RÉPUBLIQUE FRANÇAISE
Liberté Égalité Fraternité

Université de Toulouse

Doctorat de l'Université de Toulouse

préparé à l'ISAE-SUPAERO

Approche volume fini centrée au noeud : reformulation du schéma V4 original pour l'adaptation de maillage.

Thèse présentée et soutenue, le 14 janvier 2026 par

Yoan GORSCHKA

École doctorale
MEGEP - Mécanique, Energétique, Génie civil, Procédés

Spécialité
Dynamique des fluides

Unité de recherche
ISAE-ONERA EDyF Energétique et Dynamique des Fluides

Thèse dirigée par
Guillaume PUIGT

Composition du jury
M. Rodolphe TURPAULT, Président, Bordeaux INP
M. Christophe BERTHON, Rapporteur, Nantes Université
Mme Elena COLLADO MORATA, Examinatrice, CEA CESTA
M. Bruno KOOBUS, Examinateur, Université de Montpellier
M. Alberto REMIGI, Examinateur, Safran Tech
M. Guillaume PUIGT, Directeur de thèse, ONERA Toulouse

Membres invités
M. Cédric Content , ONERA Fontenay-aux-Roses

À mon grand-père, Professeur Ladislas Raczy

«*Plus l'ascension est longue, plus la montée est difficile, plus grande sera la satisfaction et plus magnifique sera la vue une fois au sommet.*»
Jigorō Kanō

Résumé

La mécanique des fluides numérique, généralement désignée par le terme anglais Computational Fluid Dynamics (CFD), permet de représenter les écoulements par simulation numérique. C'est la stratégie qui permet aujourd'hui de dimensionner les avions ou les moteurs. Pour cela, le domaine d'intérêt est recouvert de cellules qui partagent des nœuds et on parle alors de maillage. Sans connaître *a priori* l'écoulement, le domaine d'étude est maillé par des cellules isotropes. Avec cette stratégie, le nombre de degrés de liberté peut devenir très important, notamment si l'on souhaite suivre des phénomènes anisotropes, comme des chocs ou des tourbillons. Un nombre très important de degrés de liberté rend les simulations numériques impossibles en pratique. L'adaptation de maillage anisotrope consiste à faire converger dans un seul processus le couple (maillage, solution) en s'appuyant sur un critère mathématique de mesure de l'erreur numérique. L'adaptation sur maillage tétraédrique anisotrope permet d'aligner les mailles tétraédriques avec les directions préférentielles de l'écoulement tout en minimisant le nombre de mailles et cette approche démontre tout son intérêt pour des applications en milieu industriel. En conséquence, cette approche nécessite des schémas numériques robustes, puisque l'anisotropie peut dégrader la précision et la robustesse des schémas numériques. L'approche généralement utilisée pour l'adaptation de maillage est l'approche centrée au nœud (ou vertex-centred) dans laquelle les volumes de contrôle sont construits explicitement autour des nœuds du maillage initial. Jusqu'à présent, l'approche exclusivement utilisée est une approche dite edge-based, dans laquelle les inconnues du problème sont localisées aux nœuds du maillage. Le schéma V4, proposé à l'origine par Alain Dervieux, est une extension de l'approche MUSCL de van Leer à la reconstruction au centre des arêtes d'un maillage pour les flux convectif. Cette approche permet de construire un schéma d'ordre 2 au sens de la valeur au nœud sur tout type de maillage. Bien que les résultats récents pour ce schéma numérique montrent une grande robustesse sur des maillages présentant une anisotropie de l'ordre du million, il n'existe pas aujourd'hui de schéma numérique d'ordre 2 au sens des volumes finis (c'est-à-dire où les inconnues du problème sont les valeurs moyennes par cellule duale) pour l'approche centrée au nœud et de type V4 pour les écoulements compressibles. L'objectif de ce travail de thèse est de construire un tel schéma à partir du schéma V4. Bien que les inconnues du problème soient les valeurs moyennes par cellules duales, la construction du schéma V4 se fait à partir des valeurs aux nœuds. Le principe de déconvolution, proposé à l'origine par Ollivier-Gooch, permet de construire la valeur au nœud à partir des valeurs moyennes disponibles. Dans ce travail de thèse, nous avons utilisé des opérateurs proposés par Bernard *et al.*, qui conduisent à la résolution d'un système carré de petite taille. Nous avons d'abord travaillé avec l'équation de transport scalaire. Nous sommes parvenus à montrer que le schéma V4 ainsi défini est bien d'ordre 2 sur tout type de maillage au sens des volumes finis. Ce schéma conserve également ses propriétés de robustesse sur maillage adapté anisotrope. Nous avons également étendu la définition des opérateurs de déconvolution, ce qui permet une reconstruction de la valeur au nœud à l'ordre 3 sur tout type de maillage. Nous avons ensuite traité les équations d'Euler. Les résultats montrent également que le schéma numérique est d'ordre 2 sur maillage quelconque au sens des volumes finis. En se basant sur les résultats de la déconvolution, nous avons également proposé une formulation plus précise des conditions aux limites. Enfin, la formulation volume fini nous permet d'utiliser une quadrature d'ordre 2 pour les flux convectifs, ce qui conduit à une diminution de l'erreur dans le cas de maillage adapté

anisotrope.

Summary

Computational fluid dynamics (CFD) is used to represent fluid flows through numerical simulation. It is the strategy currently used to design aircraft and engines. To do this, the area of interest is covered with cells that share nodes, which is referred to as meshing. Without prior knowledge of the flow, the domain of study is meshed with isotropic cells. With this strategy, the number of degrees of freedom can become very large, especially if one wishes to track anisotropic phenomena such as shocks or vortices. Such a large number of degrees of freedom makes numerical simulations intractable in practice. Anisotropic mesh adaptation consists of converging the pair (mesh, solution) in a single process based on a mathematical criterion for measuring numerical error. Adaptation on an anisotropic tetrahedral mesh allows the tetrahedra to be aligned with the preferred flow directions while minimising the number of elements, and this approach is beneficial for industrial applications. Consequently, this approach requires robust numerical schemes, since anisotropy degrades the accuracy of numerical schemes. The approach generally used for mesh adaptation is the vertex-centred approach, in which control volumes are constructed explicitly around the nodes of the initial mesh. Until now, the approach used exclusively has been an edge-based approach, in which the unknowns of the problem are located at the nodes of the mesh. The V4 scheme, originally proposed by Alain Dervieux, is an extension of van Leer's MUSCL approach to this edge-based formalism for convective flux. This approach allows the construction of a second-order scheme in terms of node value for any mesh composed of simplices. Furthermore, recent results for this numerical scheme show great robustness on meshes with an anisotropy of the order of one million. Although recent results for this numerical scheme show great robustness on meshes with anisotropy of the order of one million, there is currently no second-order numerical scheme in the finite volume sense (*i.e.* where the unknowns of the problem are the averaged values per dual cell) for the vertex-centred V4 approach for compressible flows. The goal of this thesis is to define such a finite volume numerical scheme from the edge-based V4 scheme. Although the unknowns of the problem are the averaged values per dual cell, the V4 scheme is built from the values at the nodes. The principle of deconvolution, originally proposed by Olliver-Gooch, allows the value at the node to be built from the available averaged values. In this thesis, we used operators proposed by Bernard *et al.*, which lead to the resolution of a small square system. We first worked with the scalar advection equation. We were able to show that the V4 scheme thus defined is indeed at order 2 on any mesh in the finite volume sense. This scheme also preserves its robustness properties on adapted anisotropic meshes. We also extended the definition of deconvolution operators, which allows for a third-order reconstruction of the value at the node on any mesh. We then dealt with Euler's equations. The results also show that the numerical scheme is second-order on any mesh in the finite volume sense. Based on the deconvolution results, we also proposed a more precise formulation of the boundary conditions. Finally, the finite volume formulation allows us to use second-order quadrature for convective fluxes, which leads to a reduction in error in the case of anisotropic adapted meshes.

Remerciements

Mes tout premiers remerciements s'adressent à mon directeur de thèse, Guillaume Puigt. Ses connaissances scientifiques ont remarquablement contribué à cette recherche et ses qualités humaines ont rendu la relation de travail très agréable. Qu'il trouve ici l'expression de ma gratitude.

Ce travail de thèse a bénéficié de l'encadrement d'Alberto Remigi. Je tiens à lui exprimer toute ma reconnaissance pour son suivi durant ces trois années, ainsi que pour ses nombreuses idées qui ont contribué positivement à cette étude.

J'ai également bénéficié de l'encadrement de Cédric Content et de Bruno Maugars. Je leur suis reconnaissant pour leur aide et pour leurs conseils durant ces trois années.

Je tiens à exprimer ma reconnaissance envers les membres de mon jury qui ont accepté de juger ce travail de thèse, et je les remercie pour leurs remarques ainsi que pour les échanges que nous avons eus le jour de la soutenance. Je remercie Christophe Berthon et Rodolphe Turpault d'avoir accepté de rapporter ce travail. Je remercie également Elena Colado Morata et Bruno Kobus d'avoir accepté d'examiner ce travail.

Ce travail a été mené en partie au sein de l'équipe STAR de Safran Tech. Je remercie Marc Chauvy pour sa confiance et de m'avoir accueilli au sein de l'équipe. Je remercie vivement tous les membres de l'équipe pour les discussions fructueuses que nous avons eues, pour le soutien qu'ils m'ont apporté et pour leur sympathie.

Ce travail a également été mené au sein de l'équipe CLEF de l'ONERA. Je remercie Matthieu Soismier pour sa confiance et son accueil au sein de l'équipe. Je remercie également tous les membres de l'équipe pour les nombreux échanges que nous avons eus, pour leur aide et pour leur bienveillance.

J'ai eu le plaisir de partager ces années avec d'autres doctorants. Je tiens à remercier chaleureusement Kévin Ancourt, Hugo Dornier, Guillaume du Pont de Romémont, Sébastien Dubois et Rémi Guillot pour leur présence amicale. Sans eux, cette période aurait été bien plus fade.

Je remercie l'équipe GammaO de l'INRIA, à laquelle ce travail de thèse doit beaucoup. Merci à Frédéric Alauzet pour son accueil lors de mes visites à Palaiseau, ainsi que pour les différents échanges que nous avons eus. Un grand merci à Cosimo Tarsia-Morisco pour son investissement considérable dans ce travail de thèse. Nos nombreux échanges ont grandement contribué à cette étude. Je remercie également Andrea Gobbi pour son aide à la validation du code DeepDive. J'adresse toute ma sympathie aux membres de l'équipe.

Je remercie ma famille d'être venue en nombre à Châtillon pour assister à la présentation de mes travaux. Je suis très reconnaissant envers mes amis pour leur présence et leur soutien depuis de nombreuses années. Merci infiniment à mes parents ainsi qu'à mes frères, Antoine et Thomas, pour leur soutien sans faille depuis toujours. Je souhaite enfin exprimer ma plus profonde gratitude à Emma pour sa patience, sa présence et son soutien constant depuis près de dix ans.

Résumé complet

Face à l'urgence climatique et la nécessité de réduire ses émissions de gaz à effet de serre, l'aviation mondiale s'est dotée d'objectifs ambitieux, dont la neutralité carbone à l'horizon 2050. Cet objectif passe entre autres par la réduction de consommation des moteurs de l'aviation civile. Le programme CFM RISE (Revolutionary Innovation for Sustainable Engines) a été lancé le 14 juin 2021 à Paris par General Electric et Safran et ambitionne le développement de technologies permettant la réduction de plus de 20% de la consommation de carburant et des émissions de gaz à effet de serre associés par rapport aux moteurs de l'époque. Ce programme vise à développer une gamme de technologies innovantes et disruptives, avec notamment la technologie open-fan. La mise en service de ce nouveau moteur est prévue à l'horizon 2035. Les turbomachines présentent des phénomènes physiques très complexes qu'il est nécessaire d'étudier et de comprendre si l'on souhaite améliorer leur performance, que ce soit en termes de consommations de carburant, d'émissions de gaz à effet de serre ou de fiabilité. L'étude de la physique des turbomachines peut se faire par des moyens expérimentaux ainsi que par de la simulation numérique. En effet, la mécanique des fluides numérique, généralement désignée par le terme anglais Computational Fluid Dynamics (CFD), permet de représenter les écoulements par simulation numérique. C'est aujourd'hui un outil indispensable dans la stratégie de recherche et développement du groupe Safran. L'étude des phénomènes physiques en turbomachine est rendue possible par la résolution de système à loi de conservation, comme les équations d'Euler ou les équations de Navier-Stokes compressibles. Dans le cadre de cette étude, on se limitera aux équations d'Euler compressibles, qui décrivent l'écoulement d'un fluide parfait. Les équations d'Euler compressibles sans terme source peuvent s'écrire de la façon suivante :

$$\begin{cases} \dfrac{\partial \rho}{\partial t} & + & \overrightarrow{\nabla} \cdot (\rho \mathbf{u}) & = & 0, \\ \dfrac{\partial \rho \mathbf{u}}{\partial t} & + & \overrightarrow{\nabla} \cdot (\rho \mathbf{u} \otimes \mathbf{u}) + \overrightarrow{\nabla} p & = & 0, \\ \dfrac{\partial \rho E}{\partial t} & + & \overrightarrow{\nabla} \cdot (\mathbf{u}(\rho E + p)) & = & 0. \end{cases}$$

Les trois équations décrivent respectivement la conservation de la masse, le bilan de quantité de mouvement ainsi que le bilan d'énergie. Les équations d'Euler peuvent s'écrire sous la forme compacte suivante :

$$\frac{\partial \mathbf{W}}{\partial t} + \overrightarrow{\nabla} \cdot F(\mathbf{W}) = 0; \quad \mathbf{W} = \begin{pmatrix} \rho \\ \rho \mathbf{u} \\ \rho E \end{pmatrix}; \quad F(\mathbf{W}) = \begin{pmatrix} \rho \mathbf{u} \\ \rho \mathbf{u} \otimes \mathbf{u} + p \mathbb{I} \\ \mathbf{u}(\rho E + p) \end{pmatrix},$$

où $\mathbf{W}$ représente les états du problème et $F(\mathbf{W})$ représente le flux convectif. En pratique, les fluides industriels sont visqueux et on résout plutôt les équations de Navier-Stokes. La résolution de ces équations peut se faire à l'aide de différentes stratégies de modélisation. Parmi les nombreuses méthodes de résolution, on peut notamment citer l'approche DNS (Direct Numerical Simulation) qui permet de résoudre toutes les échelles de la turbulence et ne repose donc

sur aucun choix de modélisation. On peut également citer les approches de type LES (Large Eddy Simulation), dans lesquelles on résout les grandes structures de la turbulence, alors que les petites structures sont modélisées. Enfin, les approches de type RANS (Reynolds Averaged Navier-Stokes) modélisent toutes les échelles de la turbulence. Cette liste n'est pas exhaustive et d'autres méthodes existent pour la résolution des équations de Navier-Stokes (URANS, hybride RANS/LES, *etc.*). Malgré l'augmentation continue de la puissance de calcul ces soixante dix dernières années, le coût de calcul associé aux méthodes DNS et LES rend difficilement utilisables ces approches pour des applications industrielles. De ce fait, les approches de type RANS sont aujourd'hui préférées dans l'industrie. Dans le cadre des simulations RANS, le domaine d'intérêt est recouvert de cellules qui partagent des nœuds et on parle alors de maillage. Sans connaître *a priori* l'écoulement, le domaine d'étude est maillé par des cellules isotropes. Avec cette stratégie, le nombre de degrés de liberté peut devenir très important, notamment si l'on souhaite suivre des phénomènes anisotropes, comme des chocs ou des tourbillons. Pour contourner ce problème, des méthodologies d'adaptation de maillage ont été développées. Parmi ces méthodes, on peut notamment citer l'adaptation de maillage anisotrope. L'adaptation de maillage consiste ici à faire converger dans un seul processus le couple (maillage, solution) en s'appuyant sur un critère mathématique de mesure de l'erreur numérique. L'adaptation sur maillage tétraédrique anisotrope permet d'aligner les mailles tétraédriques avec les directions préférentielles de l'écoulement tout en minimisant le nombre de mailles et cette approche démontre tout son intérêt pour des applications en milieu industriel. En contrepartie, l'anisotropie des éléments dégrade la précision et la robustesse des schémas numériques. L'approche généralement utilisée pour l'adaptation de maillage est l'approche centrée au nœud (ou *vertex-centred*) dans laquelle les volumes de contrôle sont construits explicitement autour des nœuds du maillage initial. De plus, l'approche *vertex-centred* bénéficie d'un voisinage de rang 1 plus riche que l'approche centrée cellule, ce qui est intéressant pour le calcul de gradient. Cette approche bénéficie d'un alignement entre les arêtes du maillage et la direction préférentielle de l'écoulement, ce qui qui la rend très pertinente dans l'étude de phénomènes fortement anisotropes, comme les chocs ou les tourbillons. Jusqu'à présent, la discrétisation exclusivement utilisée est une approche dite edge-based, dans laquelle les inconnues du problème sont localisées aux nœuds du maillage. La discrétisation edge-based des équations d'Euler peut s'écrire de la façon suivante :

$$\frac{d\mathbf{W}_i}{dt} + \frac{1}{|C_i|} \sum_{j \in \nu_i} \Phi_{ij} \overrightarrow{n}_{ij} = 0,$$

où Φ_{ij} représente le flux numérique, calculé à l'aide du solveur de Riemann approché de Roe dans le cadre de ce travail. De plus, $|C_i|$ représenté la mesure de la cellule duale C_i, $\overrightarrow{n}_{ij}$ la normale sortante à la cellule, ν_i l'ensemble des cellules voisines de la cellule i et $\mathbf{W}_i$ la valeur de $\mathbf{W}$ au sommet i. Le schéma ainsi défini est un schéma d'ordre 1. Pour augmenter la précision des schémas numériques, van Leer propose une reconstruction par cellule des inconnues du problème. Ainsi, la méthode MUSCL permet d'atteindre une précision à l'ordre 2 en évaluant les flux convectifs à l'aide des valeurs extrapolées à la face plutôt que les valeurs aux nœuds du maillage. Les méthodes d'ordre 2 sont populaires dans un contexte industriel, puisqu'elles offrent un bon compromis entre précision et robustesse. Dans le cas de l'approche *vertex-centred*, les valeurs sont extrapolées au point M_{ij}, le milieu de l'arête ij. Le calcul des valeurs extrapolées nécessite de connaître les gradients aux nœuds du maillage. Plusieurs méthodes coexistent et conduisent à des schémas numériques différents. D'une part, on peut citer le travail de Nishikawa, dans lequel les gradients nodaux sont calculés par une méthode des moindres carrés. Cette méthode repose sur la résolution d'un système linéaire rectangulaire de taille arbitraire. Le schéma numérique ainsi construit est d'ordre 2 pour des maillages composés de simplexes. Cette approche n'est, cependant, que d'ordre 1 sur d'autres types d'éléments, sauf pour des conditions géométriques très spécifiques. On sait également que cette méthode présente des problèmes de robustesse sur des maillages triangulaires présentant un *aspect-ratio* très élevé. Une autre

classe de méthode a été proposée à l'origine par Alain Dervieux, pour des maillages composés de simplexes. Contrairement à l'approche MUSCL originale, les gradients sont calculés par arête à l'aide d'une combinaison linéaire de plusieurs gradients. Plus précisément, pour une arête donnée, on identifie les 2 triangles/tétraèdres les plus alignés avec l'arête. Puis, à partir de ces éléments, on vient calculer des gradients éléments finis sur les triangles/tétraèdres identifiés et on calcule également un gradient centré pour chaque arête du maillage. On reconstruit alors un gradient d'ordre élevé à partir des gradients précédemment calculés en introduisant un paramètre β. Ainsi, le gradient d'ordre élevé est construit dans la direction de l'arête, ce qui fait de ce schéma un schéma 1D dans la direction de chaque arête. On a donc un alignement entre la direction privilégiée de l'écoulement et les arêtes du maillage. C'est cette construction 1D du schéma numérique qui semble la rendre très robuste sur maillage fortement anisotrope. Les gradients sont évaluées de la façon suivante :

$$\begin{cases} \overrightarrow{\nabla \mathbf{W}}_i^{\beta} \cdot \overrightarrow{ij} = (1-\beta)\overrightarrow{\nabla \mathbf{W}}_{ij}^{C} \cdot \overrightarrow{ij} + \beta \overrightarrow{\nabla \mathbf{W}}_{ij}^{U} \cdot \overrightarrow{ij}, \\ \overrightarrow{\nabla \mathbf{W}}_j^{\beta} \cdot \overrightarrow{ij} = (1-\beta)\overrightarrow{\nabla \mathbf{W}}_{ij}^{C} \cdot \overrightarrow{ij} + \beta \overrightarrow{\nabla \mathbf{W}}_{ij}^{D} \cdot \overrightarrow{ij}. \end{cases}$$

Ce schéma est également connu sous le nom de schéma V4 et il est de précision maximale pour $\beta = 1/3$. Nous nous limitons donc à cette valeur dans le cadre de ce travail. Le schéma ainsi construit est d'ordre 2 avec une dissipation numérique d'ordre 4. Pour les arêtes de bord, il n'est pas toujours possible de définir un triangle amont. Pour corriger ce problème, une reconstruction MUSCL faisant intervenir le gradient de Clément a également été proposée. Enfin, les conditions aux limites sont traitées faiblement. Pour ce faire, on calcule un état à l'aide des valeurs aux nœuds et des conditions limites. Puis on calcule un flux analytique pour chaque face duale des cellules de bord. Puisqu'il n'est pas possible de définir une normale en un point du maillage, on calcule dans les faits un flux par facette duale en utilisant la normale à chaque facette duale.

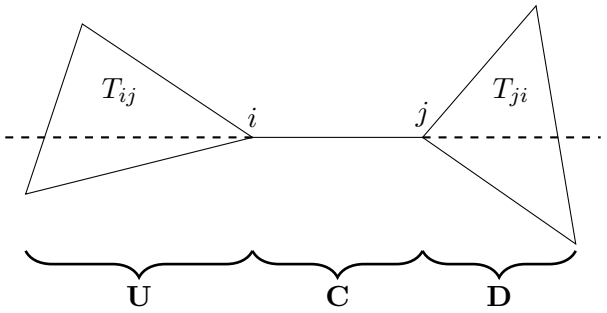


Figure 1: Triangles associés à l'arête ij pour la reconstruction MUSCL

Le schéma V4 étant très robuste sur maillage anisotrope, il est l'un des schémas les plus utilisés en adaptation de maillage aujourd'hui. Nous étudions donc ce schéma sur des cas de complexité croissante. Dans un premier temps, nous traitons le cas de test d'advection linéaire. Cette étude est menée dans le code DeepDive, un code de calcul 2D développé pour les besoins de ce travail. Pour ce cas de test, nous définissons quatre types de maillages (régulier, perturbé, Friedrich-Keller et Union-Jack) et nous menons, pour chaque type de maillage, des études de convergence en ordre pour l'erreur de troncature et l'erreur de discrétisation. Les résultats nous montrent que le schéma V4 est au moins d'ordre 2 sur tout type de maillage si l'on considère une reconstruction MUSCL spécifique pour les arêtes ne présentant pas d'élément amont. Pour le cas particulier du maillage Union-Jack, le schéma V4 est d'ordre 3. Nous définissons également une hiérarchie de maillages adaptés anisotropes afin d'étudier le comportement du schéma V4 sur ce type de maillage. Les résultats montrent une grande robustesse du schéma V4 sur ce type de maillage même si l'ordre 2 n'est pas conservé si l'on se réfère à l'erreur $\mathcal{L}_\infty$. Dans un deuxième temps, nous

étudions le schéma V4 pour les équations d'Euler. Cette étude est menée dans le code SoNICS, le code CFD de nouvelle génération actuellement en cours de développement entre l'ONERA et Safran. Le cas d'advection d'un vortex isentropique nous permet d'étudier le comportement du schéma V4 sur le point courant, puisque dans notre cas, il n'y a pas d'interaction entre le tourbillon et le bord du domaine. Ce cas de test possède une solution analytique pour les équations d'Euler. En effet, le vortex isentropique reste inchangé au cours de son advection à sa vitesse initiale. Ainsi, il est possible de mener une analyse d'erreur en comparant la solution exacte avec la solution issue du calcul CFD. On peut ainsi mesurer la précision et l'ordre de convergence de notre schéma. Pour ce cas de test, nous définissons trois types de maillages (régulier, perturbé et Friedrich-Keller) et pour chaque type de maillage nous menons des études de convergence en ordre pour l'erreur de discrétisation. Les résultats nous montrent également que le schéma V4 est d'ordre 2 sur tout type de maillage. Enfin, nous traitons le cas de test de Ringleb. Le cas de l'écoulement de Ringleb est un cas largement utilisé pour la validation des schémas d'ordre élevé. C'est un cas de test compressible, non visqueux et non conduisant la chaleur pour les équations d'Euler. Par transformation hodographique des équations d'Euler, une solution exacte du problème est connue, qui est une solution stationnaire. Le cas de Ringleb est généralement résolu sur un domaine curviligne représentant un canal autour d'un obstacle. Dans ces conditions, l'écoulement est transsonique, avec une large région supersonique près du nez de l'obstacle, mais sans choc. Ce cas de test nous permet d'inclure le traitement des conditions limites dans l'analyse du schéma V4. Nous définissons deux types de maillages, à savoir des maillages réguliers et des maillages perturbés. Une étude de la convergence en ordre de l'erreur de discrétisation nous permet d'arriver à la même conclusion que pour les cas de test précédents, à savoir que le schéma V4 et d'ordre 2 sur tout type de maillage si l'on considère la valeur au nœud comme solution de référence.

Malgré les bonnes propriétés du schéma V4, nous avons identifié plusieurs limites à la méthode. D'une part, il semble que cette discrétisation est au maximum d'ordre 3 et que donc il ne serait pas possible de construire un schéma numérique edge-based d'ordre supérieur à trois avec un seul flux numérique par arête. Cette limitation proviendrait de la discrétisation de la divergence des flux. D'autre part, le traitement des conditions limites peut être source d'erreur, notamment lorsqu'une cellule duale partage deux conditions aux limites. Pour contourner ces problèmes, nous proposons d'utiliser une discrétisation volumes finis des équations du problème. La discrétisation volumes finis repose sur la formulation intégrale des équations du problème. Pour ce faire, on intègre les équations sur chaque volume dual, ce qui nous donne l'expression suivante :

$$\frac{d}{dt}\int_{C_i} \mathbf{W} d\Omega + \int_{\partial C_i} F(\mathbf{W}) \cdot \overrightarrow{n_{C_i}}\, d\Gamma = 0, \quad \forall i \in \{1, .., N_c\}.$$

Dans le cas de la discrétisation volumes finis, les inconnues du problèmes sont les valeurs moyennes par cellule duale. La construction du schéma numérique assure la conservativité de la discrétisation volumes finis et ce n'est pas exactement le cas de la discrétisation edge-based. La discrétisation volumes finis permet également de construire un schéma numérique de degré de précision arbitraire. Cela passe notamment par la méthode de quadrature utilisée pour le calcul du flux convectif. Une quadrature de Gauss à l'ordre 2 permet de calculer le flux des cellules de bord pour chaque facette duale et non plus par face comme c'est le cas pour la formulation edge-based. Nous faisons le constat qu'actuellement dans la littérature, il n'existe pas de schéma volumes finis V4 vertex-centred pour les équations d'Euler compressibles pensé pour l'adaptation de maillage anisotrope. Or, les schémas volumes finis présentent des avantages en comparaison avec la discrétisation edge-based des équations. L'objectif de ce travail de thèse est de construire un schéma volumes finis centré au nœud qui soit au moins d'ordre 2 au sens des volumes finis sur tout type de maillage. Cette étude étant motivée par l'adaptation

de maillage anisotrope, nous avons pour exigence de construire un schéma qui soit robuste sur maillage fortement anisotrope. Nous nous limitons dans le cadre de cette étude aux maillages composés de simplexes, c'est-à-dire de triangles en 2D et de tétraèdres en 3D. De plus, le but de cette thèse est d'implémenter le nouveau schéma numérique dans le code SoNICS.

Nous souhaitons contruire un schéma volumes finis centré au nœud qui soit robuste sur maillage fortement anisotrope. Les résultats de la littérature ainsi que l'étude que nous avons menée nous montrent que le schéma V4 est robuste sur maillage fortement anisotrope. Cependant, il n'est pas possible d'utiliser le schéma V4 en l'état pour une formulation volumes finis. En effet, les inconnues du problème sont maintenant les valeurs moyennes par celulle duale, mais le schéma V4 nécessite de manipuler les valeurs aux sommets du maillage. Pour illustrer le problème, nous calculons un gradient élément fini sur un maillage perturbé à partir des champs moyens. Les résultats montrent que le gradient ainsi défini est inconsistant et ne converge donc pas en maillage.

