strategies are given in Tables 4.5 to 4.7). Here we focus on

the L^2-like norms for the velocity and the pressure, while observing that the results for the $\ell^2(H^1)$ norm of the velocity error are similar. First-order convergence in time is observed as expected. Slightly lower rates are measured for the coarsest time steps with AC(1). Since very few time steps are considered in this situation (as few as 4), some stagnation can be expected. However, the expected slope is recovered for finer values of Δt. The results also lead to interesting observations about the influence of the parameter η on the accuracy of AC(η). As expected, higher values of η give more accurate solutions. The effect seems to be stronger for the pressure: for instance, the difference between $\eta = 10$, —o—, and $\eta = 100$, —o—, is greater for the pressure than for the velocity. This is not surprising since the parameter η impacts directly the incompressibility constraint, hence the pressure. Moreover, as expected for values of $\eta \to \infty$, the monolithic approach is recovered, and indeed the plot for AC(100), —o—, is superimposed with the monolithic one, —■—.

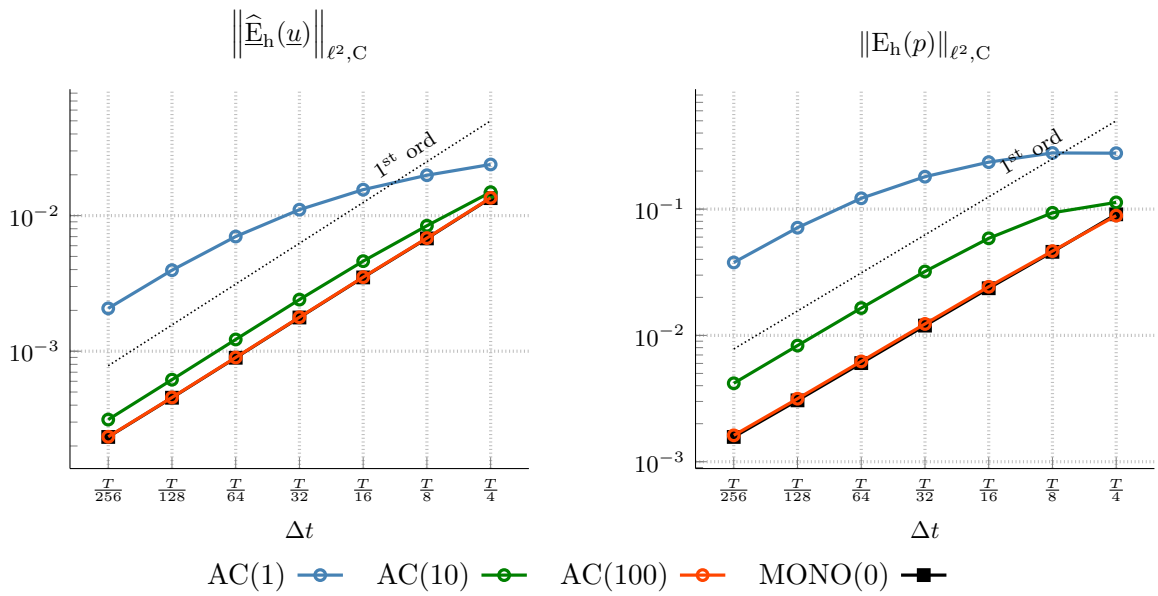


Figure 4.1 – Unsteady Stokes problem. 3D TGV solution (4.50), $T = 2$. Cartesian mesh with 256^3 cells. Convergence in time. Left: velocity L^2-error, right: pressure L^2-error.

4.4.2 Efficiency results

We are now going to compare the efficiency of the monolithic approach and the AC method on the basis of accuracy vs. computational times. As detailed above, several combinations of linear solver and preconditioners have been used, and the computations have always been run on dedicated nodes of the EDF GAIA cluster in order to ensure the fairness of the results.

Figure 4.2 shows the results of these efficiency tests. Since many combinations have been run, we decrease the opacity of some curves in Fig. 4.2 in order to make the best results stand out. We proceed to some remarks. (i) Concerning the monolithic approach, the best result is obtained with $\gamma = 0$ (which, we recall, means that no augmentation is considered) and the AMG preconditioner. Whenever $\gamma > 0$, the performance of the AMG preconditioner degrades significantly, to an extent that the computations were not able to terminate in a reasonable amount of time. Consequently, only computations with $\gamma > 0$ and a Jacobi preconditioner are presented in Fig. 4.2. The reason of this gap in the performances is likely to be found in how the grad-div operator adds some coupling between the faces and, more importantly, the Cartesian components of the velocity. An ad-hoc strategy for

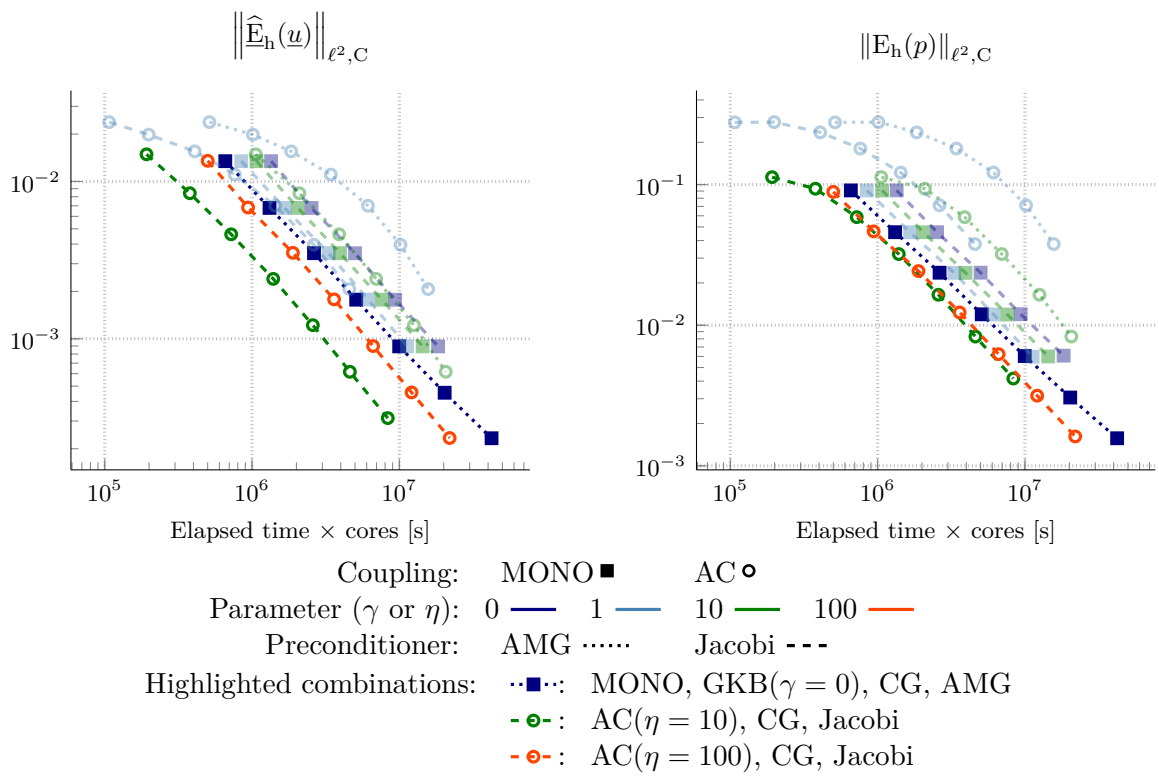
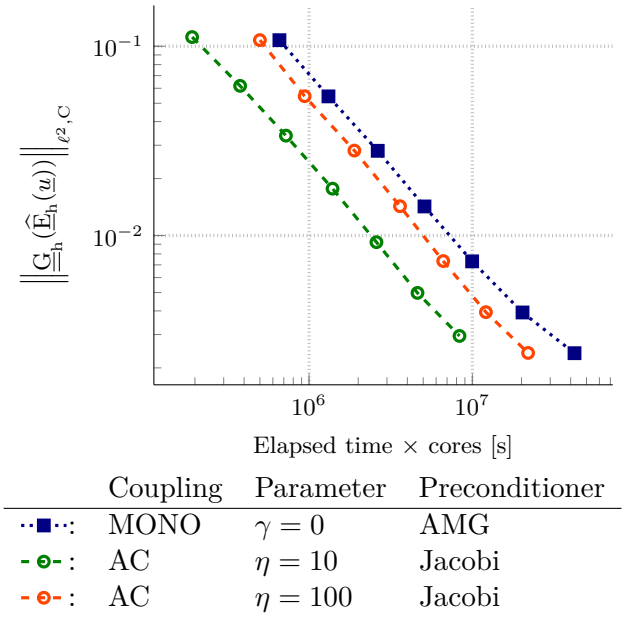


Figure 4.2 – Unsteady Stokes problem. 3D TGV solution (4.50), $T = 2$. Cartesian mesh with 256^3 cells. Cost vs. accuracy for velocity (left) and pressure (right). The most efficient combinations have more intense colors.

the construction of the coarser levels in the multigrid algorithm should then be considered so that the AMG preconditioner (which, in the current configuration, is optimized for elliptic problems) remains meaningful and efficient. (ii) Similarly, the best performances for the AC method have been observed with the Jacobi preconditioner. Moreover, as expected, a high value for η affects the conditioning of the linear system and, consequently, the performance of the iterative solvers. Compare for instance the computation times of AC(10), -⊙-, and AC(100), -⊙-. Considering the remarks in Section 4.4.1 on high values of η leading to more accurate computations, the right balance between CPU time and error level should be investigated. (iii) Considering all the results, we observe that the two leftmost (thus the best) curves are both obtained with the AC method, in particular, AC(10), -⊙-, and AC(100), -⊙-. Hence, given a target error threshold, the AC method reaches it in less computational time than the monolithic approach.

For the sake of completeness, we report in Fig. 4.3 the efficiency results with respect to the error measure $\left\|\underline{\underline{G}}_h(\underline{\widehat{E}}_h(\underline{u}))\right\|_{\ell^2,C}$ for the best three strategies. The conclusion is the same as above with the AC method being more efficient than the monolithic approach. We notice here a hint of stagnation at the end of the curves, most likely due to the spatial discretization becoming relevant for the smallest time-step values.

To summarize, the AC(η) method appears to be an efficient alternative to the monolithic approach for the unsteady Stokes problem. However, caution has to be used in the choice of the parameter η in order to recover both acceptable computation times and accurate results. The observation here is that to reach a given error threshold, AC(10) requests less computational effort than AC(100). Notice, however, that this threshold is reached with a smaller Δt for AC(10) than for AC(100). For instance, in the pressure results in Fig. 4.1,



	Coupling	Parameter	Preconditioner
··■··:	MONO	$\gamma = 0$	AMG
- ⊙- :	AC	$\eta = 10$	Jacobi
- ⊙- :	AC	$\eta = 100$	Jacobi

Figure 4.3 – Unsteady Stokes problem. 3D TGV solution (4.50), $T = 2$. Cartesian mesh with 256^3 cells. Cost vs. $\ell^2(H^1)$-accuracy.

the points of AC(10), —⊙—, are almost at the same level as those of AC(100), —⊙—, with a time step two times larger, but the former points are obtained with less computational time. Finally, having access to a preconditioner adapted to the grad-div operator may result in a significant efficiency improvement.

We would like to point out that these results, although providing a general idea about the behavior of the two considered coupling strategies, may vary if a different numerical setting is considered. For one, a tuning of the iterative linear solvers (GKB(γ) and CG) and, in particular, of the related tolerances (see Remark 4.13) may lead to different results, especially when considering the execution time. For example, we compare in Fig. 4.4 the best results of Fig. 4.2 (semi-transparent) with the results obtained with a GKB(0) procedure where the tolerance is less strict, $\epsilon_{\text{GKB}} := 10^{-8}$ (instead of 10^{-10} as above). We can see that although there is some moderate gain in the computation times, it is not enough to make the monolithic approach more efficient than the AC method. Moreover, regarding possible improvements, the computations may benefit from a preconditioner adapted to the grad-div operator: this should benefit both the AC method and the monolithic approach, for which no parameter $\gamma > 0$ was retained in the GKB procedure since the Jacobi and AMG preconditioners did not perform well. Nevertheless, we think that the efficiency results shown in Fig. 4.2 provide a reliable comparison of the performances of the monolithic approach and of the AC method, and we are confident that we would draw the same general conclusions if another (fair) numerical setting were to be used.

4.4.3 Polyhedral meshes

Having identified the best combinations of coupling, solver and preconditioner in the previous section, we are going to test them on polyhedral meshes, namely the Prismatic meshes with polygonal bases (PrG, Fig. 3.4f) and the CheckerBoard meshes (CB, Fig. 3.4b). The PrG mesh chosen for this test case has more than 4M cells, leading to a final coupled system of more than 56M unknowns (after static condensation); the CB mesh has more than 31M cells and a total system size of more than 414M unknowns. The meshes have been created in *Code_Saturne* by copying and gluing the most refined grid of each sequence used

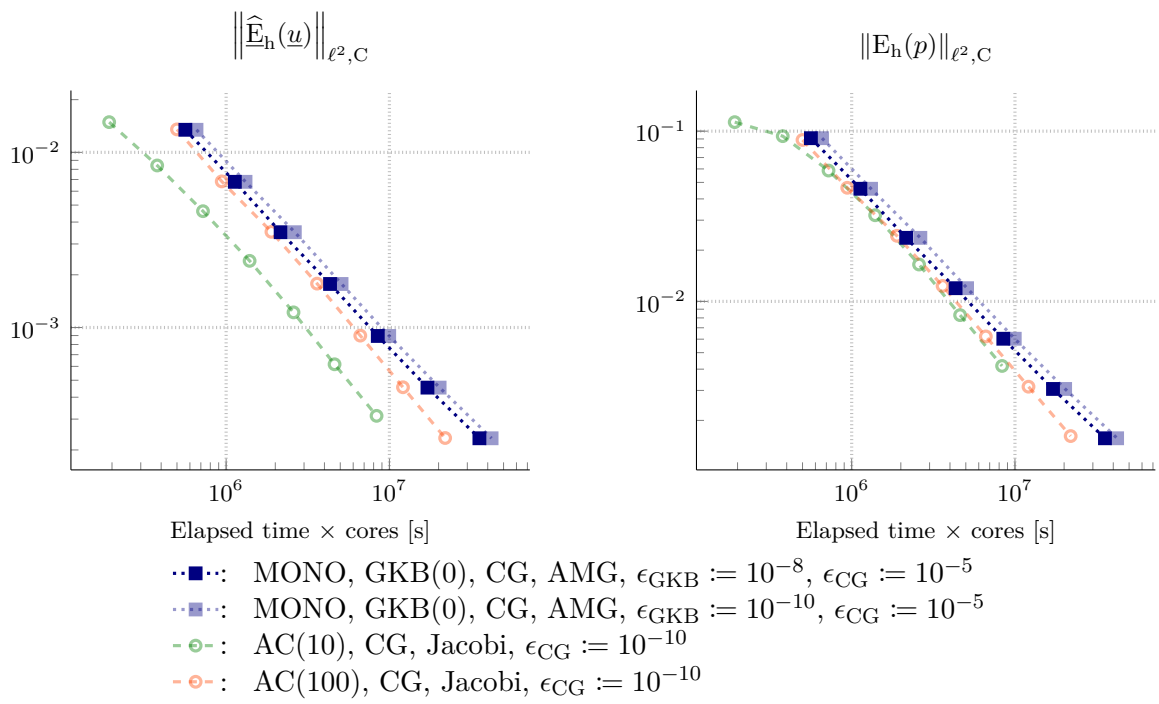


Figure 4.4 – Unsteady Stokes problem. 3D TGV solution (4.50), $T = 2$. Cartesian mesh with 256^3 cells. Cost vs. accuracy for velocity (left) and pressure (right).

so far. Only the best three combinations are tested for each mesh: a monolithic approach solved by GKB($\gamma = 0$) with a CG internal solver and AMG preconditioner, together with AC($\eta = 10$) and AC($\eta = 100$) solved with a Jacobi-preconditioned CG. The results are shown in Figs. 4.5 and 4.6, where the results obtained for the Cartesian results are redrawn as reference (denoted by H and ■). More detailed results for these two polyhedral meshes are given in Tables 4.8 to 4.13 which are postponed to Section 4.6.1.

Generally speaking, the expected first-order convergence rates are recovered for each method and mesh. However, one can notice the effect of the space discretization error which becomes dominant for low values of the time step. Especially when using CB meshes, a plateau is visible for the pressure errors. On the contrary, the velocity errors seem to be less affected. This stagnation, however, provides us with some interesting insights. In the right panel of Fig. 4.5, the stagnation for AC(10), -◆-, begins for lower values of Δt than for the other two considered procedures. If one reasonably assumes that there is no coupling error in the monolithic case, ··◆··, so that this case could be taken as reference, then the difference in the levels of the plateaus can be identified as the coupling error due to the AC technique with $\eta = 10$. Moreover, given a strategy, we observe no significant difference between the errors due to the different meshes, whereas one would expect the Cartesian meshes to be more accurate, than, for instance the PrG one, since the shape is more regular and the mesh is more refined. However, this confirms that these errors are essentially due to the time discretization.

The cost vs. accuracy results on the PrG mesh sequence confirm what has been observed in the previous section about the Cartesian grid: the AC technique with the two considered values of the parameter η are the most efficient ones. The behavior on the PrG and Cartesian meshes are actually very similar. On the CB meshes, one observes different results. Even though the AC(10) method still remains the most efficient approach, especially when looking at the velocity results, the gap between the other two considered techniques is less evident. It is worth mentioning that, in the case of a regular solution, the precision of the CB is

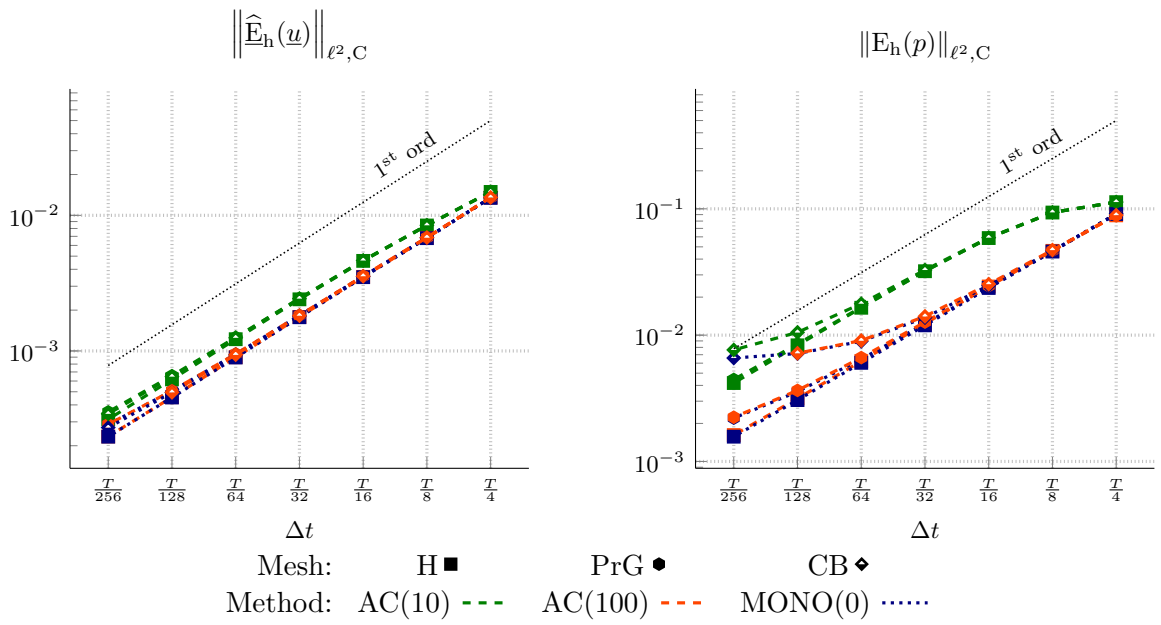


Figure 4.5 – Unsteady Stokes problem and 3D TGV solution (4.50), $T = 2$. Polyhedral meshes. Convergence in time. Error plots for velocity (left) and pressure (right).

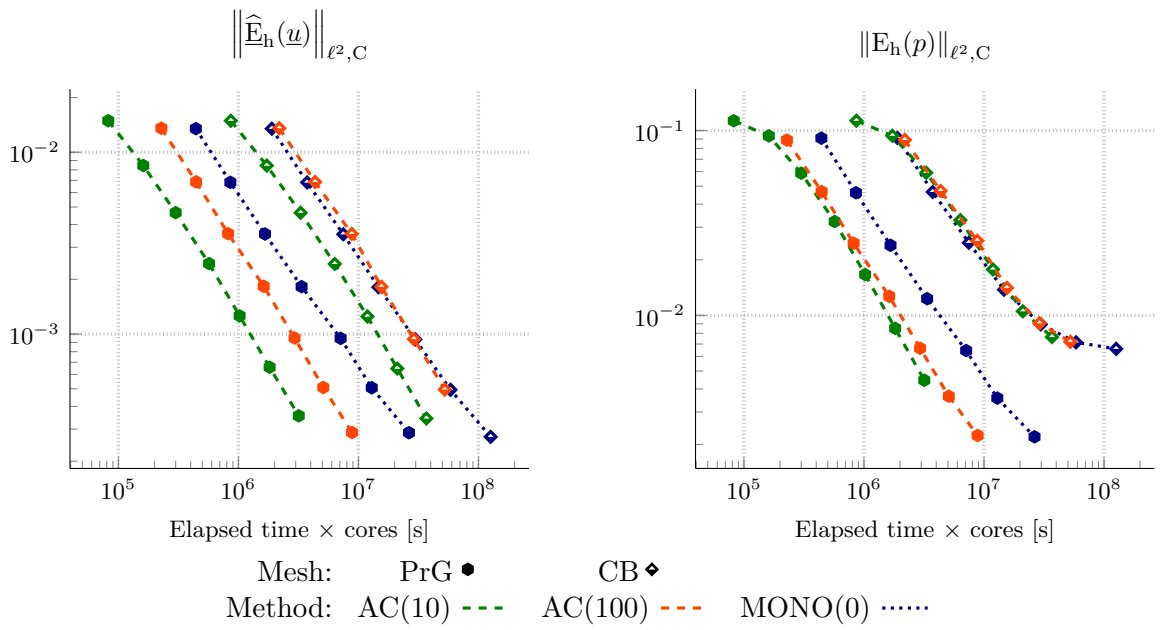


Figure 4.6 – Unsteady Stokes problem and 3D TGV solution (4.50), $T = 2$. Polyhedral meshes. Cost vs. accuracy for velocity (left) and pressure (right).

governed by the size of the larger cuboids (see Fig. 3.4b), thus obtaining the same accuracy level of a regular mesh with much fewer DoFs. The CB is hence a rather inefficient choice for this test case, but it allows us to test the discretization on meshes with hanging-nodes.

This analysis shows once again the importance of the arbitrary parameter η in the AC method. If the system is relatively easy to invert, as was the case on the Cartesian meshes, one might choose high values for η knowing that the gain in the accuracy should compensate the loss in the performance. However, when the system is more ill-conditioned due to the mesh geometry, as in the CB meshes for instance, lower values of η should be preferred in order to avoid degrading excessively the performance of the solver while still ensuring acceptable accuracy levels.

4.5 Numerical results: Navier–Stokes equations

The Taylor–Green Vortex (TGV) (Taylor and Green, 1937) is a well-known 2D test case usually considered to evaluate the performances of a Navier–Stokes unsteady solver. In fact, it constitutes an analytic solution of the 2D NSE given by

$$\begin{cases} \underline{u}_{\mathrm{TGV}}(x,y) := \exp(-2\nu t) \begin{bmatrix} \sin(x)\cos(y) \\ -\cos(x)\sin(y) \end{bmatrix}, \\ p_{\mathrm{TGV}}(x,y) := \dfrac{1}{4}\exp(-4\nu t)(\cos(2x)+\cos(2y)). \end{cases} \tag{4.59}$$

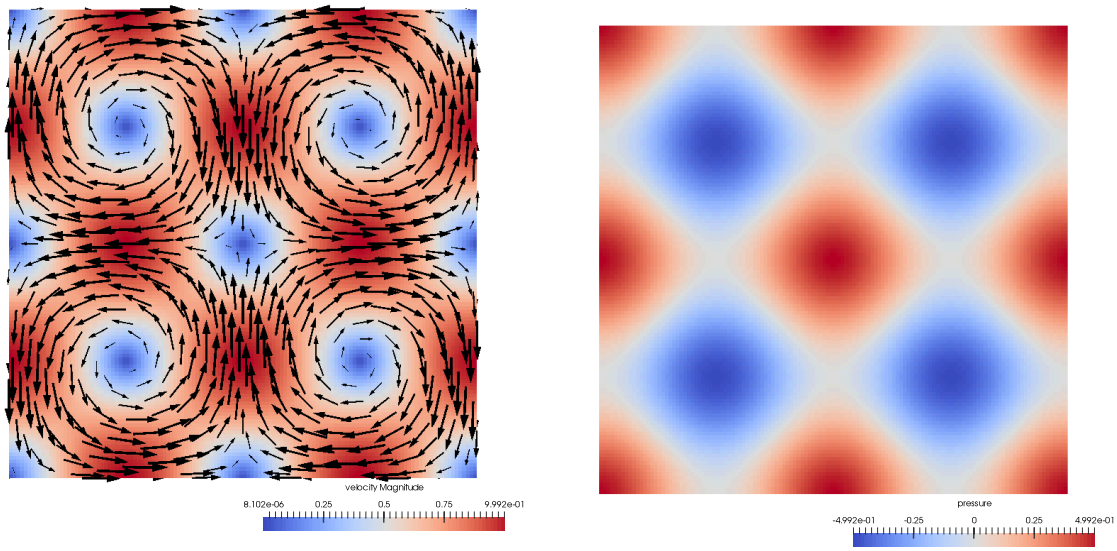
The remarkable property of the TGV test case is that the viscous term balances the time derivative, whereas the convection term balances the pressure gradient, hence avoiding the need for a source term:

$$\frac{\partial \underline{u}_{\mathrm{TGV}}}{\partial t} = \nu \underline{\Delta}\, \underline{u}_{\mathrm{TGV}}, \quad (\underline{u}_{\mathrm{TGV}} \cdot \underline{\nabla})\underline{u}_{\mathrm{TGV}} = -\underline{\nabla} p_{\mathrm{TGV}}, \quad \underline{f}_{\mathrm{TGV}} = \underline{0}. \tag{4.60}$$

The domain is $\Omega := [0, 2\pi]^2$, and the viscosity ν is varied to consider different Reynolds numbers. For the sake of simplicity, we set $L := 1$ and $U := 1$, so that $Re = \frac{1}{\nu}$. Notice that the exact solution features a separation of variables between space and time, and only the time behavior depends on the viscosity (and hence the Reynolds number). The higher the viscosity, the steeper the time decrease. The time limit T varies according to the Reynolds number in order to let the solution "evolve sufficiently". The reference solution at $t = 0$ is depicted in Fig. 4.7. We will mainly deal with fairly refined Cartesian meshes, the ones already considered for the lid-driven cavity test in Section 3.5.3: the 2D setting allows us to take advantage of direct solvers, in particular the sparse LU decomposition from MUMPS.

We pursue several aims with this test case. The first one is to evaluate the orders of convergence in time for all the combinations of coupling techniques and convection treatments presented in Sections 4.2 and 4.3, respectively. Secondly, we want to assess the conservation properties of the various schemes regarding the kinetic energy. Finally, having considered an explicit treatment of the convection, a numerical study is performed in order to find the stability limit on the time step for such a strategy and how it depends on the Reynolds number.

Contrary to the previous test case, no efficiency study will be performed. In fact, the extensive use of a direct solver levels out the performances and no insightful remarks can be drawn, except the most obvious ones. Undoubtedly, the Picard iterations will always be less efficient than the two other convection treatments (namely, the linear and explicit convection), and similarly, a resolution of a system coming from the AC technique will be faster than for a saddle-point problem. However, for instance, using a direct solver will not allow us to appreciate the difference that there could be between a system obtained



(a) Velocity: directions (on selected cells) and magnitude.

(b) Pressure.

Figure 4.7 – 2D Taylor–Green Vortex (4.59), reference solution at $t = 0$. Cartesian mesh composed of 128^2 cells (see Fig. 3.3a).

by linearization of the convection (hence nonsymmetric) and one where the convection is explicit (hence symmetric).

No upwind stabilization is considered, $\Xi^{\mathrm{upw}} := 0$. Let us also remark that $\underline{u}_{\mathrm{TGV}} \cdot \underline{n}_{\partial\Omega} = 0$ on $\partial\Omega$. Hence, owing to Lemma 2.45, if the convection field is (discretely) divergence-free, the CDO convective trilinear form is skew-symmetric. This should be the case whenever the monolithic approach is used, no matter the convection treatment (or, indeed, no matter the iteration if the Picard algorithm is considered) since the approximate convection field is always the solution of an Oseen problem. The tolerance for the Picard algorithm ε^{P} has been set to 10^{-6}.

Remark 4.14 - Choice of η for AC. The arbitrary parameter of the AC method was chosen so that $\eta = 10Re$. Indeed, in order to obtain results whose accuracy is comparable to that obtained with the monolithic approach, the Reynolds number seems to play a role in the determination of an adequate parameter η. We report in Table 4.2 the errors obtained with $Re \approx 33$ and $Re = 100$, the finest considered Δt, a linearized convection operator, and by letting η vary. All in all, the choice $\eta = 10Re$ seems to ensure a good level of accuracy comparable to the level obtained with the monolithic approach while still leading to an acceptable conditioning of the system. Hence, we will use $\eta = 10Re$ in this test case. $\diamond$

4.5.1 Convergence in time

Different combinations of coupling techniques and convection treatments are now tested to see if first-order of convergence in time is recovered. Two values of the viscosity are considered: $\nu = 0.03$ leading to $Re \approx 33$ and $\nu = 0.01$ leading to $Re = 100$. The final time is chosen so that $\max_\Omega |\underline{u}_{\mathrm{TGV}}(T, \underline{x})| \approx \frac{1}{10} \max_\Omega |\underline{u}_{\mathrm{TGV}}(0, \underline{x})|$, giving $T = 40$ for $Re \approx 33$, and $T = 120$ for $Re = 100$. The considered time step values are $\Delta t = \frac{T}{8}, \frac{T}{16}, \frac{T}{32}, \frac{T}{64}$.

The results are reported in Fig. 4.8. The data about the explicit convection for $Re = 100$ is missing since the considered values of the time step lead to the divergence of the method, both with the monolithic and AC techniques: the stability of this procedure will be further

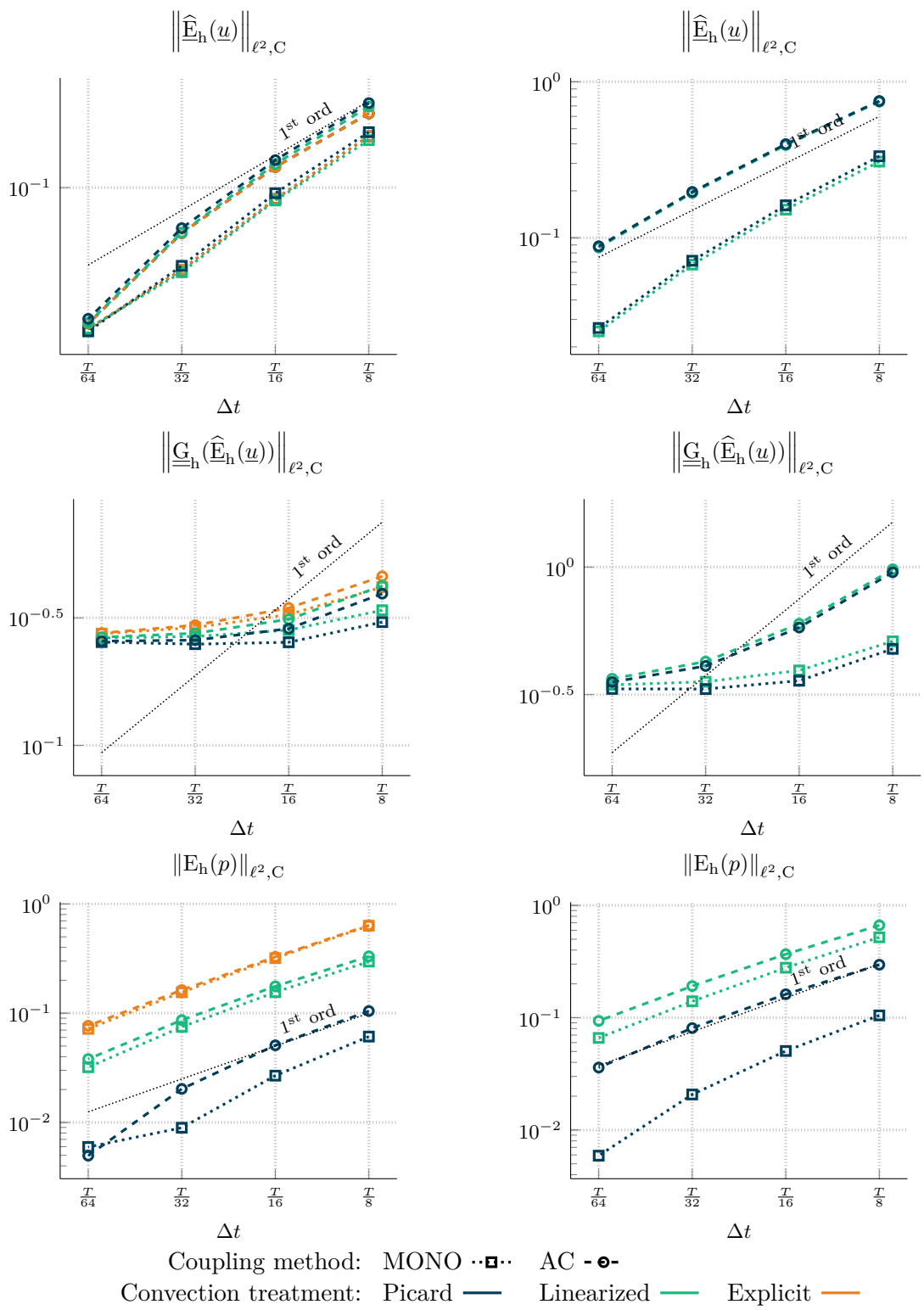


Figure 4.8 – 2D Navier–Stokes, Taylor–Green Vortex. Convergence in time. Top: velocity L^2-error; middle: velocity H^1-error; bottom: pressure L^2-error. Left column: $Re \approx 33$, $T = 40$, Cartesian mesh composed of 128^2 cells; left column: $Re = 100$, $T = 120$, Cartesian mesh composed of 512^2 cells.