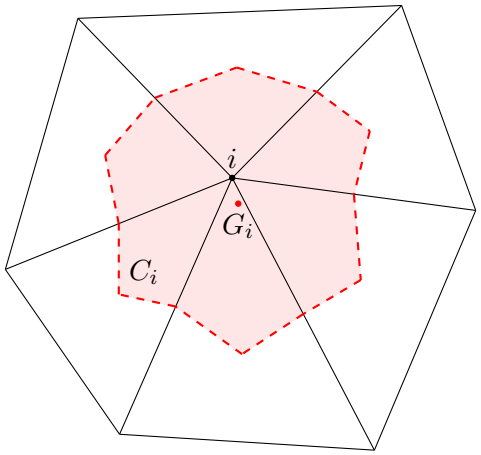


Figure 2: Centre de gravité de la cellule duale sur maillage quelconque

Dans le cadre d'un maillage quelconque, le centre de gravité de la cellule duale et le nœud du maillage ne sont pas confondus. Dans ce cas-là, considérer que la valeur moyenne par cellule duale est égale à la valeur au nœud du maillage conduit à considérer une valeur au nœud d'ordre 1, puisque les moments volumiques d'ordre 1 ne sont pas nuls. Une valeur au nœud d'ordre 1 conduit naturellement à un gradient inconsistant. Cette problématique n'existe pas dans un formalisme centré cellule puisque les moments volumiques d'ordre 1 sont toujours nuls et que donc considérer que la valeur moyenne par cellule duale est égale à la valeur au centre de gravité de la cellule conduit à considérer une valeur au centre de la cellule d'ordre 2. Pour pouvoir utiliser le schéma V4 dans un formalisme centré cellule, il est nécessaire de reconstruire des valeurs au nœud à partir des valeurs moyennes disponibles. Cela est connu sous le nom de déconvolution. À l'aide d'un développement de Taylor intégré sur chaque cellule duale du maillage, il est possible de faire le lien entre les valeurs moyennes connues et la valeur et les dérivées successives que l'on cherche à construire aux nœuds du maillage. Cela fait apparaître les moments volumiques ($\overline{x}_i$ ou $\widehat{x}_{ij}$ par exemple) et permet donc de montrer que si les moments volumiques d'ordre 1 sont nuls alors la valeur au nœud est égale à la valeur moyenne sous une hypothèse d'ordre 2 :

$$\left\{ \begin{array}{rcl} \bar{\mathbf{W}}_i & = & \mathbf{W}|_i + \dfrac{\partial \mathbf{W}}{\partial x}|_i \overline{x}_i + \dfrac{\partial \mathbf{W}}{\partial y}|_i \overline{y}_i + O(h^2), \\ \bar{\mathbf{W}}_j & = & \mathbf{W}|_i + \dfrac{\partial \mathbf{W}}{\partial x}|_i \widehat{x}_{ij} + \dfrac{\partial \mathbf{W}}{\partial y}|_i \widehat{y}_{ij} + O(h^2). \end{array} \right.$$

Plusieurs approches permettant de reconstruire la valeur et les dérivées successives au nœud sont proposées dans la littérature. Par exemple, une formulation générique de la déconvolution a été proposée par Ollivier-Gooch dans un article de 2002. Cette méthode repose sur la méthode des moindres carrés et conduit à la résolution d'un système linéaire rectangulaire de taille arbitraire. La taille du système linéaire dépend du nombre de cellules voisines considéré dans le stencil. Ce nombre doit être au moins égal au nombre de dérivées que l'on cherche à construire. Cependant, il a été montré par Bridel-Bertomeu que pour avoir un bon conditionnement de la matrice du système, il était nécessaire de considérer davantage de voisins. Par exemple, pour une reconstruction d'ordre 2 en 2D, il est nécessaire de considérer, au minimum, 15 voisins pour un maillage cartésien (structuré à pas d'espace constant). La résolution de tel système linéaire pouvant s'avérer coûteuse d'un point de vue numérique, nous avons opté pour une autre approche proposée par Bernard *et al.* dans un article de 2020. Cette approche repose sur des opérateurs compacts. Pour une déconvolution permettant de reconstruire une valeur au nœud d'ordre 2 et un gradient au nœud d'ordre 1, on commence par calculer un gradient Green-Gauss sur chaque cellule duale :

$$\overrightarrow{R}_i = \sum_{j \in \mathcal{L}_i} \frac{\bar{\mathbf{W}}_i + \bar{\mathbf{W}}_j}{2} \overrightarrow{S}_{ij}.$$

Le gradient ainsi défini est une mauvaise approximation du gradient au point i. L'idée est ensuite d'injecter les développements de Taylor intégrés sur chaque cellule duale dans l'expression du gradient. Pour fermer les cellules de bord, nous proposons de reconstruire linéairement un état au centre de chaque facette duale à partir des valeurs moyennes sur les cellules de bord. Ainsi, la méthode nous conduit à résoudre le système linéaire suivant.

$$\begin{pmatrix} \bar{\mathbf{W}}_i \\ \overrightarrow{R}_i(\bar{\mathbf{W}}_i) \end{pmatrix} = \begin{pmatrix} 1 & \overline{x}_i & \overline{y}_i \\ 0 & & \\ 0 & & \bar{\bar{\mathbf{M}}} \end{pmatrix} \begin{pmatrix} \mathbf{W}|_i \\ \frac{\partial \mathbf{W}}{\partial x}|_i \\ \frac{\partial \mathbf{W}}{\partial y}|_i \end{pmatrix}$$

La matrice de ce système linéaire est une matrice carrée de petite taille. Par exemple, pour une reconstruction d'ordre 2 en 2D, elle est de taille 3×3, ce qui contraste avec le système linéaire obtenu avec l'approche d'Ollivier-Gooch. Par construction, elle est toujours bien conditionnée et inversible. Son inverse est appelée matrice de déconvolution. Puisque la matrice de convolution ne dépend que de quantités géométriques (moments volumiques, normales aux cellules duales), il est préférable, d'un point de vue numérique, de ne stocker que la matrice de déconvolution. Nous étudions cette méthode de déconvolution sur des solutions analytiques. Les résultats sont ceux attendus, puisque cette méthode de déconvolution permet de reconstruire, à partir de la valeur moyenne par cellule duale de la solution analytique, une valeur au nœud d'ordre 2 et un gradient au nœud d'ordre 1 sur tout type de maillage, et ce, pour les cellules à l'intérieur du domaine ainsi que pour les cellules de bord. Le papier de Bernard *et al.* propose également un opérateur hessien permettant de reconstruire une valeur au nœud d'ordre 3, un gradient au nœud d'ordre 2 ainsi qu'une hessienne au nœud d'ordre 1. Pour que la matrice de convolution soit inversible en 2D, il est nécessaire qu'un sommet donné ait au moins 5 voisins, car le nombre d'inconnus du problème est alors de 6 (la valeur au nœud, les deux termes de dérivée et les trois termes d'hessienne en tenant compte de la symétrie). Puisque le nombre de voisins peut être insuffisant, notamment pour les cellules de bord, nous proposons d'étendre le stencil pris en compte dans la construction de la matrice de convolution. Cela conduit à pondérer les opérateurs gradient et hessienne. Par exemple, l'opérateur gradient est défini de la façon suivante :

$$\overrightarrow{R}_i(\mathbf{W}_i) = \sum_{j \in \mathcal{L}_i} \Big(\sum_{l \in \mathcal{L}_{ij}} \alpha_{il} \frac{\bar{\mathbf{W}}_i + \bar{\mathbf{W}}_l}{2} \overrightarrow{S}_{ij} \Big).$$

L'extension de stencil ne change pas la taille de la matrice de convolution. Comme pour la déconvolution du deuxième ordre, on teste la méthode sur des solutions analytiques. Les résultats montrent que ces nouveaux opérateurs permettent de reconstruire, à partir des valeurs moyennes disponibles, une valeur au nœud d'ordre 3, un gradient au nœud d'ordre 2 et une hessienne au nœud d'ordre 1. Ces résultats, obtenus sur maillage perturbé, sont valables pour l'ensemble des cellules du maillage.

Les résultats de la déconvolution sont utilisés pour définir le schéma V4 dans un formalisme volumes finis. Ce schéma étant un schéma d'ordre 2, les résultats issus de la déconvolution du deuxième ordre suffisent. En utilisant les valeurs au nœud d'ordre 2 issues de la déconvolution, on est capable de calculer un gradient élément fini d'ordre 1, un gradient centré d'ordre 1 et donc un gradient d'ordre élevé consistant par combinaison linéaire de ces deux gradients. Cela nous permet donc de définir une valeur extrapolée à la face par reconstruction MUSCL et donc de retrouver une formulation consistante du schéma V4. On utilise également les résultats de la déconvolution pour le traitement des conditions aux limites. En effet, on calcule une valeur extrapolée au centre de chaque facette duale à l'aide d'un développement de Taylor et des résultats de la déconvolution. Cela nous permet de traiter les conditions aux limites pour chaque facette duale, et non plus par face comme c'était le cas pour la formulation edge-based. Pour évaluer le schéma numérique ainsi défini, nous reprenons les mêmes cas de test de complexité croissante que pour la discrétisation edge-based. Les résultats pour l'équation d'advection scalaire nous montrent que le schéma ainsi construit est d'ordre 2 sur tout type de maillage au sens des volumes finis, c'est-à-dire en comparaison avec la valeur moyenne par cellule duale. Les résultats sur maillage adapté anisotrope montrent que le schéma volumes finis conserve la robustesse du schéma edge-based sur ce type de maillage. Cependant, l'ordre 2 n'est pas conservé, comme c'était le cas pour la formulation edge-based. En plus de l'étude de l'ordre global du schéma, nous menons une étude de l'ordre local de notre approche volumes finis. Pour mesurer localement l'ordre de notre schéma numérique, on injecte les développements de Taylor dans notre schéma numérique. Pour un maillage cartésien et uniforme, il est facile de faire cette étude à la main. Dans le cas de maillage quelconque en 2D, cette étude peut-être très laborieuse, voire tout simplement impossible. Cependant, en utilisant la linéarité de l'équation d'advection scalaire, il est possible de mener cette étude numériquement. Les résultats nous montrent que le schéma défini selon un formalisme volumes finis est bien localement d'ordre 2, et ce pour l'ensemble des cellules du maillage à l'exception des cellules définies autour des coins du maillage. Nous étendons ensuite la définition de notre schéma numérique aux équations d'Euler. Nous montrons que pour construire un schéma volumes finis d'ordre 2, il n'est pas nécessaire de calculer des termes de correction supplémentaires. Nous évaluons notre schéma numérique sur le cas de test d'advection de tourbillon isentropique. Les résultats nous montrent que pour l'ensemble des maillages considérés, le schéma est d'ordre 2 au sens des volumes finis. Nous complétons cette étude avec le cas de Ringleb. Nous observons également que pour ce cas de test, le schéma est d'ordre 2 sur tout type de maillage au sens des volumes finis. Enfin, nous comparons la vitesse de convergence du schéma edge-based et du schéma volumes finis. Les résultats montrent que pour des maillages perturbés, le schéma volumes finis converge légèrement plus vite vers la solution stationnaire que le schéma edge-based.

Nous nous intéressons ensuite au calcul du flux convectif. Pour l'approche volumes finis centrée au nœud, les faces sont composées de plusieurs facettes. Ces facettes sont agglomérées pour

composer des faces et on calcule le flux convectif une seule fois par face. Jusqu'à présent, le flux convectif était calculé au point M_{ij}, le milieu de l'arête ij et le point d'intersection des différentes facettes. Puisque les faces ne sont pas planes dans le cas général, le point M_{ij} n'est pas, en général, le centre de gravité de la face entre les cellules i et j. Ainsi, calculer le flux convectif une seule fois par face et au point M_{ij} revient à utiliser une formule de quadrature du premier ordre. Le flux convectif ainsi calculé est d'ordre 1. Nous proposons une approche permettant de calculer le flux convectif à l'ordre 2. Pour calculer un flux convectif à l'ordre 2, il est nécessaire d'utiliser une formule de quadrature du second ordre. Nous choisissons de calculer le flux convectif au centre de gravité des facettes duales, ce qui correspond bien à une formule de quadrature d'ordre 2. Pour continuer à ne calculer le flux convectif qu'une seule fois par face, on se ramène, pour toutes les facettes composants une face, au point M_{ij} à l'aide d'un développement en série de Taylor entre le centre de gravité des facettes et le point M_{ij}. Cela a plusieurs avantages. En effet, le calcul du flux convectif repose sur un solveur de Riemann approché, qui peut représenter un coût de calcul important. Si on choisissait de calculer le flux convectif au centre de chaque facette duale, alors on multiplierait le coût de calcul associé au solveur de Riemann approché par le nombre de facettes duales, ce qui peut rendre le coût de calcul prohibitif pour l'utilisation de la méthode en pratique. C'est d'autant plus vrai sur des configurations 3D, où le nombre de points de quadrature peut devenir important. De plus, la reconstruction MUSCL nous permet de reconstruire des états d'ordre 2 au point M_{ij}. Pour évaluer le flux numérique au milieu des facettes, il faudrait connaître, au moins à l'ordre 2, la valeur des états au centre de gravité des facettes. Il serait donc nécessaire de reconstruire autant d'états qu'il y a de facettes. Toujours dans le but de réduire le coût de calcul, il est préférable de ne reconstruire les états qu'une seule fois par face. ll peut également être difficile d'implémenter un calcul de flux convectif par facette dans un code de calcul existant, puisque cela demanderait de modifier grandement sa structure. Dans le cas 2D, le calcul du flux convectif à l'ordre 2 est donné par la formule suivante :

$$
\begin{aligned}
F(\mathbf{W})|_{G^1_{ij}}\,\overrightarrow{n}^1_{ij}|S^1_{ij}| + F(\mathbf{W})|_{G^2_{ij}}\,\overrightarrow{n}^2_{ij}|S^2_{ij}| \;&=\; \\
F(\mathbf{W})|_{M_{ij}}(\overrightarrow{n}^1_{ij}|S^1_{ij}| + \overrightarrow{n}^2_{ij}|S^2_{ij}|) \;&+\; \frac{\partial F}{\partial x}|_{M_{ij}}\Big[(x_{G^1_{ij}} - x_{M_{ij}})\overrightarrow{n}^1_{ij}|S^1_{ij}| + (x_{G^2_{ij}} - x_{M_{ij}})\overrightarrow{n}^2_{ij}|S^2_{ij}|\Big] \\
&+\; \frac{\partial F}{\partial y}|_{M_{ij}}\Big[(y_{G^1_{ij}} - y_{M_{ij}})\overrightarrow{n}_{ij1}|S^1_{ij}| + (y_{G^2_{ij}} - y_{M_{ij}})\overrightarrow{n}^2_{ij}|S^2_{ij}|\Big].
\end{aligned}
$$

Dans cette expression, on retrouve le flux convectif d'ordre 1 ainsi que des termes de correction. Ces termes de corrections dépendent notamment des moments surfaciques d'ordre 1 ainsi que de la normale à la facette duale. Ces quantités ne dépendent que du maillage et peuvent donc être précalculées. Les termes de corrections dépendent également de la dérivée du flux convectif au point M_{ij}. Pour calculer cette quantité, on se base sur la formule analytique, qui nécessite de connaître les états ainsi que leur gradient au point M_{ij}. La valeur des états est donnée par la reconstruction MUSCL, alors que la dérivée des états est calculée à l'aide des résultats de la déconvolution. En effet, le processus de déconvolution donne accès aux gradients des états aux nœuds du maillage. On calcule alors le gradient au point M_{ij} en faisant la moyenne des gradients des sommets i et j.

Pour tester cette nouvelle approche, nous utilisons une base de test dont la complexité est croissante, la même que précédemment. Nous testons d'abord notre schéma sur un cas d'advection scalaire. Les premiers résultats obtenus sur certains maillages sans anisotropie suggèrent une légère réduction de l'erreur de discrétisation pour les maillages les plus fins. Les résultats obtenus avec un maillage adapté montrent toutefois une réduction plus significative de l'erreur de discrétisation. Nous testons ensuite le schéma sur les équations d'Euler. Les résultats montrent également une diminution plus marquée de l'erreur de discrétisation pour les maillages les plus

raffinés. De plus, le calcul du flux convectif au second ordre réduit de manière significative le nombre d'itérations nécessaires à la convergence des résidus. Enfin, nous démontrons que le calcul du flux convectif au second ordre permet de définir un schéma numérique d'ordre 2 pour toutes les cellules d'un maillage donné, ce qui n'était pas le cas avec un flux convectif d'ordre 1.

Ce travail nous a permis de construire, à partir du schéma V4, un schéma volumes finis d'ordre 2 sur tout type de maillage. Pour ce faire, nous utilisons une formule de déconvolution d'ordre 2, nous permettant de construire une valeur au nœud d'ordre 2 à partir des valeurs moyennes disponibles. Nous obtenons finalement un calcul du flux convectif à l'ordre 2 sur toutes les cellules du maillage. De plus, le schéma volumes finis conserve la robustesse du schéma edge-based sur maillage fortement anisotrope. Ce travail nous permet de considérer plusieurs perspectives pour un futur travail. Premièrement, aucune étude sur les maillages anisotropes adaptés aux équations d'Euler n'a été menée dans le cadre de ce travail. Plusieurs applications dans les domaines des turbomachines et de l'aérodynamique externe sont envisageables. Il semble indispensable de mener de telles études pour confirmer que la déconvolution n'est pas associée à un manque de robustesse. De plus, les cas de test examinés dans cette étude ne nous ont pas permis de démontrer la supériorité de l'approche volumes finis sur l'approche edge-based pour la gestion des conditions aux limites. Il serait donc intéressant de comparer ces deux approches à cet égard et de déterminer laquelle se démarque. Cette étude se limite également aux équations d'Euler; elle n'aborde donc pas la partie visqueuse des équations de Navier-Stokes. Cependant, la formulation MEV (Mixed Element Volume) permet déjà de résoudre les équations de Navier-Stokes par discrétisation edge-based. Sur la base des résultats de la déconvolution, il serait possible de traiter la partie visqueuse dans notre formulation volumes finis. La principale exigence pour le flux visqueux serait ici l'intégration d'une déconvolution de troisième ordre, car la dérivation est effectuée deux fois et nécessite une reconstruction plus précise de la solution. Les travaux présentés dans cette thèse répondent à une demande importante de révision des schémas pour des solutions stationnaires, généralement issues d'équations RANS (Reynolds Averaged Navier Stokes). On peut envisager d'étendre la déconvolution à la simulation aux grandes échelles. Enfin, les travaux proposés pourraient être étendus à la prise en compte des fortes discontinuités. Pour cela, on peut viser plusieurs applications. Tout d'abord, l'adaptation de maillage est actuellement analysée pour traiter les écoulements hypersoniques, en se concentrant sur la distribution du flux thermique pariétal. Utiliser une méthode de déconvolution sur ce type de cas nécessite de se questionner sur la façon d'étendre la méthode en présence de discontinuité. Dans le même domaine, l'adaptation de maillage anisotrope est appliquée avec succès à l'arc électrique, où l'arc est associé à une forte discontinuité et requiert de la robustesse.

Contents

1 Introduction **25**
- 1.1 Context and motivation . . . 25
- 1.2 Outline of the thesis . . . 27
- 1.3 Communications and publications . . . 28
  - 1.3.1 Communications . . . 28
  - 1.3.2 Publications . . . 28

2 State of the art **29**
- 2.1 Governing equations . . . 29
  - 2.1.1 Conservation laws . . . 29
  - 2.1.2 Scalar advection equation . . . 30
  - 2.1.3 Euler equations . . . 30
  - 2.1.4 Navier-Stokes equations . . . 31
- 2.2 Discretisation of equations . . . 33
  - 2.2.1 Space discretisation . . . 33
  - 2.2.2 Integral form of the equations . . . 34
  - 2.2.3 First order scheme for vertex-centred finite volume approach . . . 35
    - Riemann problem . . . 37
  - 2.2.4 Higher-order scheme for vertex-centred finite volume approach . . . 38
  - 2.2.5 Edge-based scheme for conservation laws . . . 38
    - First order scheme . . . 38
    - Second order scheme . . . 39
  - 2.2.6 Cell-centred finite volume scheme . . . 42
    - First order scheme . . . 42
    - Second order scheme . . . 42
  - 2.2.7 Boundary conditions . . . 43
    - Scalar advection equation . . . 44
    - Euler equations . . . 44
  - 2.2.8 Time integration . . . 44
    - Explicit time integration . . . 44
    - Implicit time integration . . . 46
  - 2.2.9 Residual computation . . . 46
- 2.3 Comparison between cell-centred and vertex-centred finite volume approaches . . . 46
- 2.4 Limitations and areas for improvement . . . 48

3 Edge-based V4 scheme **51**
- 3.1 Numerical scheme . . . 51
  - 3.1.1 Construction of the numerical scheme . . . 51
  - 3.1.2 Boundary condition . . . 59
- 3.2 Numerical results : 2D scalar transport test case . . . 60
  - 3.2.1 Test case description . . . 60

3.2.2 Truncation error . . . 61
3.2.3 Discretization error . . . 65
Results on standard meshes . . . 66
Results on adapted meshes . . . 68
3.3 Numerical results : 2D Euler test case . . . 69
3.3.1 The convection of a vortex . . . 69
Test case description . . . 69
Numerical results . . . 72
3.3.2 The Ringleb flow . . . 73
Test case description . . . 73
Numerical results . . . 76
3.4 Conclusion . . . 79

4 Finite volume V4 scheme **81**
4.1 Initial approach to deconvolution . . . 84
4.2 A more compact approach to deconvolution . . . 86
4.2.1 Introduction of the method . . . 86
4.2.2 Numerical results . . . 91
4.3 Extension to deconvolution . . . 94
4.3.1 Presentation of the method . . . 94
4.3.2 Numerical results . . . 96
4.4 Application of the deconvolution . . . 99
4.4.1 MUSCL reconstruction . . . 100
4.4.2 Boundary state reconstruction . . . 100
4.5 Numerical results : 2D scalar transport test case . . . 101
4.5.1 Results on standard meshes . . . 101
4.5.2 Results on adapted meshes . . . 102
4.6 Error analysis . . . 103
4.7 Extension to Euler equation . . . 109
4.7.1 Velocity . . . 110
4.7.2 Pressure . . . 111
4.7.3 Temperature . . . 112
4.8 Numerical results : 2D Euler test case . . . 113
4.8.1 Advection of an isentropic vortex . . . 113
4.8.2 Ringleb . . . 115
4.9 Conclusion . . . 118

5 Flux correction **121**
5.1 Geometric error . . . 121
5.2 Scalar advection equation . . . 124
5.2.1 Numerical results on standard meshes . . . 126
5.2.2 Numerical results on adapted meshes . . . 127
5.3 Euler equations . . . 127
5.3.1 Numerical results: the convection of a vortex . . . 128
5.3.2 Numerical results: Ringleb test case . . . 130
5.4 Error analysis . . . 133
5.5 Conclusion . . . 134

6 Conclusion and perspectives **135**

A Initialisation of the Ringleb test case **139**

B CFD software **141**
B.1 SoNICS 141
B.2 DeepDive 142

Chapter 1

Introduction

Contents

1.1 Context and motivation **25**
1.2 Outline of the thesis **27**
1.3 Communications and publications **28**
1.3.1 Communications 28
1.3.2 Publications 28

1.1 Context and motivation

In the face of the climate emergency and the urgent need to reduce greenhouse gas emissions, the global aviation industry has set itself ambitious targets, including achieving carbon neutrality by 2050 [1]. This objective includes reducing the fuel consumption of civil aviation engines, among other things. On 14 June 2021, General Electric and Safran launched the CFM RISE (Revolutionary Innovation for Sustainable Engines) programme in Paris. The programme aims to develop a range of innovative and disruptive technologies, including open-fan technology that will reduce fuel consumption and associated greenhouse gas emissions by more than 20% compared to current engines. The new engine is scheduled to enter service by 2035 [2]. Turbomachinery involves complex physical phenomena that must be studied and understood to improve performance in terms of fuel consumption, greenhouse gas emissions, and reliability. A typical aircraft engine consists of an air inlet, known as a fan, which allows air to enter the engine. The pressure of a portion of the incoming air is then increased using the compressor. This compressed air then passes through the combustion chamber, where it is mixed with kerosene and ignited. The resulting gases are then expanded through a turbine. Some of this energy is used to power the fan and compressor. Finally, the gases are accelerated through a nozzle to produce the thrust necessary to move the aircraft. The physics of turbomachines can be studied through experimentation as well as numerical simulation. Indeed, computational fluid dynamics (CFD) allows flows to be represented by numerical simulation. This is now an essential tool in the Safran Group's research and development strategy. This thesis exclusively deals with fluid dynamics issues related to fans, compressors, turbines, and nozzles in the context of turbomachinery. Studying the physical phenomena in turbomachinery involves solving the compressible Navier-Stokes equations. These equations can be solved using various modelling strategies. One notable method is the DNS (Direct Numerical Simulation) [3] approach, which solves turbulence at all scales and therefore does not rely on any modelling choices. Another approach is LES (Large Eddy Simulation) [4], in which the large structures of turbulence are resolved while the small structures are modelled. Additionally, RANS (Reynolds-averaged Navier-Stokes) [5] approach models all turbulence scales. This list is not exhaustive, as there are other methods for solving

the Navier-Stokes equations (URANS [6], hybrid RANS/LES [7], etc.). Despite continuous increases in computing power over the last seventy years, the computational cost of DNS and LES methods makes them difficult to use for industrial applications. The RANS methodology also enables us to understand certain physical phenomena. Consequently, RANS-type approaches are now favoured in industry. However, numerical simulations remain imprecise because errors arise at every stage of the simulation process. These include modelling errors resulting from the choice of turbulence model [8], geometric errors resulting from the CAD representation of the geometry of the problem [9], mesh discretisation errors resulting from the discretisation of the geometry of the problem [10], numerical scheme discretisation errors resulting from the choice of numerical scheme and its associated errors [11], and finally solver errors resulting from the implementation of the software [8]. In this thesis, we focus on two key areas: mesh discretization error and numerical scheme discretization error. In CFD simulations, the domain of interest is covered by cells that share nodes and is then referred to as a mesh. In the absence of prior knowledge regarding the flow, the study domain is meshed by isotropic cells. This strategy can result in a significant increase in the number of degrees of freedom, particularly when investigating anisotropic phenomena such as shocks or vortices (see figure 1.1). A vast number of degrees of freedom makes numerical simulations impossible in practice, even for RANS modelling.



Figure 1.1: RANS simulation of a high-pressure compressor, final anisotropic mesh (left) and density field (right) from [12]

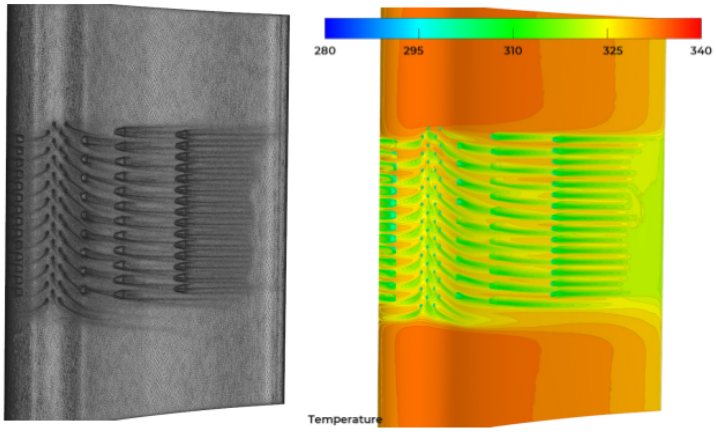


Figure 1.2: RANS simulation of the TATEF2 NGV blade, final anisotropic mesh (left) and temperature field (right) from [13]

To overcome this problem, mesh adaptation methodologies have been developed. Among these methods, we can notably cite anisotropic mesh adaptation [14]. Mesh adaptation consists of converging the pair (mesh, solution) in a single process based on a mathematical criterion for measuring the numerical error. Adaptation with an anisotropic tetrahedral mesh allows

the tetrahedral meshes to be aligned with the preferred flow directions while minimizing the number of vertices, and this approach demonstrates its interest for applications in industrial environments. Recent work involving the use of anisotropic mesh adaptation in turbomachinery configurations has demonstrated this approach's full potential in an industrial context. In particular, this work has focused on studying film cooling of a turbine blade [13] or the performance of a compressor [12]. Finally, Tarsia-Morisco *et al.* present in [15] several industrial applications of anisotropic mesh adaptation. On the other hand, the anisotropy of the elements can degrade the accuracy and robustness of the numerical schemes. There is therefore a real need for accurate and robust numerical schemes.

The most common approach to anisotropic mesh adaptation is the vertex-centred approach, in which control volumes are explicitly constructed around the nodes of the initial mesh. Until now, the only approach used was an edge-based approach, in which the unknowns of the problem are point-value solutions at the nodes of the mesh. The V4 scheme, originally proposed by Alain Dervieux [16], extends van Leer's MUSCL [17] approach to this edge-based formalism for convective flux. This approach enables a second-order scheme to be constructed based on the node value on any type of mesh. Moreover, recent results for this numerical scheme demonstrate its robustness on meshes exhibiting anisotropy on the order of a million [18]. While the edge-based approach and, more specifically, the V4 scheme has produced excellent results on highly anisotropic meshes, we have identified several limitations to the method, and we wish to avoid them using a finite volume discretization of the equations, which is based on an integral formulation of the problem to be solved. However, we have found that there is currently no vertex-centred finite volume V4 scheme designed for anisotropic adapted meshes. The objective of this thesis is to construct a V4 vertex-centred finite volume scheme that is, at least, of order two in the sense of finite volumes. As this study is motivated by the adaptation of anisotropic meshes, it is necessary to construct a robust scheme for highly anisotropic meshes. In this study, we limit our analysis to meshes composed of simplexes, *i.e.*, triangles in 2D and tetrahedra in 3D. Additionally, this thesis aims to implement the new numerical scheme in SoNICS [19], the new-generation CFD code co-developed by ONERA and Safran.

1.2 Outline of the thesis

This thesis is organised into six chapters :

- Chapter 1 provides the context and motivation behind this thesis.
- Chapter 2 outlines all the numerical methods employed. Several conservation law systems for fluid mechanics are presented. We also describe the finite volume discretisation of the equations, as well as the edge-based discretisation approach, and compare the two.
- Chapter 3 provides a thorough analysis of the V4 scheme. We detail all the elements of its construction. We then conduct a convergence study of the scheme based on truncation and discretisation errors. To achieve this, we consider cases of increasing complexity, ranging from the scalar transport equation to the Euler equations.
- Chapter 4 presents the construction of a vertex-centred finite volume scheme based on the V4 scheme. We introduce the deconvolution principle, which enables node values to be reconstructed from available mean values, and propose an extension to the existing operators in the literature. We then present an approach to constructing a vertex-centred finite-volume scheme based on the V4 scheme. We analyse the resulting scheme using test cases of increasing complexity, ranging from the scalar transport equation to the Euler equations. We also conduct a local analysis of the error of the finite volume scheme.

- Chapter 5 introduces a novel method for calculating second-order convective flux, which is based on the finite-volume discretisation of the equations. We analyse the resulting scheme using test cases of increasing complexity, ranging from the scalar transport equation to the Euler equations.
- Chapter 6 summarises the results obtained and presents perspectives for future work.

1.3 Communications and publications

1.3.1 Communications

- Y. Gorschka, G. Puigt, C. Content, B. Maugars, A. Remigi, *Improvement and numerical analysis of cell vertex finite volume schemes*, ECCOMAS Congress 2024, Lisbon, Portugal.
- Y. Gorschka, G. Puigt, C. Content, B. Maugars, A. Remigi, C. Tarsia-Morisco, *Reviewing the vertex centred V4 scheme for more accurate mesh adaptation*, AIAA Scitech 2025 Forum, Orlando, FL, United States of America, January 6-10, 2025.

1.3.2 Publications

- Y. Gorschka, G. Puigt, C. Content, B. Maugars, A. Remigi, C. Tarsia-Morisco, *Reviewing the vertex centred V4 scheme for more accurate mesh adaptation*, AIAA Paper 2025-1108 [20], 2025.

Chapter 2

State of the art

Contents

2.1 Governing equations 29
2.1.1 Conservation laws 29
2.1.2 Scalar advection equation 30
2.1.3 Euler equations 30
2.1.4 Navier-Stokes equations 31
2.2 Discretisation of equations 33
2.2.1 Space discretisation 33
2.2.2 Integral form of the equations 34
2.2.3 First order scheme for vertex-centred finite volume approach 35
2.2.4 Higher-order scheme for vertex-centred finite volume approach 38
2.2.5 Edge-based scheme for conservation laws 38
2.2.6 Cell-centred finite volume scheme 42
2.2.7 Boundary conditions 43
2.2.8 Time integration 44
2.2.9 Residual computation 46
2.3 Comparison between cell-centred and vertex-centred finite volume approaches 46
2.4 Limitations and areas for improvement 48

2.1 Governing equations

In this work, we study several conservation laws systems for fluid mechanics. After introducing a general framework, we will present several conservation laws systems.

2.1.1 Conservation laws

Let us consider a conservation law system without source term written in the following compact form.

$$\frac{\partial \mathbf{W}}{\partial t} + \overrightarrow{\nabla} \cdot \mathcal{F}(\mathbf{W}) = 0 \quad \text{in} \quad \Omega \times [0, t_f], \tag{2.1}$$

where $\mathbf{W} = (w_1, \ldots, w_m)^\top$ represents the m-vector of conservatives variables, $\mathbf{x} = (x, y, z)^\top$ defines the position in space, $\mathcal{F} = (f_1(\mathbf{W}), \ldots, f_m(\mathbf{W}))^\top$ represents the tensor of the conservative

fluxes. Moreover, $\Omega \times [0, t_f]$ is the space-time domain in which the solutions are sought. To close the problem, an initial condition and boundary conditions are given by:

$$\begin{cases} \mathbf{W}(\mathbf{x}, t = 0) = \mathbf{W}_0(\mathbf{x}) & \text{in} \quad \Omega, \\ \mathbf{W}(\mathbf{x} \in \partial\Omega, t) = \mathbf{W}_{\text{bc}}(t) & \text{in} \quad \partial\Omega \times [0, t_f]. \end{cases} \tag{2.2}$$

The Jacobian matrix of the flux function $\mathcal{F}(\mathbf{W})$ in 2.1 is the matrix:

$$\bar{\bar{A}}(\mathbf{W}) = \frac{\partial \mathcal{F}}{\partial \mathbf{W}} = \begin{pmatrix} \frac{\partial f_1}{\partial w_1} & \cdots & \frac{\partial f_1}{\partial w_m} \\ \frac{\partial f_2}{\partial w_1} & \cdots & \frac{\partial f_2}{\partial w_m} \\ \vdots & \vdots & \vdots \\ \frac{\partial f_m}{\partial w_1} & \cdots & \frac{\partial f_m}{\partial w_m} \end{pmatrix}. \tag{2.3}$$

The Jacobian matrix allows equation 2.1 to be rewritten in the quasi-linear form given in (2.4). This is achieved by applying the chain rule. We refer to this as quasi-linear form because the Jacobian matrix depends on vector W.

$$\frac{\partial \mathbf{W}}{\partial t} + \sum_{j=1}^{d} \bar{\bar{A}}(\mathbf{W}) \frac{\partial \mathbf{W}}{\partial x_j} = 0 \quad \text{in} \quad \Omega \times [0, t_f]. \tag{2.4}$$

The system thus defined is said to be hyperbolic at point (x, t) if the Jacobian matrix $\bar{\bar{A}}$ has m real eigenvalues $\lambda_1, \ldots, \lambda_m$ corresponding to a set of m linearly independent eigenvectors $\mathbf{K}_1, \ldots, \mathbf{K}_m$. If all the eigenvalues are distinct, the system is said to be strictly hyperbolic. In physical terms, eigenvalues represent the speed at which information propagates. We will now present several systems of conservation laws.

2.1.2 Scalar advection equation

The scalar advection can be written in the following general form:

$$\frac{\partial w}{\partial t} + \mathbf{a} \cdot \vec{\nabla} w = 0, \tag{2.5}$$

where $\vec{\mathbf{a}} = (a_1, a_2, a_3)^\top$ is the transport velocity. The transport velocity will depend on the space location but the transport velocity is such that its divergence $\vec{\nabla} \cdot \mathbf{a}$ is 0. Such equation is a standard model for hyperbolic problem and it can be solved analytically using characteristics once boundary and initial conditions are provided. Equation (2.5) can be written in the compact form 2.1. Indeed, the state vector and fluxes are given by:

$$\mathbf{W} = w; \quad \mathcal{F}(\mathbf{W}) = w\mathbf{a}. \tag{2.6}$$

2.1.3 Euler equations

The Euler equations are a system of nonlinear hyperbolic conservation laws. They govern the dynamics of a compressible fluid for which the effects of volume forces, viscous stresses, and heat flow are negligible. Here, we consider the conservation of mass and the balance of momentum and energy in the absence of a source term:

$$\begin{cases} \frac{\partial \rho}{\partial t} + \vec{\nabla} \cdot (\rho \mathbf{u}) = 0, \\ \frac{\partial \rho \mathbf{u}}{\partial t} + \vec{\nabla} \cdot (\rho \mathbf{u} \otimes \mathbf{u}) + \vec{\nabla} p = 0, \\ \frac{\partial \rho E}{\partial t} + \vec{\nabla} \cdot (\mathbf{u}(\rho E + p)) = 0, \end{cases} \tag{2.7}$$

with t the physical time (s), ρ the density ($\mathrm{kg}\cdot\mathrm{m}^{-3}$), $\mathbf{u}$ the velocity ($\mathrm{m}\cdot\mathrm{s}^{-1}$), E the total specific energy ($\mathrm{m}^2\cdot\mathrm{s}^{-2}$), and p the pressure ($\mathrm{N}\cdot\mathrm{m}^{-2}$). To close the problem, the gas is assumed to be a perfect gas in this work. Then, the pressure is defined as

$$p = (\gamma - 1)(\rho E - \frac{1}{2}\rho\|\mathbf{u}\|^2),$$

with $\gamma = \frac{C_p}{C_v}$ the ratio of specific heat, where C_p is the specific heat at constant pressure ($\mathrm{J}\cdot\mathrm{kg}^{-1}\cdot\mathrm{K}$) and C_v is the specific heat at constant volume ($\mathrm{J}\cdot\mathrm{kg}^{-1}\cdot\mathrm{K}^{-1}$). For air, the ratio of specific heat γ is constant for a calorically ideal gas and $\gamma = 1.4$.

Equation (2.7) can be written in the compact form 2.1 and the state vector and the convectif flux are as follows

$$\mathbf{W} = \begin{pmatrix} \rho \\ \rho\mathbf{u} \\ \rho E \end{pmatrix}; \quad \mathcal{F}(\mathbf{W}) = F(\mathbf{W}) = \begin{pmatrix} \rho\mathbf{u} \\ \rho\mathbf{u}\otimes\mathbf{u} + p\mathbb{I} \\ \mathbf{u}(\rho E + p) \end{pmatrix}, \tag{2.8}$$

where $F(\mathbf{W})$ dentotes the convectif flux and $\mathbb{I}$ is the identity matrix.

2.1.4 Navier-Stokes equations

The Navier-Stokes equations are a system of non-linear partial differential equations that describe the motion of a viscous fluid. In the same way as for the Euler equations, we consider the Navier-Stokes equations in the case of a compressible fluid for the conservation of mass and for the momentum and energy balance without source term:

$$\begin{cases} \dfrac{\partial\rho}{\partial t} + \vec{\nabla}\cdot(\rho\mathbf{u}) = 0, \\ \dfrac{\partial(\rho\mathbf{u})}{\partial t} + \vec{\nabla}\cdot(\rho\mathbf{u}\otimes\mathbf{u}) + \vec{\nabla}p - \vec{\nabla}\cdot\bar{\bar{\tau}} = 0, \\ \dfrac{\partial\rho E}{\partial t} + \vec{\nabla}\cdot(\mathbf{u}(\rho E + p)) - \vec{\nabla}\cdot(\bar{\bar{\tau}}\cdot\mathbf{u}) - \vec{\nabla}\cdot\mathbf{q} = 0. \end{cases} \tag{2.9}$$

The gas considered is always assumed to be a perfect gas and the temperature T (K) is defined as:

$$T = \frac{1}{C_v}(E - \frac{1}{2}\|\mathbf{u}\|^2),$$

where λ represents the laminar conductivity ($\mathrm{W}\cdot\mathrm{m}^{-1}\cdot\mathrm{K}^{-1}$) and $\bar{\bar{\tau}}$ the viscous stress tensor ($\mathrm{N}\cdot\mathrm{m}^{-3}$). The relationship between pressure p, temperature T, and density ρ is described by the equation of state for perfect gas:

$$p = \rho R T, \tag{2.10}$$

where R is the gas constant as $R = C_p - C_v = \frac{\mathcal{R}}{M}$. $\mathcal{R}$ is the universal gas constant and $\mathcal{R} = 8.134(\mathrm{J}\cdot\mathrm{K}^{-1}\cdot\mathrm{mol}^{-1})$, M is the molar mass ($\mathrm{kg}\cdot\mathrm{mol}^{-1}$).

In the case of a Newtonian fluid, $\bar{\bar{\tau}}$ is linearly related to the derivatives of the velocity field $\vec{u}$ (components u, v, and w) via the deformation tensor $\bar{\bar{\mathbf{D}}}$ by:

$$\bar{\bar{\mathbf{D}}} = \frac{1}{2}(\overrightarrow{\nabla}\mathbf{u} + \overrightarrow{\nabla}\mathbf{u}^{\top}) = \begin{pmatrix} u_x & \frac{1}{2}(v_x + u_y) & \frac{1}{2}(w_x + u_z) \\ \frac{1}{2}(u_y + v_x) & v_y & \frac{1}{2}(w_y + u_z) \\ \frac{1}{2}(u_z + w_x) & \frac{1}{2}(v_z + w_y) & w_z \end{pmatrix},$$

with μ the dynamic viscosity ($\mathrm{kg}\cdot\mathrm{m}^{-1}\mathrm{w}\cdot\mathrm{s}^{-1}$) and $u_x = \dfrac{\partial u}{\partial x}$. The Newtonian assumption allows you to write the viscous stress tensor as:

$$\bar{\bar{\tau}} = \lambda\overrightarrow{\nabla}\cdot\mathbf{u}\mathbb{I} + 2\mu\bar{\bar{\mathbf{D}}}. \tag{2.11}$$

The Stokes hypothesis enables the formulation of the equality $3\lambda + 2\mu = 0$, which allows the viscous stress tensor to be expressed by:

$$\bar{\bar{\tau}} = 2\mu(\bar{\bar{\mathbf{D}}} - \frac{1}{3}\overrightarrow{\nabla}\cdot\mathbf{u}\mathbb{I}). \tag{2.12}$$

In order to fully characterise the viscous stress tensor, it is essential to determine the dynamic viscosity as a function of temperature. For a colorically perfect gas, this relationship is given by the Sutherland's relationship [21]:

$$\mu(T) = \mu_{\mathrm{ref}}\Big(\frac{T}{T_{\mathrm{ref}}}\Big)^{\frac{3}{2}}\frac{T_{\mathrm{ref}} + S}{T + S}.$$

For air, the three constants are $\mu_{\mathrm{ref}} = 1.716\times10^{-5}(\mathrm{kg}\cdot\mathrm{m}^{-1}\cdot\mathrm{s}^{-1}), T_{\mathrm{ref}} = 273.11(\mathrm{K})$ and $S = 110.56(\mathrm{K})$.

The relationship between the heat flux $\overrightarrow{\mathbf{q}}(\mathrm{W}\cdot\mathrm{m}^{-2})$ and the temperature T is governed by Fourier's law [22]:

$$\mathbf{q} = -\kappa\overrightarrow{\nabla}T, \tag{2.13}$$

where κ is the thermal conductivity of the fluid ($\mathrm{W}\cdot\mathrm{m}^{-1}\cdot\mathrm{K}^{-1}$). By introducing the Prandtl number Pr, $Pr = \frac{\mu C_p}{\kappa}$, Fourier's law becomes:

$$\mathbf{q} = -\frac{\mu C_p}{Pr}\overrightarrow{\nabla}T, \tag{2.14}$$

with the Prandtl number assumed to be constant and equal to $Pr = 0.72$ for air.

Equation (2.9) can be written in the compact form 2.1. To achieve this, the fluxes are split in an inviscid (convective) part and a viscous part:

$$\frac{\partial\mathbf{W}}{\partial t} + \overrightarrow{\nabla}\cdot F(\mathbf{W}) - \overrightarrow{\nabla}\cdot G(\mathbf{W}, \overrightarrow{\nabla}\mathbf{W}) = 0 \quad \text{in} \quad \Omega\times[0, t_f], \tag{2.15}$$

where $\mathbf{W}$ represents the vector of conservatives variables, F is the convective flux, G the diffusion flux quantity. The convection term is associated with the hyperbolic nature of Euler's equations. The viscous terms depend on solution and gradient and are associated with the elliptic or parabolic nature of the Navier-Stokes equations. They are defined, for the Navier-Stokes equations, by:

$$\mathbf{W} = \begin{pmatrix} \rho \\ \rho\mathbf{u} \\ \rho E \end{pmatrix}, \quad F(\mathbf{W}) = \begin{pmatrix} \rho\mathbf{u} \\ \rho\mathbf{u}\otimes\mathbf{u} + p\mathbf{I}_d \\ \mathbf{u}(\rho E + p) \end{pmatrix} \quad \text{and} \quad G(\mathbf{W}, \overrightarrow{\nabla}\mathbf{W}) = \begin{pmatrix} 0 \\ \bar{\bar{\tau}} \\ \bar{\bar{\tau}}\cdot\mathbf{u} + \mathbf{q} \end{pmatrix}. \tag{2.16}$$

This definition of the vector **W** represents the conservative variables of the problem. The vector **U** represents the primitive variables of the problem and is of strong importance for the hyperbolic terms:

$$\mathbf{U} = \begin{pmatrix} \rho \\ \mathbf{u} \\ p \text{ or } T \end{pmatrix}. \tag{2.17}$$

The natural set of equations for this work is the compressible Navier-Stokes equations. However, the main focus is on the hyperbolic equations, as is the case with the Euler or scalar advection equations. Indeed, the scheme we developed is dedicated to the hyperbolic equations.

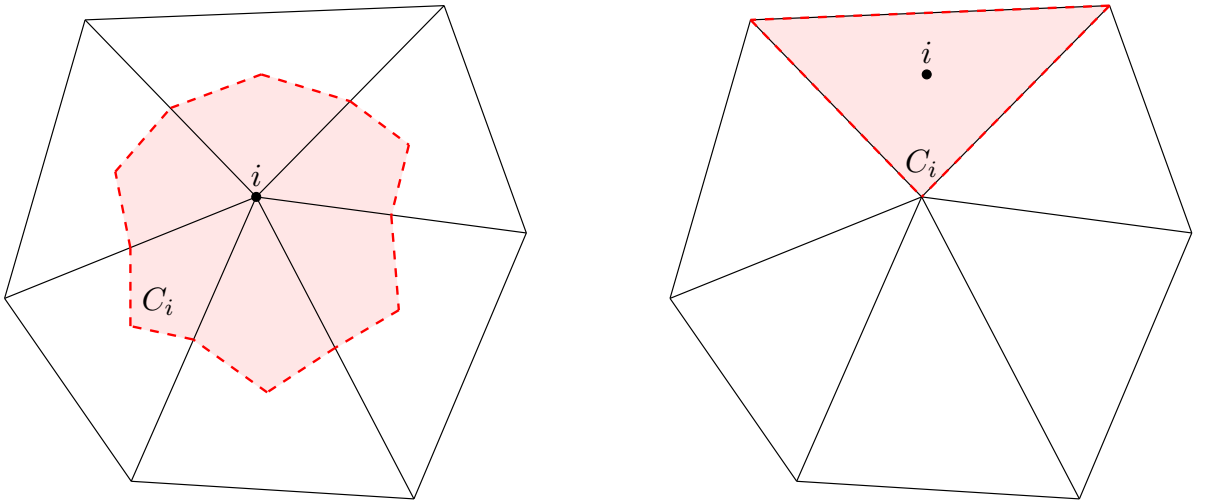
2.2 Discretisation of equations

2.2.1 Space discretisation

Let $\Omega \subset \mathbb{R}^d$ ($d = 1, 2$ or 3) the domain of interest. To close the problem, boundary conditions are imposed on $\partial\Omega$. The domain Ω is discretized into a set of elements $(K_i)_{1 \leq i \leq N_k}$ such that $\mathring{K}_i \cap \mathring{K}_j = \varnothing \ \forall i \neq j$, where $\mathring{K}_i$ represents the interior of element K_i. This discretisation conforms to the subdivision of the boundary $\partial\Omega$ and defines the primal mesh $\mathcal{H}$. From the set of elements, a set of control volumes $(C_i)_{1 \leq i \leq N_c}$ can be constructed from the mesh $\mathcal{H}$. The discretized domain Ω_h can finally be covered by two sets of elements:

$$\Omega_h = \bigcup_{i=1}^{N_k} K_i = \bigcup_{i=1}^{N_c} C_i,$$

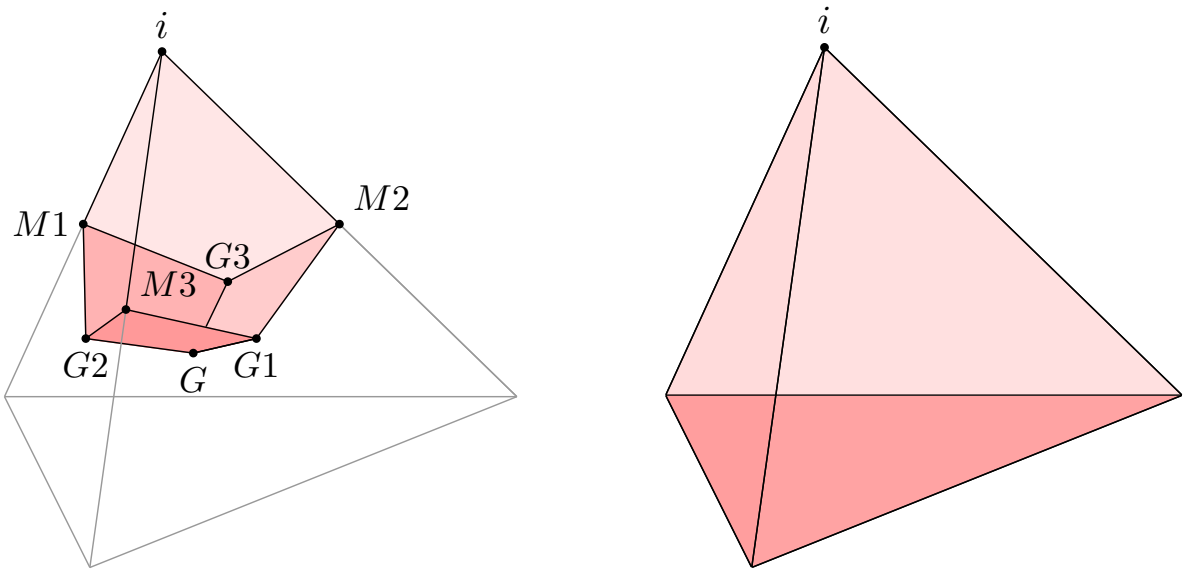
with N_k (respectively N_c) the number of mesh elements (respectively the number of control volumes). This introduction of two sets of control elements enables the definition of both cell-centred and vertex-centred formulations. Indeed, if $K_i = C_i$, $\forall i$ and $N_c = N_k$, one recovers the cell-centred finite volume formulation where control volume are mesh elements themselves. If the set of elements is built from mesh nodes and if N_c represents the number of mesh nodes, it is possible to define two different sets of elements. The set K_i of the standard mesh elements of the primal grid, and the set C_j of control volumes build around mesh nodes. In the following, the median cell approach is considered to define the dual cells. In 2D, dual cells are built using edges that link cell center and edge mid-nodes. Note that in 3D, dual cells are built from cell center, face center, edge mid-nodes so boundary faces are quadrilaterals. They have the property that these four nodes all behave in the same plane. This is a strong advantage since there aren't any issue with geometric quantities (area, normal vector...). The proof that the four nodes are in the same place is easily performed by barycentre definitions, and is described in [23]. Figs. 2.1 and 2.2 show such dual cell definitions.



(a) vertex-centred mesh

(b) Cell-centred mesh

Figure 2.1: Control volumes in the case of an unstructured 2D mesh for approaches vertex-centred (a) and cell-centred (b). The dual cell is shown in red.



(a) vertex-centred mesh

(b) Cell-centred mesh

Figure 2.2: Control volumes in the case of an unstructured 3D mesh for approaches vertex-centred (a) and cell-centred (b). The dual cell is shown in red.

2.2.2 Integral form of the equations

We present here the finite-volume discretisation of the Navier-Stokes equations, as given in Section 2.15. The equation is integrated over each control volume C_i:

$$\int_{C_i} \frac{\partial \mathbf{W}}{\partial t} d\Omega + \int_{C_i} \overrightarrow{\nabla} \cdot F(\mathbf{W})\, d\Omega + \int_{C_i} \overrightarrow{\nabla} \cdot G(\mathbf{W}, \overrightarrow{\nabla}\mathbf{W})\, d\Omega = 0 \quad \forall i \in \{1, ..., N_c\}. \tag{2.18}$$

Applying a Green-Gauss formula to equation (2.18), the volume integral of the divergence term is replaced by an integral over the boundary of C_i, denoted by ∂C_i:

$$\int_{C_i} \frac{\partial \mathbf{W}}{\partial t} d\Omega + \int_{\partial C_i} F(\mathbf{W}) \cdot \overrightarrow{n_{C_i}}\, d\Gamma + \int_{\partial C_i} G(\mathbf{W}, \vec{\nabla}\mathbf{W}) \cdot \overrightarrow{n_{C_i}}\, d\Gamma = 0 \quad \forall i \in \{1, ..., N_c\}, \tag{2.19}$$

where $\overrightarrow{n_{C_i}}$ represents the outgoing normal vector to the control volume C_i. Finally, Fubini's theorem allows us to interchange the volume integral and the time derivative:

$$\frac{d}{dt}\int_{C_i} \mathbf{W} d\Omega + \int_{\partial C_i} F(\mathbf{W}) \cdot \overrightarrow{n_{C_i}}\, d\Gamma + \int_{\partial C_i} G(\mathbf{W}, \vec{\nabla}\mathbf{W}) \cdot \overrightarrow{n_{C_i}}\, d\Gamma = 0 \quad \forall i \in \{1, ..., N_c\}. \tag{2.20}$$

A purely finite volume discretisation enables us to work with the averaged value of the unknowns in the problem. This approach is common to both cell-centred and vertex-centred formulations. Only the definition of the dual cell differs. The averaged value per dual cell is defined as:

$$\bar{\mathbf{W}}_i = \frac{1}{|C_i|}\int_{C_i} \mathbf{W}\, d\Omega.$$

One obtains the following set of partial differential equations:

$$\frac{d(|C_i|\bar{\mathbf{W}}_i)}{dt} + \underbrace{F_i}_{\substack{\textbf{Convective} \\ \textbf{flux}}} + \underbrace{G_i}_{\substack{\textbf{Diffusive} \\ \textbf{flux}}} + \underbrace{\Gamma_i}_{\substack{\textbf{Boundary} \\ \textbf{conditions}}} = 0 \tag{2.21}$$

where $\bar{\mathbf{W}}$ holds for cell-averaged solution. Moreover, $F_i = \int_{\partial C_i} F(\mathbf{W}) \cdot \vec{n}_{C_i}\, d\Gamma$, $G_i = \int_{\partial C_i} G(\mathbf{W}, \vec{\nabla}\mathbf{W}) \cdot \overrightarrow{n_{C_i}}\, d\Gamma$, and $\Gamma_i = \int_{\partial C_i \cap \partial\Omega} \gamma_i(\mathbf{W}_i)\, d\Gamma$ with γ_i given by the boundary condition.

In this study, the focus is on convective flux for the vertex-centred approach. We will therefore primarily present the vertex-centred approach. This is why we will provide a detailed description of the construction of a first-order scheme before switching to a more detailed overview of higher order approaches.

2.2.3 First order scheme for vertex-centred finite volume approach

To compute convective flux, the boundary ∂C_i is decomposed into many facets. For a given dual cell C_i, ν_i denotes all the facets of C_i. This leads to the definition:

$$\int_{\partial C_i} F(\mathbf{W}) \cdot \vec{n}_{C_i}\, d\Gamma = \sum_{j \in \nu_i} \int_{f_j} F(\mathbf{W}) \cdot \vec{n}_{ij}\, d\Gamma. \tag{2.22}$$

A standard way to perform the flux computation is to perform a loop on the facets. For a first-order computation, the interface quantities are the right and left states in the dual cell that share the facet. In practice, in 2D, there are two facets that share the edge ij midpoint between dual cells C_i and C_j. The situation is almost the same in 3D, where all four-nodes facets using the edge ij midpoint have the same cell indices i and j.
Finally, first-order accurate flux can be computed on any point of the facet, since the normal vector is always constant, and a judicious choice is the edge midpoint shared by all the facets

between cells C_i and C_j. So, if we want to calculate the numerical flux $F(\mathbf{W})$ through the S_{ij1} facet, we can write:

$$\int_{S_{ij1}} F(\mathbf{W}) \cdot \vec{n}\, ds = F(\mathbf{W})|_{M_{ij}} . \vec{n}_{ij1} |S_{ij1}|$$

Therefore, in the 2D case, two facets are agglomerated to form a face:

$$\int_{S_{ij1}} F(\mathbf{W}) \cdot \vec{n}\, ds + \int_{S_{ij2}} F(\mathbf{W}) \cdot \vec{n}\, ds = F(\mathbf{W})|_{M_{ij}} (\vec{n}_{ij1} |S_{ij1}| + \vec{n}_{ij2} |S_{ij2}|).$$

The previous expression highlights the fact that, for order 1, only one numerical flux can be calculated per face, at the integration point M_{ij}. The numerical flux is linear to the normal, which justifies the previous sum. All that then remains to be done is to evaluate the sum of the normals of each facet making up the face. The very same development can be applied to 3D dual cells.

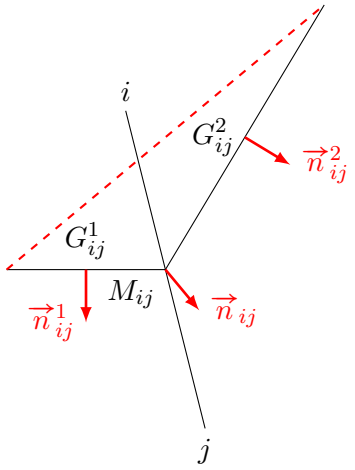


Figure 2.3: Flux geometry

In the 2D case, we have:

$$\vec{n}_{ij1} |S_{ij1}| = \begin{pmatrix} y_{M_{ij}} - y_{G_{ij1}} \\ x_{G_{ij1}} - x_{M_{ij}} \end{pmatrix} \quad \text{and} \quad \vec{n}_{ij2} |S_{ij2}| = \begin{pmatrix} y_{G_{ij2}} - y_{M_{ij}} \\ x_{M_{ij}} - x_{G_{ij2}} \end{pmatrix},$$

so:

$$\vec{n}_{ij1} |S_{ij1}| + \vec{n}_{ij2} |S_{ij2}| = \begin{pmatrix} y_{G_{ij2}} - y_{G_{ij1}} \\ x_{G_{ij1}} - x_{G_{ij2}} \end{pmatrix}. \tag{2.23}$$

To have a first order numerical flux, we can evaluate the convective flux at the point M_{ij} only once.

Riemann problem

As the unknowns of the problem are averaged values over the dual cell, this leads to a discontinuity at the interfaces. A Riemann problem is therefore solved for the calculation of convective fluxes. The definition of a Riemann problem is provided in equation (2.24):

$$\begin{cases} \dfrac{\partial \mathbf{W}}{\partial t} + \nabla \cdot F(\mathbf{W}) & = 0, \\ \mathbf{W}(x,0) = \mathbf{W}_0(x) & = \begin{cases} \mathbf{W}_L \text{ if } x < x_0, \\ \mathbf{W}_R \text{ if } x \geq x_0. \end{cases} \end{cases} \tag{2.24}$$

The solutions for $\mathbf{W}$ are known analytically and it is possible to analytically calculate the flux $F(\mathbf{W})$. However, solving the hyperbolic problem (2.24) exactly requires the determination of all shock waves, rarefaction waves, and contact discontinuities and this is extremely costly in terms of computational resources. For this reason, this study uses approximate Riemann solvers to solve problem (2.24). A distinction can be made between linear equations, such as the scalar transport equation, and non-linear equations, such as Euler's equations.

Scalar advection equation Since the scalar transport equation is linear, the Godunov scheme is easily defined by very simple expressions. It is an upwind formulation and for two cells C_i and C_j, we have:

$$\Phi_{ij} = \Phi(\mathbf{W}_i, \mathbf{W}_j, \vec{n}_{ij}) = \int_{f_j} F(\mathbf{W}) \cdot \vec{n}_{f_j} ds \approx S_{ij} \begin{cases} \mathbf{a}_{ij} \cdot \vec{n}_{ij} \mathbf{W}_i & \text{if} \quad \mathbf{a}_{ij} \cdot \vec{n}_{ij} > 0, \\ \mathbf{a}_{ij} \cdot \vec{n}_{ij} \mathbf{W}_j & \text{otherwise,} \end{cases} \tag{2.25}$$

where S_{ij} is the area of the edge ij and $\mathbf{a}_{ij}$ is the velocity at M_{ij}, the integration point of the convectif flux. The flux balance, for a given cell i can then be written as:

$$\int_{\partial C_i} F(\mathbf{W}) \cdot \vec{n}_{C_i} ds \approx \sum_{f_j \in \nu_i} S_{ij} \begin{cases} \mathbf{a}_{ij} \cdot \vec{n}_{ij} \mathbf{W}_i & \text{if} \quad \mathbf{a}_{ij} \cdot \vec{n}_{ij} > 0, \\ \mathbf{a}_{ij} \cdot \vec{n}_{ij} \mathbf{W}_j & \text{otherwise.} \end{cases}$$

Riemann solver In this study, we opted to solve the Riemann problem (2.24) using an approximate Riemann solver. Other approaches are described in the literature. One notable example is the SUPG method [24], which involves making the solution artificially continuous and derivable via regularisation. There are many approximate Riemann solvers in the literature. These include the HLLC solver [25] and the one of Roe [26], which we will briefly describe below [27]. The idea behind Roe's solver is to replace the Jacobian matrix A with a matrix $\tilde{A}(\mathbf{W}_i, \mathbf{W}_j)$, calculated using states $\mathbf{W}_i$ and $\mathbf{W}_j$. The numerical flux Φ_{ij} is given (2.26):

$$\Phi_{ij} = \frac{1}{2}(F(\mathbf{W}_i) + F(\mathbf{W}_j)) + |\tilde{A}|(\mathbf{W}_j - \mathbf{W}_i). \tag{2.26}$$

The numerical flux defined in this way must verify the following two assumptions.

1. Consistency : $\Phi(\mathbf{W}, \mathbf{W}) = \mathbf{W} \quad \forall \mathbf{W}$,
2. Conservation : $\Phi(\mathbf{W}_i, \mathbf{W}_j, \vec{n}_{ij}) = -\Phi(\mathbf{W}_j, \mathbf{W}_i, \vec{n}_{ji})$.

2.2.4 Higher-order scheme for vertex-centred finite volume approach

We presented the construction of a first-order finite volume scheme. Here, we detail the construction of a higher-order scheme. Of course, any scheme devoted to cell-centred formulation should be applicable to vertex-centred formulation. But, to our knowledge, none of them can take into account the structure of the dual cell and use one Riemann solution per mesh edge. However, work has been carried out on time-dependent incompressible Navier-Stokes equations by F. Stezwein [28, 29, 30, 31]. More specifically, this method is based on the k-exact [32, 33] approach and proposes extending the k-exact multiple-correction method of Pont *et al.* [34, 35] to a vertex-centred approach on unstructured meshes. To solve equation (2.20) using a k-exact reconstruction proposed by Stezwein, three steps are required.

1. Reconstruct the solution of the primitive field variables from known volume-averages in the neighborhood of the dual cell C_i at a time step t^n.

2. Approximate the fluxes at element interfaces with a higher order of accuracy using the reconstructed solution of the adjacent elements.