Table 4.2 – Space-time velocity and pressure errors. Linearized convection. AC method with $\eta \in \{Re, 10Re, 100Re\}$. The errors obtained with the monolithic approach are given in parenthesis as reference. First group of columns: $Re \approx 33$, Cartesian mesh composed of 128^2 cells, $T = 40$, $\Delta t = \frac{T}{64} = 0.625$. Second group of columns: $Re = 100$, Cartesian mesh composed of 512^2 cells, $T = 120$, $\Delta t = \frac{T}{64} = 1.875$.

	$Re \approx 33$				$Re = 100$			
η	Re	$10Re$	$100Re$	(MONO)	Re	$10Re$	$100Re$	(MONO)
$\left\lVert \widehat{\underline{\mathrm{E}}}_{\mathrm{h}}(\underline{u}) \right\rVert_{\ell^2,\mathrm{C}}$	$1.35e{-1}$	$1.80e{-2}$	$1.59e{-2}$	$(1.58e{-2})$	$6.44e{-1}$	$8.67e{-2}$	$3.06e{-2}$	$(2.51e{-2})$
$\left\lVert \underline{\underline{\mathrm{G}}}_{\mathrm{h}}(\widehat{\underline{\mathrm{E}}}_{\mathrm{h}}(\underline{u})) \right\rVert_{\ell^2,\mathrm{C}}$	$3.21e{-1}$	$2.65e{-1}$	$2.64e{-1}$	$(2.64e{-2})$	$8.91e{-1}$	$3.65e{-1}$	$3.46e{-1}$	$(3.45e{-1})$
$\lVert \mathrm{E}_{\mathrm{h}}(p) \rVert_{\ell^2,\mathrm{C}}$	$8.99e{-2}$	$3.80e{-2}$	$3.31e{-2}$	$(3.21e{-2})$	$3.34e{-1}$	$9.36e{-2}$	$6.92e{-2}$	$(6.60e{-2})$

discussed in Section 4.5.3. We observe in Fig. 4.8 that the expected orders of convergence in time are recovered for all the strategies, except for the H^1-norm of the velocity which shows an early stagnation likely due to the spatial error becoming dominant. As usual, we observe a slight difference between the errors obtained with AC($10Re$) method and the monolithic approach. Another relevant observation is that the convection treatment has a significant impact on the pressure error since the errors curves are grouped by the convection treatment. This behavior may be explained by the particular setting of the TGV test case in which the convection term compensates the gradient of the pressure. As expected, the Picard algorithm is the most accurate, followed by the linearized convection and, finally, the explicit convection.

4.5.2 Convection treatments and dissipativity

We now test the dissipativity of the scheme with respect to the kinetic energy $\mathcal{E}_{\mathrm{kin,h}}(\widehat{\underline{w}}_{\mathrm{h}})$ defined in (4.16). We recall that the monolithic approach combined with an implicit or linearized convection has been proved to be dissipative (see Lemmas 4.8 and 4.11), whereas one has no a priori knowledge about the monolithic approach with the explicit convection or with the AC method (whichever convection treatment is considered). Let us notice that we will investigate $\mathcal{E}_{\mathrm{kin,h}}(\widehat{\underline{w}}_{\mathrm{h}})$ and not $\mathcal{E}_{\mathrm{AC,h}}(\widehat{\underline{w}}_{\mathrm{h}}, q_{\mathrm{h}}) := \mathcal{E}_{\mathrm{kin,h}}(\widehat{\underline{w}}_{\mathrm{h}}) + \frac{\Delta t}{2\nu\eta} \lVert q_{\mathrm{h}} \rVert^2_{L^2(\Omega)}$ when dealing with the AC method.

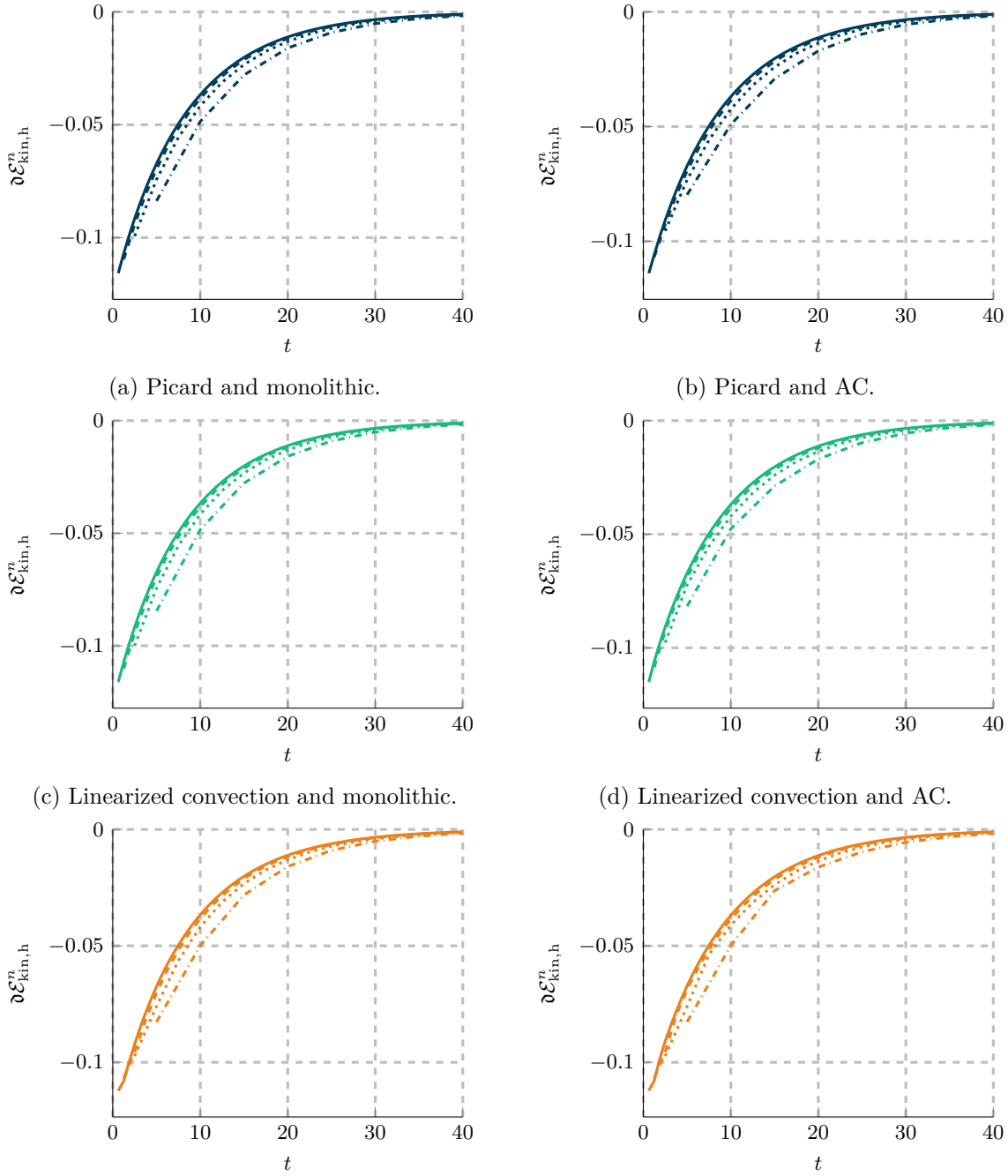
We investigate the following quantity:

$$\mathfrak{d}\mathcal{E}^n_{\mathrm{kin,h}} := \frac{\mathcal{E}_{\mathrm{kin,h}}(\widehat{\underline{v}}^n_{\mathrm{h}}) - \mathcal{E}_{\mathrm{kin,h}}(\widehat{\underline{v}}^{n-1}_{\mathrm{h}})}{\Delta t}, \qquad n = 1, \ldots, N. \tag{4.61}$$

$\mathfrak{d}\mathcal{E}^n_{\mathrm{kin,h}}$ is a first-order measure of the time-derivative of the kinetic energy at the discrete time node t^n. Hence, one expects $\mathfrak{d}\mathcal{E}^n_{\mathrm{kin,h}}$ to be negative whenever the scheme is dissipative. The results concerning $\mathfrak{d}\mathcal{E}^n_{\mathrm{kin,h}}$ for $Re \approx 33$ and 100 are shown respectively in Figs. 4.9 and 4.10. We notice that the negativity of $\mathfrak{d}\mathcal{E}^n_{\mathrm{kin,h}}$ is recovered for the monolithic approach as expected, but also for the AC method. No remarkable difference due to the convection treatment is observed: see, for instance, Fig. 4.11, which, for a given coupling technique and a time step value, compares the data obtained for the different convection treatments.

4.5.3 Stability results with an explicit convection

As expected, some stability issues are observed in combination with the explicit convection strategy in the results presented in Sections 4.5.1 and 4.5.2. We shed here some light by means of a numerical study. For both the monolithic approach and the AC method, we let Δt vary, and we seek the critical time-step value, Δt_{s}, that is the greatest Δt ensuring that the computation does not diverge (see (vi) below). For these tests, we take (i) a Cartesian mesh composed of 128^2 cells, (ii) three Reynolds numbers, $Re \in \{200, 500, 1000\}$ (notice that



(a) Picard and monolithic.

(b) Picard and AC.

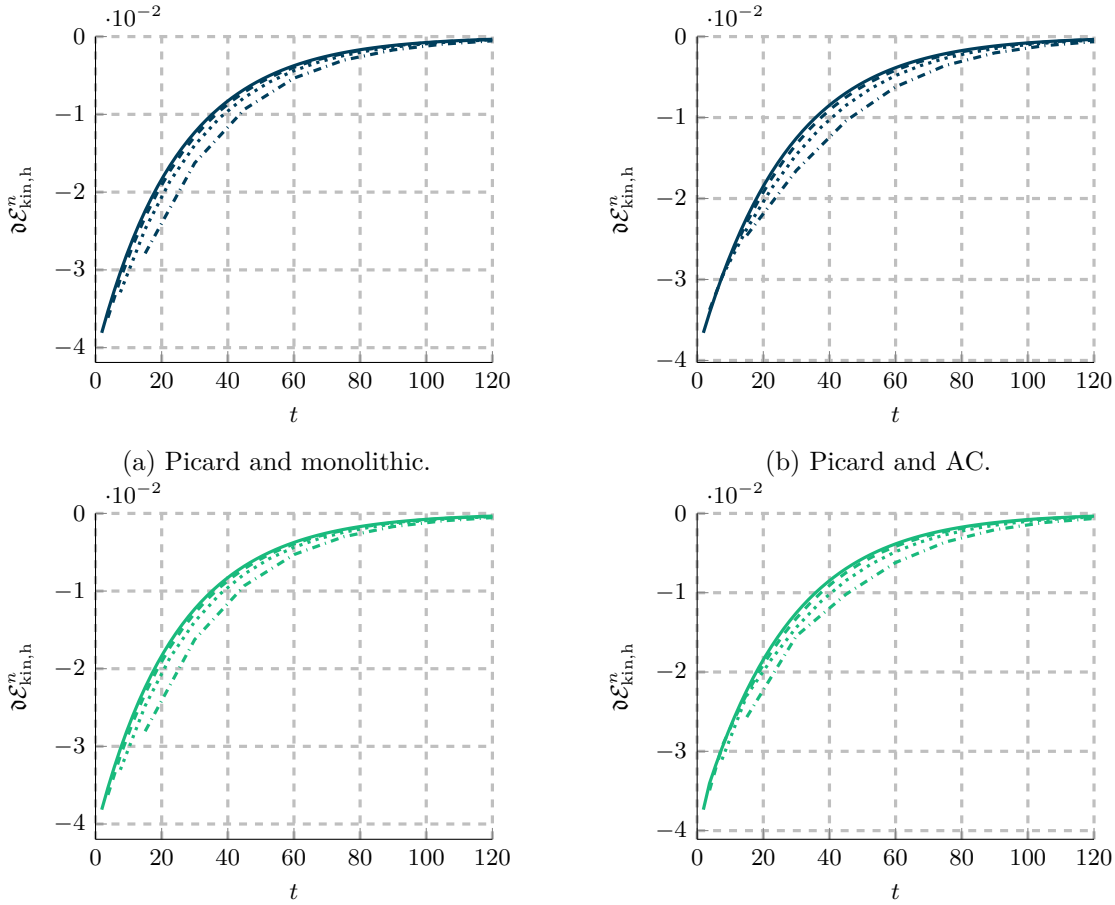
(c) Linearized convection and monolithic.

(d) Linearized convection and AC.

(e) Explicit convection and monolithic.

(f) Explicit convection and AC.

Figure 4.9 – 2D Navier–Stokes, Taylor–Green Vortex. $Re \approx 33$. $T = 40$. Mesh: Cartesian composed of 128^2. Discrete time-derivative of the kinetic energy at t^n, $\mathfrak{d}\mathcal{E}^n_{\mathrm{kin,h}}$, see (4.61). $\Delta t = \frac{T}{8}$ -·-·-, $\frac{T}{16}$ ·····, $\frac{T}{32}$ - - -, $\frac{T}{64}$ ——.



(a) Picard and monolithic.

(b) Picard and AC.

(c) Linearized convection and monolithic.

(d) Linearized convection and AC.

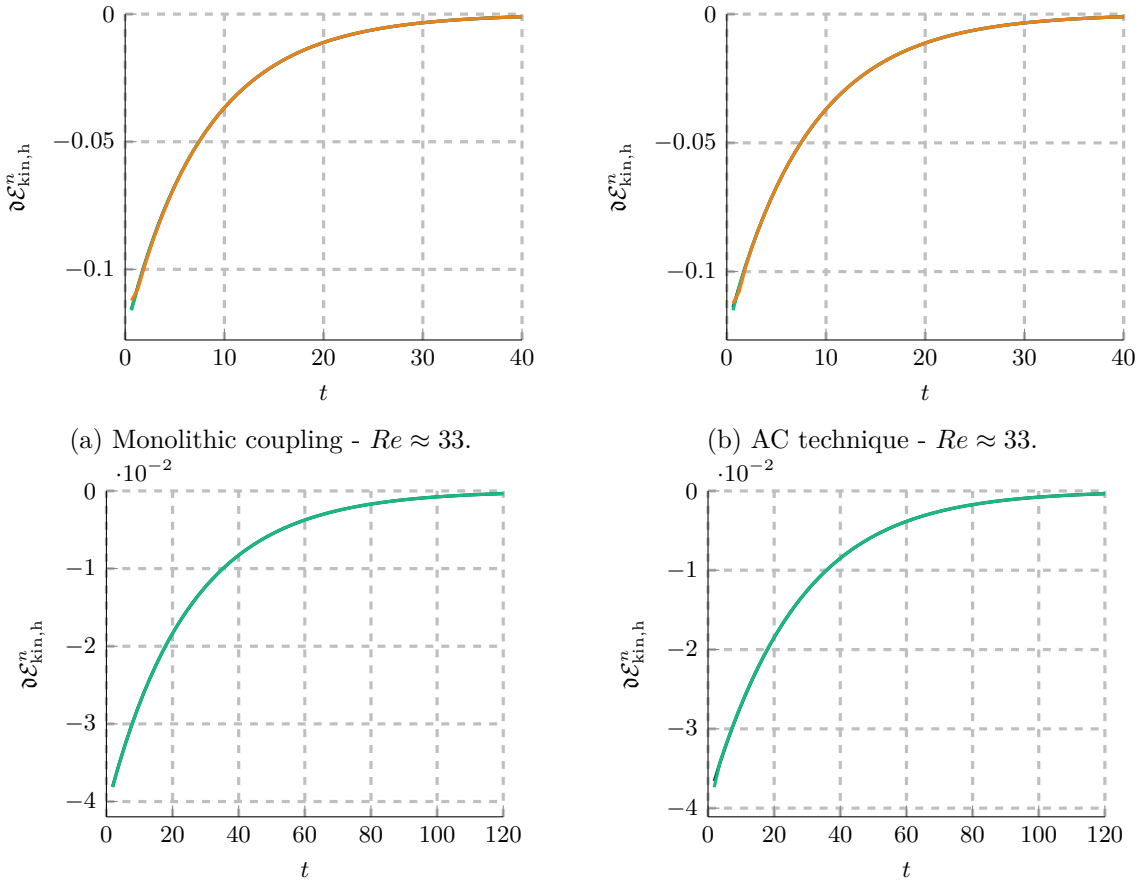
Figure 4.10 – 2D Navier–Stokes, Taylor–Green Vortex. $Re = 100$. $T = 120$. Mesh: Cartesian composed of 512^2. Discrete time-derivative of the kinetic energy at t^n, $\mathfrak{d}\mathcal{E}^n_{\mathrm{kin,h}}$, see (4.61). $\Delta t = \frac{T}{8}$ -·-·-, $\frac{T}{16}$ ·····, $\frac{T}{32}$ - - -, $\frac{T}{64}$ ——.

they are all higher than the numbers considered in Sections 4.5.1 and 4.5.2), (iii) T such that $T\,Re = 10^4$, (iv) $\eta = 10\,Re$ whenever the AC method is used (cf. Remark 4.14), (v) we seek a resolution of 1%, meaning that the gap between Δt_{s} and the smallest Δt leading to divergence is less than 1% of Δt_{s}, (vi) and we flag a computation as having diverged if, for some $n \geq 1$, we have

$$\mathcal{E}_{\mathrm{kin,h}}(\underline{\widehat{u}}^n_{\mathrm{h}}) > 1.1\,\mathcal{E}_{\mathrm{kin,h}}(\underline{\widehat{u}}^0_{\mathrm{h}}) = 1.1\,\mathcal{E}_{\mathrm{kin,h}}(\underline{\widehat{\pi}}_{\mathrm{h}}(\underline{u}_0))\,. \tag{4.62}$$

Notice that the solution (4.59) goes exponentially towards 0 with respect to time. Thus a failure to satisfy (4.62) is a symptom of numerical issues. Whenever $\Delta t > \Delta t_{\mathrm{s}}$, we define the divergence time, T_{d}, as the smallest t^n satisfying (4.62).

The results for, respectively, the monolithic approach and the AC method (with $\eta = 10Re$) are shown in Figs. 4.12 and 4.13, a one-glance summary with all the obtained Δt_{s} is given in Table 4.3 (where the results with other values of η for the AC method are presented as well). We observe that the two coupling strategies show similar results: as a matter of fact, the critical time-step values of the AC method lie in the confidence interval (recall that we chose a resolution of 1%) of the values obtained with the monolithic approach. Moreover, as shown in the bottom right panels of Figs. 4.12 and 4.13, we recover a dependency of Δt_{s} on the inverse of the Reynolds number. Finally, computations have been run by using the smallest Δt leading to divergence but considering a linearized convection (instead of explicit),



(a) Monolithic coupling - $Re \approx 33$.

(b) AC technique - $Re \approx 33$.

(c) Monolithic coupling - $Re = 100$.

(d) AC technique - $Re = 100$.

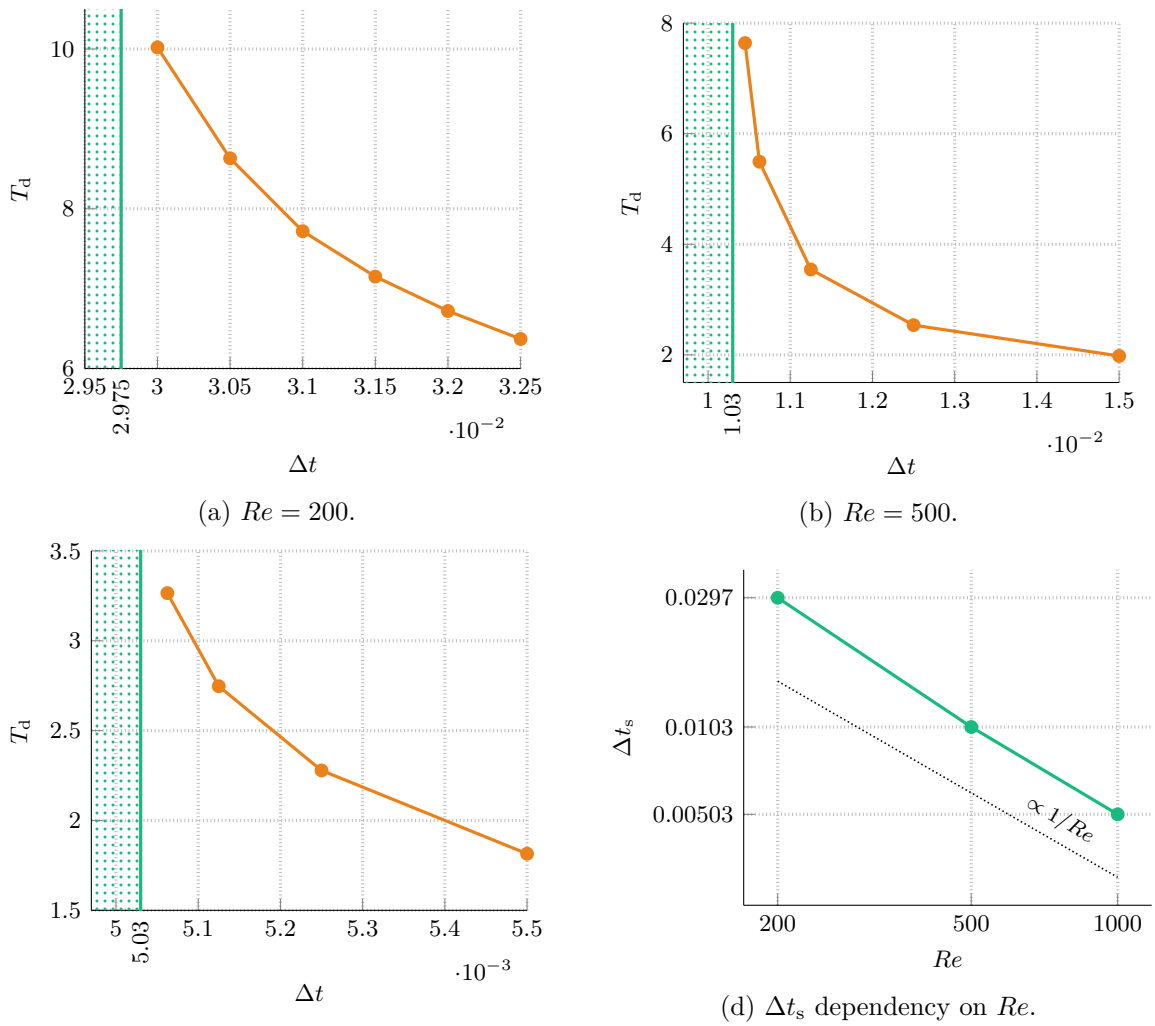
Figure 4.11 – 2D Navier–Stokes, Taylor–Green Vortex. Discrete time-derivative of the kinetic energy at t^n, $\mathfrak{d}\mathcal{E}^n_{\mathrm{kin,h}}$, see (4.61). Picard algorithm —, linearized — or explicit — convection Left: monolithic approach, right: AC technique. Top: $Re \approx 33$, $T = 40$, Cartesian mesh composed of 128^2 cells. Bottom: $Re = 100$, $T = 120$, Cartesian mesh composed of 512^2 cells. $\Delta t = \frac{T}{64}$.

Table 4.3 – Stability limits, Δt_{s}, up to a resolution of 1% obtained on a Cartesian mesh composed of 128^2 cells with the monolithic approach or the AC method for three Reynolds numbers, Re.

Re	MONO	AC($1Re$)	AC($10Re$)	AC($100Re$)
200	$2.98e{-2}$	$3.00e{-2}$	$2.98e{-2}$	$2.97e{-2}$
500	$1.03e{-2}$	$1.04e{-2}$	$1.04e{-2}$	$1.03e{-2}$
1000	$5.03e{-3}$	$5.05e{-3}$	$5.03e{-3}$	$5.00e{-3}$

and these computations did not diverge, thus confirming that the explicit treatment of the convection is the central issue. Concerning the influence of the parameter η on the stability of the AC method (see last columns of Table 4.3), we notice a slight decrease in Δt_{s} when moving to a higher value of η (we were expecting the opposite behavior). However, we observe that the results obtained for $\eta := Re$ lie just outside the 1%-resolution intervals of Δt_{s} obtained for $\eta := 100Re$.

Conservation of the kinetic energy is not the only criterion for stability. We now investigate a second criterion based on the enstrophy, $\phi(\omega) := \int_\Omega |\omega|^2$, where ω is the (scalar-valued)



(a) $Re = 200$.

(b) $Re = 500$.

(c) $Re = 1000$.

(d) Δt_s dependency on Re.

Figure 4.12 – 2D Navier–Stokes, Taylor–Green Vortex. Divergence time, T_d, for different choices of Δt with the monolithic approach and explicit convection for three Reynolds numbers, Re, and dependency of the stability limit, Δt_s, (up to 1% resolution) on Re. T_d are in orange, Δt_s in green. Cartesian mesh composed of 128^2 cells.

vorticity. At discrete level, the vorticity is calculated as follows:

$$\omega_\mathrm{c} := [\underline{\underline{\mathrm{G}}}^0_\mathrm{c}(\underline{\widehat{u}}_\mathrm{c})]_{yx} - [\underline{\underline{\mathrm{G}}}^0_\mathrm{c}(\underline{\widehat{u}}_\mathrm{c})]_{xy} \qquad \forall \mathrm{c} \in \mathrm{C}\,. \tag{4.63}$$

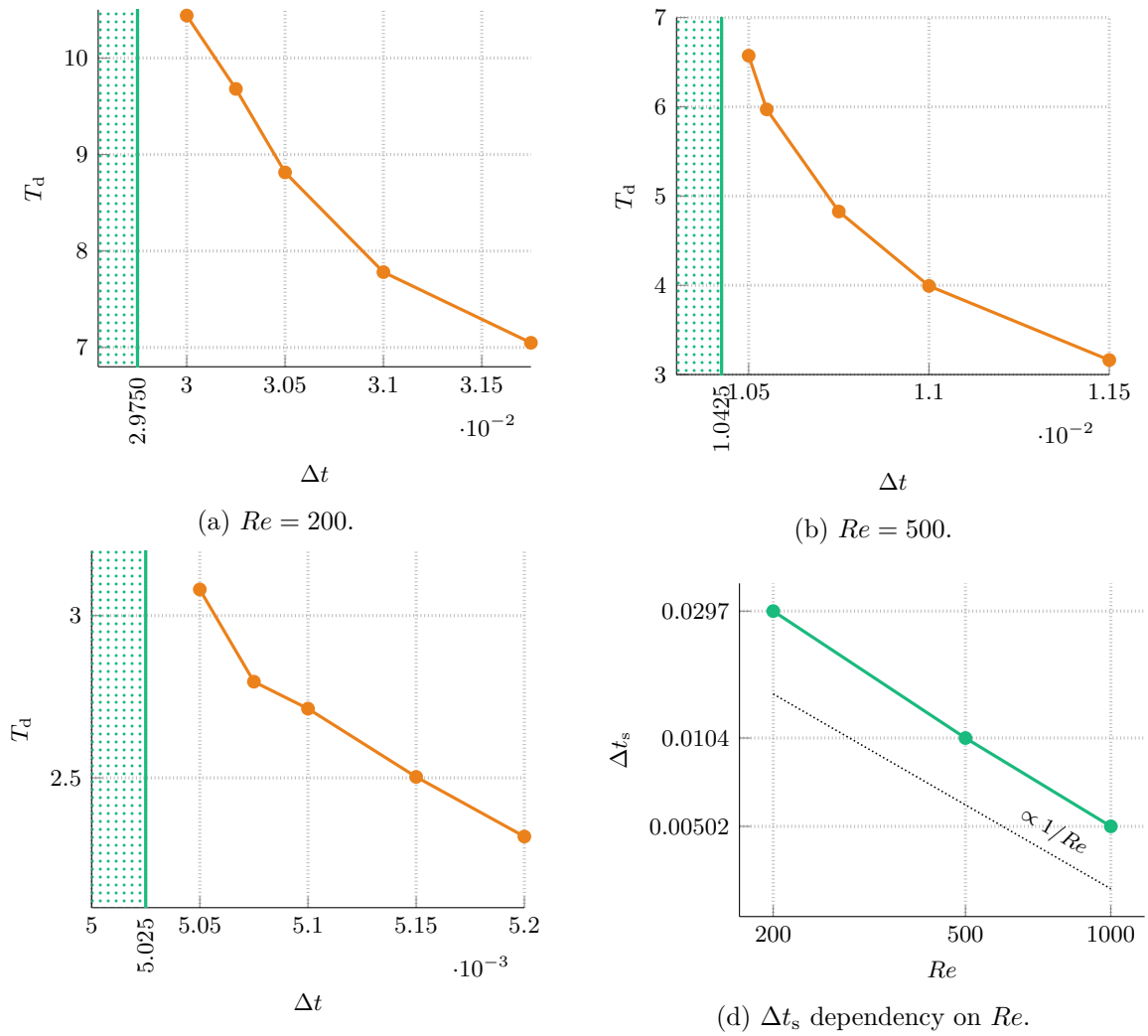
and we define the discrete enstrophy as

$$\phi_\mathrm{h}(\omega_\mathrm{C}) := \sum_{\mathrm{c}\in\mathrm{C}} |\mathrm{c}|\,|\omega_\mathrm{c}|^2\,. \tag{4.64}$$

We thus extend the setting of this numerical experiment described in the hypotheses (i)-(vi) above by adding: (vii) we flag a computation as having diverged if, for some $n \geq 1$, we have

$$\phi_\mathrm{h}(\omega^n_\mathrm{C}) > 1.1\,\phi_\mathrm{h}(\omega^0_\mathrm{C})\,. \tag{4.65}$$

A comparison of the results obtained with the setting (i)-(vi) and (i)-(vii) is given in Table 4.4. The addition of the new criterion on the enstrophy seems to affect only the higher Reynolds numbers, but the critical time-step values Δt_s obtained for the two settings still remain close.



(a) $Re = 200$.

(b) $Re = 500$.

(c) $Re = 1000$.

(d) Δt_s dependency on Re.

Figure 4.13 – 2D Navier–Stokes, Taylor–Green Vortex. Divergence time, T_d, for different choices of Δt with the AC method with $\eta = 10Re$ and explicit convection for three Reynolds numbers, Re, and dependency of the stability limit, Δt_s, (up to 1% resolution) on Re. T_d are in orange, Δt_s in green. Cartesian mesh composed of 128^2 cells.

Table 4.4 – 2D Navier–Stokes, Taylor–Green Vortex. Stability limits, Δt_s, up to a resolution of 1% obtained on a Cartesian mesh composed of 128^2 cells with the monolithic approach or the AC($10Re$) method for three Reynolds numbers, Re. Comparison between a divergence criterion concerning the kinetic energy only, or the kinetic energy and the enstrophy.