3. Let the system of equations evolve in time to estimate new volume-averages at a new time step t^{n+1}.

2.2.5 Edge-based scheme for conservation laws

First order scheme

Until now, conservation law systems have been solved using a finite volume approach, where the unknowns in the problem are the averaged values per dual cell. Following the same spatial discretisation and therefore the same definition of the primal and dual meshes, it is possible to define another class of numerical scheme. In this class of numerical schemes, the unknowns of the problem are the values at the mesh nodes. These numerical schemes are referred to as edge-based formulations in the work of Nishikawa [36] and as finite-difference-like formulations in the work of Debiez and Dervieux [37]. To construct this scheme, we start from the conservation laws system:

$$\frac{\partial \mathbf{W}}{\partial t} + \overrightarrow{\nabla} \cdot \mathcal{F}(\mathbf{W}) = 0. \tag{2.27}$$

The continuous problem (2.27) is replaced by the discrete problem (2.28) by introducing the value at the mesh node, which is called point-value solution $\mathbf{W}_i = \mathbf{W}(x_i, y_i)$:

$$\frac{d\mathbf{W}_i}{dt} + \overrightarrow{\nabla} \cdot \mathcal{F}(\mathbf{W}_i) = 0. \tag{2.28}$$

In (2.28), the time partial derivative becomes a full derivative since the unknown is located at a mesh node, and therefore does not explicitly depends on position. The objective is to calculate quantity $\overrightarrow{\nabla} \cdot \mathcal{F}(\mathbf{W}_i)$. In the reference work [37, 36], relation (2.28) is then integrated in space over the dual cell. This is required to introduce the Green-Gauss relation and transfer the divergence integral over C_i into a surface integral over ∂C_i:

$$\int_{C_i} \overrightarrow{\nabla} \cdot \mathcal{F}(\mathbf{W}) d\Omega = \sum_{j \in \nu_i} \int_{\partial C_i \cap \partial C_j} \mathcal{F}(\mathbf{W}) \overrightarrow{n}_{ij} d\Gamma \approx \sum_{j \in \nu_i} \frac{1}{2} (\mathcal{F}_i + \mathcal{F}_j) \overrightarrow{n}_{ij} |S_{ij}|, \tag{2.29}$$

where $|C_i|$ denotes the mesure of the dual cell, $\mathcal{F}_i = \mathcal{F}(\mathbf{W}_i)$ is the conservative flux at point $\mathbf{x}_i$, ν_i is the set of neighbour nodes of the node i, $\overrightarrow{n}_{ij}|S_{ij}|$ is the directed area vector as defined in (2.23). The flux is then the averaged of contributions from cells i and j. This expression takes advantage of the construction of dual volumes. Indeed, point M_{ij} is always the midpoint of edge ij. Thus, quantity $\frac{1}{2}(\mathcal{F}_i + \mathcal{F}_j)$ is always a second-order approximation of $\mathcal{F}_{M_{ij}}$. By circulating around the edge of the cell, it is easy to verify that expression (2.29) is correct for polynomials of degree 0, and therefore that this formula is at least of order 1. We will now show that this formula is of order 2 on triangular/tetrahedral meshes, which is always the case in our study. To do so, let $\mathcal{F} : (x, y) \mapsto \mathcal{F}(x, y) = \alpha x + \beta y + \delta$, where $(\alpha, \beta, \delta \in \mathbb{R}^3)$. Applying relation (2.29) to the function $\mathcal{F}$, we obtain:

$$\begin{aligned}\frac{1}{|C_i|}\sum_{j\in\nu_i}\frac{1}{2}(\mathcal{F}_i+\mathcal{F}_j)\overrightarrow{n}_{ij}|S_{ij}| &= \frac{1}{|C_i|}\sum_{j\in\nu_i}\frac{1}{2}(\alpha(x_i+x_j)+\beta(y_i+y_j)+\delta)\overrightarrow{n}_{ij}|S_{ij}| \\ &= \frac{\alpha x_i+\beta y_i+\delta}{2|C_i|}\sum_{j\in\nu_i}\overrightarrow{n}_{ij}|S_{ij}| + \frac{1}{|C_i|}\sum_{j\in\nu_i}\frac{1}{2}(\alpha x_j+\beta y_j)\overrightarrow{n}_{ij}|S_{ij}|.\end{aligned}$$

However, by circulating along the edge of the cell, we have $\sum_{j\in\nu_i}\overrightarrow{n}_{ij}|S_{ij}| = 0$. Furthermore, for meshes composed of simplexes, it can be proven that $\sum_{j\in\nu_i}\frac{1}{2}(x_j - x_i)\overrightarrow{n}_{ij}|S_{ij}| = |C_i|$. The proof starts from the integral of the function 1 over the dual cell C_i and by means of the Green-Gauss relation, it becomes the integral of the x function over the boundary. The development of the expressions is described in [38]. Finally, we can write

$$\frac{1}{|C_i|}\sum_{j\in\nu_i}\frac{1}{2}(\mathcal{F}_i+\mathcal{F}_j)\overrightarrow{n}_{ij}|S_{ij}| = \alpha+\beta = \overrightarrow{\nabla}\cdot\mathcal{F}.$$

Thus, the formula proposed in (2.29) is a second-order approximation of the divergence operator on each dual cell. It is even shown in [36] that this operator is a third-order approximation. Numerically, quantity $\frac{1}{2}(\mathcal{F}_i+\mathcal{F}_j)$ is not calculated, as this leads to an unstable numerical scheme. A Riemann solver is therefore used to calculate this quantity. Finally, we have the following discrete problem:

$$\frac{d\mathbf{W}_i}{dt} + \frac{1}{|C_i|}\sum_{j\in\nu_i}\Phi_{ij}\overrightarrow{n}_{ij} = 0, \tag{2.30}$$

where Φ_{ij} is the numerical flux, representing both the convective and diffusive fluxes. If the numercial flux is evaluated using nodal values $\mathbf{W}_i$ and $\mathbf{W}_j$, this leads to a first-order numerical scheme. Several methods have been developed to improve the accuracy of numerical schemes.

Second order scheme

To increase the accuracy of numerical schemes, van Leer proposes a cell-based reconstruction of the unknowns of the problem [17, 39, 40]. Thus, the MUSCL (Monotone Upwind Scheme for Conservation Law) method achieves second-order accuracy by evaluating convective flux using extrapolated values. In the case of the vertex-centred approach, the values are extrapolated to point M_{ij}, the midpoint of edge ij. The extrapolated values are denoted $\mathbf{W}_{ij}$ and $\mathbf{W}_{ji}$ and are defined by:

$$\begin{cases}\mathbf{W}_{ij} = \mathbf{W}_i + \frac{1}{2}(\overrightarrow{r_{ij}}\cdot\overrightarrow{\nabla\mathbf{W}_i}), \\ \mathbf{W}_{ji} = \mathbf{W}_j - \frac{1}{2}(\overrightarrow{r_{ij}}\cdot\overrightarrow{\nabla\mathbf{W}_j}).\end{cases} \tag{2.31}$$

The vector $\overrightarrow{r_{ij}}$ represents the vector from vertex i to vertex j, as can be seen in Figure 2.4. The factor $1/2$ comes from the fact that the numerical flux is always calculated in the middle of the edge connecting vertex i to vertex j. The extrapolated values $\mathbf{W}_{ij}$ needs the gradient $\nabla \mathbf{W}_i$.

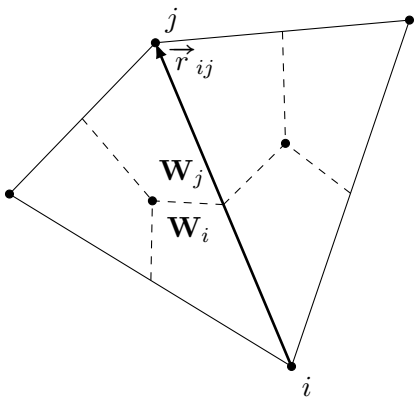


Figure 2.4: MUSCL reconstruction for the vertex-centred approach

In order to calculate the extrapolated values, it is necessary to have knowledge of the gradients at the mesh nodes. There are several methods that can be used to derive different numerical schemes.

On the one hand, we can cite the work of Nishikawa [41, 42, 43, 44], in which the nodal gradients are calculated by a least-squares method [45]. This method is based on the resolution of a rectangular linear system of any size. Indeed, in order to decrease the conditioning of the matrix of the linear system, it is possible to choose the stencil as large as is required. Bridel-Bertomeu's work [46] analyses the conditioning of the matrix as a function of the stencil under consideration for Cartesian grids and shows that more than 15 cells (in 2D) are required to maintain stability. The calculation cost of this type of method can therefore quickly become significant. The numerical scheme thus constructed is of order 2 for meshes composed of simplexes [47]. This approach is only of order 1 for other types of elements, with the exception of very specific geometric conditions [48]. It has also been established that this method is susceptible to robustness issues in the context of triangular meshes characterised by a high aspect ratio [49]. Some works propose solutions to these issues [50]. Research has demonstrated that certain studies have attained order 3 for this type of numerical scheme, whether for the Euler equations [51] or the Navier-Stokes equations [52]. In order to achieve this, it is necessary to use a gradient of order 3, such as the QIEBG approach [53], for example, and a linear reconstruction of the flux [54, 55].

Another class of method was originally proposed by Alain Dervieux [16], for meshes composed of simplexes. The novel approach differs from the original MUSCL method in that the gradients $\overrightarrow{\nabla \mathbf{W}_i}$ and $\overrightarrow{\nabla \mathbf{W}_j}$ are calculated per edge using a linear combination of multiple gradients. To be more precise, for a given edge, we identify the two triangles/tetrahedra that are most aligned with it, as illustrated in Figure 2.5. We then calculate different gradients from these elements and subsequently reconstruct a high-order gradient from the gradients that have been previously calculated. This approach enables to construct a second-order scheme with minimal numerical dissipation.

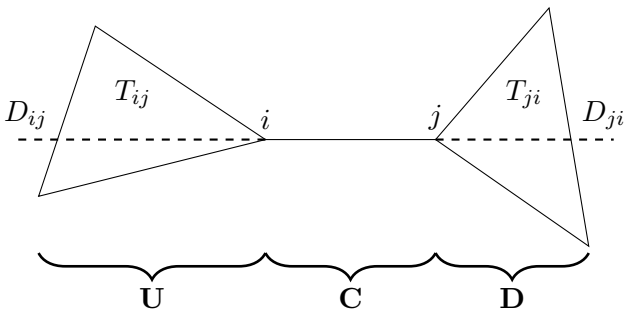


Figure 2.5: Reconstruction of a β gradient in the edge direction

The first scheme constructed from this methodology is the $\beta-$scheme. This is achieved by calculating an edge-centred gradient and a triangle/tetrahedron gradient. A "high-order" gradient is then calculated using a linear combination of the centred and upwind/downwind gradients (2.32):

$$\begin{cases} \overrightarrow{\nabla \mathbf{W}}_i^{\beta} \cdot \overrightarrow{ij} = (1-\beta)\overrightarrow{\nabla \mathbf{W}}_{ij}^{C} \cdot \overrightarrow{ij} + \beta \overrightarrow{\nabla \mathbf{W}}_{ij}^{U} \cdot \overrightarrow{ij}, \\ \overrightarrow{\nabla \mathbf{W}}_j^{\beta} \cdot \overrightarrow{ij} = (1-\beta)\overrightarrow{\nabla \mathbf{W}}_{ij}^{C} \cdot \overrightarrow{ij} + \beta \overrightarrow{\nabla \mathbf{W}}_{ij}^{D} \cdot \overrightarrow{ij}. \end{cases} \tag{2.32}$$

This scheme is also known as the V4 scheme and is of maximum accuracy for $\beta = 1/3$ [56, 57]. The constructed scheme is of order 3 on Cartesian mesh (this justifies the naming of "high-order" gradient that will be kept in the following) for the linear advection equation. For any mesh, this scheme is of order 2 with a numerical dissipation of order 4. For a comprehensive review of this scheme, refer to Chapter 3. Recent work has extended this scheme to hybrid meshes [58, 59].
In order to reduce the numerical dissipation of the V4 scheme, other numerical schemes built on the same principle have been proposed. The V6 scheme [37] is a numerical scheme with a numerical dissipation of order 6 (2.33). To do this, two additional points, D_{ij} and D_{ji} defined as the intersection between the edge ij and the elements T_{ij} and T_{ji} are introduced. A new high-order gradient is built by a more complex combination of the defined gradients. This numerical scheme remains of order 2 on any mesh. On the other hand, it reaches order 5 of convergence on Cartesian meshes for the linear advection equation. The final high-order gradient is:

$$\begin{cases} \overrightarrow{\nabla \mathbf{W}}_i^{V_6} \cdot \overrightarrow{ij} &= \Big[\Big(\frac{2}{3}\overrightarrow{\nabla \mathbf{W}}_{ij}^{C} + \frac{1}{3}\overrightarrow{\nabla \mathbf{W}}_{ij}^{U}\Big) - \frac{1}{30}\Big(\overrightarrow{\nabla \mathbf{W}}_{ij}^{U} - 2\overrightarrow{\nabla \mathbf{W}}_{ij}^{C} + \overrightarrow{\nabla \mathbf{W}}_{ij}^{D}\Big) \\ &\qquad - \frac{1}{15}\Big(\overrightarrow{\nabla \mathbf{W}}_{D_{ij}} - 2\overrightarrow{\nabla \mathbf{W}}_{i} + \overrightarrow{\nabla \mathbf{W}}_{j}\Big)\Big] \cdot \overrightarrow{ij}, \\ \overrightarrow{\nabla \mathbf{W}}_i^{V_6} \cdot \overrightarrow{ij} &= \Big[\Big(\frac{2}{3}\overrightarrow{\nabla \mathbf{W}}_{ij}^{C} + \frac{1}{3}\overrightarrow{\nabla \mathbf{W}}_{ij}^{D}\Big) - \frac{1}{30}\Big(\overrightarrow{\nabla \mathbf{W}}_{ij}^{D} - 2\overrightarrow{\nabla \mathbf{W}}_{ij}^{C} + \overrightarrow{\nabla \mathbf{W}}_{ij}^{U}\Big) \\ &\qquad - \frac{1}{15}\Big(\overrightarrow{\nabla \mathbf{W}}_{D_{ji}} - 2\overrightarrow{\nabla \mathbf{W}}_{j} + \overrightarrow{\nabla \mathbf{W}}_{i}\Big)\Big] \cdot \overrightarrow{ij}, \end{cases} \tag{2.33}$$

where $\overrightarrow{\nabla \mathbf{W}}_i$ and $\overrightarrow{\nabla \mathbf{W}}_j$ are the nodal gradients at vertices i and j. Their definition is out of the scope of the thesis.

All the schemes presented in this section are based on a MUSCL reconstruction of the problem states. All lead to a second-order scheme on any mesh. However, they can be of higher order on a Cartesian mesh for the linear advection equation. Finally, they do not all present the same numerical dissipation. In order to compare the vertex-centred approach with the cell-centred approach, the principles of the cell-centred approach are introduced in the following section.

2.2.6 Cell-centred finite volume scheme

In this section, we provide a detailed overview of the cell-centred finite volume approach, and more particularly the calculation of convective flux. For this approach, the dual cells are mesh elements. Therefore, in contrast to the vertex-centred approach, where one has the option to transport the averaged value per dual cell or the value at the node, with the cell-centred approach the only transported value is the average value per dual cell.

First order scheme

To compute convective flux, the boundary ∂C_i is decomposed into many faces. For a given dual cell C_i, ν_i denotes all the faces of C_i. This leads to a the definition

$$\int_{\partial C_i} F(\mathbf{W}) \cdot \vec{n}_{C_i} \, d\Gamma = \sum_{f_j \in \nu_i} \int_{f_j} F(\mathbf{W}) \cdot \vec{n}_{ij} \, d\Gamma. \tag{2.34}$$

As the unknowns of the problem are the averaged values per dual cell, this results in a Riemann problem being solved per interface for the calculation of the convective flux. The solution to the problem is identical to that presented in 2.24. Finally, to calculate the convective flux to order 1, we solve a Riemann solver per face by assuming that the value at the face is equal to the averaged value per dual cell. There are a number of methods in existence which can be used to increase the accuracy of cell-centred schemes. In the following section, we present the MUSCL approach in cell-centred formalism. This method, based on a linear reconstruction of the unknowns of the problem, allows us to construct a second-order numerical scheme.

Second order scheme

As with the vertex-centred approach, the aim of the MUSCL approach in cell-centred formalism is to construct values extrapolated to the face from the available averaged values. For a given cell i, a Taylor series expansion is performed between the centre of gravity G_i of cell C_i and the middle of the face where the convective flux is to be calculated. In cell-centred formalism, the averaged value is transported by dual cell and the value of the states at the centre of each cell is not known a priori. However, considering that the averaged value is equal to the value at the centre of the cell is valid under a second-order assumption. In concrete terms, we can write:

$$\bar{\mathbf{W}}_i = \mathbf{W}_i + O(h^2). \tag{2.35}$$

Hypothesis (2.35) is sufficient to allow a linear representation of the unknowns in the problem. Finally, the linear extrapolation of the unknowns in the problem is given by (2.36):

$$\begin{aligned} \mathbf{W}_{ij} &= \mathbf{W}_i + \overrightarrow{\Delta r_i} \cdot \overrightarrow{\nabla \mathbf{W}_i}, \\ \mathbf{W}_{ji} &= \mathbf{W}_j + \overrightarrow{\Delta r_j} \cdot \overrightarrow{\nabla \mathbf{W}_j}. \end{aligned} \tag{2.36}$$

The vectors $\overrightarrow{\Delta r_i}$ and $\overrightarrow{\Delta r_j}$ represent the vectors between the centres of gravity of the cells and the midpoint of the edge separating volumes i and j, as can be seen in Figure 2.6.

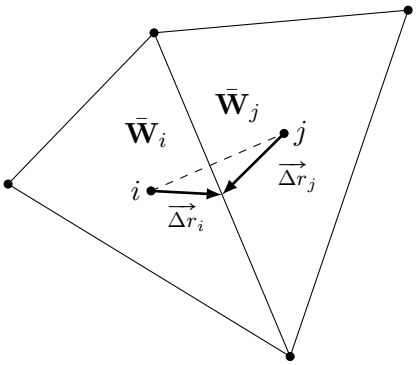


Figure 2.6: MUSCL reconstruction for the cell-centred approach

2.2.7 Boundary conditions

In this section, we detail the treatment of boundary conditions for the vertex-centred finite volume approach. The flux balance for a boundary cell is written as follows:

$$\sum_{f_j \in \nu_i} \int_{f_j} F(\mathbf{W}) \cdot \overrightarrow{n}_{ij} \, d\Gamma = \sum_{f_j \in \nu_i^*} \int_{f_j} F(\mathbf{W}) \cdot \overrightarrow{n}_{ij} \, d\Gamma + \sum_{f_j \in \nu_i^b} \int_{f_j} F(\mathbf{W}) \cdot \overrightarrow{n}_{ij} \, d\Gamma, \tag{2.37}$$

where ν_i^* represents all the facets inside the domain, and ν_i^b represents all the facets on the boundary. The convective flux for the interior facets is calculated using a Riemann solver, as is the case for the other interior faces of the domain. Boundary conditions are applied weakly to the facets on the edge of the domain. Thus, we do not use a Riemann solver and we calculate an analytical flux per boundary face. Specifically, if we denote the boundary state by $\mathbf{W}_b$, then we calculate $F(\mathbf{W}_b) \cdot \overrightarrow{n}_{ij} |S_{ij}|$, where F is the analytical convective flux specific to each system of equations, $\overrightarrow{n}_{ij}$ the outgoing normal to the facet and $|S_{ij}|$ the measure of the facet, as shown in Figure 2.7. The state $\mathbf{W}_b$ will depend on the boundary condition considered. We then distinguish between the scalar advection equation and the Euler equations.

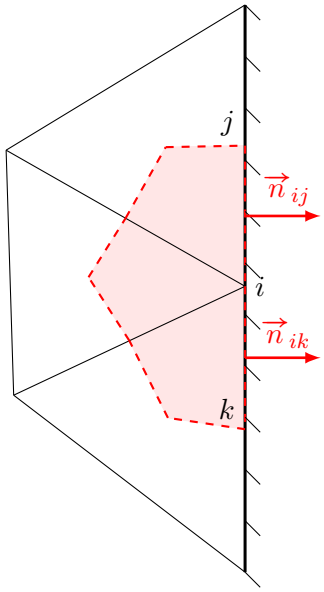


Figure 2.7: Boundary cell

Scalar advection equation

Two boundary conditions are used for the scalar transport equation, which depend on the sign of $\vec{a} \cdot \vec{n}$. Here, $\vec{a}$ represents the advection velocity and $\vec{n}$ represents the outgoing normal to the domain. If $\vec{a} \cdot \vec{n} < 0$, the theory of characteristics shows that the state is required and the Dirichlet boundary condition applies. The boundary state $\mathbf{W}_b$ is then imposed.
If $\vec{a} \cdot \vec{n} > 0$, we have an outgoing flux boundary condition, considering the boundary state to be equal to the state of cell i (i.e. $\mathbf{W}_b = \mathbf{W}_i$). We then calculate the flux at the boundary facet with this defined boundary state.

Euler equations

According to [18], three types of boundary conditions are used for calculations in aeronautics: no-slip boundary condition, symmetry plane boundary condition (or slip boundary condition) and far-field boundary condition.

- No-slip boundary condition:

  This boundary condition states that the tangential velocity of the fluid is zero at the surface of a solid. In other words, the fluid does not slide across the surface of the solid. However, this remains an assumption, and there are cases where this condition is not satisfied. This is equivalent to saying $\mathbf{u}_{\text{fluid}} = \mathbf{u}_{\text{wall}}$, where $\mathbf{u}_{\text{fluid}}$ is the velocity of the fluid and $\mathbf{u}_{\text{wall}}$ is the velocity of the wall. This boundary condition is an example of a Dirichlet boundary condition.

- Symmetry plane boundary condition:

  This condition allows us to reduce the domain of study by taking into account the symmetries of the problem. For this condition, we weakly impose $\mathbf{u} \cdot \mathbf{n} = 0$, with $\mathbf{n}$ the outward normal to the domain. In concrete terms, an intermediate state $\mathbf{W}_b$ is calculated between the states of the fluid domain $\mathbf{W}_i$ and the mirror states $\mathbf{W}_{\text{mirr}} = (\rho, \rho\mathbf{u} - 2(\rho\mathbf{u}\cdot\mathbf{n})\mathbf{n}, \rho E)^\top$. The analytical flow is then calculated from the intermediate state $\mathbf{W}_b$.

- Far-field boundary condition:

  The far-field boundary condition specifies the free flow conditions calculated from the given variables [60]. In concrete terms, an intermediate state $\mathbf{W}_b$ is calculated between the states of the fluid domain $\mathbf{W}_i$ and the states describing the flow at infinity $\mathbf{W}_\infty$. The analytical flow is then calculated from the intermediate state $\mathbf{W}_b$.

2.2.8 Time integration

Following the spatial discretisation of the equations, an ordinary differential equation (or a system of ordinary differential equations) must be solved. To do this, we can distinguish between two classes of methods: explicit methods and implicit methods. Due to practical constraints, both classes of methods have been employed in this study. We introduce the time step, defined as $\Delta t^n = t^{n+1} - t^n$. Additionally, we note the time dependence as follows: $F(\mathbf{W}_i, t^n) = F_i^n$.

Explicit time integration

For explicit time integration, the semi-discretised system for a hyperbolic equation system can be written under the form:

$$|C_i| \frac{d(\tilde{\mathbf{W}}_i)}{dt} + F_i^n + \Gamma_i^n = 0, \tag{2.38}$$

where F_i^n is the convective flux at time t^n and Γ_i^n is given by the boundary condtions. Among all the explicit schemes available in the literature, we have limited ourselves in this study to strong stability preserving schemes, which are based on the concept of stability preserving. Any sequence $(\mathbf{W}^n)$ is said to be strongly stable for a given norm $\|\|$ if, for all n, the condition (2.39) is satisfied:

$$\|\mathbf{W}^{n+1}\| \leq \|\mathbf{W}^n\|, \quad \forall n \geq 0. \tag{2.39}$$

The choice of a strong stability preserving scheme is relevant in the case of a hyperbolic equation (or a system of hyperbolic equations). Written under the divergence form, such equations present the property to ensure that new extrema cannot appear in the domain. As introduced in [61], this class of equations has a range-diminishing property that prevents existing maxima from increasing, existing minima from decreasing, and new maxima or extrema from forming. Although the strong-stability property is not exactly the discrete version of the range-diminishing property, it is very useful for determining solutions to a hyperbolic problem. Indeed, the use of numerical schemes based on this property makes it possible to suppress the formation of parasitic oscillations, while respecting a constraint on the time step. These oscillations are called non-linear instabilities and are often a precursor to the complete instability of the numerical solution itself [61]. Among strong-stability preserving schemes, we can mention strong-stability preserving Runge-Kutta schemes, that we can note RKSSP. An explicit s-step Runge-Kutta scheme can be written in the following general form.

$$\begin{aligned} \mathbf{W}^{(0)} &= \mathbf{W}^n, \\ \mathbf{W}^{(i)} &= \sum_{k=0}^{i-1} (\alpha_{ik}\mathbf{W}^{(k)} + \Delta t \beta_{ik} \mathcal{L}(\mathbf{W}^{(k)})), \\ \mathbf{W}^{n+1} &= \mathbf{W}^{(s)}, \end{aligned} \tag{2.40}$$

where $\mathcal{L}$ represents the right-hand side of the partial differential equation, $\alpha_{ik} \geq 0$ and $\alpha_{ik} = 0$ only if $\beta_{ik} = 0$ [62]. Moreover, for consistency, we must have $\sum_{k=0}^{i-1} \alpha_{ik} = 1,\ i = 1, ..., s$. Numerous Runge-Kutta methods coexist in the literature, and the most commonly used ones are compiled in [63]. In this work, we used the order-3 SSPRK developped by Shu and Osher in [64]. The optimality of the scheme was proved by Gottlied and Shu in [65]. If we denote Δt the time step and $\mathcal{L}$ the right-hand side, this scheme is defined as follows

$$\begin{aligned} \mathbf{W}^{(1)} &= \mathbf{W}^n + \Delta t \mathcal{L}(\mathbf{W}^n), \\ \mathbf{W}^{(2)} &= \frac{3}{4}\mathbf{W}^n + \frac{1}{4}\mathbf{W}^{(1)} + \frac{1}{4}\Delta t \mathcal{L}(\mathbf{W}^{(1)}), \\ \mathbf{W}^{n+1} &= \frac{1}{3}\mathbf{W}^n + \frac{2}{3}\mathbf{W}^{(2)} + \frac{2}{3}\Delta t \mathcal{L}(\mathbf{W}^{(2)}). \end{aligned}$$

Explicit schemes are conditionally stable schemes. As a result, these schemes are only stable if a time step constraint is respected. For example, the SSP-RK3 scheme is stable under the following CFL condition [66]:

$$\Delta t \leq \frac{\max_i \Delta x_i}{|\lambda|_{i,\max}}, \tag{2.41}$$

where $|\lambda|_{i,\max}$ is the magnitude of the largest eigenvalue of the matrix $\bar{\bar{A}}$ of the hyperbolic system (2.3) in the cell i.

Implicit time integration

For implicit time integration, the semi-discretised system for a hyperbolic equation system is now:

$$|C_i|\frac{d(\tilde{\mathbf{W}}_i)}{dt} + F_i^{n+1} + \Gamma_i^{n+1} = 0. \tag{2.42}$$

System (2.42) is linearised since the flux is unknown at time t^{n+1}:

$$\begin{cases} F_i^{n+1} &= F_i^n + \dfrac{\partial F}{\partial \mathbf{W}}\delta\mathbf{W}_i^{n+1} \\ \Gamma_i^{n+1} &= \Gamma_i^n + \dfrac{\partial \Gamma}{\partial \mathbf{W}}\delta\mathbf{W}_i^{n+1} \end{cases} \tag{2.43}$$

where $\delta\mathbf{W}^{n+1} = \mathbf{W}^{n+1} - \mathbf{W}^n$. Thus, we can rewrite system (2.42) in its linearised form using a backward Euler formula:

$$\Big(\frac{|C_i|}{\Delta t^n}\mathbb{I} + \frac{\partial F}{\partial \mathbf{W}} + \frac{\partial \Gamma}{\partial \mathbf{W}}\Big)\delta\mathbf{W}^{n+1} = -F_i^n - \Gamma_i^n \tag{2.44}$$

where $\mathbb{I}$ is the identity matrix. We now wish to solve the resulting linear system. Of the many methods available in the literature, we use the lower-upper symmetric successive overrelaxation (LU-SSOR) scheme. This scheme, proposed by Jameson and Yoon [67], was developed for stationary Euler and Navier-Stokes equations. This method, applied to transonic flows, has proven to be effective and robust.

2.2.9 Residual computation

In this work, we will look for stationary solutions to some test cases. For a stationary solution, the time derivative of the problem must be zero, meaning that the flux balance on each dual cell must also be zero. We denote the residual for each dual cell i by Res_i and define it as follows.

$$Res_i = \frac{1}{|C_i|}(F_i + \Gamma_i) \tag{2.45}$$

where $|C_i|$ is the measure of the dual cell, F_i is the convective flux and Γ_i is given by the boundary conditions. We calculate the norm of the residuals to evaluate the convergence of a solution to the stationary solution. We consider the calculation to have converged to the stationary solution if the residual norm is less than a given ε. In this work, we set ε to 10^{-14}, which we consider to be equal to machine zero.

2.3 Comparison between cell-centred and vertex-centred finite volume approaches

This difference leads to different numerical methods and therefore different results. Currently, the cell-centred method is widely used in industrial calculation codes, such as the elsA [68] code co-owned by ONERA, SAFRAN and AIRBUS. Despite several decades of experience with

the cell-centred approach, certain limitations are prompting a shift towards the vertex-centred approach. We can therefore seek to compare the two approaches in order to understand the advantages of the vertex-centred approach over the cell-centred approach.

Several articles propose comparing cell-centred and vertex-centred approaches. In their article [69], Wang, Schwöppe, and Heinrich show that the two approaches give similar results for structured or quasi-structured meshes. This can be explained by the fact that, for these meshes, the number of degrees of freedom (i.e., the number of dual cells) is of the same order of magnitude for both approaches. We have also seen that the calculation of the gradient does not degrade the order of the numerical scheme in the case of structured or quasi-structured meshes. It is therefore consistent to see similar results. However, this article suggests that the cell-centred approach gives better results in the case of unstructured meshes. Although this is true for a given mesh, this result should be qualified. Indeed, for a 3D tetrahedral mesh, there are 5 to 6 times more degrees of freedom in the cell-centred case than in the vertex-centred case. The greater accuracy of the cell-centred method can be explained by the greater number of degrees of freedom. As the solutions are not compared for the same number of degrees of freedom, this does not provide an answer as to the accuracy of one method compared to the other. Other articles propose comparing the two approaches [70, 71, 72]. What emerges from these documents is the difficulty of determining a metric for comparison, since the results seem to depend on the comparison method used. There is no consensus on which comparison method to use.

In the context of anisotropic mesh adaptation, highly anisotropic elements may appear. This anisotropy has consequences for the accuracy and robustness of numerical schemes, particularly for the cell-centred approach. In fact, to construct a higher-order scheme for convective fluxes or to evaluate the diffusive flux, it is necessary to know the gradients of the problem states. One of the classic approaches for calculating gradients in the cell-centred formulation is the Green-Gauss gradient derived from the divergence theorem. For a given cell C_i, the gradient is calculated by:

$$\nabla \mathbf{W}_i = \frac{1}{|C_i|} \sum_{f_j \in \nu_i} \mathbf{W}_{f_j} \overrightarrow{n}_{ij} |S_{ij}|, \tag{2.46}$$

where $\mathbf{W}_{f_j}$ is the value at the centre of the face. The challenge lies in constructing a value for the face based on the available averaged values. Under a second-order assumption, we can assume that the averaged value per dual cell is equal to the value at the centre of the cell. We can then extrapolate a value for the face from these values at the center of the cell. One approach would be to perform a linear extrapolation of the face value. This extrapolation can be written as follows:

$$\mathbf{W}_{f_j} = \alpha \mathbf{W}_i + (1-\alpha)\mathbf{W}_j \quad \text{where} \quad \alpha = \frac{\|\overrightarrow{if}\|}{\|\overrightarrow{ij}\|}.$$

Figure 2.8 shows how the value is reconstructed at the face. We denote the value at the centre of the face as $\mathbf{W}_f$ and the extrapolated value as $\mathbf{W}_{f_j}$. When calculating the gradient, we assume that $\mathbf{W}_{f_j}$ equals $\mathbf{W}_f$. As can be seen, for regular meshes with little or no anisotropy, this assumption is valid and yields a gradient of order 1. However, for deformed meshes, this method provides an inaccurate approximation of the face value and results in an inconsistent gradient [73]. Methods have been developed to correct the gradients obtained using the Green-Gauss method on any mesh; in particular, the quasi-Green method proposed by P. Brenner [74]. The same problem does not arise with the vertex-centred approach. Indeed, due to the construction

of the dual cell, two adjacent cells always intersect their common boundary at its centre. There are other arguments in favour of the vertex-centred approach. On the one hand, the rank-1 neighbourhood of a cell is generally richer. In the case of a triangular mesh, the cell-centred approach gives a rank 1 neighbourhood of 3 (one neighbour per edge of the triangle), whereas the cell-vertex approach gives a rank 1 neighbourhood of at least 3 (the number of triangles to which a given vertex belongs) and is generally 5 or 6. This rank 1 neighbourhood provides more information for calculating the gradient on the dual cell. On the other hand, in the case of the cell-centred approach, convective fluxes are calculated in the direction of the cell faces. In contrast, for the vertex-centred approach, convective fluxes are calculated in the direction of the mesh edges. Thus, in the context of anisotropic mesh adaptation, there is an alignment between the edges of the mesh and the preferred direction of flow. This is very interesting for the study of highly anisotropic phenomena such as shocks or vortices. All these factors make the vertex-centred approach the preferred approach in the context of anisotropic mesh adaptation, as these methods have proven to be accurate and robust on highly anisotropic meshes [18].

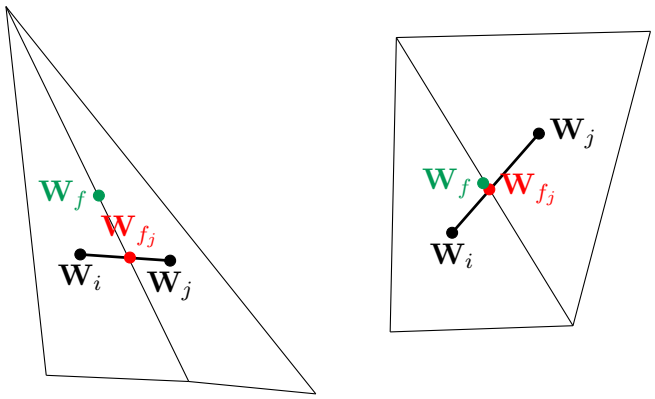


Figure 2.8: Error cell-centred gradient

2.4 Limitations and areas for improvement

The V4 scheme, which was originally proposed by A. Dervieux [37], has been used in recent work by Alauzet *et al.* [18, 75], demonstrating robustness and accuracy on meshes with anisotropy factors of up to one million. This scheme belongs to a class described by Nishikawa as 'edge-based'[36]. Starting from a system of hyperbolic conservation equations, the equations are discretised, replacing the continuous problem with a discrete one where the unknowns are the values at the mesh vertices. This approach is not a finite volume approach, since the unknowns are not the averaged values per dual cell. It should be noted that, to our knowledge, there is currently no vertex-centred finite volume V4 scheme for compressible Euler equations in the literature designed for mesh adaptation. However, finite volume schemes offer advantages over the edge-based discretisation of equations. Firstly, the unknowns are the averaged values per dual cell. This ensures that the numerical scheme is conservative by construction. On the other hand, the edge-based formulation results in the discretisation (2.30) of the equations. It has been proven that this discretisation is of order at most 3 [36], meaning it seems impossible to construct an edge-based numerical scheme of a higher order with a single numerical flux per edge. Conversely, the finite volume discretisation enables the construction of a numerical scheme of arbitrary degree. Furthermore, the finite volume approach enables better treatment of boundary conditions. Within the context of edge-based discretisation, boundary conditions are generally calculated for each edge vertex. As it is not possible to define a normal vector at a vertex, the normal vector is instead defined using the normal vectors of the dual facets.

This method of defining boundary conditions can introduce errors, particularly when a dual cell shares two boundary conditions. Such a situation also occurs for adiabatic or isothermal walls for which the no slip boundary condition is enforced at the node. This is a prerequite to recover the good gradients used for the inviscid terms. In a finite volume approach, however, the boundary conditions can be treated for each dual facet. Calculating the convective flux for each dual facet ensures that the boundary conditions are always well posed. This pushes us towards a finite volume discretisation rather than an edge-based formulation. We therefore propose constructing a vertex-centred finite volume scheme based on the V4 scheme. The aim is to maintain the precision and robustness of the V4 scheme on an anisotropic, adapted mesh, while exploiting the benefits of a finite volume scheme.
The next chapters are devoted to the analysis of the edge-based V4 scheme and its extension using the finite volume formulation.

Chapter 3

Edge-based V4 scheme

Contents

3.1 Numerical scheme 51
3.1.1 Construction of the numerical scheme 51
3.1.2 Boundary condition 59
3.2 Numerical results : 2D scalar transport test case 60
3.2.1 Test case description 60
3.2.2 Truncation error 61
3.2.3 Discretization error 65
3.3 Numerical results : 2D Euler test case 69
3.3.1 The convection of a vortex 69
3.3.2 The Ringleb flow 73
3.4 Conclusion 79

3.1 Numerical scheme

Anisotropic mesh adaptation enables the automatic alignment of mesh edges with preferred flow directions, while minimising discretisation error, and so minimising the required mesh nodes for a given accuracy. This approach requires robust numerical schemes, since anisotropy and strong mesh size variations generally reduce the accuracy of numerical schemes. In this work, we focus on meshes composed exclusively of simplices (*i.e.* triangles in 2D and tetrahedra in 3D). Until now, the approach that has shown the best robustness is the edge-based approach, in which the unknowns of the problem are located at the mesh nodes, and in particular the V4 scheme. The V4 scheme is an edge-based numerical scheme proposed by A. Dervieux and Debiez [37]. Such a scheme performs very well on the highly anisotropic meshes encountered during the mesh adaptation loop, with an anisotropy factor of around one million [18, 75]. The V4 scheme has also been employed in other studies involving anisotropic mesh adaptation. We can cite the work of Lugrin *et al.* [76] or Puigt *et al.* [77] on the study of hypersonic flows around a vehicle, or the work of Barreau *et al.* [78, 79] on the simulation of electric arcs. In this chapter, we present a comprehensive study of the V4 numerical scheme. We will detail its construction in a 2D configuration before studying its behaviour.

3.1.1 Construction of the numerical scheme

Second-order schemes are popular because they strike a good balance between accuracy and robustness, particularly on highly anisotropic meshes. One popular approach to constructing second-order schemes is the MUSCL technique[17, 39, 40], originally developed by Van Leer.

This approach improves the precision of the numerical scheme. In the context of the vertex-centred formulation, the idea is to replace the fields $\mathbf{W}_i$ and $\mathbf{W}_j$ by the extrapolated values $\mathbf{W}_{ij}$ and $\mathbf{W}_{ji}$ to calculate the flux between cells C_i and C_j:

$$\begin{cases} \mathbf{W}_{ij} = \mathbf{W}_i + \dfrac{1}{2}(\overrightarrow{r_{ij}} \cdot \overrightarrow{\nabla \mathbf{W}_i}), \\ \mathbf{W}_{ji} = \mathbf{W}_j - \dfrac{1}{2}(\overrightarrow{r_{ij}} \cdot \overrightarrow{\nabla \mathbf{W}_j}). \end{cases} \tag{3.1}$$

The vector $\overrightarrow{r_{ij}}$ represents the vector from vertex i to vertex j. The factor $1/2$ comes from the fact that the numerical flux is always calculated in the middle of the edge connecting vertex i to vertex j. The extrapolated values $\mathbf{W}_{ij}$ needs the gradient $\nabla \mathbf{W}_i$.

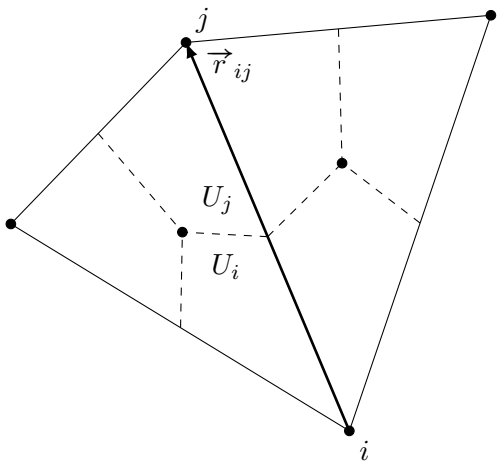


Figure 3.1: Construction of extrapolated values

Dervieux proposes obtaining a high-order gradient by performing a linear combination of several gradients. This leads to the V4 and V6 schemes [37]. Here, we detail the construction of the high-order gradient for the V4 scheme. For a given edge ij, the two triangles/tetrahedra most aligned with the edge are identified, as depicted in Figure 3.2. These two elements are labelled T_{ij} and T_{ji} in Figure 3.2 and D (Down), C (centred), and U (Up) correspond to the locations relative to edge ij. The gradient for the MUSCL reconstruction is a linear combination of the centred, upwind (defined on the U element) and downwind (defined on the D element) gradients.

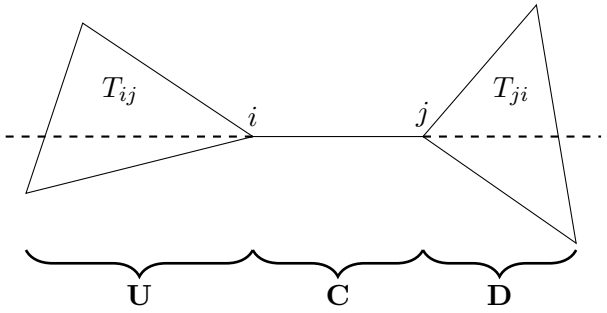


Figure 3.2: Downwind triangle T_{ij} and upwind triangle T_{ji} associated with edge ij.

In this way, the high-order gradient is built in the direction of the edge, making the V4 scheme a 1D scheme in the direction of each edge. We therefore have an alignment between the preferred

direction of the flow and the edges of the mesh. It is this 1D construction of the numerical scheme that seems to explain the robustness on a highly anisotropic mesh:

$$\begin{cases} \overrightarrow{\nabla \mathbf{W}}_i^{\beta} \cdot \overrightarrow{ij} = (1-\beta)\overrightarrow{\nabla \mathbf{W}}_{ij}^{C} \cdot \overrightarrow{ij} + \beta \overrightarrow{\nabla \mathbf{W}}_{ij}^{U} \cdot \overrightarrow{ij} \\ \overrightarrow{\nabla \mathbf{W}}_j^{\beta} \cdot \overrightarrow{ij} = (1-\beta)\overrightarrow{\nabla \mathbf{W}}_{ij}^{C} \cdot \overrightarrow{ij} + \beta \overrightarrow{\nabla \mathbf{W}}_{ij}^{D} \cdot \overrightarrow{ij}. \end{cases} \tag{3.2}$$

The parameter $\beta \in [0,1]$ is introduced to define a family of schemes, called β-scheme. For $\beta = 0$ the scheme is centred and unstable, and for $\beta = 1$, the scheme is fully upwind. This scheme is third-order accurate for the linear scalar transport equation on two-dimensional Friedrich-Keller mesh when β is set to 1/3 [37]. In the general case, *i.e.* on any mesh and for a non-linear equation, the scheme is of order 2 [44, 42, 41]. Moreover, it can be shown that, for $\beta = 1/3$ and for unstructured meshes composed of simplexes, this scheme is of order 2 with a numerical diffusion of order 4 [18]. The scheme with $\beta = 1/3$ is the V4 scheme.

The centred gradient, which is relative to the edge ij, is implicitly defined by:

$$\overrightarrow{\nabla \mathbf{W}}_{ij}^{C} \cdot \overrightarrow{ij} = \mathbf{W}_j - \mathbf{W}_i \quad \text{and} \quad \overrightarrow{\nabla \mathbf{W}}_{ij}^{C} \cdot \overrightarrow{ji} = \mathbf{W}_i - \mathbf{W}_j.$$

The upwind and downwind contributions are computed using the P^1-Galerkin gradient. Let K be an element of the primal mesh based on vertices labeled a, b and c. Let V_h be the approximation space associated with the P^1 Lagrange finite element. We denote by ϕ_a, ϕ_b and ϕ_c the basis function of the associated vertex [80]. The P^1-Galerkin gradient on triangle/tetrahedron K is:

$$\overrightarrow{\nabla \mathbf{W}}|_K = \sum_{v \in \{a,b,c\}} \mathbf{W}_v \overrightarrow{\nabla \phi}_v, \tag{3.3}$$

where the basis function are defined:

$$\begin{cases} \phi_a = \dfrac{1}{2|T|}\big(x_b y_c - x_c y_b + (y_b - y_c)x + (x_c - x_b)y\big) \\ \phi_b = \dfrac{1}{2|T|}\big(x_c y_a - x_a y_c + (y_c - y_a)x + (x_a - x_c)y\big) \\ \phi_c = \dfrac{1}{2|T|}\big(x_a y_b - x_b y_a + (y_a - y_b)x + (x_b - x_a)y\big). \end{cases}$$

The finite element gradient is therefore:

$$\overrightarrow{\nabla \mathbf{W}}|_K = \begin{pmatrix} \dfrac{(y_b - y_c)\mathbf{W}_a + (y_c - y_a)\mathbf{W}_b + (y_a - y_b)\mathbf{W}_c}{2|T|} \\ \dfrac{(x_c - x_b)\mathbf{W}_a + (x_a - x_c)\mathbf{W}_b + (x_b - x_a)\mathbf{W}_c}{2|T|} \end{pmatrix}.$$

The finite element gradient is considered to estimate upwind and downing contributions in Eq. 3.2: $\overrightarrow{\nabla \mathbf{W}}_{ij}^{U} = \overrightarrow{\nabla \mathbf{W}}|_{K_{ij}}$ and $\overrightarrow{\nabla \mathbf{W}}_{ij}^{D} = \overrightarrow{\nabla \mathbf{W}}|_{K_{ji}}$.

Remark :
It can be shown that the P^1-Galerkin gradient is equivalent to the Green-Gauss gradient on a triangle/tetrahedron when the fields are known at the mesh nodes and the solution supposed linear over the element. Indeed, the Green-Gauss gradient is given by the relation 3.4:

$$\overrightarrow{\nabla \mathbf{W}}|_K = \frac{1}{|T|} \sum_{f \in \nu_f} \mathbf{W}_f \overrightarrow{S}_f, \tag{3.4}$$

where ν_f is the set of faces of the element K. Since the fields are defined at the mesh nodes, the value at the face $\mathbf{W}_f$ is defined as the average of the quantities at the edge vertices i and j of the element K:

$$\mathbf{W}_f = \frac{1}{2}(\mathbf{W}_i + \mathbf{W}_j),$$

and it comes:

$$\begin{aligned}
\overrightarrow{\nabla \mathbf{W}}|_K &= \begin{pmatrix} \dfrac{(y_b - y_a)(\mathbf{W}_b + \mathbf{W}_a) + (y_c - y_b)(\mathbf{W}_c + \mathbf{W}_b) + (y_a - y_c)(\mathbf{W}_a + \mathbf{W}_c)}{2|T|} \\ \dfrac{(x_a - y_b)(\mathbf{W}_b + \mathbf{W}_a) + (x_b - x_c)(\mathbf{W}_c + \mathbf{W}_b) + (x_c - x_a)(\mathbf{W}_a + \mathbf{W}_c)}{2|T|} \end{pmatrix} \\
&= \begin{pmatrix} \dfrac{(y_b - y_c)\mathbf{W}_a + (y_c - y_a)\mathbf{W}_b + (y_a - y_b)\mathbf{W}_c}{2|T|} \\ \dfrac{(x_c - y_b)\mathbf{W}_a + (x_a - x_c)\mathbf{W}_b + (x_b - x_a)\mathbf{W}_c}{2|T|} \end{pmatrix}.
\end{aligned}$$

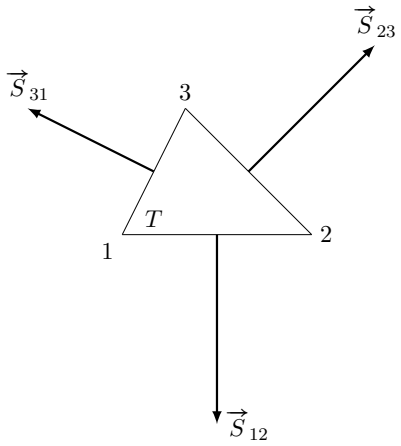
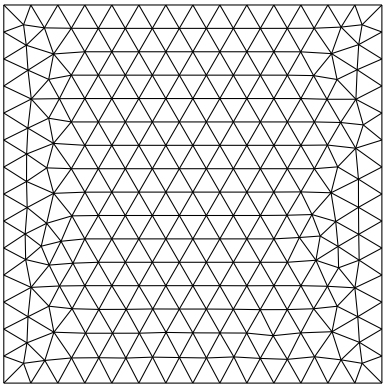
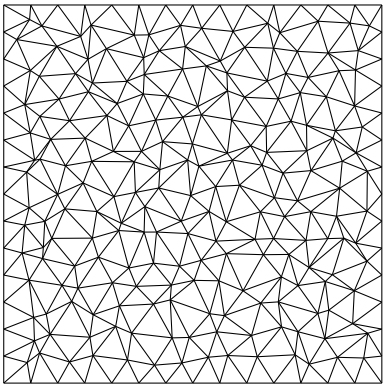


Figure 3.3: Notations for calculating gradients on triangular mesh.

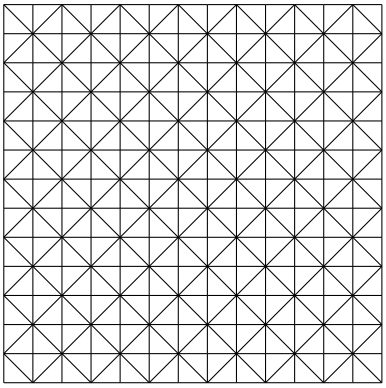
The convergence order of the finite element gradient is assessed. In order to achieve this objective, four distinct types of meshes are defined: regular, perturbed, Friedrich-Keller and Union Jack. An illustration of the meshes is given in Figure 3.4. The regular meshes are generated using GMSH software [81]. Their generation is based on an advancing front algorithm [82]. To obtain the perturbed meshes, the regular meshes are randomly deformed in order to create irregularity in the mesh. Finally, the Friedrich-Keller and Union Jack meshes are generated by hand, using scripts. For Union Jack and Friedrich-Keller meshes, the dual centres and nodes of the mesh are at the same location, except at the boundary. This is also the case for regular meshes, although the difference in location between vertices and cell centre at the boundary is more irregular than for Union Jack and Friedrich-Keller meshes. Finally, disturbed meshes do not exhibit regularity and the dual centres and nodes of the mesh do not coincide. For each type of mesh, a hierarchy is defined in order to compute the order of convergence on different types of mesh.



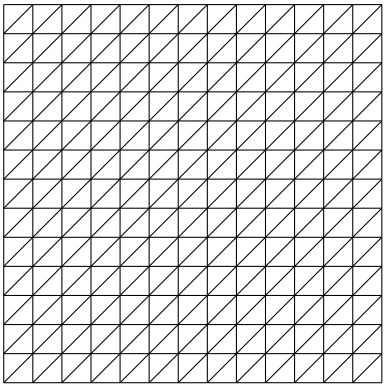
(a) Regular mesh



(b) Disturbed mesh



(c) Union jack mesh



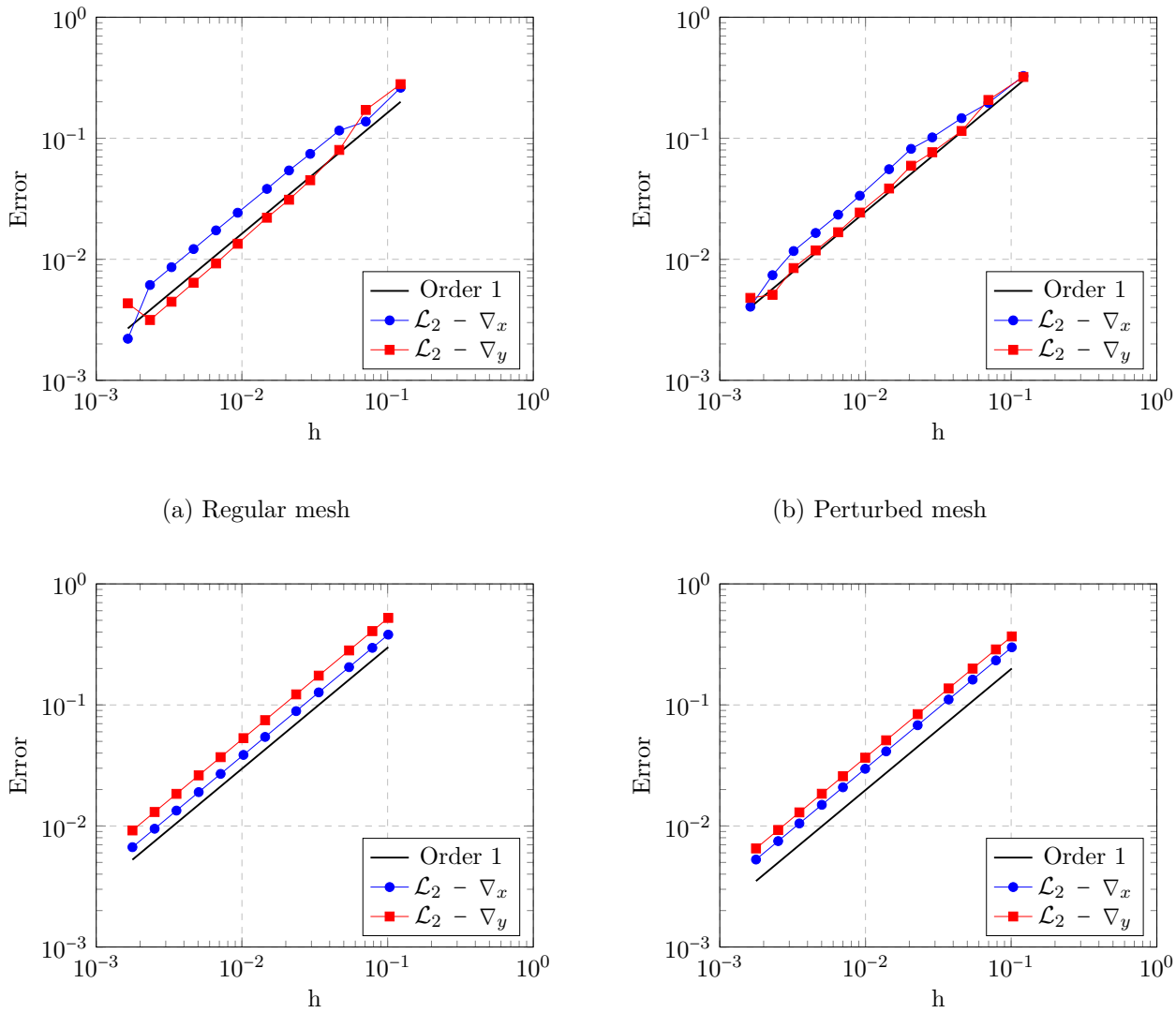
(d) Friedrich-Keller mesh

Figure 3.4: Exemple regular, perturbed, Friedrich-Keller and Union Jack meshes, clockwise from top left.

Table 3.1 recalls the number of elements for each mesh type.

	Regular	**Disturbed**	**Friedrich-Keller**	**Union Jack**
1	44	44	64	64
2	118	118	100	100
3	259	259	196	196
4	621	621	400	400
5	1190	1190	1024	1024
6	2354	2354	2704	2704
7	5850	5850	5184	5184
8	11523	11523	10404	10404
9	23332	23332	20164	20164
10	46683	46683	40804	40804
11	91941	91941	79524	79524
12	184880	184880	160000	160000

Table 3.1: Number of vertices for each mesh of the unit square



(a) Regular mesh

(b) Perturbed mesh

(c) Friedrich-Keller mesh

(d) Union Jack mesh

Figure 3.5: $\mathcal{L}_2$ error for finite element gradient on regular, perturbed, Friedrich-Keller and Union Jack meshes, clockwise from top left.

Figure 3.5 shows the evolution of the $\mathcal{L}_2$ error norm for the finite element gradient as a function of mesh size. In our analysis, the mesh size will always be linked with dual cell area through the following relation:

$$h_{eff} = \frac{1}{N} \sum_i |C_i|^{\frac{1}{d}}, \tag{3.5}$$

where $|C_i|$ is the area of the dual cell, and d the dimension of the problem (here $d = 2$). Error analysis is performed using an analytical function. The function given in 3.6 is evaluated at each vertex of the mesh. Then, a gradient is calculated for each triangle. Finally, the numerical results are compared with the theoretical gradient evaluated at the centre of each triangle. As

can be seen in Figure 3.5, the finite element gradient is of order 1 on all types of mesh.

$$\begin{aligned} f(x,y) &= \sin(xy), \\ \overrightarrow{\nabla} f &= \begin{pmatrix} y\cos(xy) \\ x\cos(xy) \end{pmatrix}, \\ \bar{\bar{\Delta}} f &= \begin{pmatrix} -y^2\sin(yx) & \cos(xy) - xy\sin(xy) \\ \cos(xy) - xy\sin(xy) & -x^2\sin(xy) \end{pmatrix}. \end{aligned} \tag{3.6}$$

Since second-order scheme are not monotonous, they can lead to oscillations. These oscillations can lead to a solution of poor quality or lead to non-respect of the admissible states (negative density for example). To counter this, a widely used technique is to guarantee the TVD property (*total variation diminishing*) in 1D [83] or the LED property in 2D/3D [84]. This guarantees that the values $\mathbf{W}_{ij}$ and $\mathbf{W}_{ji}$ are admissible solutions of the initial problem. Concretely, the value of the gradient $(\nabla \mathbf{W}_i)$ is replaced by the value $(\nabla \mathbf{W}_i)^{\text{Lim}}$. A well-suited limiter for the V4 scheme is the one of Piperno [85], which is the extension of the Van Albada limiter to the β scheme:

$$\overrightarrow{\nabla \mathbf{W}}^{\text{Lim}} = \overrightarrow{\nabla \mathbf{W}}^{U} \psi_{PI}(R),$$

with

$$R = \frac{\overrightarrow{\nabla \mathbf{W}}^{C} \cdot \overrightarrow{ij}}{\overrightarrow{\nabla \mathbf{W}}^{U} \cdot \overrightarrow{ij}} \quad \text{and} \quad \psi_{PI}(R) = \left(\frac{1}{3} + \frac{2}{3}R\right) \begin{cases} \dfrac{3\frac{1}{R^2} - 6\frac{1}{R} + 19}{\frac{1}{R^3} - 3\frac{1}{R} + 18} & \text{if} \quad R < 1, \\ 1 + \left(\dfrac{3}{2}\dfrac{1}{R} + 1\right)\left(\dfrac{1}{R} - 1\right)^3 & \text{if} \quad R \geq 1. \end{cases}$$

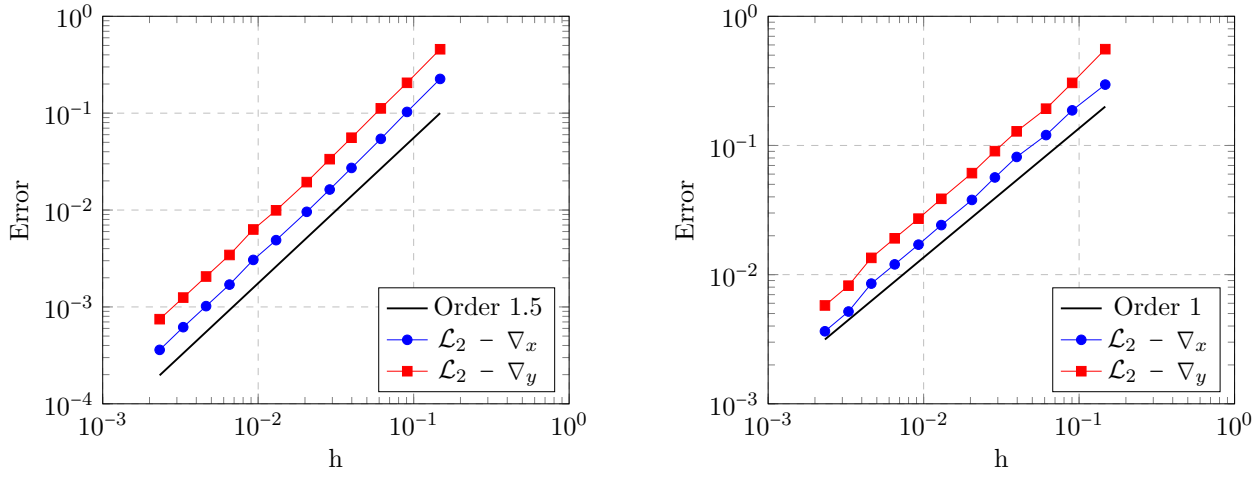
For some edges with one or two vertices on the boundary of the domain, it is not always possible to define an upwind element because the element is behind the boundary condition. Two possibilities can be found to overcome this difficulty. One can locally switch to a numerical scheme of order 1 or, alternatively, a new high-order gradient can be defined, using an alternative definition of the upwind gradient. This last approach was proposed in [18]. It requires Clément's gradient [86] that can be seen as a weighted average of the P^1 finite element in all triangles that share a given vertex i:

$$\overrightarrow{\nabla \mathbf{W}}_i = \frac{\sum\limits_{K_j \ni i} |K_j| (\nabla \mathbf{W})|_{K_j}}{\sum_{K_j \ni i} |K_j|} = \sum_{K_j \ni i} \frac{|K_j|}{(d+1)|C_i|} (\nabla \mathbf{W})|_{K_j},$$

where $\overrightarrow{\nabla \mathbf{W}}|_{K_j}$ is the P^1-Galerkin gradient of element K_j given by relation 3.3, and $|K_j|$ and $|C_i|$ denote the volume of the primal element K_j and dual element C_i. For a dual volume defined using the median cell approach (see Figure 2.1), $(d+1)|C_i| = \sum_{K_j \ni P_i} |K_j|$, where d is the dimension of the problem.