	MONO		AC($10Re$)	
Re	$\mathcal{E}_\mathrm{kin,h}$	$\mathcal{E}_\mathrm{kin,h}$ & ϕ_h	$\mathcal{E}_\mathrm{kin,h}$	$\mathcal{E}_\mathrm{kin,h}$ & ϕ_h
200	$2.98e{-2}$	$2.98e{-2}$	$2.98e{-2}$	$2.98e{-2}$
500	$1.03e{-2}$	$1.03e{-2}$	$1.04e{-2}$	$1.03e{-2}$
1000	$5.03e{-3}$	$4.96e{-3}$	$5.03e{-3}$	$4.96e{-3}$

4.6 Detailed results

We give in this section additional results on the test cases presented in Section 4.4 and Section 4.5.

4.6.1 Stokes equations

In Section 4.4, several simulations of the modified 3D Taylor–Green Vortex test case for the unsteady Stokes equation have been run. Their results for the different meshes (Cartesian, CheckerBoard, and Prismatic with polygonal bases), the two coupling techniques (monolithic approach and AC method), and several combinations of iterative solvers and preconditioners are collected in Tables 4.5 to 4.13. In addition to the velocity and error values already presented in Section 4.4, the tables also provide insights on the computational time and the performances of the iterative solvers. We refer to the introduction in Section 4.4 for more details on the algebraic setting.

Table 4.5 – Unsteady Stokes problem. 3D TGV solution (4.50), $T = 2$. Cartesian mesh with 256^3 cells. Detailed results for the monolithic approach solved with a GKB algorithm with an arbitrary parameter $\gamma = 0$, and a CG solver equipped with an AMG preconditioner - ··■·· in Fig. 4.2.

$\Delta t = \frac{T}{\cdots}$	$\left\lVert \widehat{\underline{\mathrm{E}}}_{\mathrm{h}}(\underline{u}) \right\rVert_{\ell^2,\mathrm{C}}$	Order	$\left\lVert \underline{\underline{\mathrm{G}}}_{\mathrm{h}}(\widehat{\underline{\mathrm{E}}}_{\mathrm{h}}(\underline{u})) \right\rVert_{\ell^2,\mathrm{C}}$	Order	$\lVert \mathrm{E}_{\mathrm{h}}(p) \rVert_{\ell^2,\mathrm{C}}$	Order	Elapsed [s]	#cores	Elps×#rnk [s]	Elps×#rnk/#TS [s]	Solver Iter/#TS
4	$1.35e{-2}$	–	$1.08e{-1}$	–	$9.09e{-2}$	–	$1.88e{+3}$	350	$6.58e{+5}$	$1.65e{+5}$	237
8	$6.79e{-3}$	0.99	$5.44e{-2}$	0.99	$4.59e{-2}$	0.99	$3.76e{+3}$	350	$1.31e{+6}$	$1.64e{+5}$	238
16	$3.50e{-3}$	0.96	$2.80e{-2}$	0.96	$2.36e{-2}$	0.96	$7.53e{+3}$	350	$2.63e{+6}$	$1.65e{+5}$	233
32	$1.77e{-3}$	0.98	$1.43e{-2}$	0.98	$1.20e{-2}$	0.98	$1.45e{+4}$	350	$5.09e{+6}$	$1.59e{+5}$	229
64	$8.95e{-4}$	0.99	$7.31e{-3}$	0.96	$6.04e{-3}$	0.99	$2.85e{+4}$	350	$9.97e{+6}$	$1.56e{+5}$	229
128	$4.54e{-4}$	0.98	$3.92e{-3}$	0.90	$3.06e{-3}$	0.98	$3.87e{+4}$	525	$2.03e{+7}$	$1.59e{+5}$	238
256	$2.33e{-4}$	0.96	$2.39e{-3}$	0.72	$1.57e{-3}$	0.96	$8.05e{+4}$	525	$4.22e{+7}$	$1.65e{+5}$	251

Table 4.6 – Unsteady Stokes problem. 3D TGV solution (4.50), $T = 2$. Cartesian mesh with 256^3 cells. Detailed results for the AC technique, arbitrary parameter $\eta = 10$, solved with a CG iterative solver and a Jacobi preconditioner - -⊙- in Fig. 4.2.

$\Delta t = \frac{T}{\cdots}$	$\left\lVert \widehat{\underline{\mathrm{E}}}_{\mathrm{h}}(\underline{u}) \right\rVert_{\ell^2,\mathrm{C}}$	Order	$\left\lVert \underline{\underline{\mathrm{G}}}_{\mathrm{h}}(\widehat{\underline{\mathrm{E}}}_{\mathrm{h}}(\underline{u})) \right\rVert_{\ell^2,\mathrm{C}}$	Order	$\lVert \mathrm{E}_{\mathrm{h}}(p) \rVert_{\ell^2,\mathrm{C}}$	Order	Elapsed [s]	#cores	Elps×#rnk [s]	Elps×#rnk/#TS [s]	Solver Iter/#TS
4	$1.49e{-2}$	–	$1.12e{-1}$	–	$1.13e{-1}$	–	$5.49e{+2}$	350	$1.92e{+5}$	$4.81e{+4}$	2631
8	$8.42e{-3}$	0.82	$6.18e{-2}$	0.86	$9.35e{-2}$	0.27	$1.08e{+3}$	350	$3.78e{+5}$	$4.73e{+4}$	2613
16	$4.61e{-3}$	0.87	$3.37e{-2}$	0.87	$5.87e{-2}$	0.67	$2.06e{+3}$	350	$7.22e{+5}$	$4.51e{+4}$	2504
32	$2.40e{-3}$	0.94	$1.77e{-2}$	0.93	$3.21e{-2}$	0.87	$3.98e{+3}$	350	$1.39e{+6}$	$4.35e{+4}$	2388
64	$1.22e{-3}$	0.98	$9.23e{-3}$	0.94	$1.65e{-2}$	0.96	$7.39e{+3}$	350	$2.59e{+6}$	$4.04e{+4}$	2239
128	$6.17e{-4}$	0.99	$4.98e{-3}$	0.89	$8.31e{-3}$	0.99	$1.32e{+4}$	350	$4.61e{+6}$	$3.60e{+4}$	2012
256	$3.13e{-4}$	0.98	$2.95e{-3}$	0.76	$4.18e{-3}$	0.99	$1.59e{+4}$	525	$8.34e{+6}$	$3.26e{+4}$	1730

Table 4.7 – Unsteady Stokes problem. 3D TGV solution (4.50), $T = 2$. Cartesian mesh with 256^3 cells. Detailed results for the AC technique, arbitrary parameter $\eta = 100$, solved with a CG iterative solver and a Jacobi preconditioner - -⊙- in Fig. 4.2.

$\Delta t = \frac{T}{\cdots}$	$\left\lVert \widehat{\underline{\mathrm{E}}}_{\mathrm{h}}(\underline{u}) \right\rVert_{\ell^2,\mathrm{C}}$	Order	$\left\lVert \underline{\underline{\mathrm{G}}}_{\mathrm{h}}(\widehat{\underline{\mathrm{E}}}_{\mathrm{h}}(\underline{u})) \right\rVert_{\ell^2,\mathrm{C}}$	Order	$\lVert \mathrm{E}_{\mathrm{h}}(p) \rVert_{\ell^2,\mathrm{C}}$	Order	Elapsed [s]	#cores	Elps×#rnk [s]	Elps×#rnk/#TS [s]	Solver Iter/#TS
4	$1.35e{-2}$	–	$1.08e{-1}$	–	$8.88e{-2}$	–	$9.52e{+2}$	525	$5.00e{+5}$	$1.25e{+5}$	6965
8	$6.83e{-3}$	0.99	$5.45e{-2}$	0.98	$4.65e{-2}$	0.93	$1.79e{+3}$	525	$9.40e{+5}$	$1.17e{+5}$	6909
16	$3.52e{-3}$	0.96	$2.81e{-2}$	0.95	$2.43e{-2}$	0.94	$3.61e{+3}$	525	$1.89e{+6}$	$1.18e{+5}$	6612
32	$1.78e{-3}$	0.98	$1.43e{-2}$	0.98	$1.23e{-2}$	0.98	$6.87e{+3}$	525	$3.60e{+6}$	$1.13e{+5}$	6241
64	$8.99e{-4}$	0.99	$7.33e{-3}$	0.96	$6.23e{-3}$	0.99	$1.26e{+4}$	525	$6.63e{+6}$	$1.04e{+5}$	5786
128	$4.56e{-4}$	0.98	$3.94e{-3}$	0.90	$3.16e{-3}$	0.98	$2.31e{+4}$	525	$1.21e{+7}$	$9.46e{+4}$	5231
256	$2.34e{-4}$	0.96	$2.40e{-3}$	0.72	$1.62e{-3}$	0.96	$4.18e{+4}$	525	$2.19e{+7}$	$8.57e{+4}$	4656

Table 4.8 – Unsteady Stokes problem. 3D TGV solution (4.50), $T = 2$. Detailed results for the monolithic approach solved with a GKB algorithm with an arbitrary parameter $\gamma = 0$, and a CG solver equipped with an AMG preconditioner. Prismatic-Polygonal mesh. ··●·· in Figs. 4.5 and 4.6.

$\Delta t = \frac{T}{\cdots}$	$\left\lVert \widehat{\underline{\mathrm{E}}}_{\mathrm{h}}(\underline{u}) \right\rVert_{\ell^2,\mathrm{C}}$	Order	$\lVert \mathrm{E}_{\mathrm{h}}(p) \rVert_{\ell^2,\mathrm{C}}$	Order	Elapsed [s]	#cores	Elps×#rnk [s]	Elps×#rnk/#TS [s]	Solver Iter/#TS
4	1.35e−2	–	9.10e−2	–	1.26e+3	350	4.42e+5	1.10e+5	436
8	6.84e−3	0.98	4.61e−2	0.98	2.45e+3	350	8.57e+5	1.07e+5	435
16	3.55e−3	0.95	2.39e−2	0.95	4.76e+3	350	1.67e+6	1.04e+5	429
32	1.82e−3	0.96	1.23e−2	0.96	9.60e+3	350	3.36e+6	1.05e+5	440
64	9.47e−4	0.95	6.46e−3	0.93	1.35e+4	525	7.11e+6	1.11e+5	438
128	5.07e−4	0.90	3.57e−3	0.86	2.46e+4	525	1.29e+7	1.01e+5	459
256	2.87e−4	0.82	2.20e−3	0.70	5.02e+4	525	2.64e+7	1.03e+5	477

Table 4.9 – Unsteady Stokes problem. 3D TGV solution (4.50), $T = 2$. Detailed results for the AC technique, arbitrary parameter $\eta = 10$, solved with a CG iterative solver and a Jacobi preconditioner. Prismatic-Polygonal mesh. -●- in Figs. 4.5 and 4.6.

$\Delta t = \frac{T}{\cdots}$	$\left\lVert \widehat{\underline{\mathrm{E}}}_{\mathrm{h}}(\underline{u}) \right\rVert_{\ell^2,\mathrm{C}}$	Order	$\lVert \mathrm{E}_{\mathrm{h}}(p) \rVert_{\ell^2,\mathrm{C}}$	Order	Elapsed [s]	#cores	Elps×#rnk [s]	Elps×#rnk/#TS [s]	Solver Iter/#TS
4	1.49e−2	–	1.13e−1	–	2.35e+2	350	8.23e+4	2.06e+4	2710
8	8.46e−3	0.82	9.36e−2	0.27	4.61e+2	350	1.61e+5	2.02e+4	2657
16	4.65e−3	0.86	5.88e−2	0.67	8.57e+2	350	3.00e+5	1.88e+4	2469
32	2.44e−3	0.93	3.22e−2	0.87	1.63e+3	350	5.70e+5	1.78e+4	2327
64	1.26e−3	0.96	1.66e−2	0.95	1.95e+3	525	1.02e+6	1.60e+4	2071
128	6.57e−4	0.94	8.51e−3	0.97	3.47e+3	525	1.82e+6	1.42e+4	1826
256	3.55e−4	0.89	4.46e−3	0.93	6.06e+3	525	3.18e+6	1.24e+4	1585

Table 4.10 – Unsteady Stokes problem. 3D TGV solution (4.50), $T = 2$. Detailed results for the AC technique, arbitrary parameter $\eta = 100$, solved with a CG iterative solver and a Jacobi preconditioner. Prismatic-Polygonal mesh. -●- in Figs. 4.5 and 4.6.

$\Delta t = \frac{T}{\cdots}$	$\left\lVert \widehat{\underline{\mathrm{E}}}_{\mathrm{h}}(\underline{u}) \right\rVert_{\ell^2,\mathrm{C}}$	Order	$\lVert \mathrm{E}_{\mathrm{h}}(p) \rVert_{\ell^2,\mathrm{C}}$	Order	Elapsed [s]	#cores	Elps×#rnk [s]	Elps×#rnk/#TS [s]	Solver Iter/#TS
4	1.36e−2	–	8.89e−2	–	6.52e+2	350	2.28e+5	5.70e+4	7557
8	6.87e−3	0.98	4.68e−2	0.93	1.27e+3	350	4.44e+5	5.55e+4	7426
16	3.57e−3	0.95	2.46e−2	0.93	2.34e+3	350	8.21e+5	5.13e+4	7063
32	1.83e−3	0.96	1.27e−2	0.96	4.64e+3	350	1.62e+6	5.08e+4	6686
64	9.51e−4	0.95	6.64e−3	0.94	8.39e+3	350	2.94e+6	4.59e+4	6052
128	5.09e−4	0.90	3.65e−3	0.86	1.46e+4	350	5.09e+6	3.98e+4	5328
256	2.88e−4	0.82	2.24e−3	0.71	2.53e+4	350	8.85e+6	3.46e+4	4581

Table 4.11 – Unsteady Stokes problem. 3D TGV solution (4.50), $T = 2$. Detailed results for the monolithic approach solved with a GKB algorithm with an arbitrary parameter $\gamma = 0$, and a CG solver equipped with an AMG preconditioner. CheckerBoard mesh. ··◆·· in Figs. 4.5 and 4.6.

$\Delta t = \frac{T}{\cdots}$	$\left\lVert \widehat{\underline{\mathrm{E}}}_{\mathrm{h}}(\underline{u}) \right\rVert_{\ell^2,\mathrm{C}}$	Order	$\lVert \mathrm{E}_{\mathrm{h}}(p) \rVert_{\ell^2,\mathrm{C}}$	Order	Elapsed [s]	#cores	Elps×#rnk [s]	Elps×#rnk/#TS [s]	Solver Iter/#TS
4	1.35e−2	–	9.13e−2	–	5.42e+3	350	1.90e+6	4.75e+5	217
8	6.82e−3	0.98	4.65e−2	0.97	1.07e+4	350	3.74e+6	4.67e+5	212
16	3.54e−3	0.95	2.47e−2	0.91	2.14e+4	350	7.48e+6	4.68e+5	212
32	1.81e−3	0.97	1.38e−2	0.84	4.20e+4	350	1.47e+7	4.59e+5	206
64	9.35e−4	0.96	8.94e−3	0.62	5.67e+4	525	2.98e+7	4.65e+5	205
128	4.94e−4	0.92	7.15e−3	0.32	1.11e+5	525	5.85e+7	4.57e+5	197
256	2.72e−4	0.86	6.59e−3	0.12	1.44e+5	875	1.26e+8	4.94e+5	202

Table 4.12 – Unsteady Stokes problem. 3D TGV solution (4.50), $T = 2$. Detailed results for the AC technique, arbitrary parameter $\eta = 10$, solved with a CG iterative solver and a Jacobi preconditioner. CheckerBoard mesh. -◆- in Figs. 4.5 and 4.6.

$\Delta t = \frac{T}{\cdots}$	$\left\lVert \widehat{\underline{\mathrm{E}}}_{\mathrm{h}}(\underline{u}) \right\rVert_{\ell^2,\mathrm{C}}$	Order	$\lVert \mathrm{E}_{\mathrm{h}}(p) \rVert_{\ell^2,\mathrm{C}}$	Order	Elapsed [s]	#cores	Elps×#rnk [s]	Elps×#rnk/#TS [s]	Solver Iter/#TS
4	1.49e−2	–	1.13e−1	–	2.47e+3	350	8.65e+5	2.16e+5	2829
8	8.45e−3	0.82	9.37e−2	0.27	4.92e+3	350	1.72e+6	2.15e+5	2817
16	4.64e−3	0.86	5.91e−2	0.67	9.43e+3	350	3.30e+6	2.06e+5	2697
32	2.43e−3	0.93	3.28e−2	0.85	1.81e+4	350	6.35e+6	1.98e+5	2565
64	1.25e−3	0.96	1.77e−2	0.89	2.26e+4	525	1.19e+7	1.86e+5	2405
128	6.46e−4	0.95	1.05e−2	0.75	4.02e+4	525	2.11e+7	1.65e+5	2154
256	3.44e−4	0.91	7.64e−3	0.46	7.02e+4	525	3.69e+7	1.44e+5	1864

Table 4.13 – Unsteady Stokes problem. 3D TGV solution (4.50), $T = 2$. Detailed results for the AC technique, arbitrary parameter $\eta = 100$, solved with a CG iterative solver and a Jacobi preconditioner. CheckerBoard mesh. -◆- in Figs. 4.5 and 4.6.

$\Delta t = \frac{T}{\cdots}$	$\left\lVert \widehat{\underline{\mathrm{E}}}_{\mathrm{h}}(\underline{u}) \right\rVert_{\ell^2,\mathrm{C}}$	Order	$\lVert \mathrm{E}_{\mathrm{h}}(p) \rVert_{\ell^2,\mathrm{C}}$	Order	Elapsed [s]	#cores	Elps×#rnk [s]	Elps×#rnk/#TS [s]	Solver Iter/#TS
4	1.35e−2	–	8.91e−2	–	6.26e+3	350	2.19e+6	5.48e+5	7159
8	6.86e−3	0.98	4.72e−2	0.92	1.25e+4	350	4.36e+6	5.45e+5	7131
16	3.56e−3	0.95	2.53e−2	0.90	2.51e+4	350	8.80e+6	5.50e+5	6815
32	1.82e−3	0.97	1.41e−2	0.85	4.44e+4	350	1.55e+7	4.86e+5	6398
64	9.38e−4	0.96	9.06e−3	0.64	5.53e+4	525	2.90e+7	4.54e+5	5922
128	4.95e−4	0.92	7.19e−3	0.33	9.99e+4	525	5.25e+7	4.10e+5	5363

4.6.2 Navier–Stokes equations

The classical 2D Taylor–Green Vortex for the NSE has been considered in Section 4.5. The same coupling techniques as in the previous test case (monolithic approach and AC method) are combined with the three classical convection treatments (Picard iterations, linearized and explicit convection).

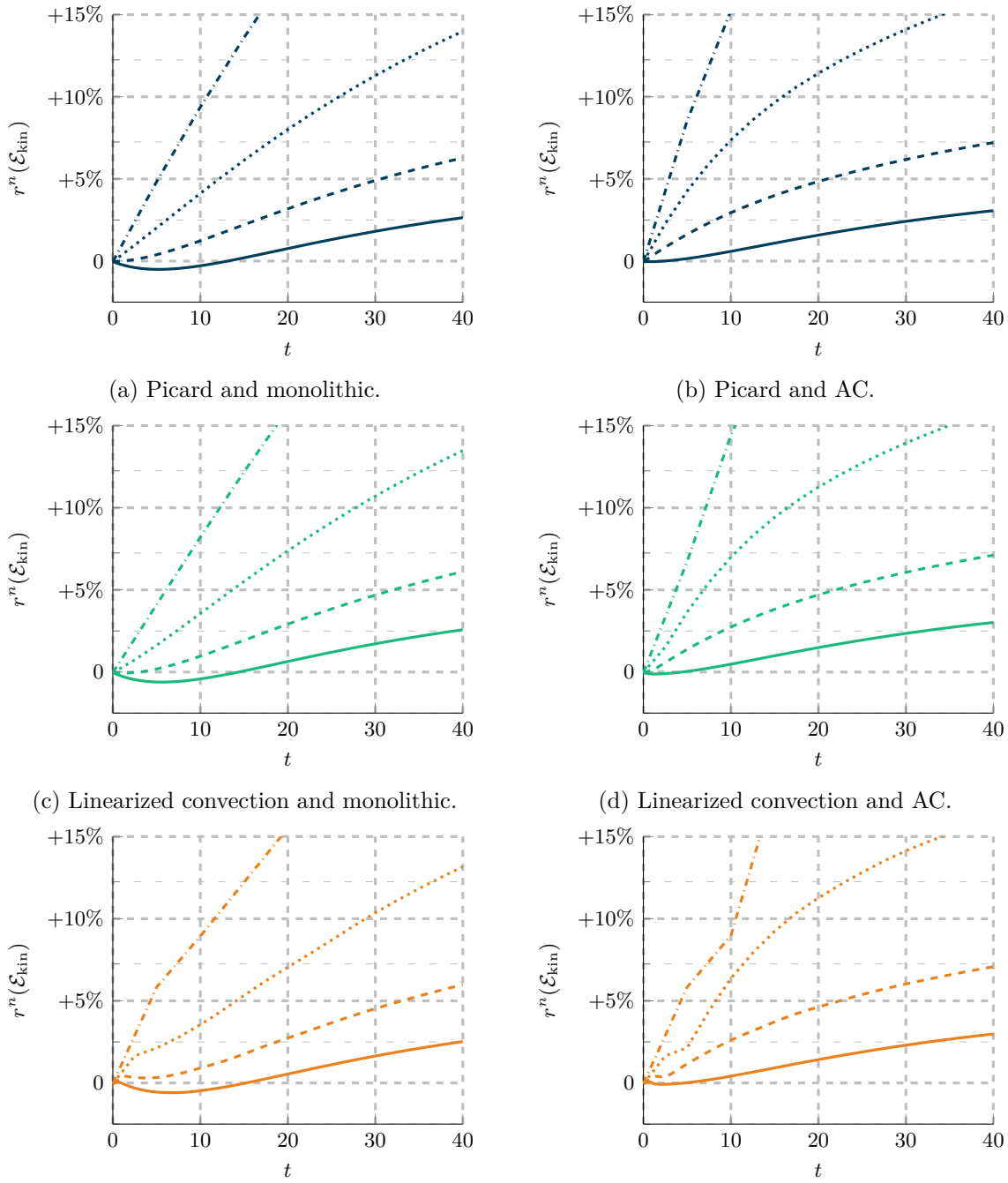
We investigate the following quantity

$$r^n(\mathcal{E}_{\text{kin}}) := \frac{\mathcal{E}_{\text{kin,h}}(\widehat{\underline{v}}_{\text{h}}^n) - \mathcal{E}_{\text{kin}}(t^n)}{\mathcal{E}_{\text{kin}}(t^n)}\,, \qquad n = 0, \ldots, N\,, \tag{4.66}$$

where

$$\mathcal{E}_{\text{kin}}(t) := \frac{1}{2}\int_\Omega |\underline{u}|_2^2 = \pi^2 \exp(-4\nu t)\,, \tag{4.67}$$

is the exact kinetic energy of the TGV flow, readily inferred from (4.59). $r^n(\mathcal{E}_{\text{kin}})$ is the normalized difference (with sign) of the computed and exact kinetic energy. We report in Figs. 4.14 and 4.15 the results obtained for $r^n(\mathcal{E}_{\text{kin}})$ at, respectively, $Re \approx 33$ and $Re = 100$. The difference between the coarsest and finest time step values that have been considered is significant, with the latter being remarkably closer to the reference curve. This shows the time discretization errors are dominant. Conversely, neither the coupling technique nor the convection treatment lead to notable differences. This remark is confirmed by the results in Fig. 4.16 which compare the data obtained for the smallest time step values between the different convection treatments.



(a) Picard and monolithic.

(b) Picard and AC.

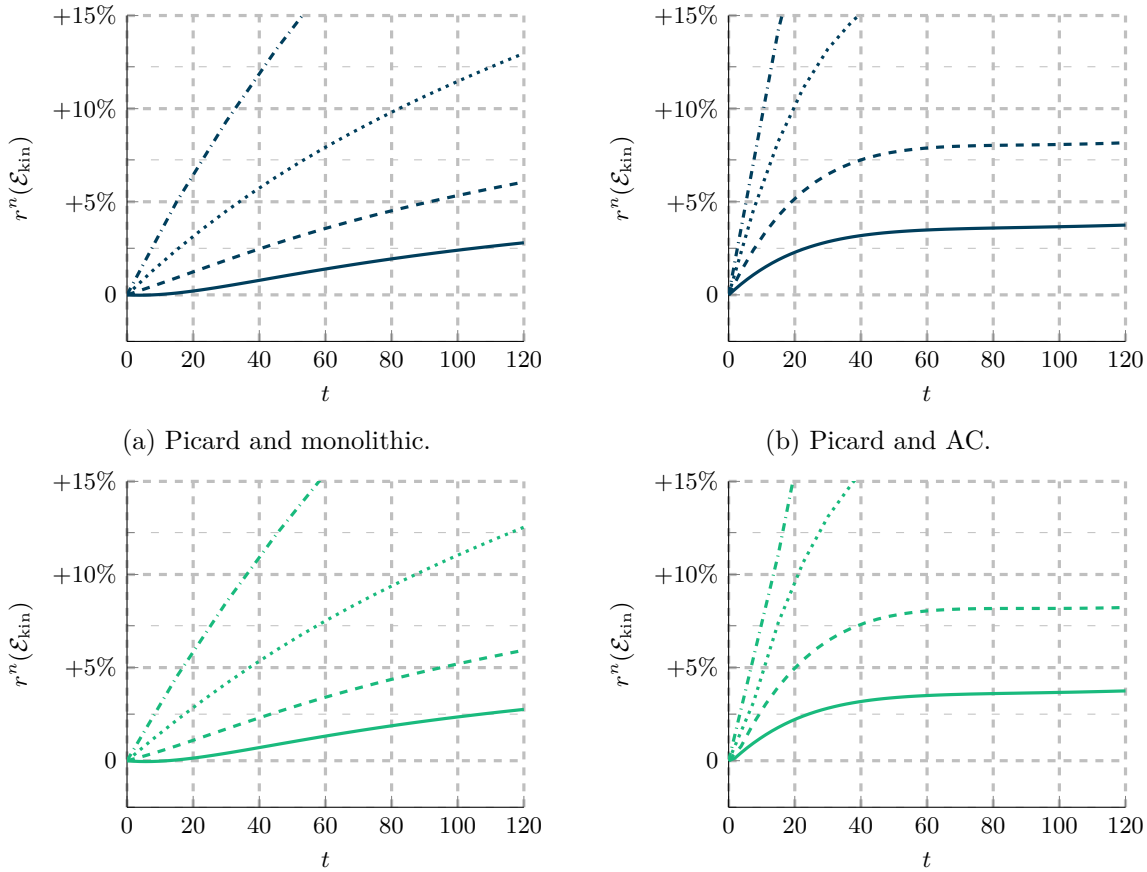
(c) Linearized convection and monolithic.

(d) Linearized convection and AC.

(e) Explicit convection and monolithic.

(f) Explicit convection and AC.

Figure 4.14 – 2D Navier–Stokes, Taylor–Green Vortex. $Re \approx 33$. $T = 40$. Mesh: Cartesian composed of 128^2 cells. Ratio (in percentage) between the computed and reference kinetic energy $r(\mathcal{E}_{\text{kin,h}})$ (see (4.66)) for different combinations of coupling technique and convection treatment. $\Delta t = \frac{T}{8}$ -·-·-, $\frac{T}{16}$ ·····, $\frac{T}{32}$ - - -, $\frac{T}{64}$ ——.



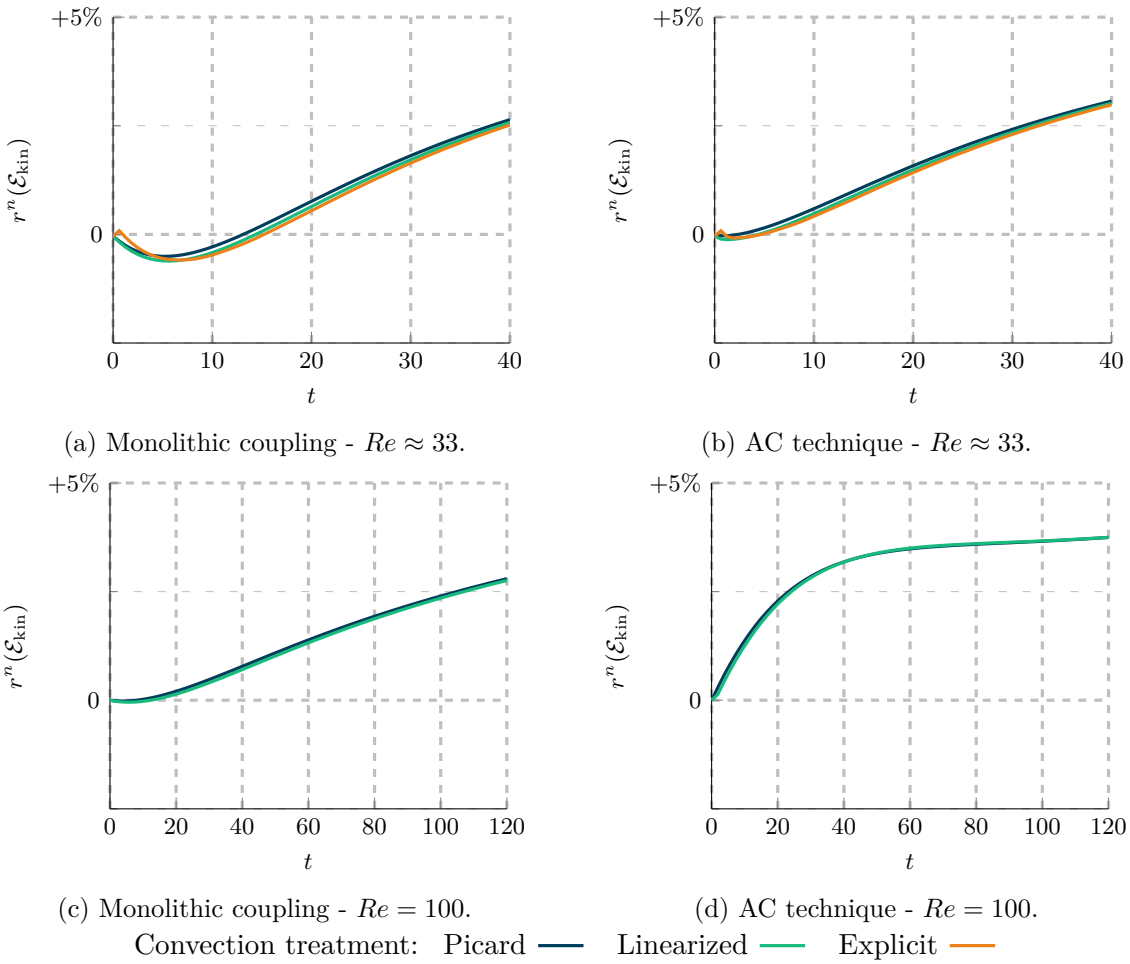
(a) Picard and monolithic.

(b) Picard and AC.

(c) Linearized convection and monolithic.

(d) Linearized convection and AC.

Figure 4.15 – 2D Navier–Stokes, Taylor–Green Vortex. $Re = 100$. $T = 120$. Mesh: Cartesian 512^2 cells. Ratio (in percentage) between the computed and reference kinetic energy $r\left(\mathcal{E}_{\text{kin,h}}\right)$ (see (4.66)) for different combinations of coupling technique and convection treatment. $\Delta t = \frac{T}{8}$ -·-·-, $\frac{T}{16}$ ·····, $\frac{T}{32}$ - - -, $\frac{T}{64}$ ——.



(a) Monolithic coupling - $Re \approx 33$.

(b) AC technique - $Re \approx 33$.

(c) Monolithic coupling - $Re = 100$.

(d) AC technique - $Re = 100$.

Figure 4.16 – 2D Navier–Stokes, Taylor–Green Vortex. Ratio (in percentage) between the computed and reference kinetic energy $r\left(\mathcal{E}_{\text{kin,h}}\right)$ (see (4.66)) for different combinations of coupling technique and convection treatment. Top: $Re \approx 33$, $T = 40$, Cartesian 128×128 mesh. Bottom: $Re = 100$, $T = 120$, Cartesian 512×512 mesh. $\Delta t = \frac{T}{64}$.

Chapter 5

Extension to second-order time-stepping

Contents

5.1 Second-order time-schemes 131

5.1.1 Monolithic approach 132

5.1.2 Artificial Compressibility 134

5.2 Numerical results: Stokes equations 136

5.2.1 Comparison: Monolithic approach vs. Artificial Compressibility . . 137

5.2.2 Comparison: first- vs. second-order time discretizations 139

5.3 Numerical results: Navier–Stokes equations 141

5.3.1 Convergence in time 141

5.3.2 Convection treatments and dissipativity 141

5.3.3 Stability results with an explicit convection 145

5.4 Detailed results 148

5.4.1 Stokes equations 148

5.4.2 Navier–Stokes equations 152

The framework introduced in Chapter 4 constitutes a minimal setting to deal with the time discretization of the Navier-Stokes equations (NSE). The setting is now extended to second-order time-schemes, both for the classical monolithic approach and the Artificial Compressibility (AC) method. As before, several strategies resulting from combinations of coupling techniques, time-schemes and convection treatments are compared both in terms of accuracy and efficiency. Two second-order time-schemes are presented in the first part of this chapter: a Backward Differentiation Formula (BDF2) applied to the monolithic approach and a bootstrapping technique applied to the AC method.

5.1 Second-order time-schemes

As in the previous chapter, two velocity-pressure coupling techniques are considered: the monolithic approach and the Artificial Compressibility method. We now present second-order time-schemes for both of these coupling techniques.

The model problem is: Find $(\underline{u}, p)$ such that

$$\begin{cases} \dfrac{\partial \underline{u}}{\partial t} - \nu \underline{\Delta}\, \underline{u} + \xi^{\mathrm{NS}} (\underline{u} \cdot \underline{\nabla}\,) \underline{u} + \underline{\nabla}\, p = \underline{f} & \text{in } \Omega \times (0,T)\,, \\ \underline{\nabla} \cdot \underline{u} = 0 & \text{in } \Omega \times (0,T)\,, \\ \underline{u} = \underline{u}_\partial & \text{on } \partial\Omega \times (0,T)\,, \\ \cdot \underline{u}_{|t=0} = \underline{u}_0 & \text{on } \Omega\,, \end{cases} \tag{5.1}$$

For the sake of simplicity, we take $\underline{u}_\partial := \underline{0}$ in the presentation of the schemes.

5.1.1 Monolithic approach

Two of the most common second-order time-schemes are undoubtedly the Crank–Nicolson scheme (CN) and the second-order Backward Differentiation Formula (BDF2).

One of the main advantage of the CN scheme is that only the evaluation of the solution at the previous time step is needed (as it is the case for the first-order Implicit Euler used in Chapter 4). Moreover, its analysis is well established (Kim and Moin, 1985; Heywood and Rannacher, 1990; Charnyi *et al.*, 2017).

However, we prefer to focus on the BDF2 time-scheme because of its stronger stability properties. The counterpart is that one needs to store two evaluations of the solution and thus consider an adapted initialization. Avoiding for the moment the spatial discretization and focusing on the Stokes equation ($\xi^{\mathrm{NS}} := 0$ in (5.1)), using the BDF2 scheme for the time discretization leads to

$$\begin{cases} \dfrac{3\underline{u}^n - 4\underline{u}^{n-1} + \underline{u}^{n-2}}{2\Delta t} - \nu \underline{\Delta}\, \underline{u}^n + \underline{\nabla}\, p^n = \underline{f}^n\,, \\ \underline{\nabla} \cdot \underline{u}^n = 0\,. \end{cases} \tag{5.2}$$

Thus, one takes advantage of the solution at the two previous time steps to build a second order approximation of the time-derivative at t^n. Equation (5.2) is valid only if the discrete time nodes t^n are uniformly spaced (Δt is constant): alternative forms are available if Δt varies.

Remark 5.1 - Initialization. The BDF2 time-scheme needs two initial conditions to start: consider for instance $n = 1$ in (5.2), then $\underline{u}^0$ and $\underline{u}^{-1}$ are required. However, the latter is not available. In order to overcome the problem, the most common strategy is to replace at $n = 1$ the BDF2 scheme by an Implicit Euler time discretization (see for instance (4.5)) which needs only $\underline{u}^0$. The complete procedure hence reads:

$$\begin{cases} \dfrac{\underline{u}^n - \underline{u}^{n-1}}{\Delta t} - \nu \underline{\Delta}\, \underline{u}^n + \underline{\nabla}\, p^n = \underline{f}^n\,, \\ \underline{\nabla} \cdot \underline{u}^n = 0\,, \end{cases} \quad n = 1\,,$$
$$\begin{cases} \dfrac{3\underline{u}^n - 4\underline{u}^{n-1} + \underline{u}^{n-2}}{2\Delta t} - \nu \underline{\Delta}\, \underline{u}^n + \underline{\nabla}\, p^n = \underline{f}^n\,, \\ \underline{\nabla} \cdot \underline{u}^n = 0\,, \end{cases} \quad n \geq 2\,. \tag{5.3}$$

One can show, at least for the heat equation, that second-order in time error estimates are recovered in the energy norm.

Another possible strategy for the initialization is the Richardson's extrapolation. Let us briefly detail it for the case at hand. Suppose that $(\underline{u}^{1,i}, p^{1,i})$, $i = 1, 2$, are two approximations obtained by using a first-order time-scheme (e.g. Implicit Euler) with two different time steps, $\Delta t_1 \neq \Delta t_2$. Then, $(\underline{u}^{1,i}, p^{1,i})$, $i = 1, 2$, can be combined in order to

obtained a (final-step) solution $(\underline{u}^1, p^1)$ which is second-order in time. Take for instance $2\Delta t_1 = \Delta t_2 = \Delta t$; then, $\underline{u}^1 := 2\underline{u}^{1,2} - \underline{u}^{1,1}$ is second-order accurate. A similar relation holds for the pressure. Thus, Richardson's extrapolation allows one to recover a sequence of solutions which is second-order in time from the first time step. The price to pay consists in one additional system to be solved initially. ◇

Face-based CDO discretization and algebraic version

All the tools needed in order to recover a face-based CDO (CDO-Fb) discretization of the system (5.2) have been already discussed in Section 3.1.3 and Section 4.1. The CDO problem reads: For all $n = 1, \dots, N$, find $(\widehat{\underline{u}}_{\mathrm{h}}^n, p_{\mathrm{h}}^n) \in \widehat{\underline{\mathcal{U}}}_{\mathrm{h},0} \times \mathcal{P}_{\mathrm{h},*}$ such that

$$\begin{cases} \dfrac{1}{2\Delta t} m(3\underline{u}_{\mathrm{C}}^n - 4\underline{u}_{\mathrm{C}}^{n-1} + \underline{u}_{\mathrm{C}}^{n-2}, \underline{v}_{\mathrm{C}}) + \nu a_{\mathrm{h}}(\widehat{\underline{u}}_{\mathrm{h}}^n, \widehat{\underline{v}}_{\mathrm{h}}^n) + b_{\mathrm{h}}(\widehat{\underline{v}}_{\mathrm{h}}, p_{\mathrm{h}}^n) = l^n(\underline{v}_{\mathrm{C}}) \,, \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad b_{\mathrm{h}}(\widehat{\underline{u}}_{\mathrm{h}}^n, q_{\mathrm{h}}) = 0 \,, \end{cases} \tag{5.4}$$

for all $(\widehat{\underline{v}}_{\mathrm{h}}, q_{\mathrm{h}}) \in \widehat{\underline{\mathcal{U}}}_{\mathrm{h},0} \times \mathcal{P}_{\mathrm{h},*}$.

The local system corresponding to (5.4) has the same structure as (4.12) where the matrices are built as follows:

$$\begin{aligned} \mathbf{A}_{\mathrm{c}} &= \frac{3}{2\Delta t}\mathbf{M}_{\mathrm{c}} + \nu \mathbf{G}_{\mathrm{c}} \,, \\ \mathbf{B}_{\mathrm{c}} &= \mathbf{D}_{\mathrm{c}} \,, \\ \mathbf{F}_{\mathrm{c}}^n &= \mathbf{S}_{\mathrm{c}}^n + \frac{2}{\Delta t}\mathbf{R}_{\mathrm{c}}^{n-1} - \frac{1}{2\Delta t}\mathbf{R}_{\mathrm{c}}^{n-2} \,. \end{aligned} \tag{5.5}$$

Remark 5.2 - Energy balance. We cannot proceed as in the proof of Lemma 4.3 to recover an equation for the kinetic energy associated with $\widehat{\underline{u}}_{\mathrm{h}}^n$. However, we believe that one can proceed as in Ern and Guermond (2020) to recover a stability result on the kinetic energy, see in particular Lemma 36.1 therein for the heat equation. ◇

Convection treatments

The convection treatments presented in Section 4.3 need to be slightly adapted so that second-order time accuracy is maintained. All the techniques presented below hinge on the following extrapolation result. Consider a smooth function $g(t)$ and a sequence of equispaced time nodes $\{t^n := n\Delta t\}_{n=0,\dots,N}$. Using Taylor expansions at t^n, we readily see that

$$\widetilde{g}^n := 2g^{n-1} - g^{n-2} \tag{5.6}$$

is a second order approximation of g^n.

Explicit convection Applying (5.6) to $g := (\underline{u} \cdot \underline{\nabla})\underline{u}$ gives

$$2(\underline{u}^{n-1} \cdot \underline{\nabla})\underline{u}^{n-1} - (\underline{u}^{n-2} \cdot \underline{\nabla})\underline{u}^{n-2} \tag{5.7}$$

as a second-order explicit extrapolation of the convection operator. The CDO-Fb problem then reads: For $n = 1, \dots, N$, find $(\widehat{\underline{u}}_{\mathrm{h}}^n, p_{\mathrm{h}}^n) \in \widehat{\underline{\mathcal{U}}}_{\mathrm{h},0} \times \mathcal{P}_{\mathrm{h},*}$ solving

$$\begin{cases} \dfrac{1}{2\Delta t} m(3\underline{u}_{\mathrm{C}}^n - 4\underline{u}_{\mathrm{C}}^{n-1} + \underline{u}_{\mathrm{C}}^{n-2}, \underline{v}_{\mathrm{C}}) + \nu a_{\mathrm{h}}(\widehat{\underline{u}}_{\mathrm{h}}^n, \widehat{\underline{v}}_{\mathrm{h}}) + b_{\mathrm{h}}(\widehat{\underline{v}}_{\mathrm{h}}, p_{\mathrm{h}}^n) \\ \qquad = l^n(\underline{v}_{\mathrm{C}}) - \Big(2t_{\mathrm{h}}(\widehat{\underline{u}}_{\mathrm{h}}^{n-1}; \widehat{\underline{u}}_{\mathrm{h}}^{n-1}, \widehat{\underline{v}}_{\mathrm{h}}) - t_{\mathrm{h}}(\widehat{\underline{u}}_{\mathrm{h}}^{n-2}; \widehat{\underline{u}}_{\mathrm{h}}^{n-2}, \widehat{\underline{v}}_{\mathrm{h}})\Big) \,, \\ b_{\mathrm{h}}(\widehat{\underline{u}}_{\mathrm{h}}^n, q_{\mathrm{h}}) = 0 \,, \end{cases} \tag{5.8}$$

for all $\widehat{\underline{v}}_{\mathrm{h}} \in \widehat{\underline{\mathcal{U}}}_{\mathrm{h},0}$ and all $q_{\mathrm{h}} \in \mathcal{P}_{\mathrm{h},*}$.

Linearized convection This time, the target term is only the convection field. Hence, applying (5.6) to $\underline{u}$, approximating the convection term at t^n in the NSE by

$$(\widetilde{\underline{u}}^n \cdot \underline{\nabla})\underline{u}^n, \qquad \widetilde{\underline{u}}^n := 2\underline{u}^{n-1} - \underline{u}^{n-2}, \tag{5.9}$$

and considering a CDO-Fb discretization, one gets: For $n = 1, \ldots, N$, find $(\widehat{\underline{u}}_{\mathrm{h}}^n, p_{\mathrm{h}}^n) \in \widehat{\underline{\mathcal{U}}}_{\mathrm{h},0} \times \mathcal{P}_{\mathrm{h},*}$ solving

$$\begin{cases} \dfrac{1}{2\Delta t} m(3\underline{u}_{\mathrm{C}}^n - 4\underline{u}_{\mathrm{C}}^{n-1} + \underline{u}_{\mathrm{C}}^{n-2}, \underline{v}_{\mathrm{C}}) + \nu a_{\mathrm{h}}(\widehat{\underline{u}}_{\mathrm{h}}^n, \widehat{\underline{v}}_{\mathrm{h}}) + t_{\mathrm{h}}(\widetilde{\widehat{\underline{u}}}_{\mathrm{h}}^{n-1}; \widehat{\underline{u}}_{\mathrm{h}}^n, \widehat{\underline{v}}_{\mathrm{h}}) + b_{\mathrm{h}}(\widehat{\underline{v}}_{\mathrm{h}}, p_{\mathrm{h}}^n) = l^n(\underline{v}_{\mathrm{C}}), \\ b_{\mathrm{h}}(\widehat{\underline{u}}_{\mathrm{h}}^n, q_{\mathrm{h}}) = 0, \end{cases} \tag{5.10}$$

for all $\widehat{\underline{v}}_{\mathrm{h}} \in \widehat{\underline{\mathcal{U}}}_{\mathrm{h},0}$ and all $q_{\mathrm{h}} \in \mathcal{P}_{\mathrm{h},*}$. We used $\widetilde{\widehat{\underline{u}}}_{\mathrm{h}}^n := 2\widehat{\underline{u}}_{\mathrm{h}}^{n-1} - \widehat{\underline{u}}_{\mathrm{h}}^{n-2}$.

Picard iteration Being an iterative method, the Picard algorithm, whenever it converges, should automatically recover second-order time accuracy. Thus no special modifications are needed. The CDO-Fb formulation then reads: For $n = 1, \ldots, N$, iterate on $k \geq 1$ until convergence: find $(\widehat{\underline{u}}_{\mathrm{h}}^{n,k}, p_{\mathrm{h}}^{n,k}) \in \widehat{\underline{\mathcal{U}}}_{\mathrm{h},0} \times \mathcal{P}_{\mathrm{h},*}$ such that

$$\begin{cases} \dfrac{1}{2\Delta t} m(3\underline{u}_{\mathrm{C}}^{n,k} - 4\underline{u}_{\mathrm{C}}^{n-1,\infty} + \underline{u}_{\mathrm{C}}^{n-2,\infty}, \underline{v}_{\mathrm{C}}) + \nu a_{\mathrm{h}}(\widehat{\underline{u}}_{\mathrm{h}}^{n,k}, \widehat{\underline{v}}_{\mathrm{h}}) \\ \qquad\qquad + t_{\mathrm{h}}(\widehat{\underline{u}}_{\mathrm{h}}^{n,k-1}; \widehat{\underline{u}}_{\mathrm{h}}^{n,k}, \widehat{\underline{v}}_{\mathrm{h}}) + b_{\mathrm{h}}(\widehat{\underline{v}}_{\mathrm{h}}, p_{\mathrm{h}}^{n,k}) = l^n(\underline{v}_{\mathrm{C}}), \\ b_{\mathrm{h}}(\widehat{\underline{u}}_{\mathrm{h}}^{n,k}, q_{\mathrm{h}}) = 0, \end{cases} \tag{5.11}$$

for all $\widehat{\underline{v}}_{\mathrm{h}} \in \widehat{\underline{\mathcal{U}}}_{\mathrm{h},0}$ and all $q_{\mathrm{h}} \in \mathcal{P}_{\mathrm{h},*}$. Recall that $\widehat{\underline{u}}_{\mathrm{h}}^{n-1,\infty}$ (respectively $\widehat{\underline{u}}_{\mathrm{h}}^{n-2,\infty}$) denotes the solution given by the Picard algorithm at the time step $n-1$ (resp. $n-2$). Our numerical experiments indicate that the Picard iteration might take a lot of time to converge: using $\widehat{\underline{u}}_{\mathrm{h}}^{n,0} \widetilde{\widehat{\underline{u}}}_{\mathrm{h}}^n$ as initialization might improve its performances.

Remark 5.3 - Energy balance. A stability result for the BDF2 time-scheme similar to the one presented in Ern and Guermond (2020, Lemma 67.1) and discussed Remark 5.2 can be obtained whenever the convection treatment is dissipative (or neutral with respect to the kinetic energy) since the analysis requires testing the momentum equation with $\widehat{\underline{v}}_{\mathrm{h}} := \widehat{\underline{u}}_{\mathrm{h}}^n$ for (5.10) or $\widehat{\underline{v}}_{\mathrm{h}} := \widehat{\underline{u}}_{\mathrm{h}}^{n,k}$ for (5.11). This is the case for the Picard algorithm and the linearized convection (one proceeds as in the proofs of Lemmas 4.8 and 4.11). ◇

5.1.2 Artificial Compressibility

We recall that the AC method hinges on a perturbation of the mass balance which enables one to decouple velocity and pressure. In Chapter 4, in the case of a Stokes problem, a first-order time-scheme has been used leading to (see (4.24))

$$\begin{cases} \dfrac{\underline{u}^n - \underline{u}^{n-1}}{\Delta t} - \nu(\underline{\Delta}\,\underline{u}^n + \eta \underline{\nabla}\,\underline{\nabla} \cdot \underline{u}^n) + \underline{\nabla}\, p^{n-1} = \underline{f}^n, \\ p^n = p^{n-1} - \nu\eta \underline{\nabla} \cdot \underline{u}^n. \end{cases} \tag{5.12}$$

A promising feature of the AC strategy is the possibility of devising time-schemes of arbitrary order of convergence (Guermond and Minev, 2015). This is achieved via a bootstrapping technique: in short, by solving k accurately-designed equations similar to (5.12) per time step, one can recover k-th order convergence in time.

Combined with the BDF technique to handle the time derivative, the bootstrapping technique leads to the following procedure for the Stokes equations: For $n \geq 1$, find $(\underline{u}_1^n, p_1^n)$

such that

$$\begin{cases} \dfrac{\underline{u}_1^n - \underline{u}_1^{n-1}}{\Delta t} - \nu\left(\underline{\Delta}\,\underline{u}_1^n + \eta\underline{\nabla}\,\underline{\nabla}\cdot\underline{u}_1^n\right) = \underline{f}^n - \underline{\nabla}\,p_1^{n-1}\,, & (5.13a) \\ p_1^n = p_1^{n-1} - \nu\eta\underline{\nabla}\cdot\underline{u}_1^n\,, \quad \delta p_1^n := p_1^n - p_1^{n-1}\,, & (5.13b) \end{cases}$$

and for $n \geq 2$, find $(\underline{u}_2^n, p_2^n)$ such that

$$\begin{cases} \dfrac{3\underline{u}_2^n - 4\underline{u}_2^{n-1} + \underline{u}_2^{n-2}}{2\Delta t} - \nu\left(\underline{\Delta}\,\underline{u}_2^n + \eta\underline{\nabla}\,\underline{\nabla}\cdot\underline{u}_2^n\right) = \underline{f}^n - \underline{\nabla}\left(p_2^{n-1} + \delta p_1^n\right), & (5.13c) \\ p_2^n = p_2^{n-1} + \delta p_1^n - \nu\eta\underline{\nabla}\cdot\underline{u}_2^n\,. & (5.13d) \end{cases}$$

Here, $\underline{u}_i^n$ (respectively p_i^n) is an approximation of order i of the velocity $\underline{u}^n$ (resp. pressure p^n). Notice that Eqs. (5.13a) and (5.13b) have the same structure as the equations used in the first-order case (5.12).

Remark 5.4 - Pressure mass matrix. We observe that, as explained in Remark 4.4, although usually a mass matrix related to the pressure is sometimes introduced in the finite element setting in order to deal with (5.13b) and (5.13d), in the CDO-Fb framework those equations simply boil down to an update step since the pressure and the divergence are cell-wise constant. ◇

Remark 5.5 - Alternative formulation. Instead of using a BDF scheme and recovering (5.13), Guermond and Minev (2015) propose also a *defect correction* technique which involves only first-order time-derivative discretizations while still ensuring second-order convergence in time. The defect correction technique relies on a Taylor expansion in which the derivatives of order higher than one are discretized as well. Applying this technique to the AC method leads to replacing Eqs. (5.13c) and (5.13d) with

$$\begin{cases} \dfrac{\underline{u}_2^n - \underline{u}_2^{n-1}}{\Delta t} - \nu\left(\underline{\Delta}\,\underline{u}_2^n + \eta\underline{\nabla}\,\underline{\nabla}\cdot\underline{u}_2^n\right) = -\underline{\nabla}\left(p_2^{n-1} + \dfrac{\delta p_1^n}{\Delta t}\right) - \dfrac{1}{2}\delta^2\underline{u}_1^{n+1}\,, & (5.14a) \\ p_2^n = p_2^{n-1} + \dfrac{\delta p_1^n}{\Delta t} - \nu\eta\underline{\nabla}\cdot\underline{u}_2^n\,,\ \delta\underline{u}_1^{n+1} := \dfrac{\underline{u}_1^{n+1} - \underline{u}_1^n}{\Delta t}\,,\ \delta^2\underline{u}_1^{n+1} := \dfrac{\delta\underline{u}_1^{n+1} - \delta\underline{u}_1^n}{\Delta t}\,. & (5.14b) \end{cases}$$

The numerical results presented in Guermond and Minev (2015) indicate that this defect correction version Eqs. (5.13a), (5.13b), (5.14a) and (5.14b) seems to be more stable than Eqs. (5.13a) to (5.13d) when used in the context of NSE and explicit convection. Yet, we decided to keep the BDF-based version in order to be consistent with what has been done for the monolithic approach. ◇

Convection treatments

For (5.13) to remain of second order, when convection is considered, one of the techniques presented in Section 4.2.2 should be applied to (5.13a), and one of the techniques discussed in Section 5.1.1 to (5.13c), preferably the same method for both equations.

Let us stress again that neither the Picard algorithm nor the linearized convection are usually considered whenever dealing with NSE and the AC method. The main reason is that this framework aims at being as efficient as possible, even starting from the Stokes problem. Secondly, one tries to obtain a final linear system as simple as possible to solve, thus avoiding peculiar numerical procedures: this is why one tends to avoid the Picard iterations, but also the linearization of the convection, which would lead to a nonsymmetric linear system. For instance, in Guermond and Minev (2015, Remark 5.3), the authors advocate the use of an explicit convection treatment describing it as the natural way of moving from the Stokes to the NSE. However, for the sake of completeness, in addition to the explicit convection, we still test the linearized convection. The resulting CDO-Fb schemes are given below.

Explicit convection Adding an explicit convection to (5.13) as suggested in Guermond and Minev (2015, Remark 5.4) leads to the following procedure: For $n \geq 1$, find $(\underline{\widehat{u}}_{1,\mathrm{h}}^n, p_{1,\mathrm{h}}^n) \in \underline{\widehat{\mathcal{U}}}_{\mathrm{h},0} \times \mathcal{P}_{\mathrm{h},*}$ such that

$$\begin{cases} \dfrac{1}{\Delta t} m(\underline{u}_{1,\mathrm{C}}^n - \underline{u}_{1,\mathrm{C}}^{n-1}, \underline{v}_{\mathrm{C}}) + \nu \left(a_{\mathrm{h}}(\underline{\widehat{u}}_{1,\mathrm{h}}^n, \underline{\widehat{v}}_{\mathrm{h}}) + \eta d_{\mathrm{h}}(\underline{\widehat{u}}_{1,\mathrm{h}}^n, \underline{\widehat{v}}_{\mathrm{h}}) \right) \\ \qquad\qquad = l^n(\underline{v}_{\mathrm{C}}) - b_{\mathrm{h}}(\underline{\widehat{v}}_{\mathrm{h}}, p_{1,\mathrm{h}}^{n-1}) - t_{\mathrm{h}}(\underline{\widehat{u}}_{1,\mathrm{h}}^{n-1}; \underline{\widehat{u}}_{1,\mathrm{h}}^{n-1}, \underline{\widehat{v}}_{\mathrm{h}}) \,, & (5.15\mathrm{a}) \\ p_{1,\mathrm{h}}^n = p_{1,\mathrm{h}}^{n-1} - \nu\eta \mathrm{D}_{\mathrm{h}}(\underline{\widehat{u}}_{1,\mathrm{h}}^n) \,, \quad \delta p_{1,\mathrm{h}}^n := p_{1,\mathrm{h}}^n - p_{1,\mathrm{h}}^{n-1} \,, & (5.15\mathrm{b}) \end{cases}$$

for all $\underline{\widehat{v}}_{\mathrm{h}} \in \underline{\widehat{\mathcal{U}}}_{\mathrm{h},0}$ and all $q_{\mathrm{h}} \in \mathcal{P}_{\mathrm{h},*}$; and, for $n \geq 2$, find $(\underline{\widehat{u}}_{2,\mathrm{h}}^n, p_{2,\mathrm{h}}^n) \in \underline{\widehat{\mathcal{U}}}_{\mathrm{h},0} \times \mathcal{P}_{\mathrm{h},*}$ such that

$$\begin{cases} \dfrac{1}{2\Delta t} m(3\underline{u}_{2,\mathrm{C}}^n - 4\underline{u}_{2,\mathrm{C}}^{n-1} + \underline{u}_{2,\mathrm{C}}^{n-2}, \underline{v}_{\mathrm{C}}) + +\nu \left(a_{\mathrm{h}}(\underline{\widehat{u}}_{2,\mathrm{h}}^n, \underline{\widehat{v}}_{\mathrm{h}}) + \eta d_{\mathrm{h}}(\underline{\widehat{u}}_{2,\mathrm{h}}^n, \underline{\widehat{v}}_{\mathrm{h}}) \right) \\ \quad = l^n(\underline{v}_{\mathrm{C}}) - b_{\mathrm{h}}(\underline{\widehat{v}}_{\mathrm{h}}, p_{2,\mathrm{h}}^{n-1} + \delta p_{1,\mathrm{h}}^n) - \left(2t_{\mathrm{h}}(\underline{\widehat{u}}_{2,\mathrm{h}}^{n-1}; \underline{\widehat{u}}_{2,\mathrm{h}}^{n-1}, \underline{\widehat{v}}_{\mathrm{h}}) - t_{\mathrm{h}}(\underline{\widehat{u}}_{2,\mathrm{h}}^{n-2}; \underline{\widehat{u}}_{2,\mathrm{h}}^{n-2}, \underline{\widehat{v}}_{\mathrm{h}}) \right) \,, & (5.15\mathrm{c}) \\ p_{2,\mathrm{h}}^n = p_{2,\mathrm{h}}^{n-1} + \delta p_{1,\mathrm{h}}^n - \nu\eta \mathrm{D}_{\mathrm{h}}(\underline{\widehat{u}}_{2,\mathrm{h}}^n) \,, & (5.15\mathrm{d}) \end{cases}$$

for all $\underline{\widehat{v}}_{\mathrm{h}} \in \underline{\widehat{\mathcal{U}}}_{\mathrm{h},0}$ and all $q_{\mathrm{h}} \in \mathcal{P}_{\mathrm{h},*}$.

Linearized convection This time, we add a convection term using in (5.13c) the extrapolated velocity $\underline{\widetilde{\widehat{u}}}_{2,\mathrm{h}}^n := 2\underline{\widehat{u}}_{2,\mathrm{h}}^{n-1} - \underline{\widehat{u}}_{2,\mathrm{h}}^{n-2}$ as the convection field. We obtain the following procedure: For $n \geq 1$, find $(\underline{\widehat{u}}_{1,\mathrm{h}}^n, p_{1,\mathrm{h}}^n) \in \underline{\widehat{\mathcal{U}}}_{\mathrm{h},0} \times \mathcal{P}_{\mathrm{h},*}$ such that

$$\begin{cases} \dfrac{1}{\Delta t} m(\underline{u}_{1,\mathrm{C}}^n - \underline{u}_{1,\mathrm{C}}^{n-1}, \underline{v}_{\mathrm{C}}) + \nu \left(a_{\mathrm{h}}(\underline{\widehat{u}}_{1,\mathrm{h}}^n, \underline{\widehat{v}}_{\mathrm{h}}) + \eta d_{\mathrm{h}}(\underline{\widehat{u}}_{1,\mathrm{h}}^n, \underline{\widehat{v}}_{\mathrm{h}}) \right) \\ \qquad\qquad + t_{\mathrm{h}}(\underline{\widehat{u}}_{1,\mathrm{h}}^{n-1}; \underline{\widehat{u}}_{1,\mathrm{h}}^n, \underline{\widehat{v}}_{\mathrm{h}}) = l^n(\underline{v}_{\mathrm{C}}) - b_{\mathrm{h}}(\underline{\widehat{v}}_{\mathrm{h}}, p_{1,\mathrm{h}}^{n-1}) \,, & (5.16\mathrm{a}) \\ p_{1,\mathrm{h}}^n = p_{1,\mathrm{h}}^{n-1} - \nu\eta \mathrm{D}_{\mathrm{h}}(\underline{\widehat{u}}_{1,\mathrm{h}}^n) \,, \quad \delta p_{1,\mathrm{h}}^n := p_{1,\mathrm{h}}^n - p_{1,\mathrm{h}}^{n-1} \,, & (5.16\mathrm{b}) \end{cases}$$

for all $\underline{\widehat{v}}_{\mathrm{h}} \in \underline{\widehat{\mathcal{U}}}_{\mathrm{h},0}$ and all $q_{\mathrm{h}} \in \mathcal{P}_{\mathrm{h},*}$; and, for $n \geq 2$, find $(\underline{\widehat{u}}_{2,\mathrm{h}}^n, p_{2,\mathrm{h}}^n) \in \underline{\widehat{\mathcal{U}}}_{\mathrm{h},0} \times \mathcal{P}_{\mathrm{h},*}$ such that

$$\begin{cases} \dfrac{1}{2\Delta t} m(3\underline{u}_{2,\mathrm{C}}^n - 4\underline{u}_{2,\mathrm{C}}^{n-1} + \underline{u}_{2,\mathrm{C}}^{n-2}, \underline{v}_{\mathrm{C}}) + +\nu \left(a_{\mathrm{h}}(\underline{\widehat{u}}_{2,\mathrm{h}}^n, \underline{\widehat{v}}_{\mathrm{h}}) + \eta d_{\mathrm{h}}(\underline{\widehat{u}}_{2,\mathrm{h}}^n, \underline{\widehat{v}}_{\mathrm{h}}) \right) \\ \qquad\qquad + t_{\mathrm{h}}(\underline{\widetilde{\widehat{u}}}_{2,\mathrm{h}}^{n-1}; \underline{\widehat{u}}_{2,\mathrm{h}}^n, \underline{\widehat{v}}_{\mathrm{h}}) = l^n(\underline{v}_{\mathrm{C}}) - b_{\mathrm{h}}(\underline{\widehat{v}}_{\mathrm{h}}, p_{2,\mathrm{h}}^{n-1} + \delta p_{1,\mathrm{h}}^n) \,, & (5.16\mathrm{c}) \\ p_{2,\mathrm{h}}^n = p_{2,\mathrm{h}}^{n-1} + \delta p_{1,\mathrm{h}}^n - \nu\eta \mathrm{D}_{\mathrm{h}}(\underline{\widehat{u}}_{2,\mathrm{h}}^n) \,, & (5.16\mathrm{d}) \end{cases}$$

for all $\underline{\widehat{v}}_{\mathrm{h}} \in \underline{\widehat{\mathcal{U}}}_{\mathrm{h},0}$ and all $q_{\mathrm{h}} \in \mathcal{P}_{\mathrm{h},*}$.

5.2 Numerical results: Stokes equations

The test case presented in Section 4.4 is considered again here in order to evaluate the second-order time-schemes presented in Section 5.1. We recall that the problem at hand consists in the unsteady 3D Stokes equations with the following analytic solution:

$$\begin{cases} \underline{u}_{\mathrm{UTGV}}(x,y,z) := \alpha(t)\underline{u}_{3\mathrm{TGV}}(x,y,z) \,, \\ p_{\mathrm{UTGV}}(x,y,z) := \alpha(t)p_{3\mathrm{TGV}}(x,y,z) \,, \\ \qquad\qquad \alpha(t) := \sin(1.7\,\pi t + \tfrac{\pi}{5}) \,, \\ \underline{u}_{3\mathrm{TGV}}(x,y,z) := \begin{bmatrix} -2\cos(2\pi x)\sin(2\pi y)\sin(2\pi z) \\ \sin(2\pi x)\cos(2\pi y)\sin(2\pi z) \\ \sin(2\pi x)\sin(2\pi y)\cos(2\pi z) \end{bmatrix} \,, \\ p_{3\mathrm{TGV}}(x,y,z) := -6\pi\sin(2\pi x)\sin(2\pi y)\sin(2\pi z) \,. \end{cases} \tag{5.17}$$

The domain is $\Omega := [0,1]^3$, the time limit is $T := 2$ and the viscosity is set to $\nu := 1$. The source term, presented in (4.51), is

$$\begin{cases} \underline{f}_{\mathrm{UTGV}}(x,y,z) := \underline{f}_{\mathrm{3TGV}}(x,y,z) + 1.7\,\pi\cos(1.7\,\pi t + \frac{\pi}{5})\underline{u}_{\mathrm{3TGV}}(x,y,z)\,, \\ \underline{f}_{\mathrm{3TGV}}(x,y,z) := [-36\pi^2\cos(2\pi x)\sin(2\pi y)\sin(2\pi z),\ 0,\ 0]^T\,. \end{cases} \tag{5.18}$$

First, the convergence rates of the time errors are measured. A comparison between the monolithic approach and the AC technique is made with a special attention on the performances. Finally, the advantages and disadvantages of the second-order time-schemes with respect to the first-order time-schemes are studied.