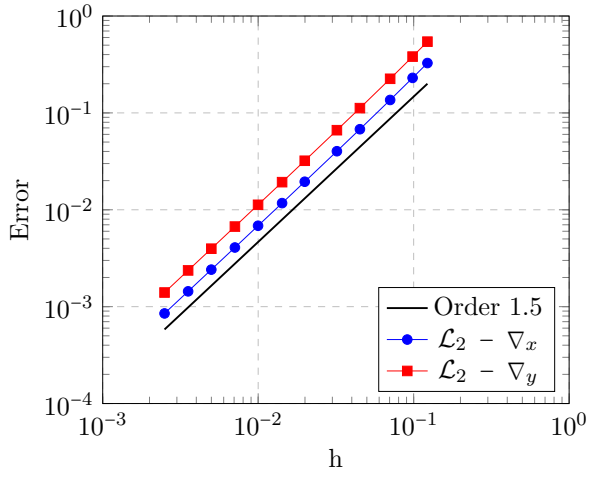
In order to verify the convergence order of Clément's gradient, let's use again the four types of meshes defined in 3.4. The reference function used is always the function given in 3.6 and it is evaluated at each vertex of the mesh. Figure 3.6 shows the evolution of the norm $\mathcal{L}_2$ of the error for the Clément gradient as a function of the mesh size. The gradient is of order 1 on disturbed meshes, and on other types of meshes, it has the hybrid order 1.5. This value does not

correspond to the power of a Taylor expansion. However, the same kind of result is provided in [87] and it is explained that this value is a blending due to a given behaviour for internal nodes, and another one for boundary vertices. In order to speak strictly of order, it is necessary to distinguish between interior and boundary points. Therefore, Clement's gradient is of order 2 for interior points and of order 1 for boundary points. Once the stencil is symmetric, we could argue that Clément's gradient is of order 2, otherwise, it is of order 1. To summarize, Clément's gradient is at least of order 1 on any mesh type, which is sufficient for the reconstruction of a high-order gradient.

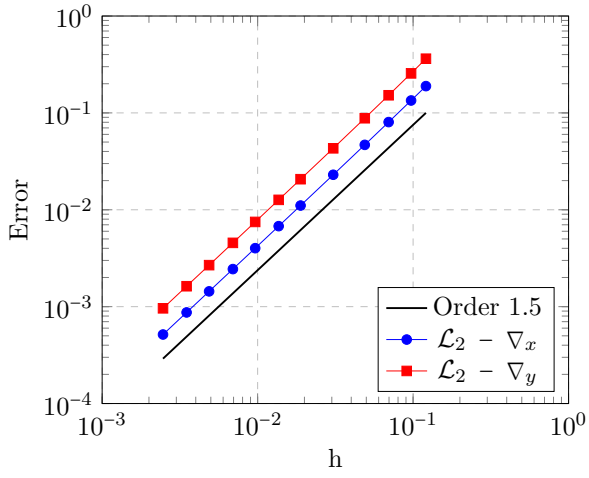


(a) Regular mesh

(b) Perturbed mesh



(c) Friedrich-Keller mesh



(d) Union Jack mesh

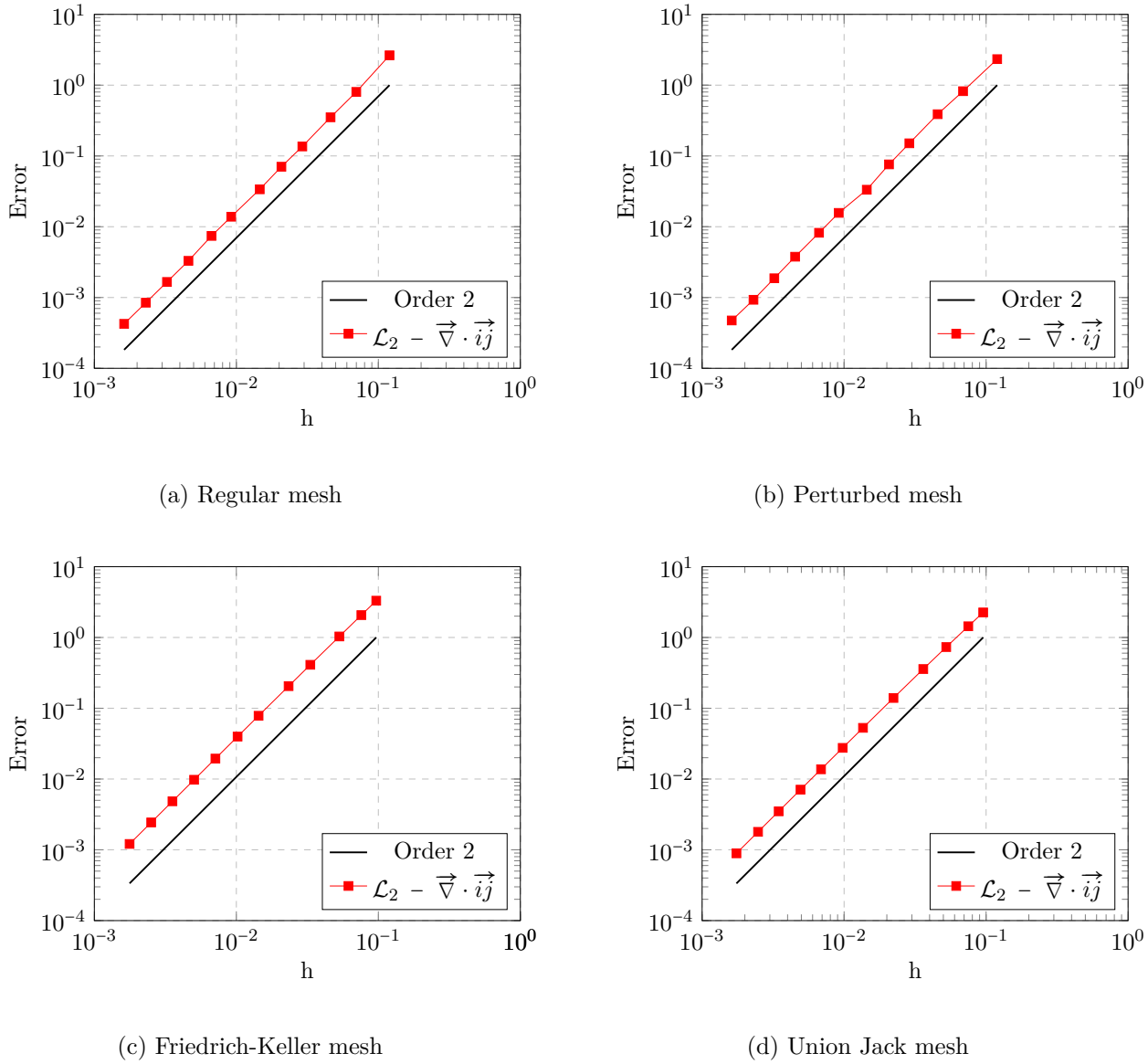
Figure 3.6: $\mathcal{L}_2$ error for the Clément gradient on regular, perturbed, Friedrich-Keller and Union Jack meshes, clockwise from top left.

We can then define the upwind gradient by:

$$\overrightarrow{\nabla \mathbf{W}}_{ij}^{U} \approx 2\overrightarrow{\nabla \mathbf{W}}_{i} - \overrightarrow{\nabla \mathbf{W}}_{ij}^{C}. \tag{3.7}$$

Finally, we want to evaluate the convergence order of upwind edge gradient. To do this, we again use the four types of meshes defined in 3.4. Figure 3.7 illustrates the evolution of the $\mathcal{L}_2$ norm of the error for the upwind edge gradient of the V4 scheme. By evaluating the function

given in 3.6 at the edge nodes, it can be seen that the resulting gradient is of order 2 for any mesh type. This is consistent with the aim of constructing a high-order, second-order gradient on any type of mesh for the V4 scheme.



(a) Regular mesh

(b) Perturbed mesh

(c) Friedrich-Keller mesh

(d) Union Jack mesh

Figure 3.7: $\mathcal{L}_2$ error for the boundary upwind gradient on regular, perturbed, Friedrich-Keller and Union Jack meshes, clockwise from top left.

3.1.2 Boundary condition

This section details the treatment of boundary conditions for an edge-based discretisation of the equations. To achieve this, we do not use a Riemann solver. For a given vertex i, the idea is to calculate a state $\mathbf{W}_b$ from the states $\mathbf{W}_i$ and $\mathbf{W}_{b_i}$, as defined by the boundary condition (see Figure 3.8). The analytical flux is then calculated from this state $\mathbf{W}_b$. As there is no normal defined at vertex i, one analytical flux is computed per each dual facet, using the normals $\vec{n}_{ik}$ and $\vec{n}_{ij}$ defined on each dual facet. To summarize, a flux is computed per boundary facet, using the normal vector associated to the boundary face. If the boundary state depends only on the value at vertex i and not on the geometry, the flux can be simplified into:

$$F(\mathbf{W}_b)|_i \overrightarrow{n}_{ik}|S_{ik}| + F(\mathbf{W}_b)|_i \overrightarrow{n}_{ij}|S_{ij}| = F(\mathbf{W}_b)|_i(\overrightarrow{n}_{ik}|S_{ik}| + \overrightarrow{n}_{ij}|S_{ij}|),$$

where $|S_{ik}|$ is the size of the dual facet ik. This can be seen as a way to apply the boundary conditions in the vertex i.

Remark :
Such a procedure is partially applied to boundary vertices at the wall for the Navier-Stokes equations: the no-slip boundary condition is enforced on the velocity components. It is mandatory to close the diffusion terms and have the right U and D gradients for the V4 scheme.

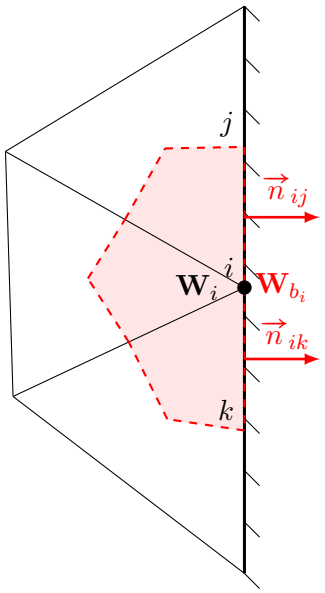


Figure 3.8: Notations for boundary conditions.

3.2 Numerical results : 2D scalar transport test case

3.2.1 Test case description

In this section, we present the results obtained for the advection equation. More specifically, we present in (3.8) the case of a circularly advected Gaussian taken from [88]. This test case has an exact stationary solution, which makes it relevant for conducting convergence studies. Furthermore, this test case has a non-constant velocity, since the velocity depends on the x and y coordinates. The exact solution is given in figure (3.9). The domain of study is the unit square of size $[0,1] \times [0,1]$. For this test case, we consider two types of boundary conditions. On the lower side, we have a Dirichlet boundary condition, where we impose the exact solution. More precisely, we calculate the exact solution at the centre of each facet, then we calculate a numerical flux per facet, which looks like a second-order finite volume formulation. For the other edges of the boundary, we consider an outflow boundary condition. Since this boundary condition only involves the internal state, we calculate a flow per dual facet with the internal state defined at the vertex. The set of equations is:

$$\begin{cases} \frac{\partial w}{\partial t} + y\frac{\partial w}{\partial x} + (1-x)\frac{\partial w}{\partial y} = 0; & \forall (x,y) \in [0,1] \times [0,1], \\ w(x,y,0) = 0; & \forall (x,y) \in [0,1] \times [0,1], \\ w(x,0,t) = \exp(-50(x-0.5)^2); & \forall x \in [0,1]; \forall t \geqslant 0. \end{cases} \tag{3.8}$$

with the exact solution

$$w_{\text{exact}}(x, y, t) = \exp(-50(r - 0.5)^2) \quad ; \quad \text{with} \quad r = \sqrt{(1-x)^2 + y^2}$$

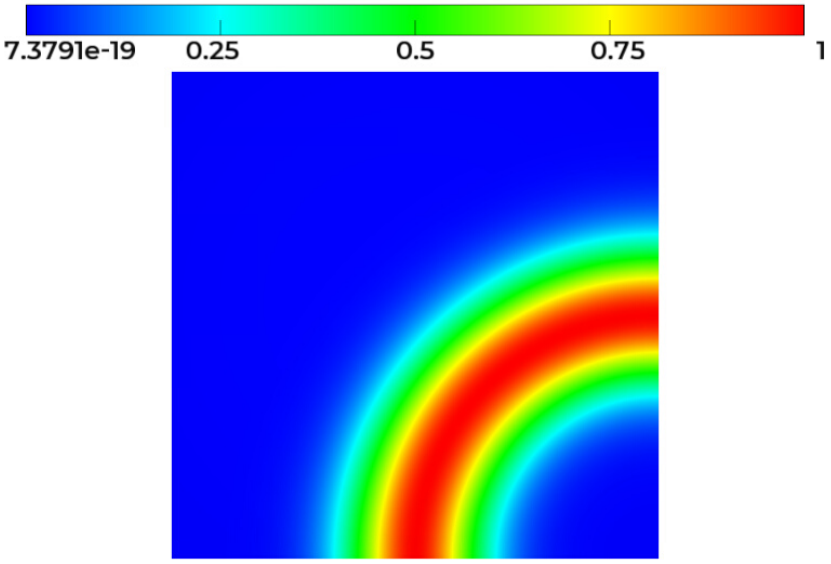


Figure 3.9: Exact solution on a regular mesh for the test case (3.8)

For the advection test case, we use the four mesh types shown in Figure 3.4, which enables us to examine how the V4 scheme behaves on different meshes.

3.2.2 Truncation error

In this section, we present results regarding the truncation error, which is defined as follows.

$$TE_p = \Big(\frac{1}{N}\sum_i \Big|\frac{Res_i}{C_i}\Big|^p\Big)^{\frac{1}{p}}; \quad TE_\infty = \max_i \Big|\frac{Res_i}{C_i}\Big|, \tag{3.9}$$

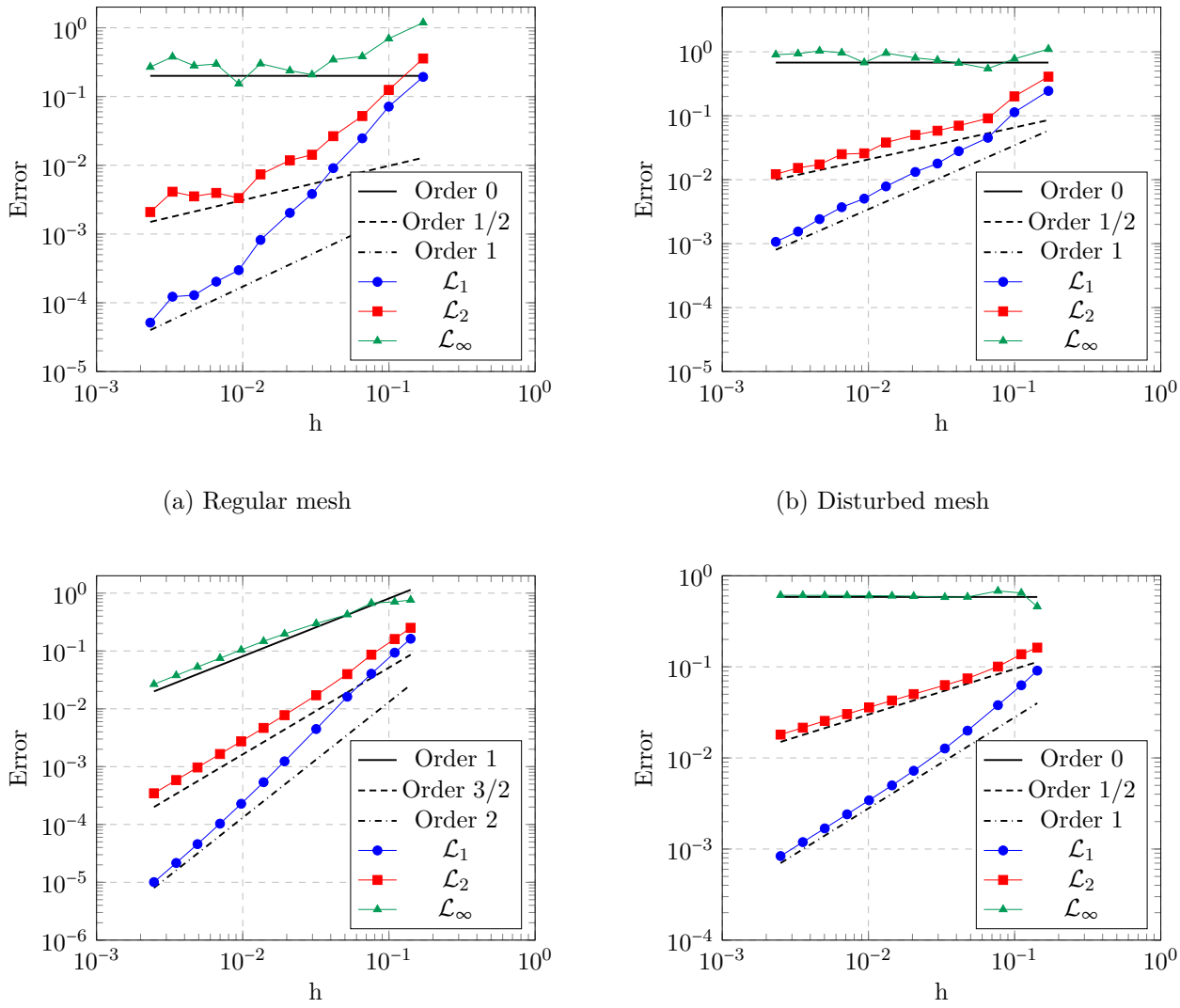
where Res_i is the RHS (2.45) of the equation computed at vertex i. Moreover, to plot the convergence results, we used again the following definition of the mesh size:

$$h_{eff} = \frac{1}{N}\sum_i |C_i|^{\frac{1}{d}}, \tag{3.10}$$

with N the number of vertices, $|C_i|$ the measure of the dual cell C_i and d the dimension of the problem. In our case, d is fixed to 2.

The truncation error allows us to easily verify the implementation of our numerical scheme. Indeed, it is enough to find the theoretical orders to validate the implementation of the code. On a regular mesh (Friedrich-Keller for example), the truncation error is equivalent to the discretization error. On an irregular mesh, the truncation error is not equivalent to the discretization error, as the truncation error is one order lower than the discretisation error on this type of mesh [48]. First, we focus on the V4 scheme without boundary reconstruction. In the event that an edge does not have Upwind elements, the scheme is reduced to a first-order one. In this section, we will examine the truncation error on the four mesh types presented in 3.4. The figure 3.10 shows the evolution of the truncation error as a function of mesh size for points inside the computational domain, while the figure 3.11 shows the results on boundary vertices. As illustrated in Figures 3.10 and 3.11, the results vary depending on the meshes considered.

For regular, perturbed and Friedrich-Keller meshes, the truncation error is inconsistent for the boundary points, meaning it does not converge with the mesh. As the numerical scheme used is of order 1 on the boundary of the domain, this result is expected. Therefore, the truncation error for the interior points is equal to $O(h^{0+1/p})$ (based on the formulation by Tarsia-Morisco and Nishikawa [87]), where p represents the norm considered. Thus, the scheme is of order 2 for points inside the domain and of order 1 for boundary points. The results obtained for the Union Jack meshes are different. Indeed, we find a truncation error equal to $O(h)$ on the boundary points, and therefore a truncation error equal to $O(h^{1+1/p})$ for the interior points. Despite a degradation in the accuracy of the numerical scheme on the boundary of the domain, the scheme behaves like a second-order scheme on all points of the mesh. The truncation error indicates that the V4 mesh without boundary reconstruction is of order 1 on all types of mesh, with the exception of Union Jack meshes, where the scheme demonstrates a second-order convergence. Reconstruction for the boundary edge 3.7 is therefore necessary for the V4 scheme to be of order 2 on all types of mesh.



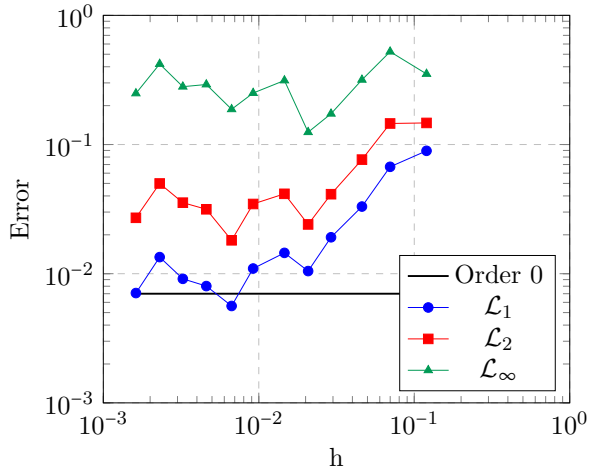
(a) Regular mesh

(b) Disturbed mesh

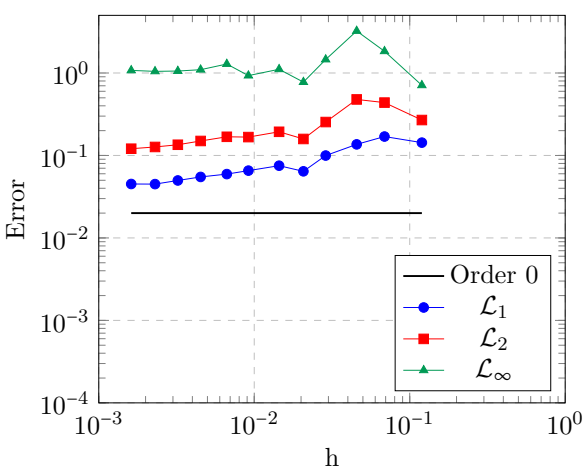
(c) Union Jack mesh

(d) Friedrich-Keller mesh

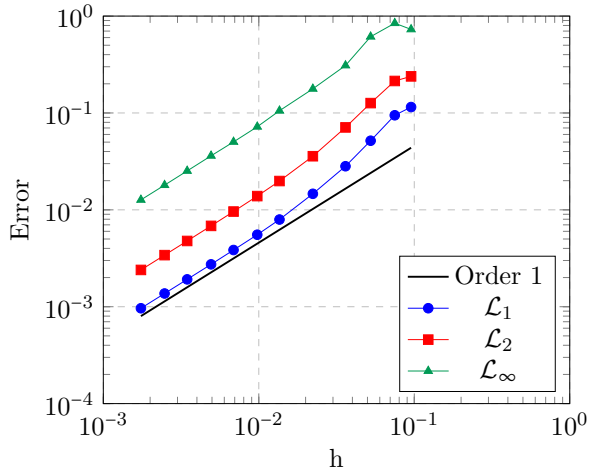
Figure 3.10: $\mathcal{L}_1$, $\mathcal{L}_2$ and $\mathcal{L}_\infty$ truncation error for interior points on regular, perturbed, Friedrich-Keller and Union Jack meshes, clockwise from top left - edge-based V4 without boundary reconstruction



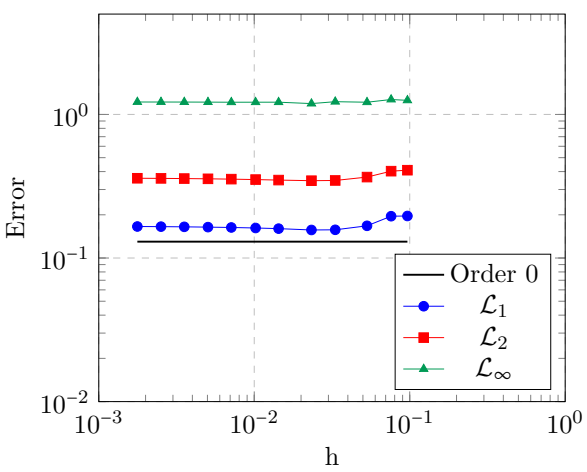
(a) Regular mesh



(b) Disturbed mesh



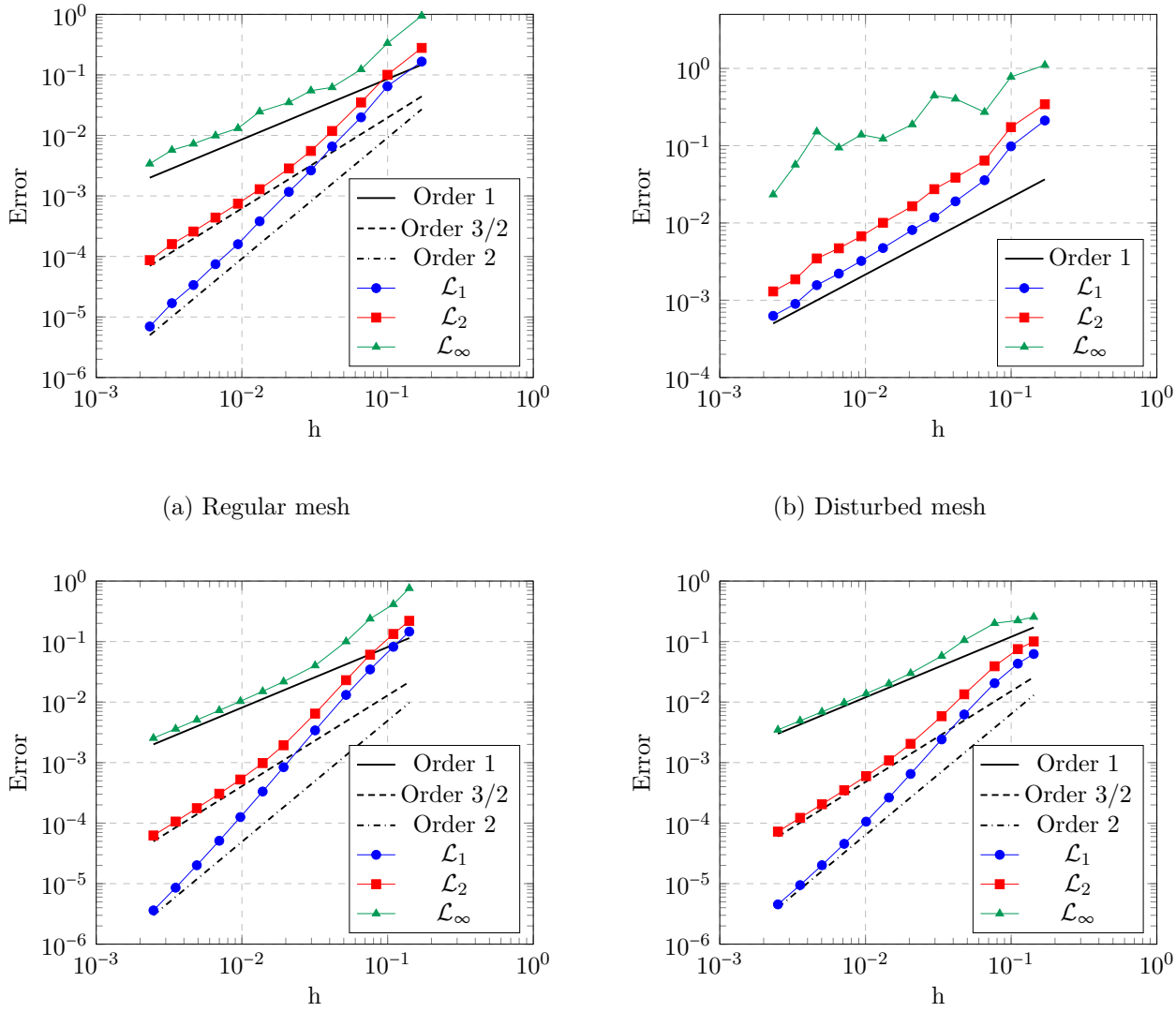
(c) Union Jack mesh



(d) Friedrich-Keller mesh

Figure 3.11: $\mathcal{L}_1$, $\mathcal{L}_2$ and $\mathcal{L}_\infty$ truncation error for boundary points on regular, perturbed, Friedrich-Keller and Union Jack meshes, clockwise from top left - edge-based V4 without boundary reconstruction

Figures 3.12 and 3.13 illustrate the plot of the truncation error against the mesh size for interior and edge points. With regard to the four types of mesh, it is important to note that the truncation error is at least of order one for interior points. The truncation error is equal to $O(h)$ for perturbed meshes. In addition, the regular, Friedrich-Keller and Union Jack meshes are meshes that can be described as regular. This means that the volume moments of order 1 are zero for almost all dual cells located in the domain. For these three types of meshes, we observe that the truncation error is equal to $O(h^{1+1/p})$ for points inside the domain. Furthermore, for the four types of mesh, the truncation error is of order 1 for boundary vertices. The truncation error analysis demonstrates that the V4 scheme, as defined, exhibits second-order convergence on any type of mesh with the correct reconstruction. These results are consistent with the literature [87].



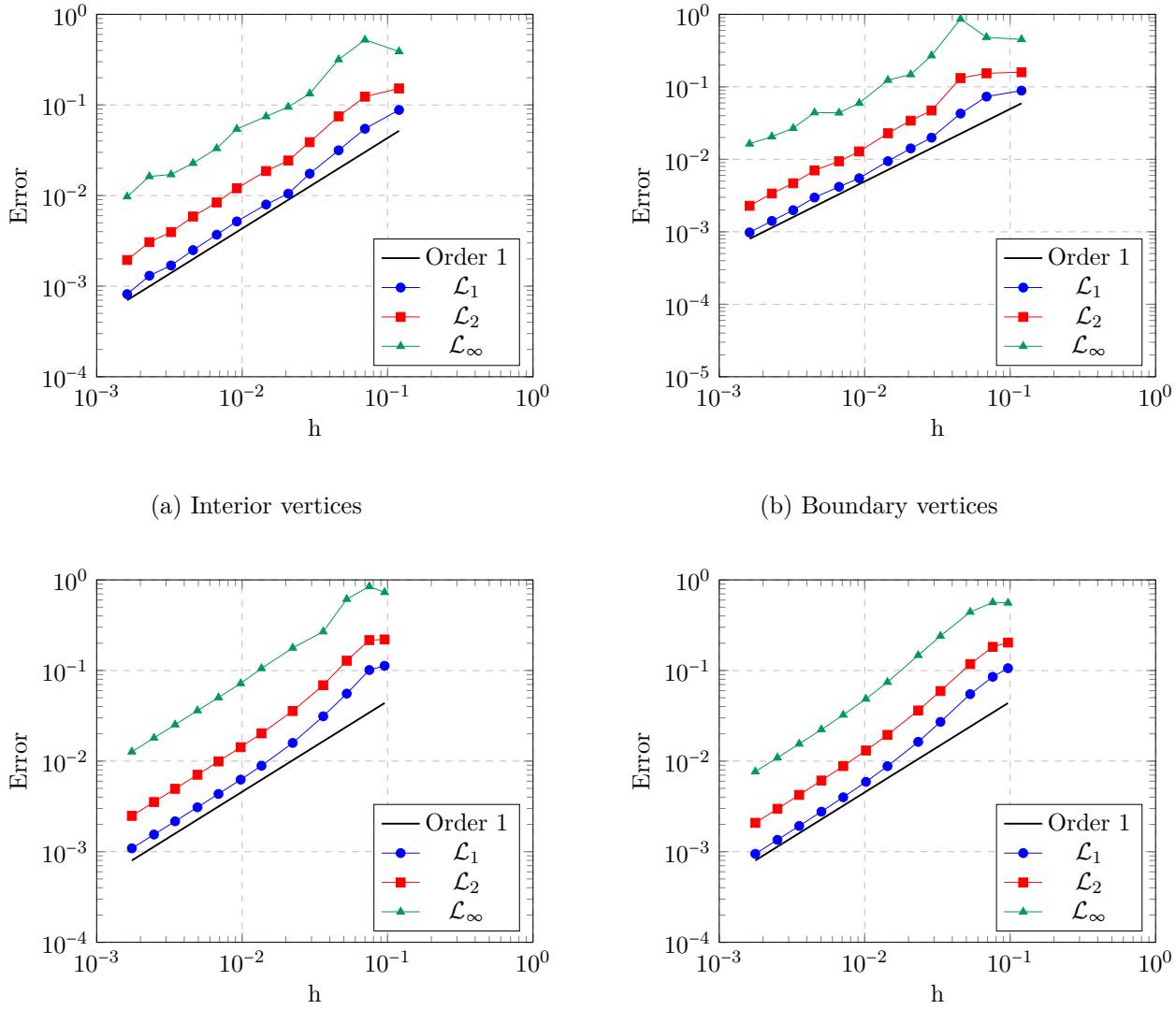
(a) Regular mesh

(b) Disturbed mesh

(c) Union Jack mesh

(d) Friedrich-Keller mesh

Figure 3.12: $\mathcal{L}_1$, $\mathcal{L}_2$ and $\mathcal{L}_\infty$ truncation error for interior points on regular, perturbed, Friedrich-Keller and Union Jack meshes, clockwise from top left - edge-base V4



(a) Interior vertices

(b) Boundary vertices

(c) Interior vertices

(d) Boundary vertices

Figure 3.13: $\mathcal{L}_1$, $\mathcal{L}_2$ and $\mathcal{L}_\infty$ truncation error for boundary points on regular, perturbed, Friedrich-Keller and Union Jack meshes, clockwise from top left - edge-based V4

The study of the truncation error has revealed that the V4 scheme, as originally described by Dervieux in [16], is not of order 2 on any mesh, at all the points of the mesh. In order to ensure order 2 on all points of the mesh, it is necessary to carry out boundary reconstruction, as described in [18] and given in 3.7. Therefore, a solution is proposed to ensure that the V4 scheme has order 2 for any mesh type with respect to truncation error. In the following section, the discretization error will be studied.