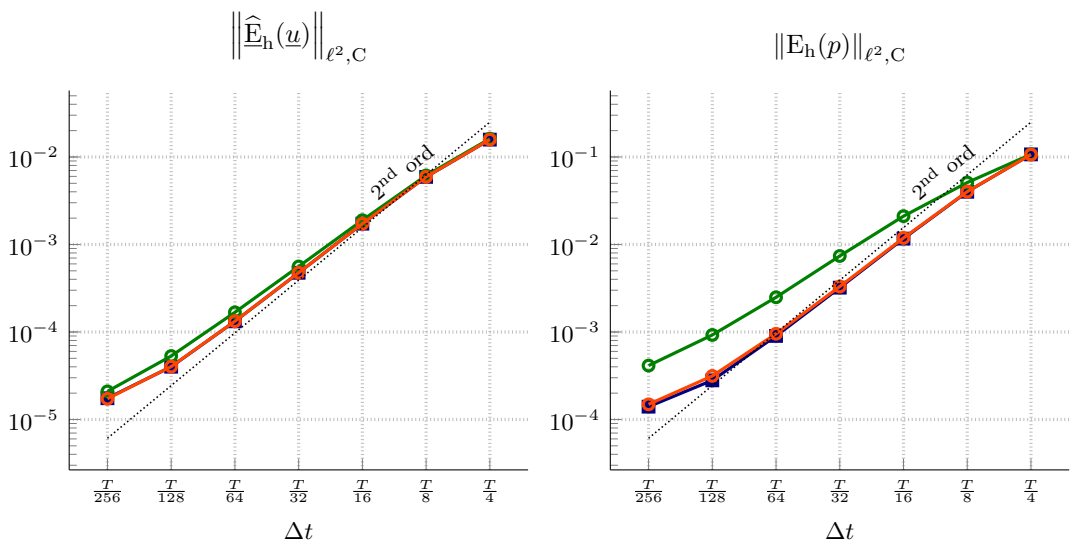
We recall that, among the several combinations of coupling, linear solver, and preconditioners, only three have been selected as being potentially the most efficient ones, namely, the monolithic approach solved with a GKB procedure (with parameter $\gamma := 0$) with a Conjugate Gradient iterative solver and a K-cycle preconditioner (MONO(0) for short), the AC(η) technique with parameter η set to 10 or 100 and solved with a Jacobi-preconditioned CG (AC(10) or AC(100) for short).

Remark 5.6 - Initialization. The initialization of the BDF2 time-scheme has been performed by means of an Implicit Euler iteration, see (5.3), so that it is more coherent with the AC scheme, see (5.13). Hence, the convergence rates for both strategies might be affected by this non-optimal start, especially if only a few time steps are considered. ◇

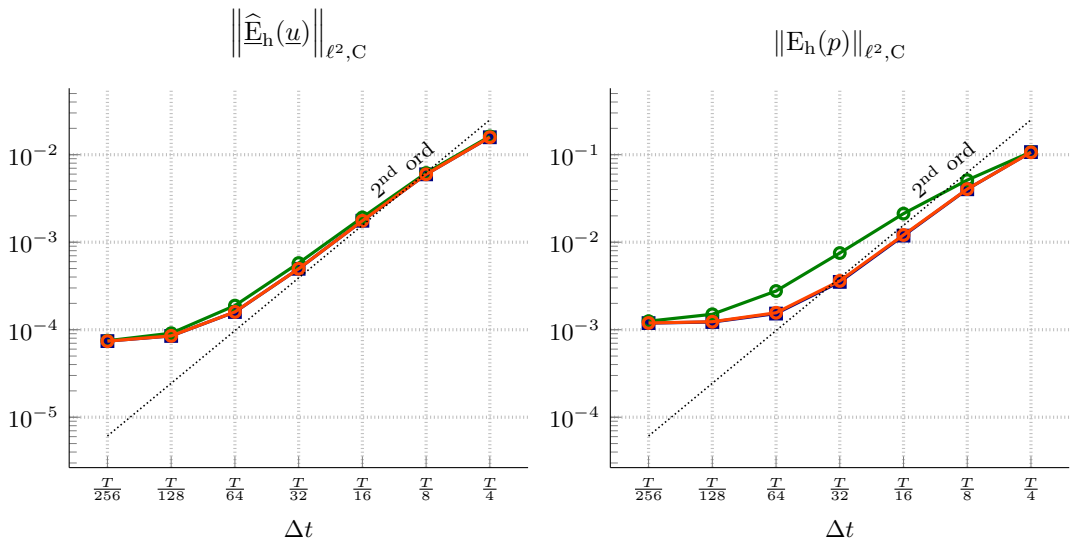
5.2.1 Comparison: Monolithic approach vs. Artificial Compressibility

Figure 5.1 collects the results obtained for the 3D unsteady TGV test case for the three strategies (AC(10), AC(100) and MONO(0)) at hand and for two different meshes, a Cartesian mesh composed of 256^3 cells and a mesh composed of prisms with polygonal bases (differently from Section 4.4.3, we do not consider here the CheckerBoard mesh for which we obtained stagnation due to the spatial error). Only the discrete $\ell^2(L^2)$-like velocity and pressure space-time errors are reported in Fig. 5.1. Insights about the $\ell^2(H^1)$ velocity error are given later in Tables 5.4 to 5.6 for the Cartesian mesh only, for which the results were less polluted (with respect to the polyhedral mesh) by the spatial error. Generally speaking, second-order convergence in time is recovered for all the considered errors, that is the discrete L^2-like norm of the velocity and the pressure. Some sub-optimal rates are observed at the two ends of the curves. The inaccuracy observed for the coarsest values of the time step is due to the non-optimal initialization (see Remark 5.6). On the other end of the curve, the stagnation is due to the spatial error dominating the temporal one. This is clear by observing the results on the polygonal meshes in Fig. 5.1b where a plateau forms starting from the third finest time step value. Moreover, focusing on the pressure errors, AC(10), —o—, which was the least accurate in the first-order setting, now reaches the level of the other two strategies, hence confirming that the spatial error dominates. The coupling does not seem to have a sizable impact on the velocity errors, at least for the three considered strategies (recall that a gap have been observed between the AC(10) and the monolithic results with the first-order schemes, see Fig. 4.1 for instance). However, the coupling still has an influence on the pressure errors, see the right column of Fig. 5.1.

Let us now move on to the efficiency results by comparing the three strategies. The results are presented in Fig. 5.2. One notices that AC(10), —o—, is the most efficient strategy: for a given error threshold, it achieves it in less computational time. Differently to what was observed with the first-order schemes (cf. Fig. 4.2), the monolithic —■— and the AC(100) —o— strategies are really close, with the latter being the least efficient one. The positions actually switched with respect to the results obtained with first-order schemes. This is due to the fact that with the monolithic approach and the BDF2 time-scheme (5.2), only one system



(a) Mesh: Cartesian 256^3.

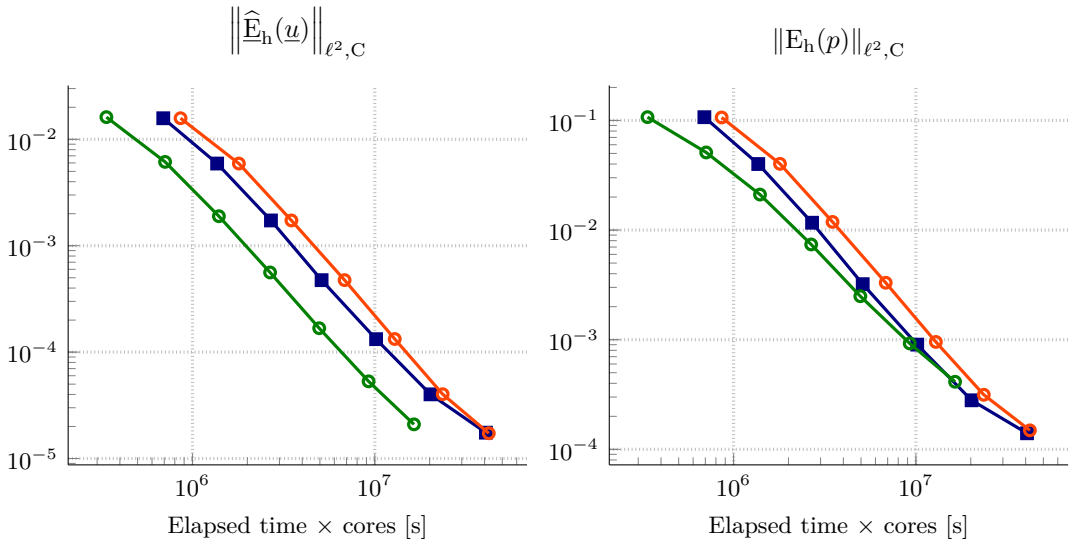


(b) Mesh: PrG160.

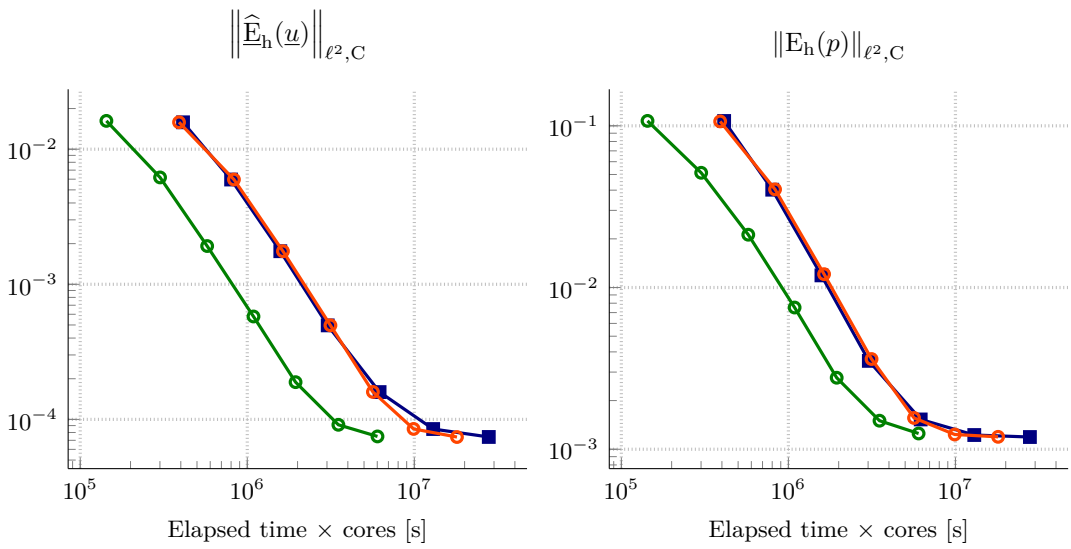
Method: AC(10)-bootstrap AC(100)-bootstrap MONO(0)-BDF2

Figure 5.1 – Unsteady Stokes problem. 3D TGV solution (5.17), $T = 2$. Convergence in time. Left: velocity L^2-error, right: pressure L^2-error. Top: Cartesian mesh, bottom: prismatic mesh with polygonal bases.

resolution per time step is needed (as it was the case for the first-order scheme, too). On the other hand, two systems need to be solved per time step in the AC-bootstrap technique, whereas only one was necessary with the standard first-order AC method. Since the linear systems resulting for the first- and second-order schemes feature the same operators, this results in an execution time which doubles from the first- to the second-order time-schemes. In Fig. 5.2, we did not show the results obtained for the error on the gradient of the velocity, which were affected by the spatial error. For this error, the conclusions are similar to those regarding the $\ell^2(L^2)$ error of the velocity itself.



(a) Mesh: Cartesian 256^3.



(b) Mesh: PrG160.

Method: AC(10)-bootstrap AC(100)-bootstrap MONO(0)-BDF2

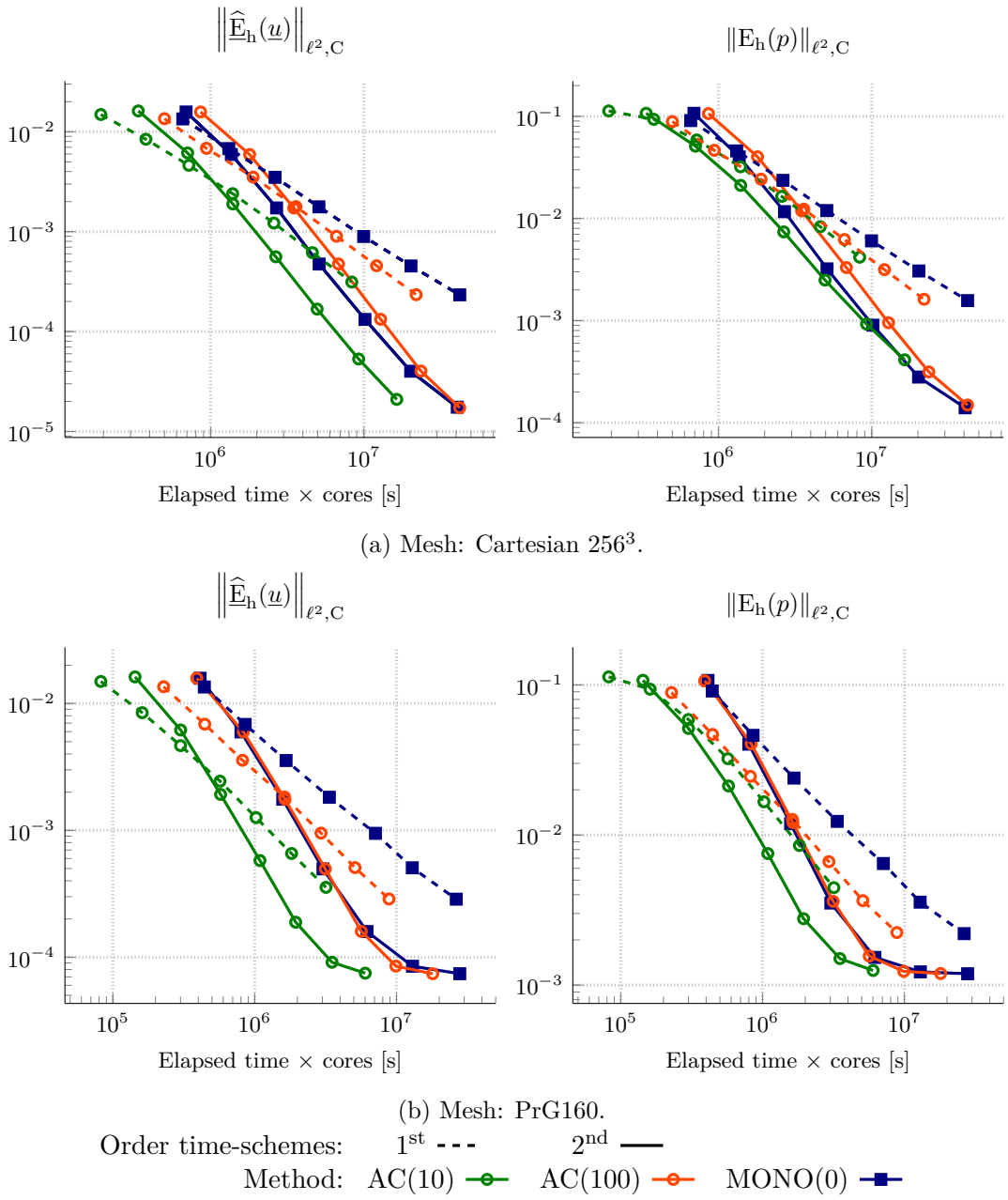
Figure 5.2 – Unsteady Stokes problem. 3D TGV solution (5.17), $T = 2$. Cost vs. accuracy. Left: velocity L^2-error, right: pressure L^2-error. Top: Cartesian mesh, bottom: prismatic mesh with polygonal bases.

5.2.2 Comparison: first- vs. second-order time discretizations

We continue here the analysis of the efficiency of the second-order time-schemes by investigating the differences between the first- and second-order time-schemes. In Fig. 5.3 we compare the results of Figs. 4.2 and 4.6 (see also Tables 4.5 to 4.10) to those of Fig. 5.2.

A detailed summary is given in Tables 5.4 to 5.9, which are collected in Section 5.4.1. As it was pointed out in Section 5.2.1, the computational times double for the bootstrapping technique with respect to the standard first-order time-scheme, while the times remain almost unchanged when moving from the Implicit Euler to the BDF2 with the monolithic approach. However, this additional computational effort is always compensated by a significant gain in the error results: those obtained with the second-order schemes are even 10

times smaller than those computed with the first-order schemes (see for instance the last lines of Tables 5.4 to 5.6).



(a) Mesh: Cartesian 256^3.

(b) Mesh: PrG160.

Figure 5.3 – Unsteady Stokes problem. 3D TGV solution (5.17), $T = 2$. Cost vs. accuracy. Left: velocity L^2-error, right: pressure L^2-error. Legend: AC(10)-bootstrap, AC(100)-bootstrap, MONO(0)-BDF2. Results obtained with second-order schemes (cf. Fig. 5.2) are in solid lines, those obtained with first-order schemes (cf. Figs. 4.2 and 4.6) in dashed lines.

The results presented here advocate for the usage of the second-order time-schemes over the first-order time-schemes. For a given coupling, in general it is more efficient to consider second-order schemes than first-order ones. If one would like to draw more general conclusions (concerning for instance the best combination of coupling technique, linear solver, and time-discretization), some uncertainties still remain. For instance, the arbitrary parameter η in the AC technique may affect significantly both the accuracy and the necessary com-

putational effort: if well chosen, η could lead to an efficient method which can compete with the saddle-point approach. The range of the values ensuring this may not be so wide, especially when the bootstrapping technique is performed. Moreover, we have seen that the linear solvers play a major role in the performance of the overall resolution procedure. Thus, the results may be different if other types of linear solvers are chosen and if the stopping criteria and thresholds of the iterative procedures change. This being said, a reasonable guess, no matter what linear solvers one has at his disposal, would be to consider a second-order AC-bootstrap scheme and a fairly small arbitrary parameter η.

5.3 Numerical results: Navier–Stokes equations

We address here the same problem as in Section 4.5, namely the Taylor–Green Vortex (TGV) (Taylor and Green, 1937). It consists in a 2D analytical solution of the NSE

$$\begin{cases} \underline{u}_{\mathrm{TGV}}(x,y) := \exp(-2\nu t) \begin{bmatrix} \sin(x)\cos(y) \\ -\cos(x)\sin(y) \end{bmatrix}, \\ p_{\mathrm{TGV}}(x,y) := \dfrac{1}{4}\exp(-4\nu t)(\cos(2x)+\cos(2y)), \end{cases} \tag{5.19}$$

in the square domain $\Omega := [0, 2\pi]^2$. We consider here two values for the viscosity: $\nu := 0.3$, and $\nu := 0.03$. Setting $L = 1$ and $U = 1$, the two viscosity values lead to $Re \approx 3$ and $Re \approx 33$ respectively. The final time is $T := 4$, if $Re \approx 3$, or $T := 40$ otherwise. The values has been chosen so that $\exp(-2\nu t) \approx \frac{1}{10}$.

As it was done in Section 4.5, we compare the monolithic approach and the AC method. For this latter, we set $\eta := 10\,Re$.

5.3.1 Convergence in time

We begin by evaluating the orders of convergence in time of the implementation of the BDF2 time-scheme and the bootstrapping technique proposed in Section 5.1.

The results are shown in Fig. 5.4. Recall that for the bootstrapping technique, we do not consider the Picard algorithm. Generally speaking, the expected second order of convergence is recovered both for the monolithic approach with the BDF2 time-scheme and for the second-order bootstrapping technique with the AC method. However, the error on the gradient of the velocity (middle) suffers from early stagnation, especially for $Re \approx 33$ (right column). The issue seems to be due to the spatial error becoming dominant, and it is probably increased by the lack of pressure-robustness of our discretization. Rates slightly lower-than-second are observed also for the pressure errors (bottom).

As for the first-order time-schemes, one can observe the influence of the convection treatment on the pressure errors (recall that for the TGV solution (5.19), the convection term compensates the pressure gradient). Stability issues prevent us from recovering data with the explicit convection treatment for $Re \approx 33$. Notice that for the Implicit Euler discretization, see Fig. 4.8, we used a more coarser grid which facilitates the stability. Yet the CFL condition seems to be more stringent with second-order time-schemes than with first-order time-schemes. A more detailed stability analysis is presented in Section 5.3.3.

5.3.2 Convection treatments and dissipativity

We follow what has been done in Section 4.5.2 and investigate the following quantities:

$$\mathfrak{d}\mathcal{E}^n_{\mathrm{kin,h}} := \frac{\mathcal{E}_{\mathrm{kin,h}}(\widehat{\underline{v}}^n_{\mathrm{h}}) - \mathcal{E}_{\mathrm{kin,h}}(\widehat{\underline{v}}^{n-1}_{\mathrm{h}})}{\Delta t}, \qquad n = 1, \ldots, N. \tag{5.20}$$

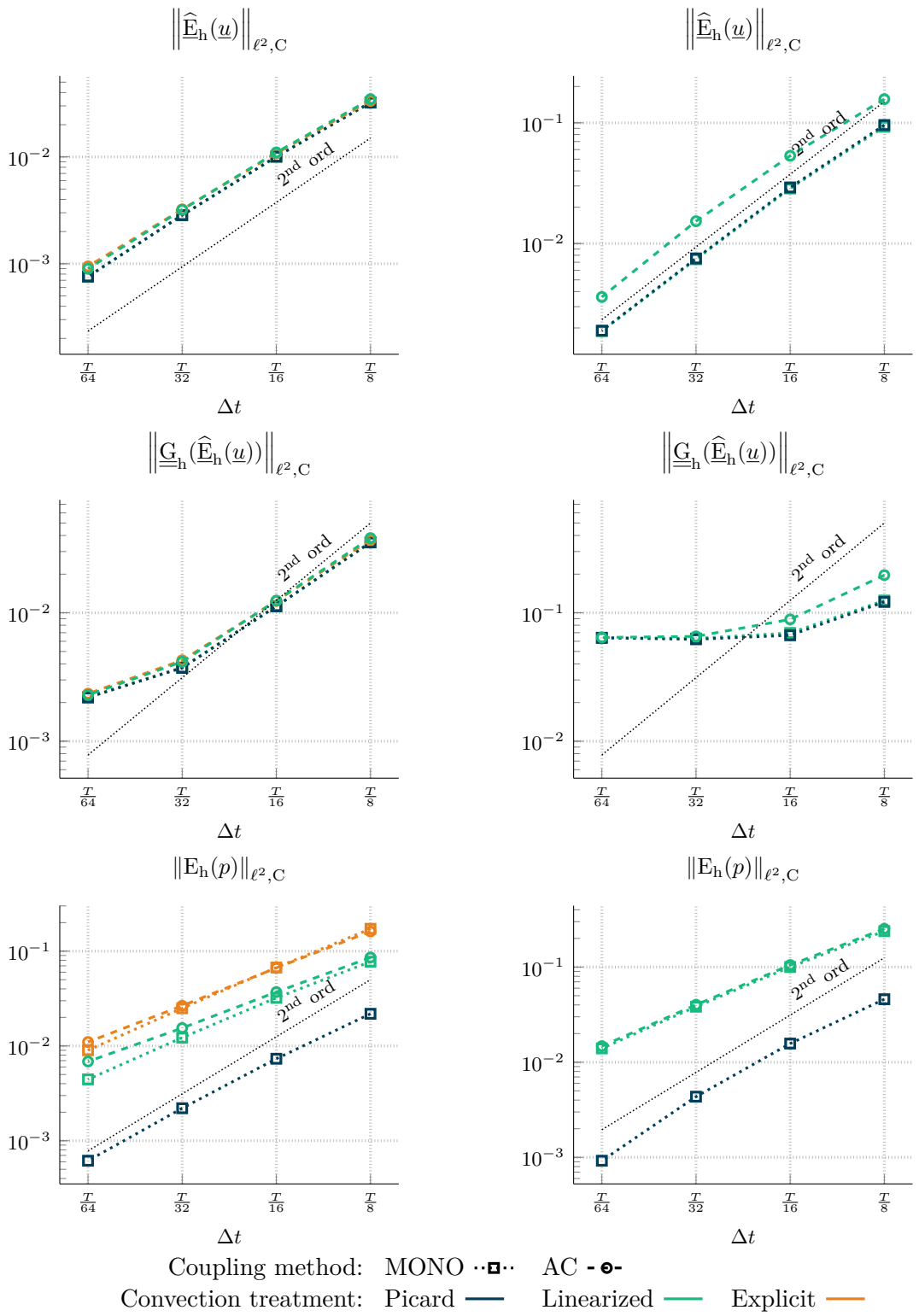
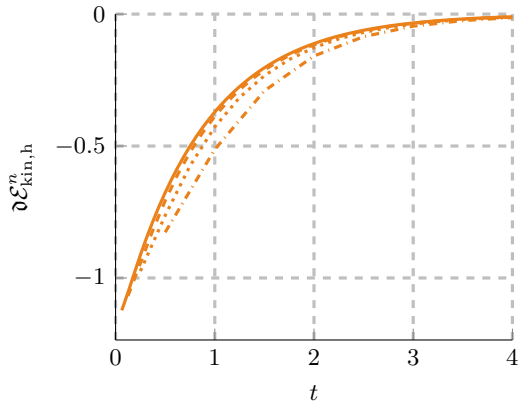
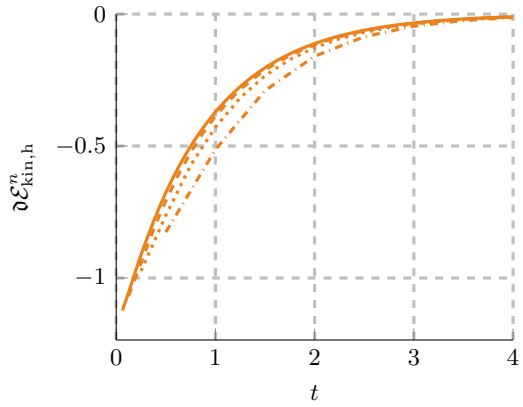


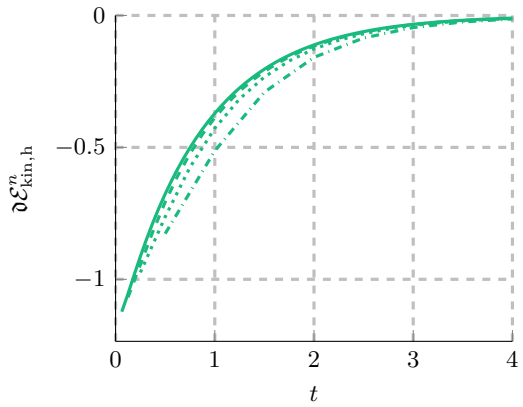
Figure 5.4 – 2D Navier–Stokes, Taylor–Green Vortex. Convergence in time. Cartesian mesh composed of 512^2 cells. Top: velocity L^2-error; middle: velocity H^1-error; bottom: pressure L^2-error. Left column: $Re \approx 3$, $T = 4$; right column: $Re \approx 33$, $T = 40$.



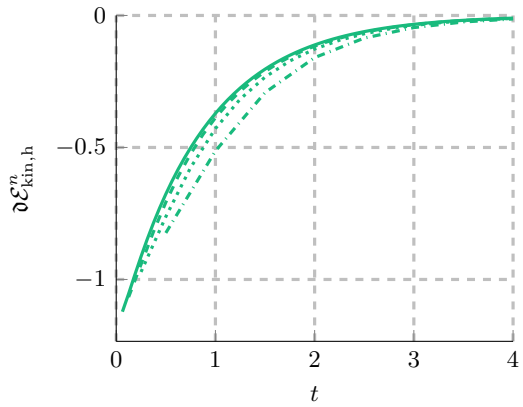
(a) Explicit convection and monolithic.



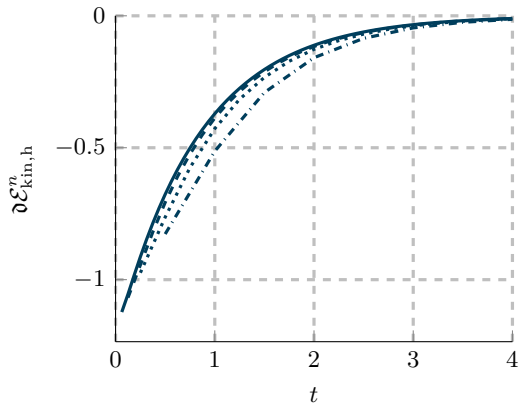
(b) Explicit convection and AC.



(c) Linearized convection and monolithic.



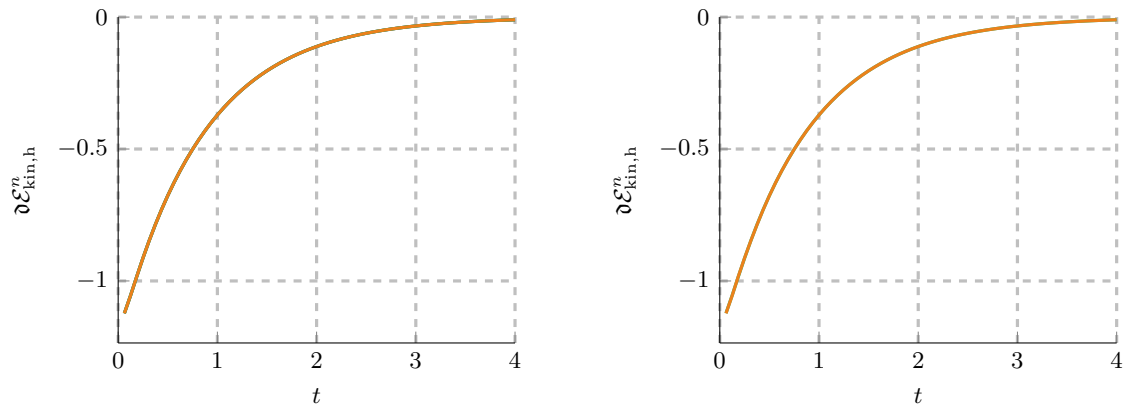
(d) Linearized convection and AC.



(e) Picard and monolithic.

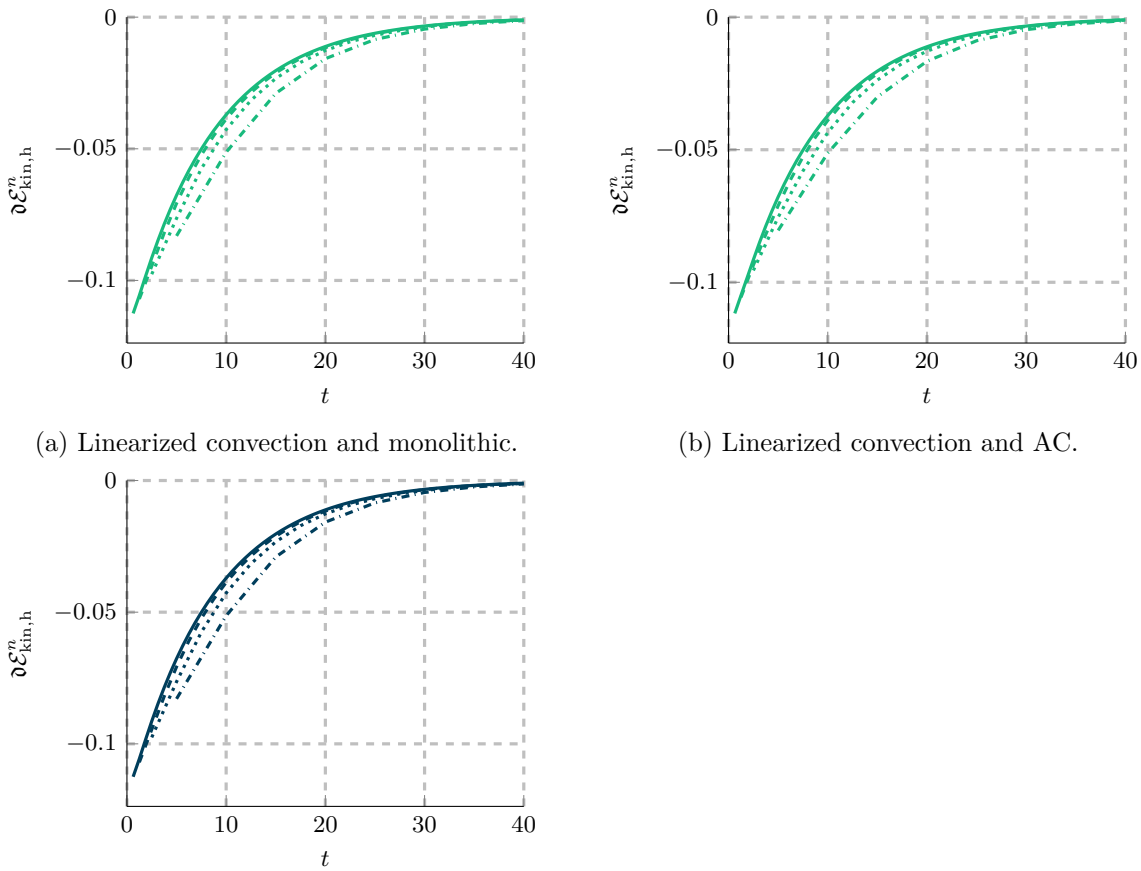
Figure 5.5 – 2D Navier–Stokes, Taylor–Green Vortex. $Re \approx 3$. $T = 4$. Cartesian mesh composed of 512^2 cells. Discrete time-derivative of the kinetic energy at t^n, $\mathfrak{d}\mathcal{E}^n_{\mathrm{kin,h}}$, see (5.20). $\Delta t = \frac{T}{8}$ -·-·-, $\frac{T}{16}$ ·····, $\frac{T}{32}$ - - -, $\frac{T}{64}$ ——.

The results are shown in Figs. 5.5 and 5.7. Even though no a priori knowledge is available, we observe that all the strategies are dissipative in this test case. Moreover, as in the first-order case, the convection treatment does not seem to lead to sizeable differences, see for instance the comparison in (5.6).



(a) Monolithic coupling. (b) AC technique - $Re \approx 3$.

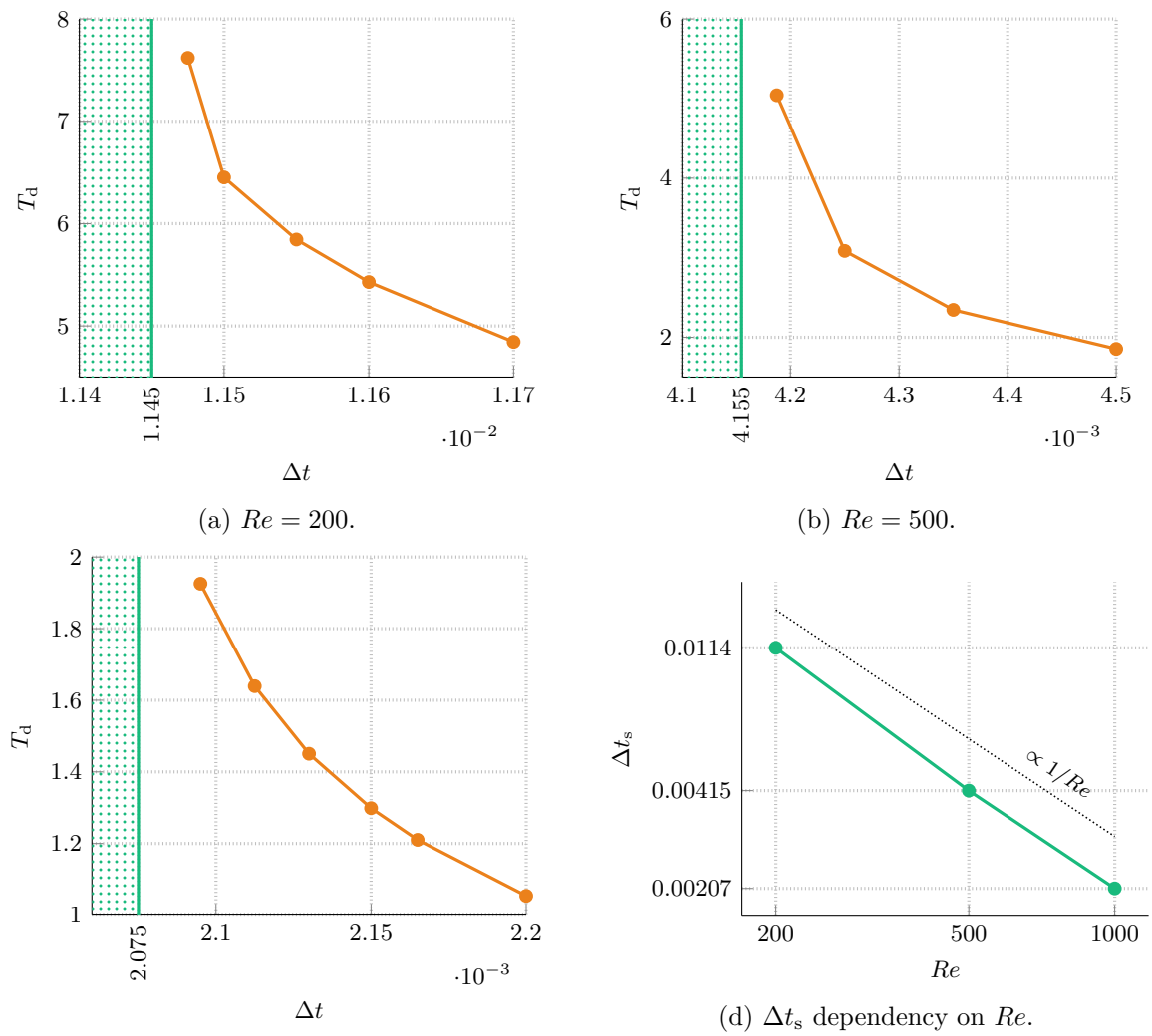
Figure 5.6 – 2D Navier–Stokes, Taylor–Green Vortex. Discrete time-derivative of the kinetic energy at t^n, $\mathfrak{d}\mathcal{E}^n_{\mathrm{kin,h}}$, see (5.20). Picard algorithm ——, linearized —— or explicit —— convection. $Re \approx 3$, $T = 4$, Cartesian mesh composed of 512^2 cells. $\Delta t = \frac{T}{64}$. Left: monolithic approach, right: AC technique.



(a) Linearized convection and monolithic. (b) Linearized convection and AC.

(c) Picard and monolithic.

Figure 5.7 – 2D Navier–Stokes, Taylor–Green Vortex. $Re \approx 33$. $T = 40$. Cartesian mesh composed of 512^2 cells. Discrete time-derivative of the kinetic energy at t^n, $\mathfrak{d}\mathcal{E}^n_{\mathrm{kin,h}}$, see (5.20). $\Delta t = \frac{T}{8}$ -·-·-, $\frac{T}{16}$ ·····, $\frac{T}{32}$ - - -, $\frac{T}{64}$ ——.



(a) $Re = 200$.

(b) $Re = 500$.

(c) $Re = 1000$.

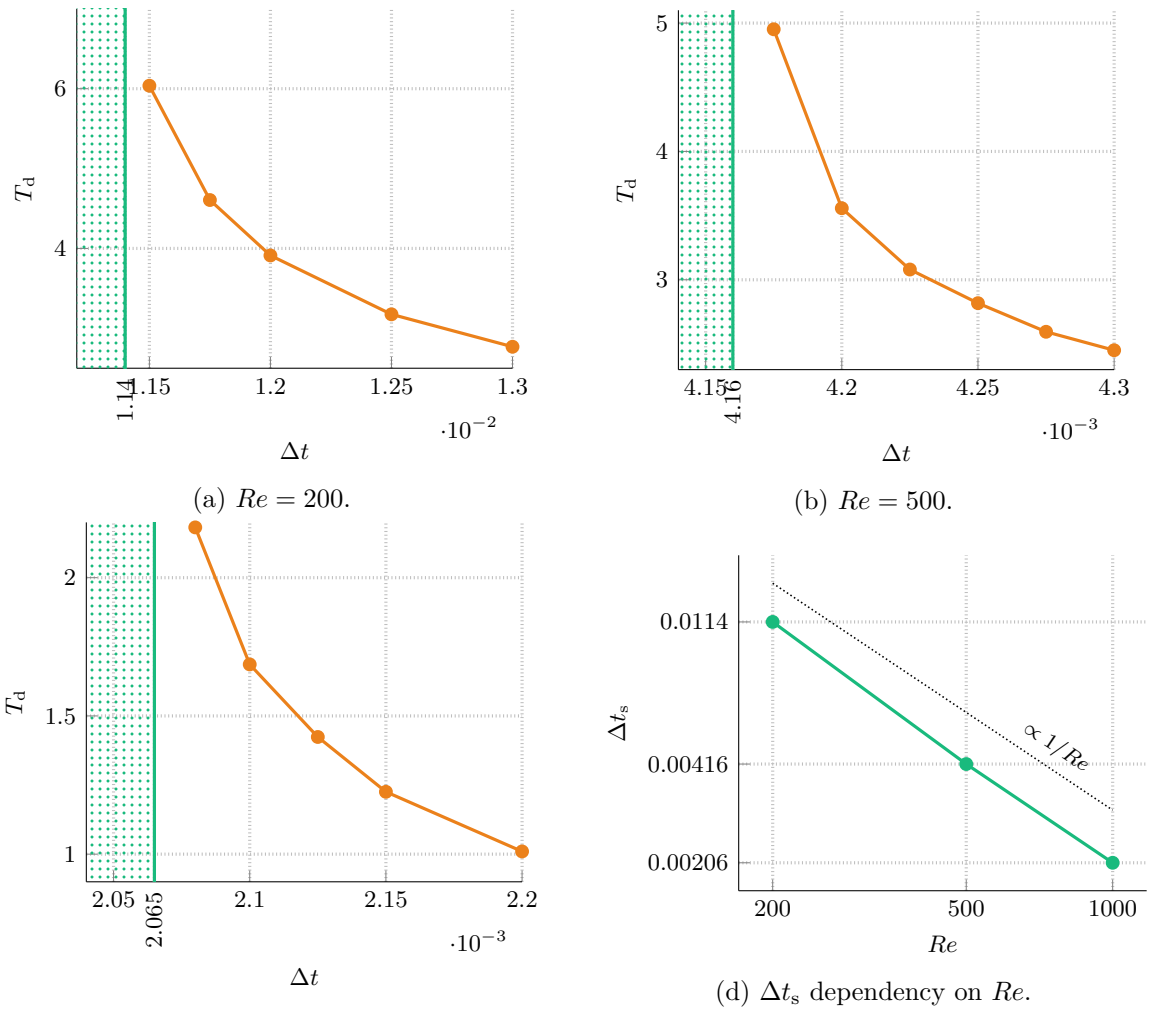
(d) Δt_s dependency on Re.

Figure 5.8 – 2D Navier–Stokes, Taylor–Green Vortex. Divergence time, T_d, for different choices of Δt with the monolithic approach with BDF2 and explicit convection for three Reynolds numbers, Re, and dependency of the stability limit, Δt_s, (up to 1% resolution) on Re. T_d are in orange, Δt_s in green. Cartesian mesh composed of 128^2 cells.

5.3.3 Stability results with an explicit convection

Following Section 4.5.3, we propose a numerical stability study of the explicit convection treatment with the two second-order time-schemes presented in Section 5.1. Let us recall the setting of these tests. Recall that Δt_s denotes the critical time-step value, that is, the greatest Δt for which we have not observed divergence, and, in a diverging computation, T_d is the first time node that satisfies the divergence criterion (5.21) below. Moreover, we consider (i) a Cartesian mesh composed of 128^2 cells, (ii) three Reynolds numbers, $Re \in \{200, 500, 1000\}$, (iii) T such that $T\,Re = 10^4$, (iv) $\eta = 10\,Re$ whenever the AC method is used, (v) we seek a resolution of 1%, meaning that the gap between Δt_s and the smallest Δt leading to divergence is less than 1% of Δt_s, (vi) and we flag a computation as having diverged if for a t^n, $n \geq 1$, we have

$$\mathcal{E}_{\mathrm{kin,h}}(\underline{\widehat{v}}_\mathrm{h}^n) > 1.1\,\mathcal{E}_{\mathrm{kin,h}}(\underline{\widehat{v}}_\mathrm{h}^0) = 1.1\,\mathcal{E}_{\mathrm{kin,h}}(\underline{\widehat{\pi}}_\mathrm{h}(\underline{u}_0))\,. \tag{5.21}$$



(a) $Re = 200$.

(b) $Re = 500$.

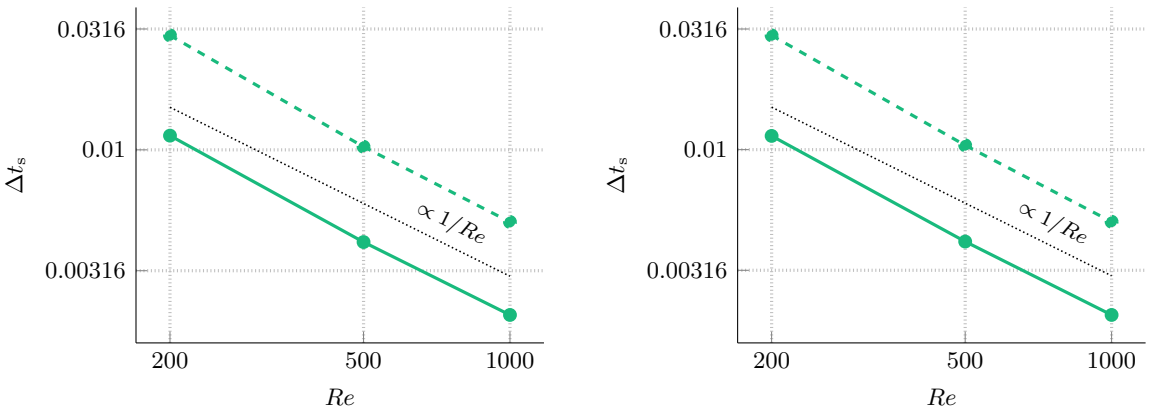
(c) $Re = 1000$.

(d) Δt_s dependency on Re.

Figure 5.9 – 2D Navier–Stokes, Taylor–Green Vortex. Divergence time, T_d, for different choices of Δt with the AC-bootstrap method with $\eta = 10Re$ and explicit convection for three Reynolds numbers, Re, and dependency of the stability limit, Δt_s, (up to 1% resolution) on Re. T_d are in orange, Δt_s in green. Cartesian mesh composed of 128^2 cells.

We present the results concerning the critical time-step values, Δt_s, and the divergence times, T_d, in Figs. 5.8 and 5.9 for, respectively, the monolithic approach with the BDF2 time-scheme and the AC method with the bootstrapping technique. As it was the case for the results concerning the first-order time-schemes (see Figs. 4.12 and 4.13), the values of Δt_s measured for the AC method are close to those obtained with the monolithic approach. Moreover, a dependency on the inverse of the Reynolds number is observed for Δt_s in the second-order case as well.

The results on Δt_s obtained with the monolithic-BDF2 and the AC-bootstrap strategies are summarized in Table 5.1. The influence of the η parameter of the AC method on the stability is less remarkable than for the first-order case. We proceed to a comparison of the stability results obtained with the first- (see Figs. 4.12 and 4.13) and second-order times-schemes. In Fig. 5.10, we observe that we can usually choose greater time-step values when considering first-order schemes: indeed, the critical time-step values Δt_s for the first-order time-schemes are more than two times higher than the Δt_s obtained at the same Reynolds number and with the same coupling strategy but with second-order time-schemes. See



(a) Monolithic approach. (b) AC method.

Figure 5.10 – Critical time-step values for stability Δt_s with respect to Reynolds number. Comparison of first- and second-order time-schemes. The results obtained with first- and second-order time-schemes are, respectively, in dashed and solid lines. Cartesian mesh composed of 128^2 cells.

Table 5.1 – 2D Navier–Stokes, Taylor–Green Vortex. Critical time-step values for stability, Δt_s, with respect to Reynolds number, Re, obtained with second order time-schemes. Cartesian mesh composed of 128^2 cells.

Re	MONO	AC(Re)	AC($10Re$)	AC($100Re$)
200	$1.15e{-2}$	$1.14e{-2}$	$1.14e{-2}$	$1.13e{-2}$
500	$4.16e{-3}$	$4.18e{-3}$	$4.16e{-3}$	$4.15e{-3}$
1000	$2.08e{-3}$	$2.07e{-3}$	$2.07e{-3}$	$2.05e{-3}$

Table 5.2 for more detailed results.

Finally, we investigate the effects of considering an additional constraint for the detection of the divergence. In fact, we track the enstrophy as well, quantity defined as follows:

$$\phi_{\mathrm{h}}(\omega_{\mathrm{C}}) := \sum_{\mathrm{c}\in\mathrm{C}} |\mathrm{c}|\,|\omega_{\mathrm{c}}|^2 \,, \qquad \omega_{\mathrm{c}} := [\underline{\underline{\mathrm{G}}}^0_{\mathrm{c}}(\widehat{\underline{u}}_{\mathrm{c}})]_{yx} - [\underline{\underline{\mathrm{G}}}^0_{\mathrm{c}}(\widehat{\underline{u}}_{\mathrm{c}})]_{xy} \qquad \forall \mathrm{c} \in \mathrm{C}\,. \tag{5.22}$$

In particular, to the hypotheses (i)-(vi) above, we add: (vii) we flag a computation as having diverged if, for some $n \geq 1$, we have

$$\phi_{\mathrm{h}}(\omega^n_{\mathrm{C}}) > 1.1\,\phi_{\mathrm{h}}(\omega^0_{\mathrm{C}})\,. \tag{5.23}$$

Table 5.3 reports the results when the new criterion (vii) is taken into account and a comparison with the results obtained without it. Differently from the first-order case where a slight decrease of Δt_s was observed for $Re = 1000$, no sizeable differences are reported here.

Table 5.2 – 2D Navier–Stokes, Taylor–Green Vortex. Critical time-step values for stability, Δt_s, with respect to the Reynolds number, Re. Comparison of first- and second-order time-schemes. Cartesian mesh composed of 128^2 cells. $\eta := 10Re$ when the AC is considered.

	Monolithic			AC		
Re	1st	2nd	$\frac{1^{\mathrm{st}}}{2^{\mathrm{nd}}}$	1st	2nd	$\frac{1^{\mathrm{st}}}{2^{\mathrm{nd}}}$
200	$2.98e{-2}$	$1.15e{-2}$	2.60	$2.98e{-2}$	$1.14e{-2}$	2.61
500	$1.03e{-2}$	$4.16e{-3}$	2.48	$1.04e{-2}$	$4.16e{-3}$	2.51
1000	$5.03e{-3}$	$2.08e{-3}$	2.42	$5.03e{-3}$	$2.07e{-3}$	2.43

Table 5.3 – 2D Navier–Stokes, Taylor–Green Vortex. Stability limits, Δt_{s}, up to a resolution of 1% obtained on a Cartesian mesh composed of 128^2 cells with the monolithic approach or the AC($10Re$) method for three Reynolds numbers, Re. Comparison between a divergence criterion concerning the kinetic energy only, or the kinetic energy and the enstrophy.

	MONO		AC($10Re$)	
Re	$\mathcal{E}_{\mathrm{kin,h}}$	$\mathcal{E}_{\mathrm{kin,h}}$ & ϕ_{h}	$\mathcal{E}_{\mathrm{kin,h}}$	$\mathcal{E}_{\mathrm{kin,h}}$ & ϕ_{h}
200	$1.15e{-2}$	$1.14e{-2}$	$1.14e{-2}$	$1.14e{-2}$
500	$4.16e{-3}$	$4.16e{-3}$	$4.16e{-3}$	$4.16e{-3}$
1000	$2.08e{-3}$	$2.06e{-3}$	$2.07e{-3}$	$2.07e{-3}$

5.4 Detailed results

We give in this section the tables collecting the details of the results presented in Section 5.2 and Section 5.3.

5.4.1 Stokes equations

The purpose of this section is to provide the complete details of the simulations presented in Section 5.2, which covers a 3D modified Taylor–Green Vortex solution to the unsteady Stokes equations. Comparisons between two second-order time-schemes, namely a BDF2 for the monolithic approach and a bootstrapping technique in the AC context, and between first- (Implicit Euler) and second-order time-schemes are done. The results are in Tables 5.4 to 5.9. The columns of the table are grouped and each third column denotes the ratio between the values obtained with the second- and first-order scheme for the strategy at hand.

Table 5.4 – Unsteady Stokes problem. 3D TGV solution (4.50), $T = 2$. Comparison of second- and first-order results of velocity and pressure errors and computational time. Coupling: monolithic. Mesh: Cartesian.

	$\left\lVert \widehat{\underline{\mathrm{E}}}_{\mathrm{h}}(\underline{u}) \right\rVert_{\ell^2,\mathrm{C}}$			$\left\lVert \underline{\underline{\mathrm{G}}}_{\mathrm{h}}(\widehat{\underline{\mathrm{E}}}_{\mathrm{h}}(\underline{u})) \right\rVert_{\ell^2,\mathrm{C}}$			$\lVert \mathrm{E}_{\mathrm{h}}(p) \rVert_{\ell^2,\mathrm{C}}$			Elapsed×cores [s]		
$\Delta t = \frac{T}{\dots}$	First	Second	Ratio $\frac{1^{\mathrm{st}}}{2^{\mathrm{nd}}}$	First	Second	Ratio $\frac{1^{\mathrm{st}}}{2^{\mathrm{nd}}}$	First	Second	Ratio $\frac{1^{\mathrm{st}}}{2^{\mathrm{nd}}}$	First	Second	Ratio $\frac{1^{\mathrm{st}}}{2^{\mathrm{nd}}}$
4	1.35e−2	1.58e−2	0.9	1.08e−1	1.27e−1	0.9	9.09e−2	1.07e−1	0.8	6.58e+5	6.92e+5	1.0
8	6.79e−3	5.92e−3	1.1	5.44e−2	4.74e−2	1.1	4.59e−2	4.00e−2	1.1	1.31e+6	1.37e+6	1.0
16	3.50e−3	1.73e−3	2.0	2.80e−2	1.39e−2	2.0	2.36e−2	1.17e−2	2.0	2.63e+6	2.69e+6	1.0
32	1.77e−3	4.75e−4	3.7	1.43e−2	4.09e−3	3.5	1.20e−2	3.21e−3	3.7	5.09e+6	5.11e+6	1.0
64	8.95e−4	1.32e−4	6.8	7.31e−3	1.84e−3	4.0	6.04e−3	9.01e−4	6.7	9.97e+6	1.02e+7	1.0
128	4.54e−4	4.01e−5	11.3	3.92e−3	1.53e−3	2.6	3.06e−3	2.80e−4	10.9	2.03e+7	2.02e+7	1.0
256	2.33e−4	1.75e−5	13.3	2.39e−3	1.51e−3	1.6	1.57e−3	1.40e−4	11.2	4.22e+7	4.07e+7	1.0

Table 5.5 – Unsteady Stokes problem. 3D TGV solution (4.50), $T = 2$. Comparison of second- and first-order results of velocity and pressure errors and computational time. Coupling: AC(10). Mesh: Cartesian.

	$\left\lVert \widehat{\underline{\mathrm{E}}}_{\mathrm{h}}(\underline{u}) \right\rVert_{\ell^2,\mathrm{C}}$			$\left\lVert \underline{\underline{\mathrm{G}}}_{\mathrm{h}}(\widehat{\underline{\mathrm{E}}}_{\mathrm{h}}(\underline{u})) \right\rVert_{\ell^2,\mathrm{C}}$			$\lVert \mathrm{E}_{\mathrm{h}}(p) \rVert_{\ell^2,\mathrm{C}}$			Elapsed×cores [s]		
$\Delta t = \frac{T}{\dots}$	First	Second	Ratio $\frac{1^{\mathrm{st}}}{2^{\mathrm{nd}}}$	First	Second	Ratio $\frac{1^{\mathrm{st}}}{2^{\mathrm{nd}}}$	First	Second	Ratio $\frac{1^{\mathrm{st}}}{2^{\mathrm{nd}}}$	First	Second	Ratio $\frac{1^{\mathrm{st}}}{2^{\mathrm{nd}}}$
4	1.49e−2	1.62e−2	0.9	1.12e−1	1.28e−1	0.9	1.13e−1	1.07e−1	1.1	1.92e+5	3.37e+5	0.6
8	8.42e−3	6.14e−3	1.4	6.18e−2	4.83e−2	1.3	9.35e−2	5.10e−2	1.8	3.78e+5	7.08e+5	0.5
16	4.61e−3	1.89e−3	2.4	3.37e−2	1.47e−2	2.3	5.87e−2	2.11e−2	2.8	7.22e+5	1.40e+6	0.5
32	2.40e−3	5.61e−4	4.3	1.77e−2	4.48e−3	4.0	3.21e−2	7.39e−3	4.3	1.39e+6	2.67e+6	0.5
64	1.22e−3	1.68e−4	7.3	9.23e−3	1.94e−3	4.7	1.65e−2	2.49e−3	6.6	2.59e+6	4.96e+6	0.5
128	6.17e−4	5.31e−5	11.6	4.98e−3	1.55e−3	3.2	8.31e−3	9.25e−4	9.0	4.61e+6	9.25e+6	0.5
256	3.13e−4	2.10e−5	14.9	2.95e−3	1.51e−3	2.0	4.18e−3	4.13e−4	10.1	8.34e+6	1.64e+7	0.5

Table 5.6 – Unsteady Stokes problem. 3D TGV solution (4.50), $T = 2$. Comparison of second- and first-order results of velocity and pressure errors and computational time. Coupling: AC(100) ⊸. Mesh: Cartesian.

	$\left\lVert \widehat{\underline{\mathrm{E}}}_{\mathrm{h}}(\underline{u}) \right\rVert_{\ell^2,\mathrm{C}}$			$\left\lVert \underline{\underline{\mathrm{G}}}_{\mathrm{h}}(\widehat{\underline{\mathrm{E}}}_{\mathrm{h}}(\underline{u})) \right\rVert_{\ell^2,\mathrm{C}}$			$\lVert \mathrm{E}_{\mathrm{h}}(p) \rVert_{\ell^2,\mathrm{C}}$			Elapsed×cores [s]		
$\Delta t = \frac{T}{\dots}$	First	Second	Ratio $\frac{1^{\mathrm{st}}}{2^{\mathrm{nd}}}$	First	Second	Ratio $\frac{1^{\mathrm{st}}}{2^{\mathrm{nd}}}$	First	Second	Ratio $\frac{1^{\mathrm{st}}}{2^{\mathrm{nd}}}$	First	Second	Ratio $\frac{1^{\mathrm{st}}}{2^{\mathrm{nd}}}$
4	1.35e−2	1.58e−2	0.9	1.08e−1	1.27e−1	0.9	8.88e−2	1.06e−1	0.8	5.00e+5	8.60e+5	0.6
8	6.83e−3	5.92e−3	1.2	5.45e−2	4.74e−2	1.1	4.65e−2	4.02e−2	1.2	9.40e+5	1.80e+6	0.5
16	3.52e−3	1.73e−3	2.0	2.81e−2	1.39e−2	2.0	2.43e−2	1.19e−2	2.0	1.89e+6	3.49e+6	0.5
32	1.78e−3	4.75e−4	3.7	1.43e−2	4.09e−3	3.5	1.23e−2	3.31e−3	3.7	3.60e+6	6.83e+6	0.5
64	8.99e−4	1.33e−4	6.8	7.33e−3	1.84e−3	4.0	6.23e−3	9.54e−4	6.5	6.63e+6	1.29e+7	0.5
128	4.56e−4	4.02e−5	11.3	3.94e−3	1.53e−3	2.6	3.16e−3	3.15e−4	10.0	1.21e+7	2.36e+7	0.5
256	2.34e−4	1.73e−5	13.5	2.40e−3	1.50e−3	1.6	1.62e−3	1.49e−4	10.9	2.19e+7	4.22e+7	0.5

Table 5.7 – Unsteady Stokes problem. 3D TGV solution (4.50), $T = 2$. Comparison of second- and first-order results of velocity and pressure errors and computational time. Coupling: monolithic ⊸. Mesh: prismatic with polygonal bases.

	$\left\lVert \widehat{\underline{\mathrm{E}}}_{\mathrm{h}}(\underline{u}) \right\rVert_{\ell^2,\mathrm{C}}$			$\lVert \mathrm{E}_{\mathrm{h}}(p) \rVert_{\ell^2,\mathrm{C}}$			Elapsed×cores [s]		
$\Delta t = \frac{T}{\dots}$	First	Second	Ratio $\frac{1^{\mathrm{st}}}{2^{\mathrm{nd}}}$	First	Second	Ratio $\frac{1^{\mathrm{st}}}{2^{\mathrm{nd}}}$	First	Second	Ratio $\frac{1^{\mathrm{st}}}{2^{\mathrm{nd}}}$
4	1.35e−2	1.58e−2	0.9	9.10e−2	1.07e−1	0.8	4.42e+5	4.12e+5	1.1
8	6.84e−3	5.97e−3	1.1	4.61e−2	4.03e−2	1.1	8.57e+5	8.00e+5	1.1
16	3.55e−3	1.76e−3	2.0	2.39e−2	1.19e−2	2.0	1.67e+6	1.58e+6	1.1
32	1.82e−3	4.97e−4	3.7	1.23e−2	3.52e−3	3.5	3.36e+6	3.03e+6	1.1
64	9.47e−4	1.59e−4	5.9	6.46e−3	1.53e−3	4.2	7.11e+6	6.20e+6	1.1
128	5.07e−4	8.49e−5	6.0	3.57e−3	1.22e−3	2.9	1.29e+7	1.30e+7	1.0
256	2.87e−4	7.41e−5	3.9	2.20e−3	1.19e−3	1.9	2.64e+7	2.79e+7	0.9

Table 5.8 – Unsteady Stokes problem. 3D TGV solution (4.50), $T = 2$. Comparison of second- and first-order results of velocity and pressure errors and computational time. Coupling: AC(10) —o—. Mesh: prismatic with polygonal bases.

	$\left\lVert \widehat{\underline{\mathrm{E}}}_{\mathrm{h}}(\underline{u}) \right\rVert_{\ell^2,\mathrm{C}}$			$\lVert \mathrm{E}_{\mathrm{h}}(p) \rVert_{\ell^2,\mathrm{C}}$			Elapsed×cores [s]		
$\Delta t = \frac{T}{\dots}$	First	Second	Ratio $\frac{1^{\mathrm{st}}}{2^{\mathrm{nd}}}$	First	Second	Ratio $\frac{1^{\mathrm{st}}}{2^{\mathrm{nd}}}$	First	Second	Ratio $\frac{1^{\mathrm{st}}}{2^{\mathrm{nd}}}$
4	$1.49e{-2}$	$1.62e{-2}$	0.9	$1.13e{-1}$	$1.07e{-1}$	1.1	$8.23e{+4}$	$1.43e{+5}$	0.6
8	$8.46e{-3}$	$6.18e{-3}$	1.4	$9.36e{-2}$	$5.12e{-2}$	1.8	$1.61e{+5}$	$3.01e{+5}$	0.5
16	$4.65e{-3}$	$1.92e{-3}$	2.4	$5.88e{-2}$	$2.12e{-2}$	2.8	$3.00e{+5}$	$5.75e{+5}$	0.5
32	$2.44e{-3}$	$5.78e{-4}$	4.2	$3.22e{-2}$	$7.52e{-3}$	4.3	$5.70e{+5}$	$1.09e{+6}$	0.5
64	$1.26e{-3}$	$1.89e{-4}$	6.7	$1.66e{-2}$	$2.77e{-3}$	6.0	$1.02e{+6}$	$1.95e{+6}$	0.5
128	$6.57e{-4}$	$9.12e{-5}$	7.2	$8.51e{-3}$	$1.50e{-3}$	5.7	$1.82e{+6}$	$3.51e{+6}$	0.5
256	$3.55e{-4}$	$7.48e{-5}$	4.7	$4.46e{-3}$	$1.25e{-3}$	3.6	$3.18e{+6}$	$6.02e{+6}$	0.5

Table 5.9 – Unsteady Stokes problem. 3D TGV solution (4.50), $T = 2$. Comparison of second- and first-order results of velocity and pressure errors and computational time. Coupling: AC(100) —o—. Mesh: prismatic with polygonal bases.