3.2.3 Discretization error

In this section, we present results regarding the discretization error. Since we are considering a numerical scheme that transports the value to the mesh nodes, we consider the following formulation for the discretization error:

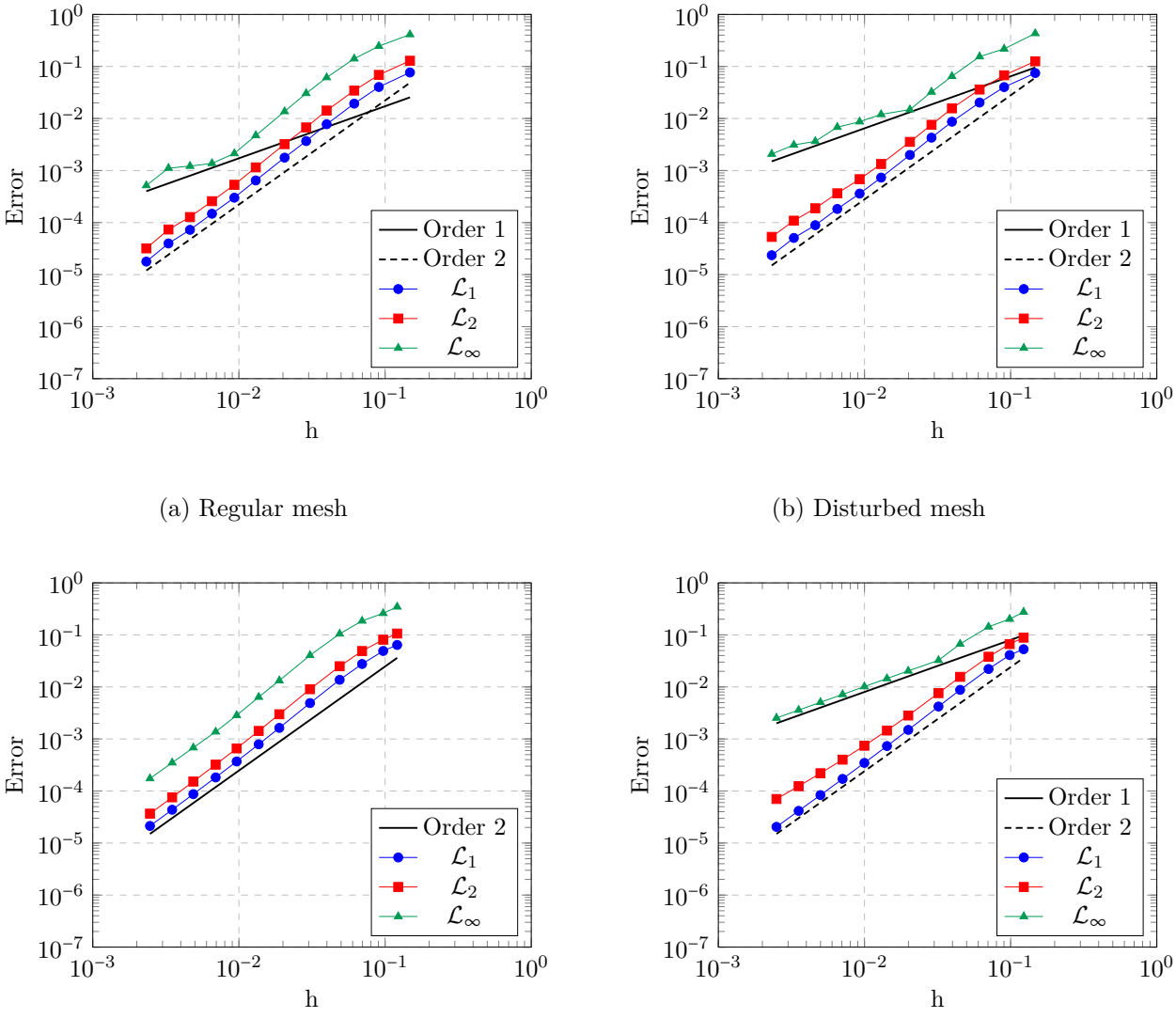
$$\|w-w^{\text{ref}}\|_{\mathcal{L}_p} = \left(\int_\Omega (w-w^{\text{ref}})^p dV\right)^{\frac{1}{p}} \approx \left(\frac{1}{N}\sum_i |w_i-w_i^{\text{ref}}|^p\right)^{\frac{1}{p}}; \quad \|w-w^{\text{ref}}\|_{\mathcal{L}_\infty} = \max_i |w_i-w_i^{\text{ref}}| \quad (3.11)$$

The nodal value, evaluated at the vertices of the mesh, is the reference value (point-value solution):

$$w_i^{\text{ref}} = w_{\text{exact}}(x_i, y_i) \quad \forall i \in \{1, ..., N_C\}$$

All the numerical results were obtained using the V4 scheme presented at the beginning of this section. The time scheme used is Shu and Osher's Rk3 scheme [64].

Results on standard meshes



(a) Regular mesh

(b) Disturbed mesh

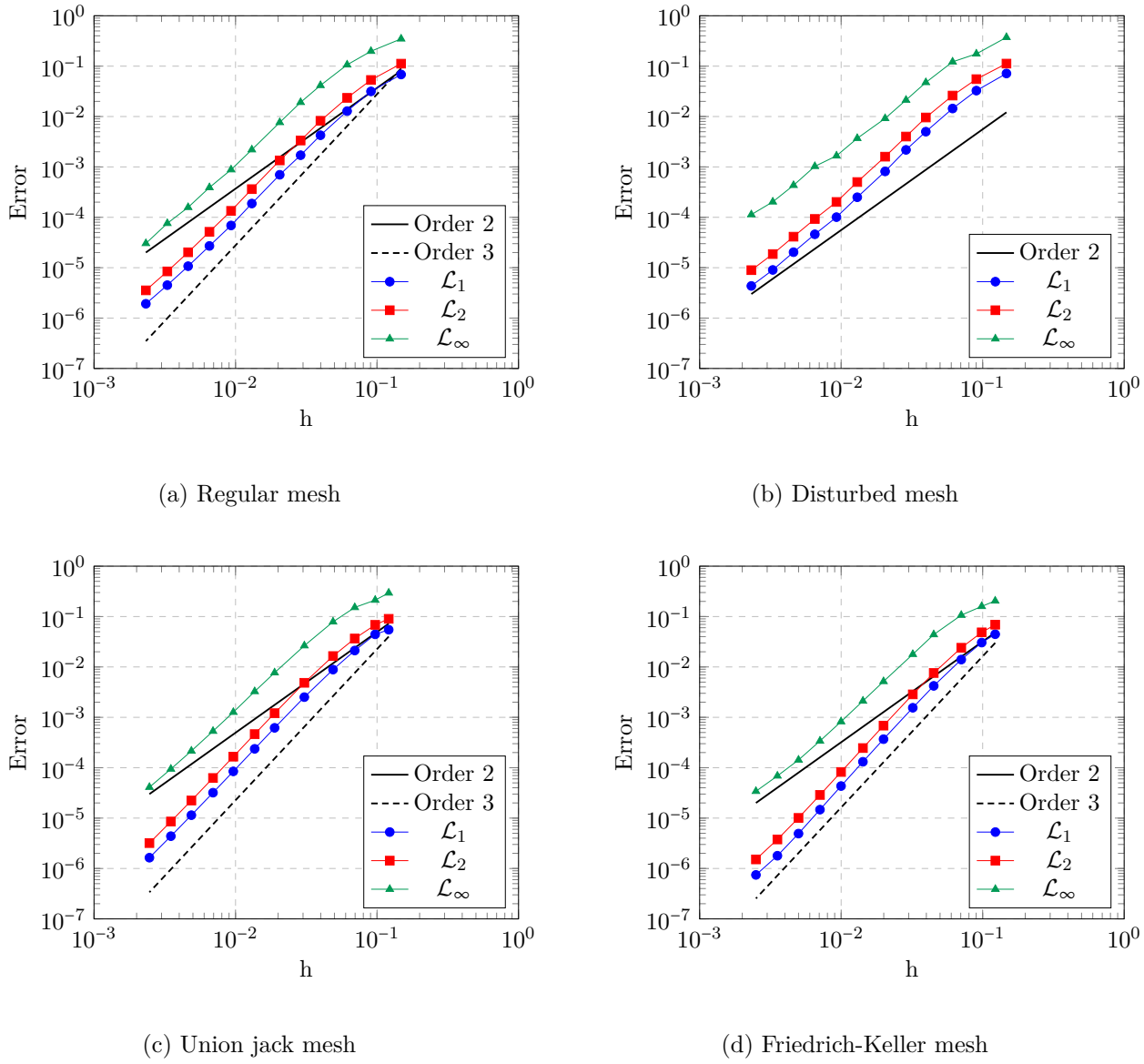
(c) Union jack mesh

(d) Friedrich-Keller mesh

Figure 3.14: $\mathcal{L}_1$, $\mathcal{L}_2$ and $\mathcal{L}_\infty$ discretization error on regular, perturbed, Friedrich-Keller and Union Jack meshes, clockwise from top left - edge-based V4 without boundary reconstruction

Our focus is on the V4 scheme without reconstruction on the boundary of the domain. Figure 3.14 represents the discretization error against the mesh size for the four types of mesh presented in 3.4. The results show that the Union Jack meshes are the only ones for which the scheme is of order 2 for all the considered norms. For the other types of meshes, the discretization error

converges to order 2 by considering the $\mathcal{L}_1$ and $\mathcal{L}_2$ norms. However, the $\mathcal{L}_\infty$ norm converges asymptotically to order 1. These results are consistent with those obtained in the study of the truncation error. Indeed, the truncation error demonstrates that the V4 scheme without boundary correction is of order 2 on all the nodes of the mesh, but only for the Union Jack meshes. For the other meshes considered, we have a scheme that is of order 1 on the boundary vertices. Therefore, the $\mathcal{L}_\infty$ norm converges to order 1. These results demonstrate that, to be of order 2 on any type of mesh, the V4 scheme must include an order 2 reconstruction for edges pointing to the boundary of the domain.



(a) Regular mesh

(b) Disturbed mesh

(c) Union jack mesh

(d) Friedrich-Keller mesh

Figure 3.15: $\mathcal{L}_1$, $\mathcal{L}_2$ and $\mathcal{L}_\infty$ discretization error on regular, perturbed, Friedrich-Keller and Union Jack meshes, clockwise from top left - edge-based V4

We now examine the results obtained for the V4 scheme with specific reconstruction for edges pointing towards the boundary of the domain. The discretization error is depicted as a function of mesh size for the four types of meshes presented in 3.4 in Figure 3.15. The results indicate that the V4 scheme is of order 2 for all the meshes considered. Specifically, the results on the perturbed meshes demonstrate convergence of order 2, consistent across all norms considered. For the remaining three types of meshes, it is observed that the error converges to order 3 when the $\mathcal{L}_1$ and $\mathcal{L}_2$ norms are considered. Furthermore, the norm of the error converges

asymptotically to order 2 if we consider the $\mathcal{L}_\infty$ norm. It can thus be concluded that the V4 scheme, as defined, is locally of order 3 on regular meshes, but that it exhibits an asymptotic convergence of order 2. It is also worth noting that the high-order reconstruction on the vertices, pointing towards the boundary of the domain, increases both the accuracy and the order of convergence of the V4 scheme. These conclusions are in accordance with the work of Nishikawa [50]. It is a known result that the V4 scheme is at least of order 2 on any type of mesh if we consider the value at the node as the reference solution.

In order to complete our study, we are now interested in the results of the V4 scheme on adapted meshes, where highly anisotropic elements can appear.

Results on adapted meshes

Here, we focus on the results obtained from the V4 scheme on adapted meshes, where highly anisotropic elements can appear. The generation of these meshes is achieved through the utilisation of an adaptation tool chain. Starting from a solution, it requires the computation of a sensor (here the solution), from which the metric constraints for the mesh generation leading to a minimum of discretization error is performed by the tool `metrix` [89] from Inria. Finally, the metric field is an input of the remesher `feflo.a` [90, 91]. A representation of an adapted mesh is given in 3.16.

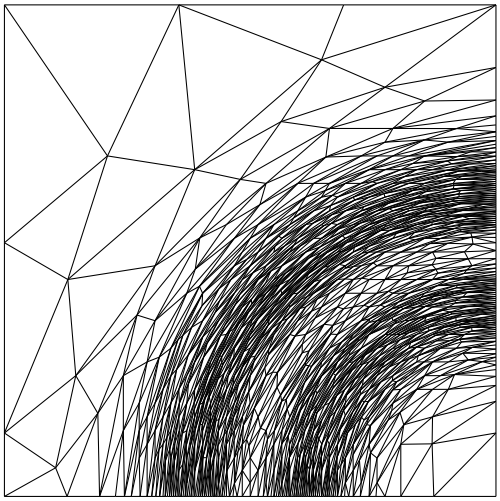


Figure 3.16: Adapted mesh for the gaussian advection test case

Figure 3.17 illustrates how the discretisation error evolves with mesh size for adapted meshes. It can be seen that the $\mathcal{L}_1$ and $\mathcal{L}_2$ norms of the discretisation error converge to order 2. On the other hand, the $\mathcal{L}_\infty$ norm of the error seems to converge to an order slightly less than 2. This loss of precision arises from the irregularity of the boundary elements. More precisely, the edges located at the domain's boundaries have an irregular size. This reduces the precision of the numerical scheme. This has been demonstrated by Nishikawa for all edge-based schemes. Methods to increase the order of convergence of the scheme at the boundary of the domain have also been proposed [36].

In this section, we have studied the scheme on a two-dimensional scalar equation. Specifically, we considered the transport equation with non-constant velocity. The results obtained show that the V4 scheme is of order 2 on all types of meshes when the reference solution considered is the value evaluated at the mesh nodes. It is important to note that specific reconstruction for boundary edges is necessary for the V4 scheme to converge to order 2 on any mesh type if the

size of the edge on the boundary remains constant. Finally, we observed that the V4 scheme maintains order 2 even on a highly anisotropic mesh. We now consider Euler's equations in order to study the V4 scheme on a system of nonlinear equations.

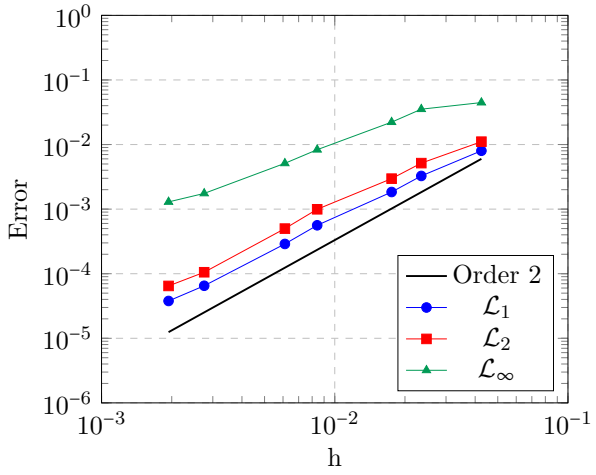


Figure 3.17: All vertices

Figure 3.18: $\mathcal{L}_1$, $\mathcal{L}_2$ and $\mathcal{L}_\infty$ discretization error on adapted mesh - edge-based V4

3.3 Numerical results : 2D Euler test case

In the previous section, we examined how the V4 scheme behaves when applied to the scalar advection equation. We demonstrated that this scheme is of order 2 for all mesh types. Additionally, we demonstrated that the V4 scheme is robust on an anisotropic, adapted mesh. We now wish to study the V4 scheme for the Euler equations. These equations are a system of conservation laws describing flows without momentum exchange by viscosity or energy exchange by thermal conduction. The Euler equations are more complex than the scalar advection equation as they are nonlinear. Moreover, we will study the V4 scheme using test cases of increasing complexity. We will start with convection of an isentropic vortex. This test does not involve the boundary treatment. This will enable us to validate the behaviour of the V4 scheme for cells within the domain. Next, we will consider the Ringleb flow case. This test case will enable us to study the behaviour of the V4 scheme for boundary vertices.

3.3.1 The convection of a vortex

Test case description

The vortex advection test case (CoVo) represents the advection of a compressible, isentropic vortex. This test case is taken from the high-order CFD method workshop [92] and the approach is largely inspired by J. Vanharen's thesis [93]. This test case has an analytical solution for the unsteady compressible Euler equations. Indeed, the isentropic vortex remains unchanged during its advection at the initial velocity $\mathbf{u}_\infty$. The CFD solver is expected to provide the initial solution advected without dissipation. It is therefore possible to carry out an error analysis by comparing the exact solution with the solution obtained from the CFD solver. In this way we can measure the accuracy and order of convergence of our scheme. This test case is unsteady, and we have decided to advect the vortex over a short distance. This means that there is no interaction between the vortex and the boundary of the domain. Consequently, this case enables us to validate the behaviour of the V4 scheme at interior points. In order to validate the boundary

conditions, it will be necessary to use another test case. Moreover, the advection of an isentropic vortex is an unsteady case. As a consequence, it is important to ensure that the predominant error is that associated with the discretization of the convective flux. The objective is therefore to minimise the error associated with the temporal scheme. For this test case, the Runge-Kutta 4 scheme is employed.

A 2D isentropic vortex can be generated using the following function.

$$\psi(x,y) = \Gamma \exp(-\frac{(x-x_c)^2 + (y-y_c)^2}{2R_c^2}), \tag{3.12}$$

where Γ is the circulation, R_c is the radius and (x_c, y_c) are the coordinates of the vertex center. From this function, we can define the primitive variables of initial flow solution:

$$\begin{aligned}
\rho(x,y) &= \rho_\infty \Big(\frac{T(x,y)}{T_\infty}\Big)^{\frac{1}{\gamma-1}}, \\
u(x,y) &= u_\infty + \frac{\partial \psi}{\partial y}, \\
v(x,y) &= v_\infty - \frac{\partial \psi}{\partial x}, \\
T(x,y) &= T_\infty - \frac{1}{2C_p}\Gamma^2 \exp\Big(-\frac{(x-x_c)^2+(y-y_c)^2}{Rc^2}\Big), \\
P(x,y) &= \rho(x,y) R_{gaz} T(x,y).
\end{aligned} \tag{3.13}$$

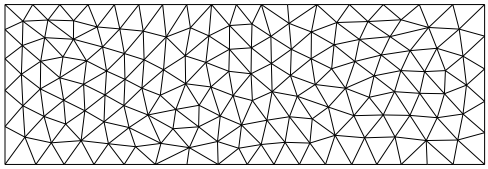
From the primitive variables, one can define the conservative variables:

$$\begin{aligned}
\rho(x,y) &= \rho_\infty \Big(\frac{T(x,y)}{T_\infty}\Big)^{\frac{1}{\gamma-1}}, \\
\rho u(x,y) &= \rho(x,y) u(x,y), \\
\rho v(x,y) &= \rho(x,y) v(x,y), \\
\rho E(x,y) &= \frac{P(x,y)}{\gamma-1} + \frac{1}{2}\rho(x,y)(u(x,y)^2 + v(x,y)^2).
\end{aligned} \tag{3.14}$$

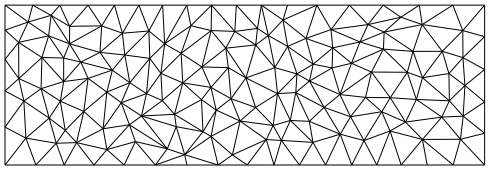
The various numerical parameters are summarized in the table 3.2.

Convected vortex parameters	
Density	$\rho_\infty = 1.16083\ \mathrm{kgm^{-3}}$
VelocityX	$u_\infty = 170.0\ \mathrm{ms^{-1}}$
VelocityY	$v_\infty = 0.0\ \mathrm{ms^{-1}}$
Pression	$p_\infty = 10290\ \mathrm{Pa}$
Circulation	$\Gamma = 34.72794\ \mathrm{m^2 s^{-1}}$
Radius	$R_c = 0.005\ \mathrm{m}$
Specific heat ratio	$\gamma = 1.4$
Gaz constant	$R_{gas} = 287.15\ \mathrm{J\,kg\,K}$
Boundary conditions	Farfield

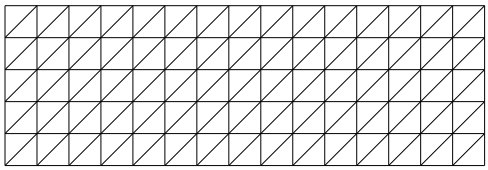
Table 3.2: Convected vortex (CoVo) parameters



(a) Regular mesh



(b) Disturbed mesh



(c) Friedrich-Keller mesh

Figure 3.19: Regular mesh (top), disturbed mesh (centre) and Friedrich-Keller (bottom) for the CoVo test case.

The domain is a rectangle of size $[0.3\text{m} \times 0.1\text{m}]$. The vortex is initialized around the point of coordinates (x_c = 0.05m, y_c = 0.05m). In this test case, we consider three types of triangular mesh: regular, disturbed and Friedrich-Keller. A hierarchy is defined for each type of mesh in order to calculate the order of convergence of our numerical schemes. This hierarchy is given in table 3.3.

	Regular	**Disturbed**	**Friedrich-Keller**
1	2919	2919	1281
2	5412	5412	4961
3	15655	15655	19521
4	66307	66307	77441
5	264171	124665	308481

Table 3.3: Number of vertices for each mesh of isentropic vortex convection test cases

Figure 3.19 shows a representation of the three different types of meshes. We have first regular meshes, for which we seek to minimise the first-order volume moments. By randomly perturbing the regular meshes, we obtain the perturbed meshes. It should be noted that these meshes are irregular and not designed to represent the flow. Consequently, they offer a valuable opportunity to assess the performance of numerical schemes. Finally, we defined Friedrich-Keller meshes. As with the advection equation, these meshes are used to verify the behaviour of the V4 scheme on a mesh without any irregularities. The vortex is advected over a short distance (1.7×10^{-2}m). The initial vortex is illustrated in Figure 3.20, where the x-component of the momentum is plotted.

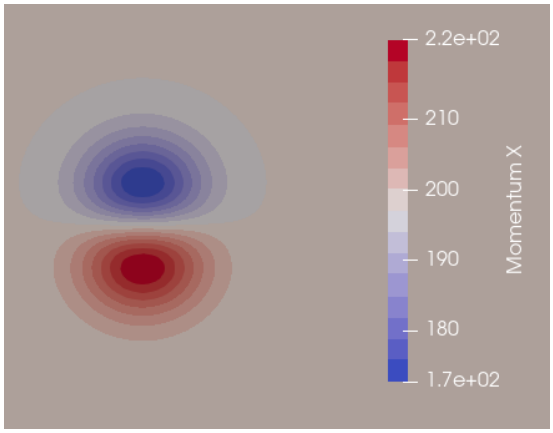
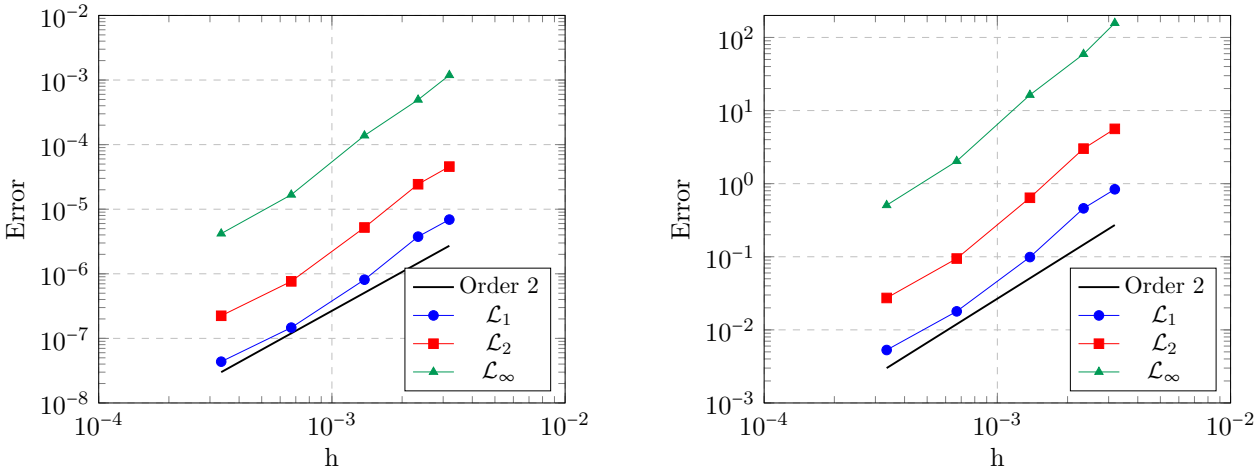


Figure 3.20: x-component of the momentum of the isentropic vortex

Numerical results

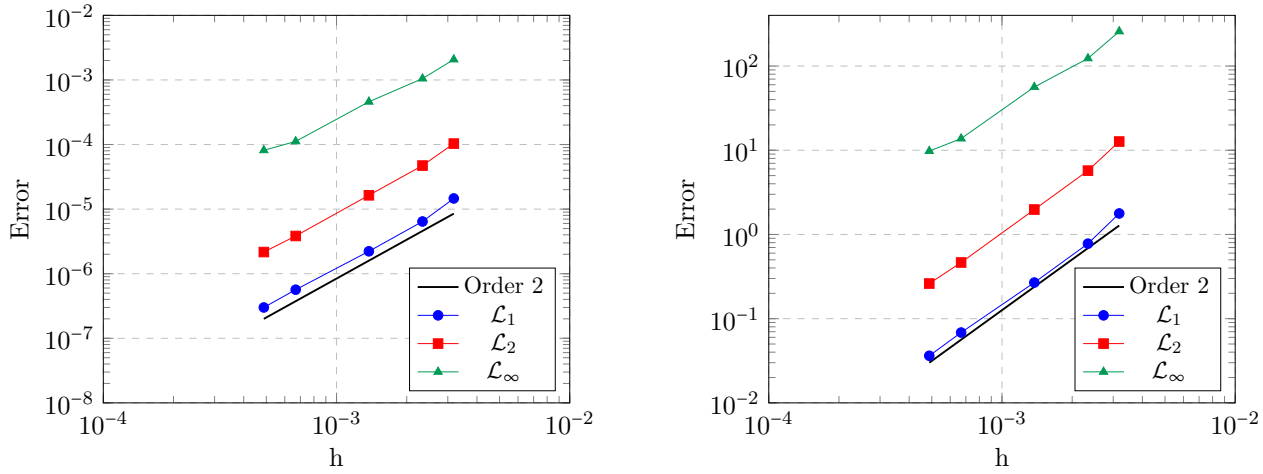
Figure 3.21 shows the norm of the error as a function of mesh size for regular meshes. The error is shown for the density and the pressure, with the reference solution evaluated at the mesh nodes. On the one hand, we recover the equivalence of all norms, that validate the numerical experiment in use. On the other hand, all the norms converge asymptotically towards order 2 for density, which shows that the V4 scheme is of order 2 for this type of mesh. Convergence for pressure is also at order 2. Figure 3.22 shows the results on disturbed meshes. Although the error level is higher than for regular meshes, here we find convergence of exactly order 2, and this is true for all the norms considered. Therefore, the V4 scheme is of order 2 for disturbed meshes. Figure 3.23 shows the evolution of the norms of the error as a function of mesh size for Friedrich-Keller meshes. The reference solution for calculating the error is always evaluated at the mesh nodes. We also observe that, for Friedrich-Keller meshes, all the considered norms are equivalent and converge asymptotically to order 2. Finally, we can conclude that the V4 scheme is of order 2 any mesh type if we consider the cells located within the domain.



(a) Density

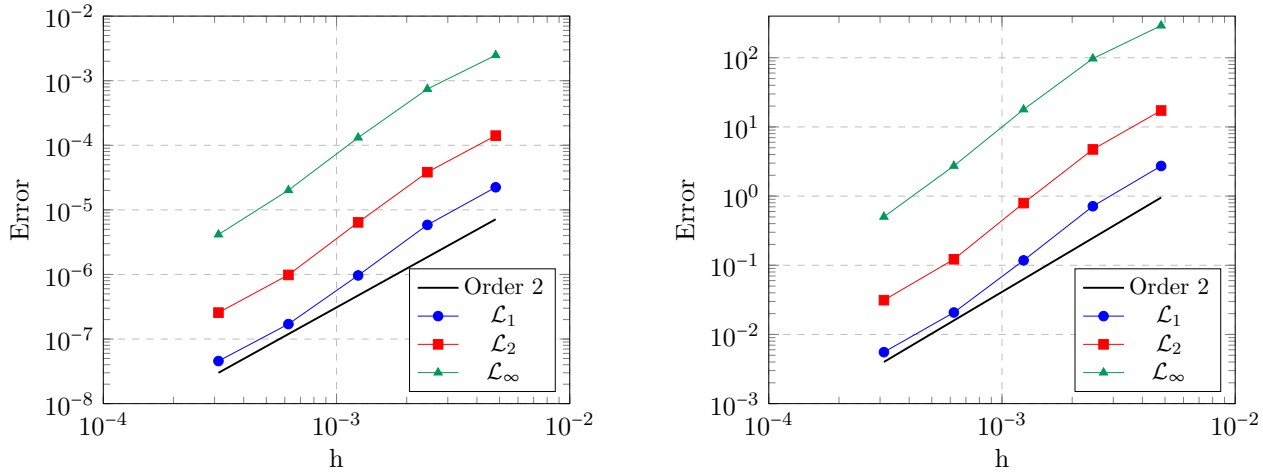
(b) Pressure

Figure 3.21: $\mathcal{L}_1$, $\mathcal{L}_2$ and $\mathcal{L}_\infty$ error for density (left) and pressure (right) on regular mesh - edge-based V4



(a) Density (b) Pressure

Figure 3.22: $\mathcal{L}_1$, $\mathcal{L}_2$ and $\mathcal{L}_\infty$ error for density (left) and pressure (right) on disturbed mesh - egge-based V4



(a) Density (b) Pressure

Figure 3.23: $\mathcal{L}_1$, $\mathcal{L}_2$ and $\mathcal{L}_\infty$ error for density (left) and pressure (right) on Friedrich-Keller mesh - edge-based V4

The vortex advection test case allowed us to show that the V4 scheme converges to order 2 on any type of mesh when the initial solution and the reference solution are evaluated at the mesh nodes. This complements the results previously obtained with the advection equation. In order to validate the behaviour of the V4 scheme at the domain boundary, the Ringleb flow case is addressed.