	$\left\lVert \widehat{\underline{\mathrm{E}}}_{\mathrm{h}}(\underline{u}) \right\rVert_{\ell^2,\mathrm{C}}$			$\lVert \mathrm{E}_{\mathrm{h}}(p) \rVert_{\ell^2,\mathrm{C}}$			Elapsed×cores [s]		
$\Delta t = \frac{T}{\dots}$	First	Second	Ratio $\frac{1^{\mathrm{st}}}{2^{\mathrm{nd}}}$	First	Second	Ratio $\frac{1^{\mathrm{st}}}{2^{\mathrm{nd}}}$	First	Second	Ratio $\frac{1^{\mathrm{st}}}{2^{\mathrm{nd}}}$
4	$1.36e{-2}$	$1.58e{-2}$	0.9	$8.89e{-2}$	$1.06e{-1}$	0.8	$2.28e{+5}$	$3.90e{+5}$	0.6
8	$6.87e{-3}$	$5.97e{-3}$	1.2	$4.68e{-2}$	$4.04e{-2}$	1.2	$4.44e{+5}$	$8.32e{+5}$	0.5
16	$3.57e{-3}$	$1.76e{-3}$	2.0	$2.46e{-2}$	$1.21e{-2}$	2.0	$8.21e{+5}$	$1.64e{+6}$	0.5
32	$1.83e{-3}$	$4.97e{-4}$	3.7	$1.27e{-2}$	$3.61e{-3}$	3.5	$1.62e{+6}$	$3.15e{+6}$	0.5
64	$9.51e{-4}$	$1.60e{-4}$	6.0	$6.64e{-3}$	$1.56e{-3}$	4.3	$2.94e{+6}$	$5.65e{+6}$	0.5
128	$5.09e{-4}$	$8.50e{-5}$	6.0	$3.65e{-3}$	$1.23e{-3}$	3.0	$5.09e{+6}$	$9.90e{+6}$	0.5
256	$2.88e{-4}$	$7.41e{-5}$	3.9	$2.24e{-3}$	$1.19e{-3}$	1.9	$8.85e{+6}$	$1.80e{+7}$	0.5

5.4.2 Navier–Stokes equations

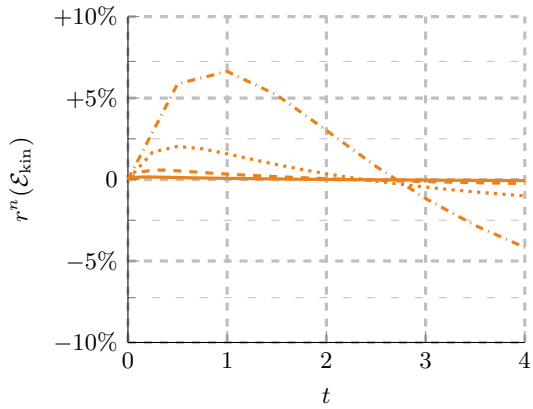
We report in this section the additional details on the Taylor–Green Vortex test case addressed with second-order time-schemes. The setting of the test case is given in Section 5.3. As it has been done in Section 4.6.2 for the first-order time setting, we compare the computed and exact kinetic energy by means of the following quantity (see (4.66))

$$r^n(\mathcal{E}_{\text{kin}}) := \frac{\mathcal{E}_{\text{kin,h}}(\widehat{\underline{v}}_{\text{h}}^n) - \mathcal{E}_{\text{kin}}(t^n)}{\mathcal{E}_{\text{kin}}(t^n)}\,, \qquad n = 0, \ldots, N\,, \tag{5.24}$$

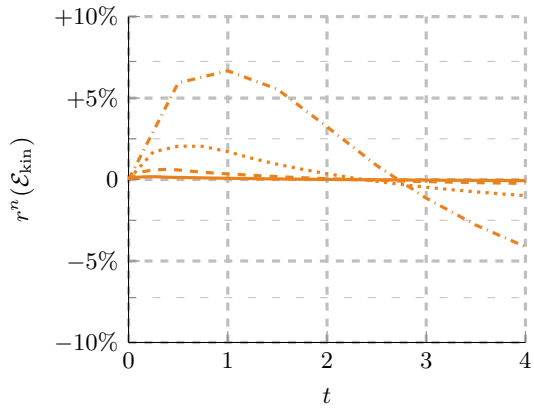
where the exact kinetic energy $\mathcal{E}_{\text{kin}}(t)$ is defined as follows:

$$\mathcal{E}_{\text{kin}}(t) := \frac{1}{2}\int_\Omega |\underline{u}|_2^2 = \pi^2 \exp(-4\nu t)\,. \tag{5.25}$$

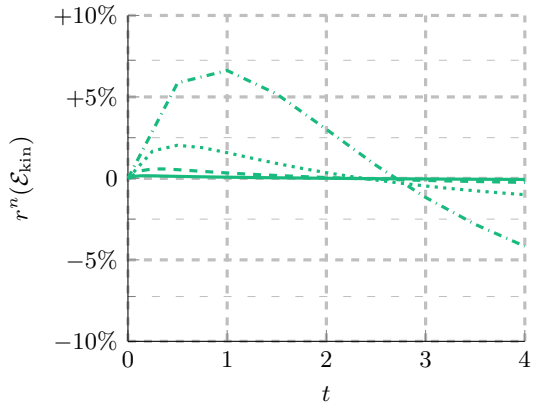
The results obtained for the different strategies are in Fig. 5.11. As in the first-order case, the discretization errors are dominant: notice the difference between the results obtained with the coarsest and finest time-step. On the contrary, differently than the first-order case, the considered second-order time-schemes seem to slightly underestimate the energy (with respect to the analytical formula (5.25)) on the long run. No sizable difference due to the convection treatment is observed.



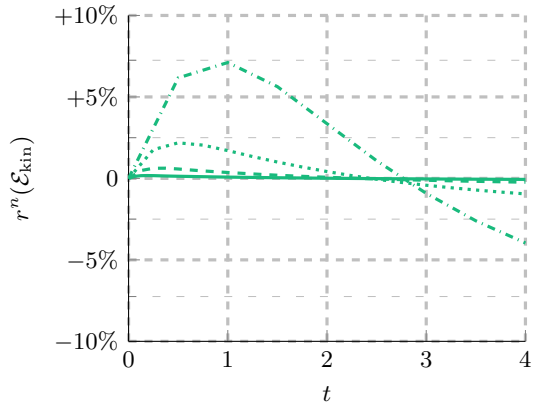
(a) Explicit convection and monolithic.



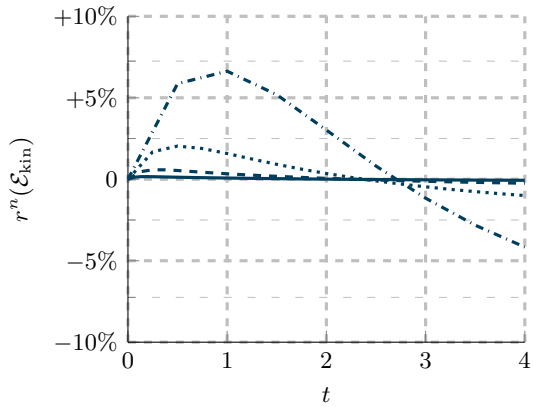
(b) Explicit convection and AC.



(c) Linearized convection and monolithic.

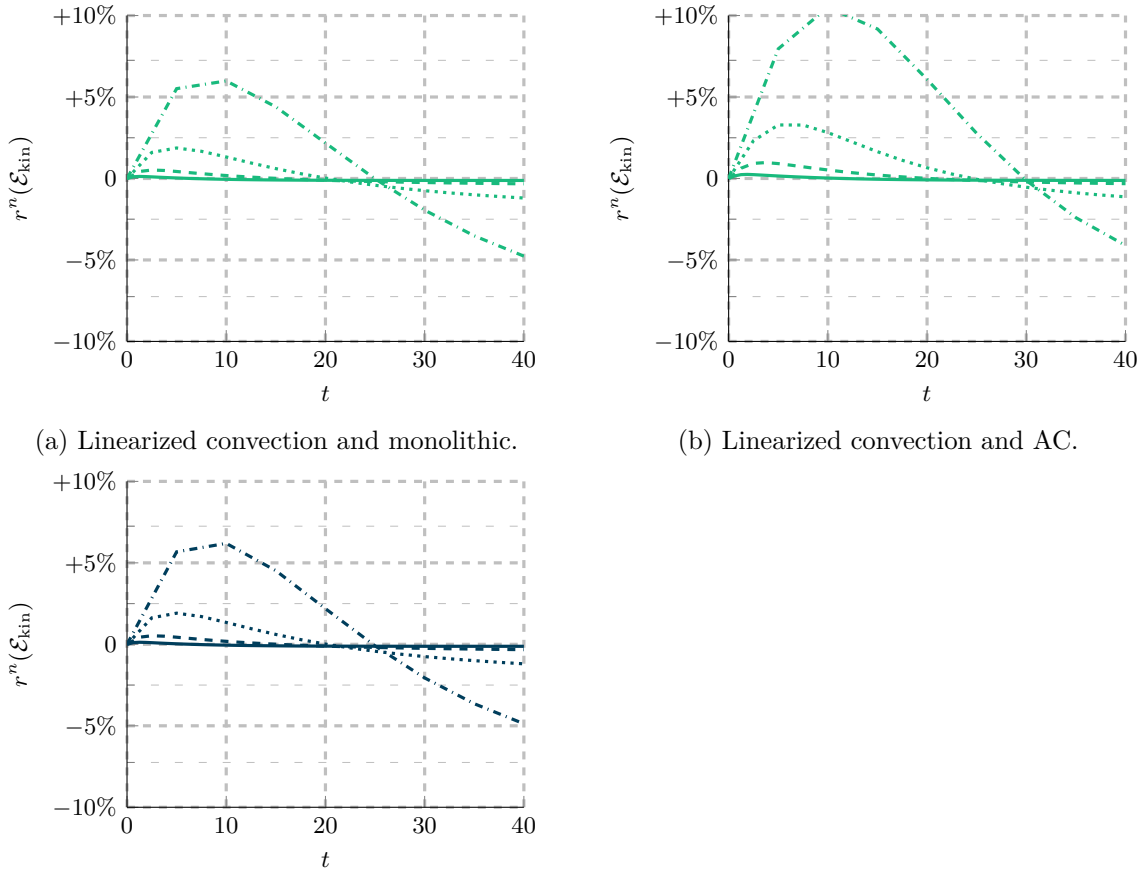


(d) Linearized convection and AC.



(e) Picard and monolithic.

Figure 5.11 – 2D Navier–Stokes, Taylor–Green Vortex. $Re \approx 3$. $T = 4$. Cartesian mesh composed of 512^2 cells. Ratio (in percentage) between the computed and reference kinetic energy $r(\mathcal{E}_{\text{kin,h}})$, see (5.24), for different combinations of coupling technique and convection treatment. $\Delta t = \frac{T}{8}$ -·-·-, $\frac{T}{16}$ ·····, $\frac{T}{32}$ - - -, $\frac{T}{64}$ ——.



(a) Linearized convection and monolithic.

(b) Linearized convection and AC.

(c) Picard and monolithic.

Figure 5.12 – 2D Navier–Stokes, Taylor–Green Vortex. $Re \approx 33$. $T = 40$. Cartesian mesh composed of 512^2. Ratio (in percentage) between the computed and reference kinetic energy $r(\mathcal{E}_{\text{kin,h}})$, see (5.24), for different combinations of coupling technique and convection treatment. $\Delta t = \frac{T}{8}$ -·-·-, $\frac{T}{16}$ ·····, $\frac{T}{32}$ - - -, $\frac{T}{64}$ ——.

Chapter 6

Conclusions and perspectives

The work presented in this Thesis allowed us to extend the face-based CDO (CDO-Fb) discretization to the unsteady, incompressible Navier–Stokes equations (NSE). Special attention has been paid to the treatment of the velocity-pressure coupling and to the treatment of the nonlinear convection term in order to improve the numerical strategy in terms of accuracy and efficiency. Concerning the velocity-pressure coupling, the monolithic approach and the Artificial Compressibility (AC) technique have been tested, whereas classical methods such as the Picard algorithm, the linearization and the explicitation of the convection term have been used to deal with the nonlinearity of the NSE.

The lowest-order discrete CDO-Fb setting and the related tools needed to build a steady Navier–Stokes problem are introduced in Chapter 2. The starting point is the work of Bonelle (2014, Section 8.3) and Bonelle and Ern (2015), whose gradient reconstruction operator has been used (in the vector-valued form) in our setting. A divergence operator satisfying an inf-sup condition and common to other lowest-other methods has been considered. The main contribution to the CDO-Fb setting is the development of two operators discretizing, respectively, the scalar-valued advection and the vector-valued convection term. Importantly, the convection operator is dissipative (and skew-symmetric under certain assumptions) and satisfies the discrete counterparts of some well-known integration-by-parts results.

Hinging on the aforementioned operators, the discrete version of the steady incompressible Stokes and NSE have been built in Chapter 3. Test cases have been considered both in two and three dimensions. The results have shown that the expected orders of convergence in space were recovered for the discrete L^2- and H^1-like norms of the velocity and the L^2-like norm of the pressure (for the pressure, even slightly higher-than-expected rates were sometimes measured). Simulations have been run on general and distorted meshes, and the theoretical orders of convergence were observed even on these latter meshes.

Having validated the steady case, we have moved in Chapter 4 to the unsteady Stokes and NSE. The performances of the monolithic approach and AC method strategies have been compared on Stokes test cases and the results showed that AC is a reliable method which ensures great efficiency while still providing good accuracy. As for the nonlinear treatment, classical strategies (Picard algorithm, linearized and explicit convection) have been compared on the well-known Taylor–Green Vortex test case. The expected orders of convergence in time have been recovered for all the considered strategies, and the behavior of the kinetic energy did not exhibit significant differences among all the considered strategies, at least in the test cases at hand. An empirical study aiming at evaluating the stability of the explicit treatment of the convection term has been performed, revealing a dependency of the critical time-step value on the inverse of the Reynolds number.

The unsteady discrete setting has been extended in Chapter 5 by applying second-order time schemes. In particular, a BDF discretization has been coupled with the monolithic approach and a bootstrapping technique with the AC method. The same test cases as in Chapter 4 have been considered. The results suggest that the second-order discretization is more efficient, in the sense that it can reach a given error threshold with less computational time than the first-order discretization. The AC method remained competitive with respect to the monolithic approach even though the gap between the two coupling techniques is less pronounced than for the first-order time-stepping scheme. These observations highlight the importance of an optimal tuning of the user-defined parameter of the AC method. The aforementioned convection treatments have been tested with second-order time-schemes, and the stability of the explicit convection has been investigated, leading to similar results as in the first-order case with, however, a reduced stability zone.

Let us now outline some perspectives to the work. Increasing the robustness of the present schemes with respect to the Reynolds number constitutes a first axis of improvement. We have already briefly discussed some techniques which could be considered to address the pressure-robustness of the CDO-Fb discretization. In particular, one possibility would be to modify the way in which the right-hand side of the discrete momentum equation is taken into account by considering a reconstruction hinging on Raviart–Thomas FEM if a simplicial mesh is considered. Other improvements can involve taking into consideration a variable viscosity and a complete stress constraint tensor, and especially the symmetric gradient.

The difficulties noticed at high Reynolds numbers are also the symptoms of a numerical setting which could need some improvements. We have chosen the Picard algorithm because of its robustness, while we are aware that it may not be the most efficient method. The development of a more adapted method, such as the Newton one or the Anderson acceleration, should be tested. Moreover, the overall efficiency would benefit from linear solvers that are more adapted to the NSE than those that we have considered (i.e. LU factorization or Jacobi-preconditioned Conjugate Gradient on the augmented system). Such improvements may include adapted preconditioners, see for instance Benzi and Olshanskii (2006) and Olshanskii and Benzi (2008), or even reliable and efficient iterative solvers for nonsymmetric matrices (see Benzi *et al.* (2005)). It must be said that, however, the aforementioned improvements are especially needed when dealing with the NSE. In fact, the performance measured for the Stokes problem where a "friendlier" setting is recovered (i.e. symmetric definite positive velocity-velocity block) gave satisfactory results when using the GKB algorithm.

The numerical tests (see especially Section 4.4) also confirmed that the AC method is an efficient and accurate alternative to the monolithic approach. Further analyses of the strategy may include a more thorough study on the influence of the user-defined parameter involved in the method. It has been shown that, on the one hand, it can have a positive impact on the accuracy but, at the same time, the performance highly depends on it (due to a possibly ill-conditioned system). Ideally, one hopes to be able to provide the range in which the optimal value lies or, even better, to let the code set it automatically. Furthermore, the perturbation of the incompressibility constraint in the AC method deserves to be further investigated. For one, in the CDO-Fb discretization, the skew-symmetry of the convection operator requires the discrete velocity to be divergence-free, which is in general not the case when the AC method is used. A first and straightforward fix is to consider the so-called Temam's trick. This could in particular ensure energy conservation, which is important at high Reynolds numbers. Another issue is the use of a non divergence-free velocity to transport a solute. Several authors warn against this lack of incompressibility (see for instance Chippada *et al.* (1997), Wheeler *et al.* (2002), Olshanskii and Reusken (2004), Linke (2009), and Galvin *et al.* (2012)). A way to overcome this problem could be to consider a postprocessing in which a divergence-free velocity field is reconstructed starting

from the one recovered from the AC step.

Even though generic BCs are allowed, only Dirichlet ones were considered in the numerical tests. Other types of BCs should be tested; among others, the symmetry and the homogeneous Neumann (outflow) BCs will allow one to extend the method on more meaningful applications. Possibly starting from classical test cases such as the backward facing step, or the flow behind a cylinder or around a NACA profile, one will then move to industrial applications. Since most of the industrial applications involve turbulent regimes, a turbulent model should be integrated into the system.

All the extensions to the CDO-Fb discussed in this Thesis have been fully integrated to the open-source, industrial CFD software *Code_Saturne*, and they are available to the end-user. NSE discretized by means of CDO-Fb schemes are currently being tested (in a setting composed of the monolithic approach and the Picard algorithm) by engineers at EDF R&D for solidification problems where the value of the Reynolds number is sufficiently low in order to avoid the turbulent regime (besides the NSE, an energy equation with a Boussinesq approximation and the transport of the solute concentration are considered). Ongoing comparisons with, for example, the legacy prediction-correction method of *Code_Saturne* hinging on a colocated Finite Volume scheme showed that the robustness and the accuracy of CDO-Fb allows the user to choose larger time-steps and to achieve a better overall performance. A prototype for CDO-Fb schemes with prediction-correction strategies is being developed at EDF R&D. At this preliminary stage, the incremental formulation (which takes into account the pressure in the prediction step as well) is considered and the correction step is performed by means of a scalar-valued diffusion problem on the pressure. So far, the CDO-Fb discretization is kept for the latter system. This means that the pressure is hybrid as well. The face-based DoFs can be simply discarded when dealing with the vector-valued velocity-related equation or one can think of using them by means of an appropriate reconstructor. In this latter case, a lifting involving both face- and cell-based pressure DoFs might be devised and used in the bilinear form ensuring the velocity-pressure coupling. If discarding pressure DoFs can be accepted, different CDO schemes may be considered for the Poisson problem for the pressure, namely, the cell-based and the vertex-based version with additional cell-based DoFs. In any case, the projection method could be included in the efficiency comparisons done between the monolith approach and the AC method.

Finally, in order to better evaluate the performance of the CDO-Fb schemes in the context of the NSE, comparisons with other spatial polyhedral methods, both classical (for instance, the colocated FV of *Code_Saturne*) and more recent schemes (lowest-order HHO or Gradient Schemes) should be considered. A possible test playground could be the recent FVCA VIII benchmark, which has been already considered for some of the numerical results presented in Sections 3.4 and 3.5.

Acronyms

Re Reynolds number

AC Artificial Compressibility

ALU Augmented Lagrangian–Uzawa algorithm

BC Boundary conditions

BDF2 Backward Differentiation Formula - 2nd order

CDO Compatible Discrete Operator schemes

CFD Computational Fluid Dynamics

CR Crouzeix–Raviart finite element

dG Discontinuous Galerkin methods

DoF Degree of Freedom

FEM Finite Element Methods

FV Finite Volume schemes

GKB Golub–Kahan Bidiagonalization

HDG Hybridizable Discontinuous Galerkin methods

HFV Hybrid Finite Volume schemes

HHO Hybrid High-Order schemes

HMM Hybrid Mixed Mimetic family

MFD Mimetic Finite Difference schemes

MFV Mixed Finite Volume schemes

NSE Navier–Stokes equations

PDE Partial differential equation

TGV Taylor–Green Vortex

VEM Virtual Element Methods

Bibliography

Abbas, M., Ern, A., and Pignet, N. (2018). "Hybrid High-Order methods for the finite deformations of hyperelastic materials". *Comput. Mech.* 62.4, pp. 909–928 (cited on p. 24).

Aghili, J. and Di Pietro, D. A. (2018). "An advection-robust Hybrid High-Order Method for the Oseen problem". *J. Sci. Comput.* 77.3, pp. 1310–1338 (cited on p. 23).

Aghili, J., Boyaval, S., and Di Pietro, D. A. (2015). "Hybridization of mixed high-order methods on general meshes and application to the Stokes equations". *Comput. Methods Appl. Math.* 15.2, pp. 111–134 (cited on p. 23).

Albensoeder, S. and Kuhlmann, H. C. (2005). "Accurate three-dimensional lid-driven cavity flow". *J. Comput. Phys.* 206.2, pp. 536–558 (cited on p. 85).

Amestoy, P. R., Duff, I. S., L'Excellent, J.-Y., and Koster, J. (2001). "A fully asynchronous multifrontal solver using distributed dynamic scheduling". *SIAM J. Matrix Anal. Appl.* 23.1, pp. 15–41 (cited on pp. 26, 75).

Andreianov, B., Bendahmane, M., Hubert, F., and Krell, S. (2012). "On 3D DDFV discretization of gradient and divergence operators. I. Meshing, operators and discrete duality". *IMA J. Numer. Anal.* 32.4, pp. 1574–1603 (cited on p. 22).

Andreianov, B., Bendahmane, M., and Hubert, F. (2013). "On 3D DDFV discretization of gradient and divergence operators: Discrete functional analysis tools and applications to degenerate parabolic problems". *Comput. Methods Appl. Math.* 13.4, pp. 369–410 (cited on p. 22).

Angeli, P.-E., Puscas, M.-A., Fauchet, G., and Cartalade, A. (2017). "FVCA8 benchmark for the Stokes and Navier–Stokes equations with the TrioCFD code – Benchmark session". In: *Finite Vol. Complex Appl. VIII; Methods Theor. Aspects.* Vol. 199. Springer Proc. Math. Stat. Lille: Springer International Publishing, pp. 181–202 (cited on pp. 77, 78, 80).

Angot, Ph. and Fabrie, P. (2012). "Convergence results for the vector penalty-projection and two-step artificial compressibility methods". *Discret. Contin. Dyn. Syst.* Series B 17.5, pp. 1383–1705 (cited on p. 32).

Angot, Ph., Caltagirone, J.-P., and Fabrie, P. (2008). "Vector Penalty-Projection Methods for the Solution of Unsteady Incompressible Flows". In: *Finite Vol. Complex Appl. V; Probl. Perspect.* Ed. by R. Eymard and J.-M. Hérard. Vol. 1. Aussois: ISTE Ltd - J. Wiley & Sons (UK, USA), pp. 169–176 (cited on p. 32).

Angot, Ph., Caltagirone, J.-P., and Fabrie, P. (2011). "A Spectacular Vector Penalty-Projection Method for Darcy and Navier–Stokes Problems". In: *Finite Vol. Complex Appl. VI; Probl. Perspect.* Ed. by J. Fořt, J. Fürst, J. Halama, R. Herbin, and F. Hubert. Vol. 1. Springer Proc. Math. Stat. Praha: Springer-Verlag (Berlin), pp. 39–47 (cited on p. 32).

Angot, Ph., Caltagirone, J.-P., and Fabrie, P. (2012). "A new fast method to compute saddle-points in constrained optimization and applications". *Appl. Math. Lett.* 25.3, pp. 245–251 (cited on p. 32).

Antonietti, F. P., Beirão da Veiga, L., Mora, D., and Verani, M. (2014). "A stream Virtual Element formulation of the Stokes problem on polygonal meshe". *SIAM J. Numer. Anal.* 52.1, pp. 386–404 (cited on p. 23).

Archambeau, F., Méchitoua, N., and Sakiz, M. (2004). "Code Saturne: A Finite Volume Code for Turbulent flows - Industrial Applications". *Int. J. Finite Vol.* 1.1 (cited on pp. 12, 75).

Arioli, M. (2013). "Generalized Golub–Kahan bidiagonalizationand stopping criteria". *SIAM J. Matrix Anal. Appl.* 34.2, pp. 571–592 (cited on pp. 24, 26, 68).

Arioli, M., Kruse, C., Ulrich, R., and Tardieu, N. (2018). *An iterative generalized Golub-Kahan algorithm for problems in structural mechanics.* Tech. rep. August (cited on p. 26).

Arnold, D. N. (1982). "An interior penalty Finite Element Method with discontinuous Elements". *SIAM J. Numer. Anal.* 19.4, pp. 742–760 (cited on p. 21).

Arnold, D. N., Brezzi, F., and Fortin, M. (1984). "A stable finite element for the Stokes equations". *Calcolo* 21.4, pp. 337–344 (cited on p. 20).

Arnold, D. N., Brezzi, F., Cockburn, B., and Marini, L. D. (2001). "Unified analysis of discontinuous Galerkin methods for elliptic problems". *SIAM J. Numer. Anal.* 29.5, pp. 1749–1779 (cited on p. 21).

Arrow, K. J., Hurwicz, L., and Uzawa, H. (1958). *Studies in Linear and Nonlinear Programming.* Standford: Cambridge University Press (cited on pp. 24, 25, 68).

Ayuso de Dios, B., Lipnikov, K., and Manzini, G. (2016). "The nonconforming Virtual Element Method". *ESAIM Math. Model. Numer. Anal.* 50.3, pp. 879–904 (cited on p. 23).

Babuška, I. (1973). "The finite element method with Lagrangian multipliers". *Numer. Math.* 20.3, pp. 179–192 (cited on p. 19).

Balay, S., Gropp, W. D., McInnes, L. C., and Smith, B. F. (1997). "Efficient Management of Parallelism in Object Oriented Numerical Software Libraries". In: *Modern Software Tools in Scientific Computing.* Ed. by E. Arge, A. M. Bruaset, and H. P. Langtangen. Birkhäuser Press, pp. 163–202 (cited on p. 75).

Bassi, F. and Rebay, S. (1997). "A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier-Stokes equations". *J. Comput. Phys.* 131.2, pp. 267–279 (cited on p. 21).

Bazilevs, Y. and Hughes, T. J. R. (2007). "Weak imposition of Dirichlet boundary conditions in fluid mechanics". *Comput. Fluids* 36.1, pp. 12–26 (cited on p. 51).

Becker, R., Capatina, D., Luce, R., and Trujillo, D. (2015). "Finite element formulation of general boundary conditions for incompressible flows". *Comput. Methods Appl. Mech. Eng.* 295, pp. 240–267 (cited on p. 18).

Beirão da Veiga, L., Lipnikov, K., and Manzini, G. (2009a). "Convergence analysis of the High-Order Mimetic Finite Difference method". *Numer. Math.* 113.3, pp. 325–356 (cited on p. 23).

Beirão da Veiga, L., Gyrya, V., Lipnikov, K., and Manzini, G. (2009b). "Mimetic Finite Difference method for the Stokes problem on polygonal meshes". *J. Comput. Phys.* 228.19, pp. 7215–7232 (cited on pp. 22, 48).

Beirão da Veiga, L., Brezzi, F., Cangiani, A., Manzini, G., Marini, L. D., and Russo, A. (2013a). "Basic principles of Virtual Element Methods". *Math. Model. Methods Appl. Sci.* 23.1, pp. 199–214 (cited on p. 23).

Beirão da Veiga, L., Brezzi, F., and Marini, L. D. (2013b). "Virtual elements for linear elasticity problems". *SIAM J. Numer. Anal.* 51.2, pp. 794–812 (cited on p. 23).

Beirão da Veiga, L., Brezzi, F., Marini, L. D., Russo, A., Brezzi, F., and Manzini, G. (2014a). "The Hitchhiker's guide to the Virtual Element Method". *Math. Model. Methods Appl. Sci.* 24.8, pp. 1541–1573 (cited on p. 23).

Beirão da Veiga, L., Lipnikov, K., and Manzini, G. (2014b). *The Mimetic Finite Difference method for elliptic problems.* Vol. 11. Springer (cited on p. 22).

Beirão da Veiga, L., Brezzi, F., Marini, L. D., and Russo, A. (2016a). "$H(div)$ and $H(curl)$-conforming Virtual Element Methods". *Numer. Math.* 133.2, pp. 303–332 (cited on p. 23).

Beirão da Veiga, L., Brezzi, F., Marini, L. D., and Russo, A. (2016b). "Mixed Virtual Element Methods for general second order elliptic problems on polygonal meshes". *ESAIM Math. Model. Numer. Anal.* 50.3, pp. 727–747 (cited on p. 23).

Beirão da Veiga, L., Lovadina, C., and Vacca, G. (2017). "Divergence free Virtual Elements for the Stokes problem on polygonal meshes". *ESAIM Math. Model. Numer. Anal.* 51.2, pp. 509–535 (cited on p. 23).

Beirão da Veiga, L., Lovadina, C., and Vacca, G. (2018). "Virtual Elements for the Navier-Stokes problem on polygonal meshes". *SIAM J. Numer. Anal.* 56.3, pp. 1210–1242 (cited on p. 23).

Benzi, M. and Olshanskii, M. A. (2006). "An augmented Lagrangian-based approach to the Oseen problem". *SIAM J. Sci. Comput.* 28.6, pp. 2095–2113 (cited on pp. 9, 26, 156).

Benzi, M., Golub, G. H., and Liesen, J. (2005). "Numerical solution of saddle point problems". *Acta Numer.* 14, pp. 1–137 (cited on pp. 9, 25, 26, 68, 156).

Benzi, M., Olshanskii, M. A., and Wang, Z. (2011). "Modified augmented Lagrangian preconditioners for the incompressible Navier–Stokes equations". *Int. J. Numer. Methods Fluids* 66, pp. 486–508 (cited on pp. 25, 26, 68).

Bercovier, M. and Engelman, M. S. (1979). "A finite element for the numerical solution of viscous incompressible flows". *J. Comput. Phys.* 30.2, pp. 181–201 (cited on p. 76).

Blanc, Ph. (1999). "Error estimate for a finite volume scheme on a MAC mesh for the Stokes problem". In: *Finite Vol. Complex Appl. II.* Hermes Science Publications Paris, pp. 117–124 (cited on p. 21).

Boffi, D., Brezzi, F., and Fortin, M. (2013). *Mixed finite element methods and applications.* Vol. 44. Springer series in Computational Mathemathics. Heidelberg: Springer (cited on pp. 5, 15, 20).

Bonelle, J. (2014). "Compatible Discrete Operator schemes on polyhedral meshes for elliptic and Stokes equations". PhD thesis. Université Paris-Est (cited on pp. 4–7, 14–16, 22, 34, 40, 43, 44, 46, 48, 65, 155).

Bonelle, J. and Ern, A. (2014). "Analysis of Compatible Discrete Operator schemes for elliptic problems on polyhedral meshes". *ESAIM Math. Model. Numer. Anal.* 48.2, pp. 553–581 (cited on pp. 4, 5, 14, 22, 40, 46, 48).

Bonelle, J. and Ern, A. (2015). "Analysis of Compatible Discrete Operator Schemes for the Stokes Equations on Polyhedral Meshes". *IMA J. Numer. Anal.* 35.4, pp. 1672–1697 (cited on pp. 4, 5, 14, 15, 155).

Bonelle, J., Di Pietro, D. A., and Ern, A. (2015). "Low-order reconstruction operators on polyhedral meshes: application to compatible discrete operator schemes". *Comput. Aided Geom. Des.* 35, pp. 27–41 (cited on pp. 16, 43).

Bonelle, J., Ern, A., and Milani, R. (2020). "Compatible Discrete Operator schemes for the steady incompressible Stokes and Navier–Stokes equations". In: *Finite Vol. Complex Appl. IX; Methods Theor. Aspects.* Ed. by R. Klöfkorn, E. Keilegavlen, F. A. Radu, and J. Fuhrmann. Vol. 323. Springer Proc. Math. Stat. Bergen: Springer International Publishing, pp. 93–101 (cited on pp. 7, 34, 62, 78, 80).

Bossavit, A. (1988). “Whitney forms: A class of finite elements for three-dimensional computations in electromagnetism”. *IEE Proc. A Phys. Sci. Meas. Instrumentation. Manag. Educ. Rev.* 135, pp. 493–500 (cited on pp. 4, 14).

Botella, O. and Peyret, R. (1998). “Benchmark spectral results on the lid-driven cavity flow”. *Comput. Fluids* 27.4, pp. 421–433 (cited on pp. 84, 86–91).

Botti, L., Di Pietro, D. A., and Droniou, J. (2019). “A Hybrid High-Order method for the incompressible Navier–Stokes equations based on Temam’s device”. *J. Comput. Phys.* 376, pp. 786–816 (cited on pp. 24, 58, 85).

Boyer, F., Krell, S., and Nabet, F. (2015). “Inf-Sup stability of the discrete duality finite volume method for the 2D Stokes problem”. *Math. Comput.* 84.296, pp. 2705–2742 (cited on p. 23).

Boyer, F., Krell, S., and Nabet, F. (2017). “Benchmark Session: The 2D Discrete Duality Finite Volume method”. In: *Finite Vol. Complex Appl. VIII; Methods Theor. Aspects.* Vol. 199. Springer Proc. Math. Stat. Lille: Springer International Publishing, pp. 181–202 (cited on pp. 77, 78).

Branin, F. H. (1966). *The algebraic-topological basis for network analogies and the vector calculus* (cited on pp. 4, 14).

Brezzi, F. (1974). “On the existence, uniqueness and approximation of saddle-point problems arising from Lagrangian multipliers”. *RAIRO* 8, pp. 129–151 (cited on p. 19).

Brezzi, F., Marini, L. D., and Süli, E. (2004). “Discontinuous Galerkin methods for first-order hyperbolic problems”. *Math. Model. Methods Appl. Sci.* 14.12, pp. 1893–1903 (cited on p. 51).

Brezzi, F., Lipnikov, K., and Shashkov, M. (2005a). “Convergence of Mimetic Finite Difference Method for Diffusion Problems on Polyhedral Meshes”. *SIAM J. Numer. Anal.* 43.5, pp. 1872–1896 (cited on p. 22).

Brezzi, F., Lipnikov, K., and Simoncini, V. (2005b). “A family of Mimetic Finite Difference methods on polygonal and polyhedral meshes”. *Math. Model. Methods Appl. Sci.* 15.10, pp. 1533–1551 (cited on p. 22).

Brezzi, F., Buffa, A., and Lipnikov, K. (2009). “Mimetic Finite Differences for elliptic problems”. *ESAIM Math. Model. Numer. Anal.* 43, pp. 277–295 (cited on pp. 22, 39, 40).

Brezzi, F., Falk, R. S., and Marini, L. D. (2014). “Basic principles of mixed Virtual Element Methods”. *ESAIM Math. Model. Numer. Anal.* 48.4, pp. 1227–1240 (cited on p. 23).

Bruneau, C.-H. and Saad, M. (2006). “The 2D lid-driven cavity problem revisited”. *Comput. Fluids* 35, pp. 326–348 (cited on pp. 84, 86, 87, 90, 91).

Burggraf, O. R. (1966). “Analytical and numerical studies of the structure of steady separated flows”. *J. Fluid Mech.* 24.1, pp. 113–151 (cited on p. 81).

Burman, E. (2012). “A penalty-free nonsymmetric Nitsche-type method for the weak imposition of boundary conditions”. *SIAM J. Numer. Anal.* 50.4, pp. 1959–1981 (cited on p. 64).

Cancès, C. and Omnes, P., eds. (2017). *Finite Volumes for Complex Applications VIII - Methods and Theoretical Aspects. FVCA 8, International Symposium.* Vol. 199. Springer Proc. Math. Stat. Lille, France: Springer International Publishing (cited on pp. 75–77).

Cangiani, A., Gyrya, V., and Manzini, G. (2016). “The nonconforming Virtual Element Method for the Stokes equations”. *SIAM J. Numer. Anal.* 54.6, pp. 3411–3435 (cited on p. 23).

Cantin, P. (2016). “Approximation of scalar and vector transport problems on polyhedral meshes”. PhD thesis. Université Paris-Est (cited on pp. 4, 5, 14, 15, 40).

Cantin, P. and Ern, A. (2016). “Vertex-based Compatible Discrete Operator schemes on polyhedral meshes for advection-diffusion equations”. *Comput. Methods Appl. Math.* 16.2, pp. 187–212 (cited on pp. 4, 5, 14, 15).

Cantin, P. and Ern, A. (2017). "An edge-based scheme on polyhedral meshes for vector advection-reaction equations". *ESAIM Math. Model. Numer. Anal.* 51.5, pp. 1561–1581 (cited on pp. 4, 5, 14, 15).

Cantin, P., Bonelle, J., Burman, E., and Ern, A. (2016). "A vertex-based scheme on polyhedral meshes for advection–reaction equations with sub-mesh stabilization". *Comput. Math. with Appl.* 72.9, pp. 2057–2071 (cited on pp. 4, 5, 14, 15).

Cazemier, W., Verstappen, R. W., and Veldman, A. E. (1998). "Proper orthogonal decomposition and low-dimensional models for driven cavity flows". *Phys. Fluids* 10.7, pp. 1685–1699 (cited on p. 84).

Charnyi, S., Heister, T., Olshanskii, M. A., and Rebholz, L. G. (2017). "On conservation laws of Navier–Stokes Galerkin discretizations". *J. Comput. Phys.* 337, pp. 289–308 (cited on pp. 18, 58, 132).

Cheng, H. M. and Droniou, J. (2019). "An HMM–ELLAM scheme on generic polygonal meshes for miscible incompressible flows in porous media". *J. Pet. Sci. Eng.* 172, pp. 707–723 (cited on p. 22).

Chénier, E., Eymard, R., Gallouët, T., and Herbin, R. (2015). "An extension of the MAC scheme to locally refined meshes: convergence analysis for the full tensor time-dependent Navier–Stokes equations". *Calcolo* 52.1, pp. 69–107 (cited on p. 21).

Chippada, S., Dawson, C. N., Martinez, M. L., and Wheeler, M. F. (1997). "A Projection Method for Constructing A Mass Conservative Velocity Field". *Comput. Methods Appl. Mech. Eng.* 157, pp. 1–10 (cited on pp. 9, 156).

Chorin, A. J. (1967). "A Numerical Method for Solving Incompressible Viscous Flow Problems". *J. Comput. Phys.* 2, pp. 12–26 (cited on p. 31).

Chorin, A. J. (1968). "Numerical Solution of the Navier–Stokes Equations". *Math. Comput.* 22.104, pp. 745–762 (cited on p. 28).

Chorin, A. J. (1969). "On the convergence of discrete approximations to the Navier–/Stokes equations". *Math. Comput.* 23.106, pp. 341–353 (cited on p. 28).

Ciarlet, Ph. G. (1978). *The Finite Element Method for Elliptic Problems.* Amsterdam: North Holland (cited on p. 40).

Cockburn, B., Dong, B., Guzmán, J., Restelli, M., and Sacco, R. (2009a). "A hybridizable discontinuous Galerkin method for steady-state convection-diffusion-reaction problems". *SIAM J. Sci. Comput.* 31.5, pp. 3827–3846 (cited on p. 23).

Cockburn, B., Gopalakrishnan, J., and Lazarov, R. (2009b). "Unified hybridization of discontinuous Galerkin, mixed, and continuous Galerkin methods for second order elliptic problems". *SIAM J. Numer. Anal.* 47.2, pp. 1319–1365 (cited on p. 23).

Cockburn, B., Nguyen, N. C., and Peraire, J. (2010). "A Comparison of HDG Methods for Stokes Flow". *J. Sci. Comput.* 45, pp. 215–237 (cited on p. 23).

Cockburn, B., Di Pietro, D. A., and Ern, A. (2016). "Bridging the Hybrid High-Order and Hybridizable Discontinuous Galerkin methods". *ESAIM Math. Model. Numer. Anal.* Polyhedral discretization for PDE 50.3, pp. 635–650 (cited on pp. 23, 24).

Crouzeix, M. and Raviart, P.-A. (1973). "Conforming and Nongonforming Finite Element Methods for Solving the Stationary Stokes Equations I". *RAIRO* 7, pp. 33–75 (cited on pp. 20, 22).

Delcourte, S. (2007). "Développement de méthodes de volumes finis pour la mécanique des fluides". French. PhD thesis. Université Paul Sabatier (cited on p. 23).

Delcourte, S. and Omnes, P. (2017). "Numerical results for a Discrete Duality Finite Volume discretization applied to the Navier–Stokes equations". In: *Finite Vol. Complex Appl. VIII; Methods Theor. Aspects.* Vol. 199. Springer Proc. Math. Stat. Lille: Springer International Publishing, pp. 141–161 (cited on pp. 76, 78).

Di Pietro, D. A. and Ern, A. (2011). *Mathematical Aspects of Discontinuous Galerkin Methods*. Ed. by G. Allaire and J. Garnier. Vol. 69. Mathématiques & applications. Springer Science & Business Media, p. 383 (cited on pp. 21, 39, 40, 50–54).

Di Pietro, D. A. and Ern, A. (2015). "A Hybrid High-Order locking-free method for linear elasticity on general meshes". *Comput. Methods Appl. Mech. Eng.* 283, pp. 1–21 (cited on pp. 6, 16, 23, 24, 39).

Di Pietro, D. A. and Ern, A. (2017). "Arbitrary-order mixed methods for heterogeneous anisotropic diffusion on general meshes". *IMA J. Numer. Anal.* 37.1, pp. 40–63 (cited on p. 23).

Di Pietro, D. A. and Krell, S. (2018). "A Hybrid High-Order Method for the Steady Incompressible Navier–Stokes Problem". *J. Sci. Comput.* 74.3, pp. 1677–1705 (cited on pp. 24, 59).

Di Pietro, D. A. and Lemaire, S. (2015). "An extension of the Crouzeix-Raviart space to general meshes with application to quasi-incompressible linear elasticity and Stokes flow". *Math. Comput.* 84.291, pp. 1–31 (cited on pp. 22, 44).

Di Pietro, D. A., Ern, A., and Lemaire, S. (2014). "An Arbitrary-Order and Compact-Stencil Discretization of Diffusion on General Meshes Based on Local Reconstruction Operators". *Comput. Methods Appl. Math.* 14.4, pp. 461–472 (cited on pp. 6, 16, 23, 24, 44, 60).

Di Pietro, D. A., Droniou, J., and Ern, A. (2015). "A discontinuous-skeletal method for advection-diffusion-reaction on general meshes". *SIAM J. Numer. Anal.* 53.5, pp. 2135–2157 (cited on pp. 24, 51, 53, 58).

Di Pietro, D. A., Ern, A., Linke, A., and Schieweck, F. (2016). "A discontinuous skeletal method for the viscosity-dependent Stokes problem". *Comput. Methods Appl. Mech. Eng.* 306, pp. 175–195 (cited on pp. 8, 24, 48, 49, 63, 64).

Di Pietro, D. A., Droniou, J., and Manzini, G. (2018). "Discontinuous skeletal gradient discretisation methods on polytopal meshes". *J. Comput. Phys.* 355 (cited on p. 24).

Domelevo, K. and Omnes, P. (2005). "A finite volume method for the Laplace equation on almost arbitrary two-dimensional grids". *Math. Model. Numer. Anal.* 39.6, pp. 1203–1249 (cited on p. 22).

Droniou, J. and Eymard, R. (2006). "A mixed finite volume scheme for anisotropic diffusion problems on any grid". *Numer. Math.* 105.1, pp. 35–71 (cited on pp. 22, 44).

Droniou, J. and Eymard, R. (2009). "Study of the mixed finite volume method for Stokes and Navier–Stokes equations". *Numer. Methods Partial Differ. Equ.* 25.1, pp. 137–171 (cited on pp. 22, 58).

Droniou, J. and Eymard, R. (2017). "Benchmark: Two hybrid mimetic mixed schemes for the lid-driven cavity". In: *Finite Vol. Complex Appl. VIII; Methods Theor. Aspects.* Vol. 199. Springer Proc. Math. Stat. Lille: Springer International Publishing, pp. 107–124 (cited on p. 58).

Droniou, J., Eymard, R., Gallouët, T., and Herbin, R. (2010). "A unified approach to Mimetic Finite Difference, Hybrid Finite Volume and Mixed Finite Volume methods". *Math. Model. Methods Appl. Sci.* 20.2, pp. 265–295 (cited on p. 22).

Droniou, J., Eymard, R., Gallouët, T., and Herbin, R. (2013). "Gradient schemes: a generic framework for the discretisation of linear, nonlinear and nonlocal elliptic and parabolic equations". *Math. Model. Methods Appl. Sci.* 23.13, pp. 2395–2432 (cited on p. 22).

Droniou, J., Eymard, R., and Feron, P. (2015). "Gradient Schemes for Stokes problem". *IMA J. Numer. Anal.* 36.4, pp. 1636–1669 (cited on pp. 22, 48).

Droniou, J., Eymard, R., Gallouët, T., Guichard, C., and Herbin, R. (2018). *The Gradient Discretisation method*. Vol. 82. Springer International Publishing, p. 497 (cited on p. 22).

Elman, H. C., Silvester, D. J., and Wathen, A. J. (2014). *Finite elements and fast iterative solvers: with applications in incompressible fluid dynamics.* Oxford University Press, USA (cited on p. 25).

Engelman, M. S., Strang, G., and Bathe, K.-J. (1981). "The application of quasi-Newton methods in fluid mechanics". *Int. J. Numer. Methods Eng.* 17.5, pp. 707–718 (cited on p. 27).

Ern, A. and Guermond, J.-L. (2004). *Theory and Practice of Finite Element.* Vol. 159. Applied mathematics sciences. New York, NY: Springer Science & Business Media (cited on pp. 19, 20, 28, 39, 54, 64, 67, 75, 94).

Ern, A. and Guermond, J.-L. (2006a). "Discontinuous Galerkin methods for Friedrichs' systems. Part I. General theory". *SIAM J. Numer. Anal.* 44.2, pp. 753–778 (cited on pp. 21, 53).

Ern, A. and Guermond, J.-L. (2006b). "Discontinuous Galerkin methods for Friedrichs' systems. Part II. Second-order elliptic PDEs". *SIAM J. Numer. Anal.* 44.6, pp. 2363–2388 (cited on p. 21).

Ern, A. and Guermond, J.-L. (2008). "Discontinuous Galerkin methods for Friedrichs' systems. Pert III. Multifield theories with partial coercivity". *SIAM J. Numer. Anal.* 46.2, pp. 776–804 (cited on p. 21).

Ern, A. and Guermond, J.-L. (2020). *Finite Elements. Volume III: First-Order and Time-Dependent PDEs.* (in press). Springer (cited on pp. 31, 100, 133, 134).

Erturk, E., Corke, T. C., and Gökçöl, C. (2005). "Numerical solutions of 2-D steady incompressible driven cavity flow at high Reynolds numbers". *Int. J. Numer. Methods Fluids* 48.7, pp. 747–774 (cited on p. 84).

Eymard, R., Gallouët, T., and Herbin, R. (2000). "Finite Volume Methods". In: *Solution of Equation in $\mathbb{R}^n$ (Part 3), Techniques of Scientific Computing (Part 3).* Ed. by J. Lions and Ph. Ciarlet. Vol. 7. Handb. Numer. Anal. Elsevier, pp. 713–1020 (cited on p. 21).

Eymard, R., Herbin, R., and Latché, J. (2006). "On a stabilized colocated Finite Volume scheme for the Stokes problem". *ESAIM Math. Model. Numer. Anal.* EDP Sciences 40.3, pp. 501–527 (cited on p. 21).

Eymard, R., Herbin, R., and Latché, J. (2007). "Convergence analysis of a colocated finite volume scheme for the incompressible Navier-Stokes equations on general 2D or 3D meshes". *SIAM J. Numer. Anal.* 45.1, pp. 1–36 (cited on p. 21).

Eymard, R., Gallouët, T., and Herbin, R. (2010). "Discretisation of heterogeneous and anisotropic diffusion problems on general nonconforming meshes. SUSHI: a scheme using stabilisation and hybrid interfaces". *IMA J. Numer. Anal.* 30.4, pp. 1009–1043 (cited on pp. 6, 16, 22, 39, 44, 46, 59).

Eymard, R., Feron, P., and Guichard, C. (2018). "Family of convergent numerical schemes for the incompressible Navier–Stokes equations". *Math. Comput. Simul.* 144, pp. 196–218 (cited on pp. 22, 58, 59).

Feron, P. (2016). "Gradient Schemes for some elliptic and parabolic, linear and non-linear problems". PhD thesis. université Paris-Est, p. 132 (cited on p. 22).

Fořt, J., Fürst, J., Halama, J., Herbin, R., and Hubert, F., eds. (2011). *Finite Volumes for Complex Applications VI - Problems & Perspectives. FVCA 6, International Symposium.* Vol. 4. Springer Proc. Math. Stat. Prague, Czech Republic: Springer Science & Business Media (cited on p. 72).

Fortin, M. and Glowinski, R. (1983). *Augmented Lagrangian methods: applications to the numerical solution of boundary-value problems.* Vol. 15. Studies in mathematics and its applications. Amsterdam: Elsevier, p. 340 (cited on pp. 24, 26, 33).

Freund, J. and Stenberg, R. (1995). "On weakly imposed boundary conditions for second order problems". In: *Proc. Ninth Int. Conf. Finite Elem. Fluids.* Venice, pp. 327–336 (cited on pp. 64, 71).

Gallouët, T., Herbin, R., Latché, J.-C., and Mallem, K. (2016). "Convergence of the Marker-and-Cell scheme for the steady-state incompressible Navier–Stokes equations on non-uniform grids". *Found. Comput. Math.* 18.1, pp. 249–289 (cited on p. 21).

Galvin, K. J., Linke, A., Rebholz, L. G., and Wilson, N. E. (2012). "Stabilizing poor mass conservation in incompressible flow problems with large irrotational forcing and application to thermal convection". *Comput. Methods Appl. Mech. Eng.* 237-240, pp. 166–176 (cited on pp. 9, 33, 156).

Gatica, G. N., Munar, M., and Sequeira, F. A. (2018). "A mixed Virtual Element Method for the Navier–Stokes equations". *Math. Model. Methods Appl. Sci.* 28.14, pp. 2863–2904 (cited on p. 23).

Ghia, U., Ghia, K. N., and Shin, C. T. (1982). "High-Re solutions for incompressible flow using the Navier–Stokes equations and a multigrid method". *J. Comput. Phys.* 48.3, pp. 387–411 (cited on pp. 84, 86–88, 91).

Girault, V. and Raviart, P.-A. (1986). *Finite Element Methods for Navier–Stokes Equations: Theory and Algorithms.* Vol. 5. Berlin: Springer-Verlag (cited on p. 20).

Glowinski, R. and Le Tallec, P. (1989). *Augmented Lagrangian and operator-splitting methods in nonlinear mechanics.* Vol. 9. Studies in applied mathematics. Philadelphia, PA: SIAM (cited on pp. 25, 26, 68).

Goda, K. (1979). "A multistep technique with implicit difference schemes for calculating two- or three-dimensional cavity flows". *J. Comput. Phys.* 30.1, pp. 76–95 (cited on p. 29).

Golub, G. H. and Kahan, W. (1965). "Calculating the singular values and pseudo-inverse of a matrix". *SIAM J. Numer. Anal.* 2.2, pp. 205–224 (cited on p. 26).

Goudon, T., Krell, S., and Lissoni, G. (2019). "DDFV method for Navier–Stokes problem with outflow boundary conditions". *Numer. Math.* 142.1, pp. 55–102 (cited on p. 23).

Gresho, Ph. M. (1990). "On the theory of semi-implicit projection methods for viscous incompressible flow and its implementation via a finite element method that also introduces a nearly consistent mass matrix. Part 1: Theory". *Int. J. Numer. Methods Fluids* 11.5, pp. 587–620 (cited on p. 28).

Guermond, J.-L. and Minev, P. D. (2015). "High-order time stepping for the incompressible Navier–Stokes equations". *SIAM J. Sci. Comput.* 37.6, A2656–A2681 (cited on pp. 8, 31, 32, 35, 100, 105, 134–136).

Guermond, J.-L. and Minev, P. D. (2017). "High-order time stepping for the Navier–Stokes equations with minimal computational complexity". *J. Comput. Appl. Math.* 310, pp. 92–103 (cited on pp. 32, 33).

Guermond, J.-L. and Shen, J. (2003). "Velocity-correction projection methods for incompressible flows". *SIAM J. Numer. Anal.* 41.1, pp. 112–134 (cited on p. 29).

Guermond, J.-L., Minev, P. D., and Shen, J. (2006). "An overview of projection methods for incompressible flows". *SIAM J. Sci. Comput.* 195, pp. 6011–6045 (cited on pp. 28, 29).

Gyrya, V. and Lipnikov, K. (2008). "High-Order Mimetic Finite Difference Method for diffusion problems on polygonal meshes". *J. Comput. Phys.* 227.20, pp. 8841–8854 (cited on p. 23).

Harlow, F. H. and Welch, E. J. (1965). "Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface". *Phys. Fluids* 8.12, pp. 2182–2189 (cited on p. 21).

Hermeline, F. (1998). “Une méhode de volumes finis pour les équations elliptiques du second ordre”. French. *Comptes Rendus l’Academie des Sci. - Ser. I Math.* 326.12, pp. 1433–1436 (cited on p. 22).

Hermeline, F. (2000). “A Finite Volume Method for the Approximation of Diffusion Operators on Distorted Meshes”. *J. Comput. Phys.* 160.2, pp. 481–499 (cited on p. 22).

Hestenes, M. R. (1969). “Multiplier and Gradient Methods”. *J. Optim. Theory Appl.* 4.5, pp. 303–320 (cited on pp. 24, 25, 33, 68).

Heywood, J. G. and Rannacher, R. (1990). “Finite-Element approximation of the nonstationary Navier–Stokes problem. IV. Error analysis for second-order time discretization”. *SIAM J. Numer. Anal.* 27.2, pp. 353–384 (cited on p. 132).

Hyman, J. M. and Shashkov, M. (1997). “Natural discretizations for the divergence, gradient, and curl on logically rectangular grids”. *Components* 33.4, pp. 81–104 (cited on p. 22).

Hyman, J. M., Morel, J. E., Shashkov, M., and Steinberg, S. (2002). “Mimetic finite difference methods for diffusion equations”. *Comput. Geosci.* 6.3-4, pp. 333–352 (cited on p. 22).

John, V., Linke, A., Merdon, C., Neilan, M., and Rebholz, L. G. (2017). “On the divergence constraint in mixed finite element methods for incompressible flows”. *SIAM Rev.* 59.3, pp. 492–544 (cited on pp. 8, 64).

Juntunen, M. and Stenberg, R. (2009). “Nitsche’s method for general boundary conditions”. *Math. Comput.* 78.267, pp. 1353–1374 (cited on p. 64).

Kim, J. and Moin, P. (1985). “Application of a Fractional-Step Method to Incompressible Navier–Stokes Equations”. *J. Comput. Phys.* 59, pp. 308–323 (cited on pp. 29, 132).

Krell, S. (2011a). “Stabilized DDFV schemes for Stokes problem with variable viscosity on general 2D meshes”. *Numer. Methods Partial Differ. Equ.* 27.6, pp. 1666–1706 (cited on p. 23).

Krell, S. (2011b). “Stabilized DDFV Schemes For The Incompressible Navier-Stokes Equations”. In: *Finite Vol. Complex Appl. VI Probl. Perspect.* Ed. by J. Fořt, J. Fürst, J. Halama, R. Herbin, and F. Hubert. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 605–612 (cited on p. 23).

Krell, S. and Manzini, G. (2012). “The Discrete Duality Finite Volume method for Stokes equations on three-dimensional polyhedral meshes”. *SIAM J. Numer. Anal.* 50.2, pp. 808–837 (cited on p. 23).

Kron, G. (1945). “Numerical solution of ordinary and Partial Differential Equations by means of equivalent circuits”. *J. Appl. Phys.* 16.3, pp. 172–186 (cited on pp. 4, 14).

Kron, G. (1953). “A set of principles to interconnect the solutions of physical systems”. *J. Appl. Phys.* 24 (cited on pp. 4, 14).

Kuznetsov, Yu., Lipnikov, K., and Shashkov, M. (2004). “The Mimetic Finite Difference method on polygonal meshes for diffusion-type problems”. *Comput. Geosci.* 8.4, pp. 301–324 (cited on p. 22).

Ladyzhenskaya, O. A. (1969). *The mathematical theory of viscous incompressible flow.* Russian. Vol. 2. New York, NY: Gordon and Breach (cited on p. 31).

Layton, W., Manica, C. C., Neda, M., Olshanskii, M. A., and Rebholz, L. G. (2009). “On the accuracy of the rotation form in simulations of the Navier–Stokes equations”. *J. Comput. Phys.* 228.9, pp. 3433–3447 (cited on p. 33).

Lederer, Ph. L., Linke, A., Merdon, C., and Schöberl, J. (2017). “Divergence-free reconstruction operators for pressure-robust Stokes discretizations with continuous pressure finite elements”. *SIAM J. Numer. Anal.* 55.5, pp. 1291–1314 (cited on pp. 8, 64).

Lesaint, P. and Raviart, P.-A. (1974). “On a Finite Element Method for solving the neutron transport equation”. In: *Math. Asp. Finite Elem. Partial Differ. Equations.* Vol. 33. New York: Academic Press, pp. 89–123 (cited on p. 21).

Linke, A. (2009). “Collision in a cross-shaped domain – A steady 2D Navier–Stokes example demonstrating the importance of mass conservation in CFD”. *Comput. Methods Appl. Mech. Eng.* 198.41-44, pp. 3278–3286 (cited on pp. 9, 156).

Linke, A. (2014). “On the role of the Helmholtz decomposition in mixed methods for incompressible flows and a new variational crime”. *Comput. Methods Appl. Mech. Engrg.* 268, pp. 782–800 (cited on pp. 8, 64).

Lipnikov, K. and Manzini, G. (2014). “A High-Order Mimetic method on unstructured polyhedral meshes for the diffusion equation”. *J. Comput. Phys.* 272, pp. 360–385 (cited on p. 23).

Nguyen, N. C., Peraire, J., and Cockburn, B. (2009). “An implicit high-order hybridizable discontinuous Galerkin method for nonlinear convection-diffusion equations”. *J. Comput. Phys.* 228.23, pp. 8841–8855 (cited on p. 23).

Nguyen, N. C., Peraire, J., and Cockburn, B. (2011). “An implicit high-order hybridizable discontinuous Galerkin method for the incompressible Navier–Stokes equations”. *J. Comput. Phys.* 230.4, pp. 1147–1170 (cited on p. 23).

Nicolaides, R. A. (1989). “Flow discretization by complementary volume techniques”. In: *9th Comput. Fluid Dyn. Conf. 1989.* American Institute of Aeronautics and Astronautics Inc (cited on p. 21).

Nicolaides, R. A. (1992). “Analysis and convergence of the MAC scheme I. The linear problem”. *SIAM J. Numer. Anal.* 29.6, pp. 1579–1591 (cited on p. 21).

Nicolaides, R. A. and Wu, X. (1996). “Analysis and convergence of the MAC scheme II. Navier-Stokes equations”. *Math. Comput.* 65.213, pp. 29–44 (cited on p. 21).

Nitsche, J. (1971). “Über ein Variationsprinzip zur Lösung von Dirichlet-Problemen bei Verwendung von Teilräumen, die keinen Randbedingungen unterworfen sind”. German. *Abhandlungen aus dem Math. Semin. der Univ. Hambg.* 36.1, pp. 9–15 (cited on pp. 64, 71).

Notay, Y. (2010). “An aggregation-based algebraic multigrid method”. *Electron. Trans. Numer. Anal.* 37, pp. 123–146 (cited on pp. 79, 106).

Olshanskii, M. A. (2002). “A low order Galerkin finite element method for the Navier–Stokes equations of steady incompressible flow: a stabilization issue and iterative methods”. *Comput. Methods Appl. Mech. Eng.* 191, pp. 5515–5536 (cited on p. 33).

Olshanskii, M. A. and Benzi, M. (2008). “An Augmented Lagrangian approach to linearized problem in hydrodynamic stability”. *SIAM J. Sci. Comput.* 30.3, pp. 1459–1473 (cited on pp. 9, 26, 33, 156).

Olshanskii, M. A. and Reusken, A. (2004). “Grad-Div stablilization for Stokes equations”. *Math. Comput.* 73.248, pp. 1699–1718 (cited on pp. 9, 33, 156).

Patankar, S. V. (1980). *Numerical Heat Transfer and Fluid Flow.* Ed. by J. W. Minkowycs and E. M. Sparrow. Vol. 80. Computational Methods in Mechanics and Thermal Sciences. Mc Graw Hill (cited on p. 21).

Poliashenko, M. and Aidun, C. K. (1995). “A direct method for computation of simple bifurcations”. *J. Comput. Phys.* 121.2, pp. 246–260 (cited on p. 84).

Rannacher, R. and Turek, S. (1992). “Simple nonconforming quadrilateral Stokes element”. *Numer. Methods Partial Differ. Equ.* 8.2, pp. 97–111 (cited on p. 21).

Raviart, P.-A. and Thomas, J.-M. (1977). “Primal Hybrid Finite Element Methods for 2nd order elliptic equations”. *Math. Comput.* 31.138, pp. 391–413 (cited on p. 22).

Reed, W. H. and Hill, T. R. (1973). *Trinagular mesh methods for the neutron trnasport equation.* Tech. rep. Los Alamos, NM: Los Alamos Scientific Laboratory (cited on p. 21).

Saad, Y. (1996). *Iterative Methods for Sparse Linear Systems.* Boston, MA: PWS Publishing (cited on pp. 9, 25).

Schieweck, F. (2008). “On the role of boundary conditions for CIP stabilization of higher order finite elements”. *Electron. Trans. Numeral. Anal.* 32, pp. 1–16 (cited on p. 51).

Shashkov, M. and Steinberg, S. (1995). *Support-Operator Finite-Difference Algorithms for General Elliptic Problems* (cited on p. 22).

Shen, J. (1992a). “On error estimates of projection methods for Navier–Stokes equations. First-order schemes”. *SIAM J. Numer. Anal.* 29.1, pp. 57–77 (cited on p. 29).

Shen, J. (1992b). “On error estimates of some higher order projection and penalty-projection methods for Navier–Stokes equations”. *Numer. Math.* 62, pp. 49–74 (cited on p. 29).

Shen, J. (1993). “A Remark on the Projection-3 Method”. *Int. J. Numer. Methods Fluids* 253.16, pp. 249–253 (cited on p. 29).

Shen, J. (1995). “On error estimates of the Penalty method for unsteady Navier–Stokes equations”. *SIAM J. Numer. Anal.* 32.2, pp. 386–403 (cited on pp. 30, 31).

Shin, D. and Strikwerda, J. C. (1997). “Inf-sup conditions for finite-difference approximations of the stokes equations”. *J. Aust. Math. Soc. Ser. B-Applied Math.* 39.1, pp. 121–134 (cited on p. 21).

Taylor, C. and Hood, P. (1973). “A numerical solution of the Navier–Stokes equations using the finite element technique”. *Comput. Fluids* 1.1, pp. 73–100 (cited on p. 20).

Taylor, G. I. and Green, A. E. (1937). “Mechanism of the production of small eddies from large ones”. *Proc. R. Soc. Lond. A* 158.895, pp. 499–521 (cited on pp. 77, 113, 141).

Temam, R. (1968). “Une méthode d’approximationdes solution des équation de Navier–-Stokes”. French. *Bull. la Société Mathématique Fr.* 96, pp. 115–152 (cited on p. 30).

Temam, R. (1969a). “Sur l’Approximation de la Solution des Equations de Navier–Stokes par la Méthode des Pas Fractionnaires (I)”. French. *Arch. Ration. Mech. Anlysis* 32.2, pp. 135–153 (cited on p. 31).

Temam, R. (1969b). “Sur l’Approximation de la Solution des Equations de Navier–Stokes par la Méthode des Pas Fractionnaires (II)”. French. *Arch. Ration. Mech. Anlysis* 33.5, pp. 377–385 (cited on p. 28).

Temam, R. (1977). *Navier–Stokes Equations: Theory and Numerical Analysis.* Vol. 2. Studies in mathematics and its applications. Amsterdam: North-Holland (cited on pp. 51, 57, 94).

Timmermans, L. J. P., Minev, P. D., and Vosse, Van de, F. N. (1996). “An Approximate Projection Scheme for Incompressible Flow Using Spectral Elements”. *Int. J. Numer. Methods Fluids* 22.7, pp. 673–688 (cited on p. 29).

Tonti, E. (1975). *On the formal structure of physical theories.* Istituto di matematica del Politecnico di Milano (cited on pp. 4, 14).

Turek, S. (1999). *Efficient Solvers for Incompressible Flow Problems.* Vol. 6. Lecture Notes in Computational Science and Engineering. Berlin: Springer Science & Business Media (cited on pp. 21, 25).

Vladimirova, N, Kuznetsov, B, and Yanenko, N. N. (1966). “Numerical calculation of the symmetrical flow of viscous incompressible liquid around a plate”. Russian. *Some Problems in Computational and Applied Mathematics, Nauka* (cited on p. 31).

Wheeler, M. F., Rivière, B., and Guillot, M. J. (2002). “Discontinuous Galerkin methods for mass conservation equations for environmental modeling”. In: *Dev. Water Sci. Comput. Methods Water Resour.* Ed. by W. Gray, S. Hassanizadeh, R. Schotting, and G. Pinder. Vol. 1. Elsevier, pp. 947–955 (cited on pp. 9, 156).

Yanenko, N. N. (1971). *The method of fractional steps.* New York, NY: Springer-Verlag (cited on p. 31).