3.3.2 The Ringleb flow

Test case description

The Ringleb [94] flow case is a widely used case for the validation of high-order schemes. It is a compressible, non-viscous and non-heat-conducting test case for the Euler equations. By

hodographic transformation of the Euler equations [95], an exact solution of the problem is known, which is a stationary solution. Starting from the unknown, here the speed of sound c, the solution of the flow for any spatial coordinates (x, y) is determined by solving the non-linear equation (3.15) [96]:

$$\left(x + \frac{J}{2}\right)^2 + y^2 = \frac{1}{4\rho^2 V^4}, \tag{3.15}$$

where the density ρ, the norm of the velocity vector V and J are defined such that:

$$\rho = c^{\frac{2}{\gamma-1}}, \quad V = \sqrt{\frac{2(1-c^2)}{\gamma-1}}, \quad J = \frac{1}{c} + \frac{1}{3c^3} + \frac{1}{5c^5} - \frac{1}{2}\log\left(\frac{1+c}{1-c}\right). \tag{3.16}$$

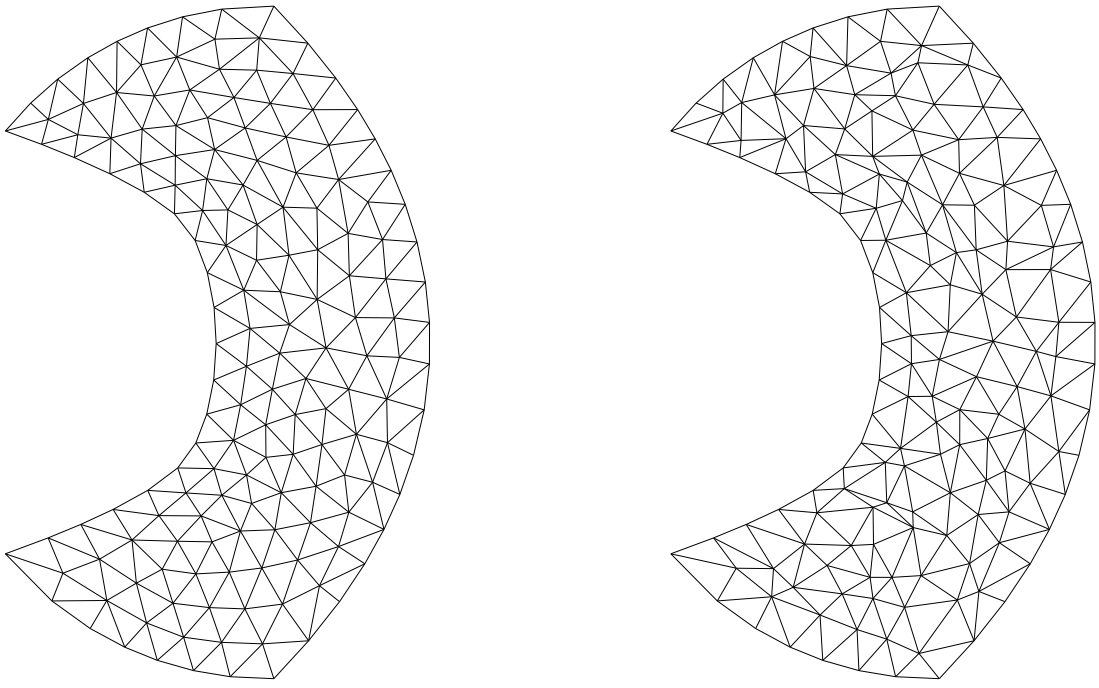
In addition, the velocity vector and the pressure field are given by the following relationships:

$$\mathbf{u} = V \begin{pmatrix} -\mathrm{sgn}(y)\cos(\theta) \\ -\sin(\theta) \end{pmatrix}, \quad p = \frac{1}{\gamma} c^{\frac{2\gamma}{\gamma-1}}, \tag{3.17}$$

where $\mathrm{sgn}(\cdot)$ is the sign function and $\sin(\theta) = \psi V$ and

$$\psi = \sqrt{\frac{1}{2V^2} + \rho\left(x + \frac{J}{2}\right)}. \tag{3.18}$$

The exact solution makes it a very suitable test case for convergence studies. The Ringleb case is generally solved on a curvilinear domain representing a channel around an obstacle. Under these conditions, the flow is transonic, with a large supersonic region near the nose of the obstacle [97]. Figure 3.25-(a) shows the Mach number for the Ringleb case. The flow is clearly supersonic with a Mach number between 0.5 and 2. However, the flow does not present any shock. Therefore, it is not necessary to use a slope limiter to perform this calculation. The results presented in this section were therefore obtained without a limiter. The mesh geometry of the problem is shown in figure 3.24. Moreover, Algorithm 1 which is given in Appendix A, describes how to compute density, velocity and pressure for a point with coordinates (x, y).



(a) Regular mesh (b) Disturbed mesh

Figure 3.24: Regular mesh (left) and disturbed mesh (rigth) for the Ringleb test case

For this test case, we also define two types of meshes: regular meshes and perturbed meshes. As with the vortex advection test case, the aim here is to show how the V4 scheme behaves on all types of meshes. A mesh hierarchy is defined for each type of mesh in order to carry out a convergence study. This hierarchy is given in table 3.4.

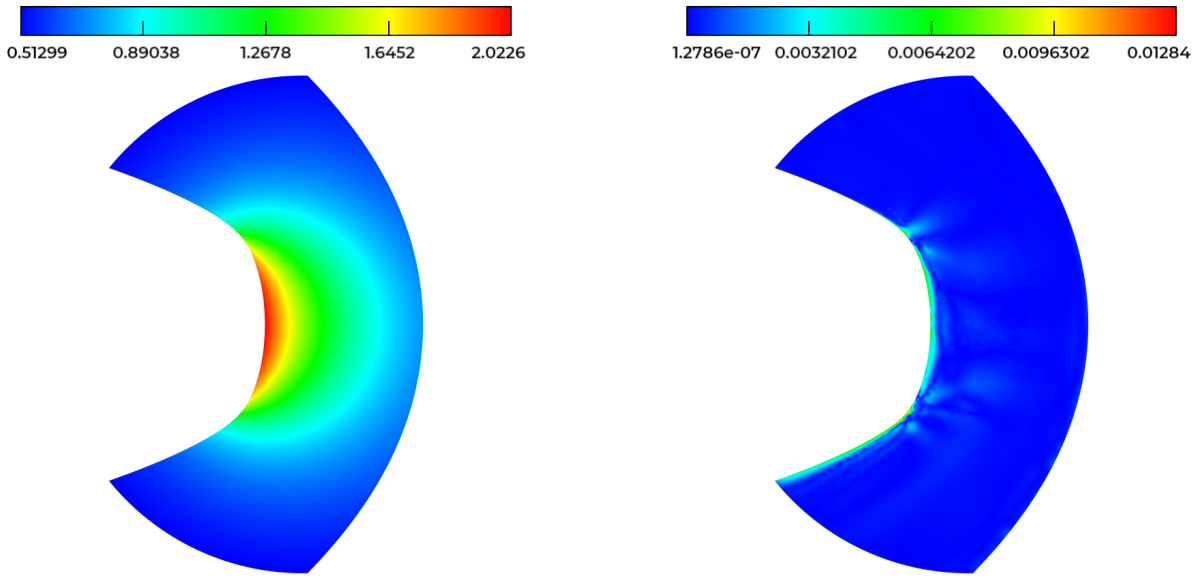
	Regular	**Disturbed**
1	68	68
2	243	243
3	879	879
4	3390	3390
5	13068	13068
6	41444	41444

Table 3.4: Number of vertices for each mesh of ringleb test cases

Moreover, the numerical parameters are given in the table 3.5.

Ringleb flow parameters	
Specific heat ratio	$\gamma = 1.4$
Gaz constant	$R_{gas} = 287.15\ \mathrm{J\,kg\,K}$
Boundary conditions	Farfield

Table 3.5: Ringleb flow parameters



(a) Mach number Ringleb test case

(b) Error on Mach number Ringleb test case

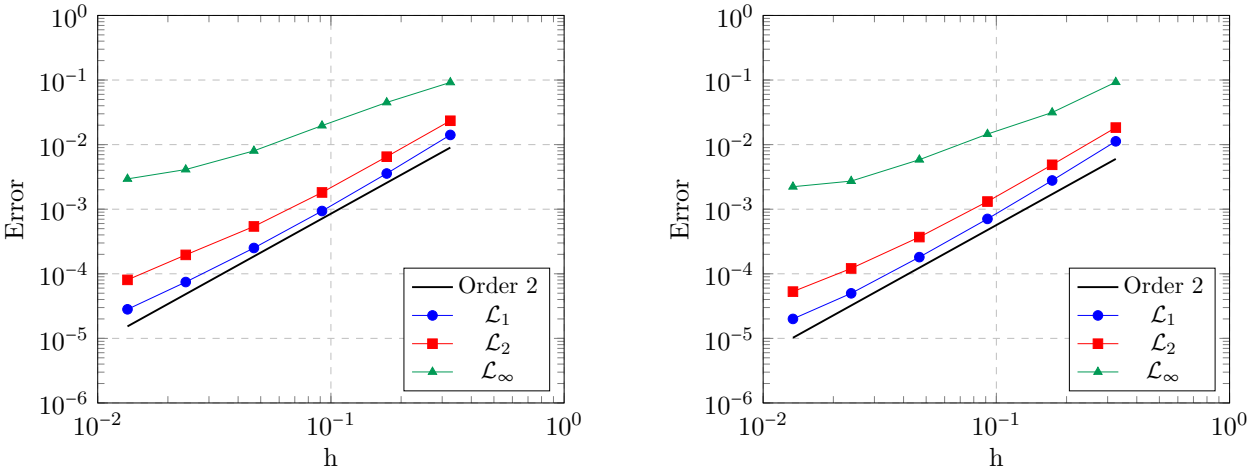
Figure 3.25: Mach number (left) and error on Mach number (rigth) for the Ringleb test case

The test case is initialised using the exact solution. Specifically, the speed of sound c, is determined by solving equation (3.15), for which a dichotomy algorithm is used in this work. For points where the x-coordinate is greater than -1.2, the speed of sound is to be sought in the interval $[0, 1]$. For all other points, and in order to ensure the correct solution is converged upon by the dichotomy algorithm, the speed of sound is to be sought in the interval $[0.8, 1]$. Once the speed of sound has been established, it is straightforward to compute all the conservative variables of the problem. Special care is taken in handling boundary conditions. The exact solution

is imposed on the boundary of the domain. To do this, a farfield condition is used. It consists of a Riemann solver for computing the flux between the exact solution and the solution at vertex i on the boundary. Since the exact solution depends on spatial coordinates, the reference point used must be chosen with care.

Numerical results

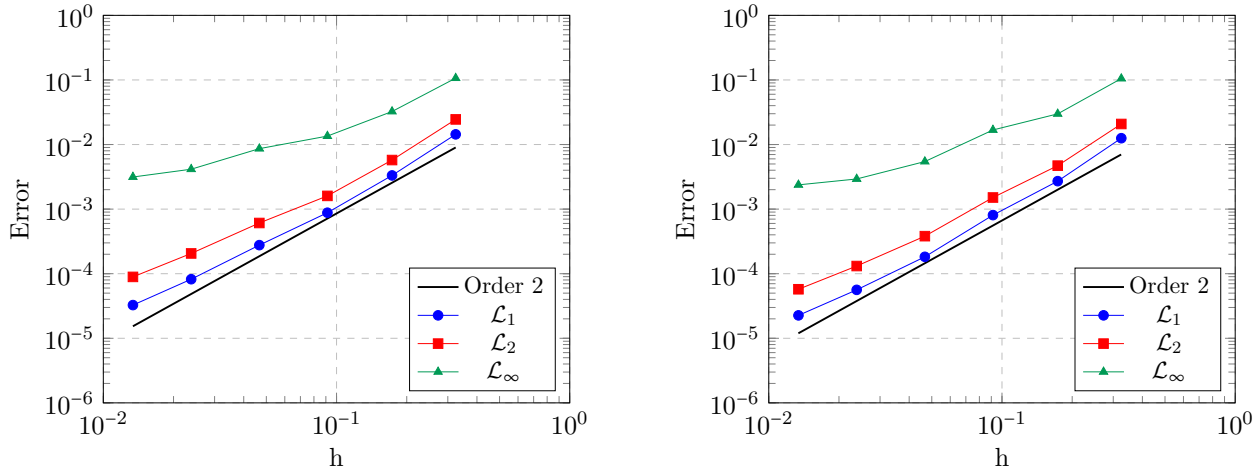
Firstly, we present the results for the V4 scheme as originally described in the literature [37]. There are two possible scenarios. If edge ij has an upwind element, then the high-order reconstruction will be used. Otherwise, the V4 scheme is degraded and we return to order 1. Figure 3.26 shows the evolution of the norm of the error as a function of mesh size for regular meshes and the Figure 3.27 shows the evolution of the norm of the error as a function of mesh size for disturbed meshes. The solution evaluated at the mesh nodes is used as the initial and reference solution in the error calculation. Errors are calculated for the density and pressure fields. The results demonstrate a loss of precision for the defined numerical scheme. Indeed, we observe that the set of norms is no longer equivalent. More precisely, the $\mathcal{L}_\infty$ error no longer converges to order 2 for either type of mesh. This is due to the fact that the numerical scheme for the vertices on boundary is only of order 1. Therefore, the numerical scheme defined is no longer of order 2 when considering all the mesh vertices.



(a) Density

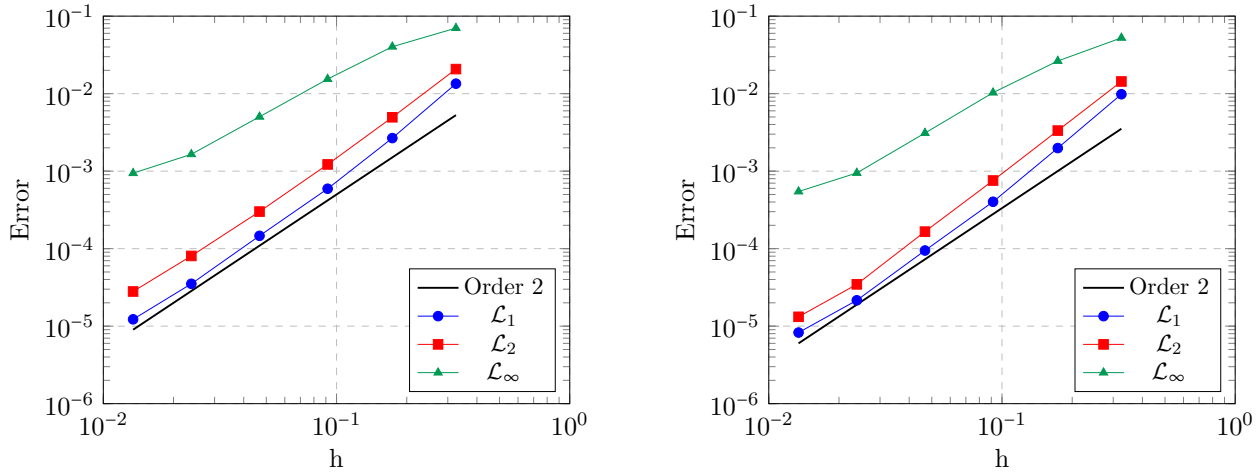
(b) Pressure

Figure 3.26: $\mathcal{L}_1$, $\mathcal{L}_2$ and $\mathcal{L}_\infty$ error for density (left) and pressure (right) on regular mesh with no boundary reconstruction - edge-based V4



(a) Density (b) Pressure

Figure 3.27: $\mathcal{L}_1$, $\mathcal{L}_2$ and $\mathcal{L}_\infty$ error for density (left) and pressure (right) on disturbed mesh with no boundary reconstruction - edge-based V4

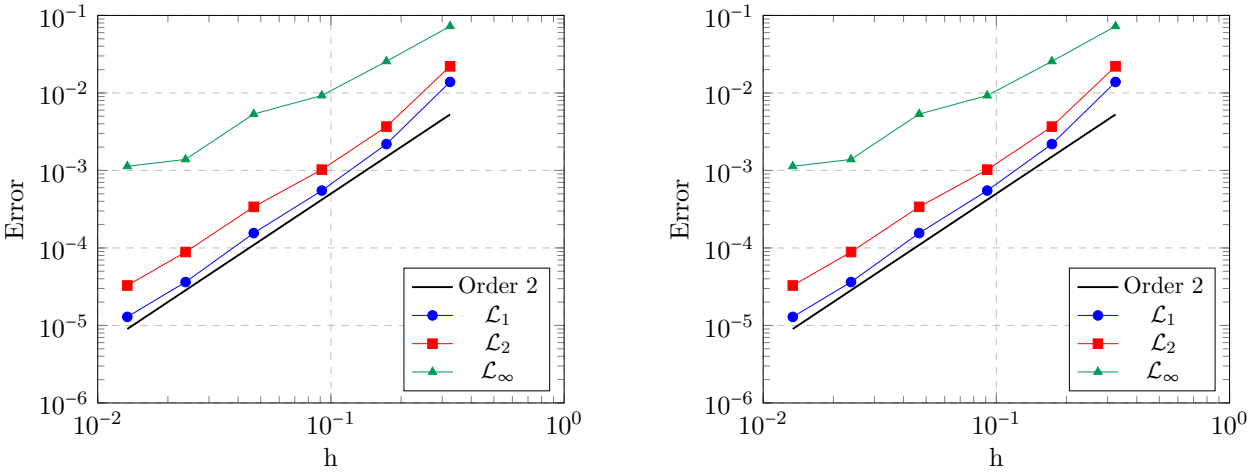


(a) Density (b) Pressure

Figure 3.28: $\mathcal{L}_1$, $\mathcal{L}_2$ and $\mathcal{L}_\infty$ error for density (left) and pressure (right) on regular mesh with boundary reconstruction - edge-based V4

Figure 3.28 shows the evolution of the $\mathcal{L}_2$ norm of the error as a function of mesh size. To obtain these results, the full V4 scheme, including modification for boundaries, is used. So, the reconstruction presented in 3.7 is used for edges that do not have an upwind element. The calculation was initialised with the exact solution evaluated at the mesh nodes. The figure on the left shows the results for density, while the figure on the right shows the results for pressure. The reference solution is evaluated at the vertices of the mesh. The results demonstrate second-order convergence for this test on regular mesh. Moreover, Figure 3.29 shows the same results for disturbed meshes using the same methodology. The conclusion remains the same, as we also have second order convergence on this type of mesh. If we consider the $\mathcal{L}_1$ and $\mathcal{L}_2$ norms, we see that the error converges globally to order 2. The $\mathcal{L}_\infty$ error reveals convergence, globally at order 2, even though the order of convergence between two successive meshes is not necessarily two. This phenomenon can be attributed to the absence of a hierarchical structure among the

distinct meshes under consideration in the convergence study. This results aligns with the earlier results, confirming that the V4 scheme, as outlined in the literature, is second-order for all types of meshes when the value at the nodes is used as the reference solution for calculating the error. Finally, when we compare the results obtained with and without boundary reconstruction, we see that reconstruction reduces the discretisation error for all error norms considered.

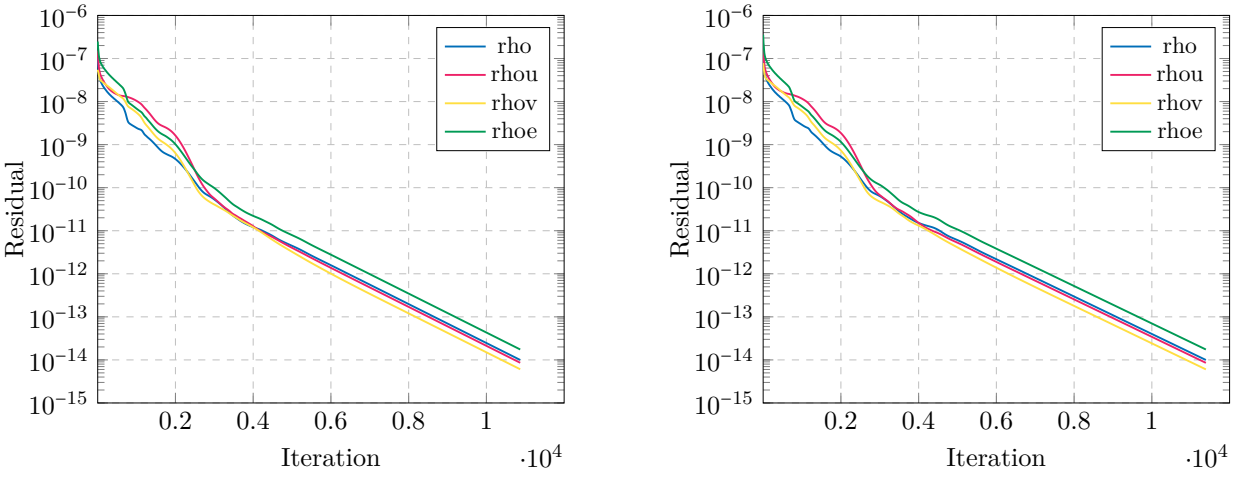


(a) Density

(b) Pressure

Figure 3.29: $\mathcal{L}_1$, $\mathcal{L}_2$ and $\mathcal{L}_\infty$ error for density (left) and pressure (right) on disturbed mesh with boundary reconstruction - edge-based V4

Furthermore, these results demonstrate the interest of the reconstruction presented in 3.7 for edges directed towards the border of the domain. This reconstruction allows us to achieve an overall convergence order of 2 on all types of meshes. Figure 3.30 illustrates the convergence of the residuals for the finest mesh. These results show us that the residuals are converged at machine zero. Moreover, to achieve this level of convergence, perturbed meshes require more iterations than regular meshes.



(a) Regular mesh

(b) Disturbed mesh

Figure 3.30: Convergence of residuals on regular mesh (left) and disturbed mesh (right) - edge-based V4

3.4 Conclusion

In this chapter, we have provided a comprehensive review of the V4 scheme. This is an edge-based numerical scheme for which the unknowns of the problem are the values at the mesh nodes. It is based on a MUSCL reconstruction for which the gradient is calculated in the direction of the mesh edges. This gradient is a linear combination of a finite element gradient calculated on the primal elements of the mesh and a centred gradient for each mesh edge. We have studied the V4 scheme on cases of increasing complexity. First, we considered the linear advection test case. For this test, we defined four types of mesh (regular, perturbed, Friedrich-Keller, and Union-Jack), and conducted convergence studies for both the truncation and discretisation errors for each mesh type. The results showed that the V4 scheme is at least of order 2 for all mesh types, provided that a specific MUSCL reconstruction is used for edges without an upwind element. For the particular case of the Union-Jack mesh, the V4 scheme is of order 3. We also defined a hierarchy of adapted, anisotropic meshes in order to study the behaviour of the V4 scheme on this type of mesh. The results demonstrate the robustness of the V4 scheme on this type of mesh, even when second-order accuracy is not preserved in terms of the $\mathcal{L}_\infty$ error. Secondly, we studied the V4 scheme for the Euler equations. The advection case of an isentropic vortex allowed us to examine the behavior of the V4 scheme at the current point, since there is no interaction between the vortex and the domain boundary in our case. For this test, we defined three types of meshes (regular, perturbed, and Friedrich-Keller), and conducted convergence studies for the discretization error for each of them. The results also show that the V4 scheme is of order 2 regardless of the mesh type. Finally, we treated the Ringleb test case. This test case enabled us to include the treatment of boundary conditions in the analysis of the V4 scheme. We defined two types of mesh: regular and perturbed meshes. A study of the convergence of the discretization error allowed us to arrive at the same conclusion as for the previous test cases, namely that the V4 scheme, in its edge-based construction, is of order 2 on any type of mesh. In the next chapter, we will detail the construction of a vertex-centred finite volume scheme based on the V4 scheme.

Of course, the work presented in this chapter is devoted to a better understanding of the V4 scheme. However, there is still a kind of bias in the definition of the scheme since it always starts from nodal quantities before switching to integral formulation. This is one of the reasons why the next chapter is dedicated to the finite volume V4 scheme.

Chapter 4

Finite volume V4 scheme

Contents

4.1 Initial approach to deconvolution 84
4.2 A more compact approach to deconvolution 86
4.2.1 Introduction of the method 86
4.2.2 Numerical results 91
4.3 Extension to deconvolution 94
4.3.1 Presentation of the method 94
4.3.2 Numerical results 96
4.4 Application of the deconvolution 99
4.4.1 MUSCL reconstruction 100
4.4.2 Boundary state reconstruction 100
4.5 Numerical results : 2D scalar transport test case 101
4.5.1 Results on standard meshes 101
4.5.2 Results on adapted meshes 102
4.6 Error analysis 103
4.7 Extension to Euler equation 109
4.7.1 Velocity 110
4.7.2 Pressure 111
4.7.3 Temperature 112
4.8 Numerical results : 2D Euler test case 113
4.8.1 Advection of an isentropic vortex 113
4.8.2 Ringleb 115
4.9 Conclusion 118

Some schemes from the edge-based formalism, such as the V4 scheme, for example, are robust on highly anisotropic meshes. This makes them a popular approach for mesh adaptation. However, this approach relies on the discretisation (2.30) of the equations, meaning it seems impossible to construct an edge-based scheme of order higher than 3 with a single numerical flux per edge. Conversely, the finite volume method, in theory, allows one to construct a scheme of any desired order. In a finite volume formalism, the quadrature formula can be augmented for the calculation of convective flux, for example. This is particularly interesting for defining the "weak" formulation of boundary conditions using integrals[1]. Furthermore, finite volume schemes are conservative by construction. These advantages encourage us to use vertex-centred finite volume schemes. Recent work by F. Stezwein has made it possible to develop such a scheme

[1]Boundary conditions can be imposed through interface values or flux. We call weak formulation an integral one.

for incompressible Euler and Navier-Stokes equations[29, 28]. However, we currently find that there is no vertex-centred finite volume scheme for compressible Euler equations based on the V4 scheme. Our objective is therefore to construct a vertex-centred, second-order, finite-volume scheme with respect to the averaged value per dual cell $\bar{\mathbf{W}}_i$. This study is motivated by the anisotropic mesh adaptation, so robustness on highly anisotropic meshes is a requirement that we retain. As shown in the previous chapter, the V4 scheme is highly robust on such meshes. This is why we have chosen to construct the finite volume scheme from the edge-based V4 scheme.

One of the first difficulties arises from the notion of averaged value per dual cell. Indeed, we want to be able to calculate, with a given degree of accuracy, the averaged value per dual cell of a function f. This is necessary, for example, for initialising a test case or calculating the error. The averaged value of a function f over a volume C is defined by:

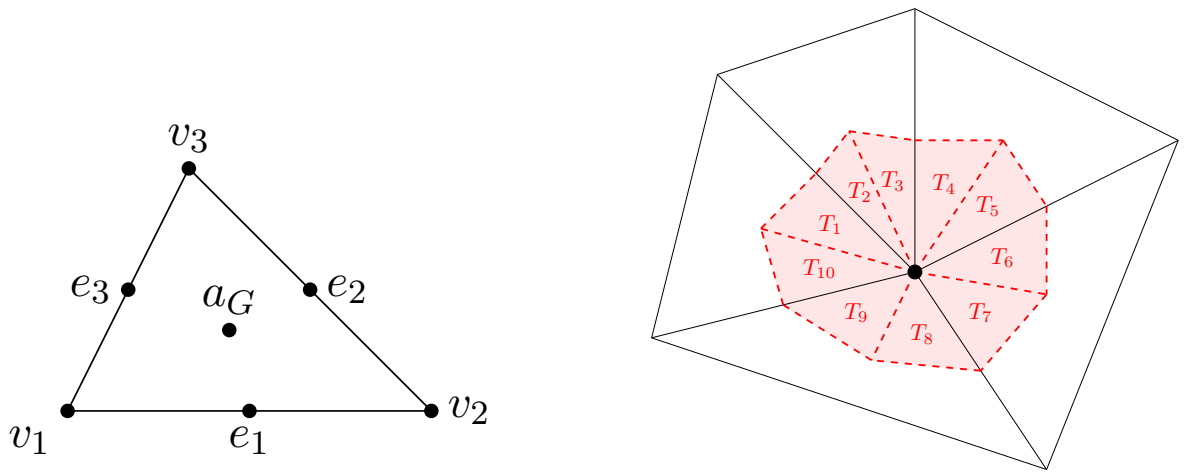
$$\bar{f}_i = \frac{1}{|C_i|}\int_{C_i} f d\Omega.$$

Thus, for each dual cell, it is necessary to be able to integrate any function f, especially using a Gauss quadrature rule. However, the median-cell approach for constructing dual volumes leads to polygonal (2D) and polyhedral (3D) cells of any shape. Thus, there is no Gaussian quadrature formula for these configurations, and it is therefore necessary to adopt another strategy. Here, we choose to divide our dual volume into a set of triangles, as shown in Figure 4.1-(b). This allows us to use a Gaussian quadrature method on each triangle and deduce the averaged value for the dual volume.

$$\bar{f}_i = \frac{1}{|C_i|}\sum_{T_j \epsilon \nu_{C_i}} \int_{T_j} f d\Omega,$$

where ν_{C_i} is the set of triangles composing C_i. Since we want to construct a finite volume scheme of at least order 2, we need a numerical integration method of at least one order higher. We use a numerical integration method of order 4 (4.1) and it is taken from [98]. Notations are provided in figure 4.1-(a):

$$I_T(f) = \frac{|T|}{60}(27f(a_G) + 3(f(v_1) + f(v_2) + f(v_3)) + 8(f(e_1) + f(e_2) + f(e_3))). \tag{4.1}$$



(a) Numerical integration points

(b) Dual split

Figure 4.1: Numerical integration

Following the calculation of the averaged value per dual cell, it is now possible to verify whether the V4 scheme is capable of transporting this averaged value. In order to find an answer, we take a set of perturbed meshes, as described in (3.4). We calculate the averaged value of function (3.6) on each dual volume, and then we try to calculate a finite element gradient, whose formula is given in (3.3), on each triangle. It is imperative that we verify that, starting from an averaged field, we are able to calculate a finite element gradient, since this is necessary in the reconstruction of the V4 scheme. The results of this test are shown in Figure 4.2.

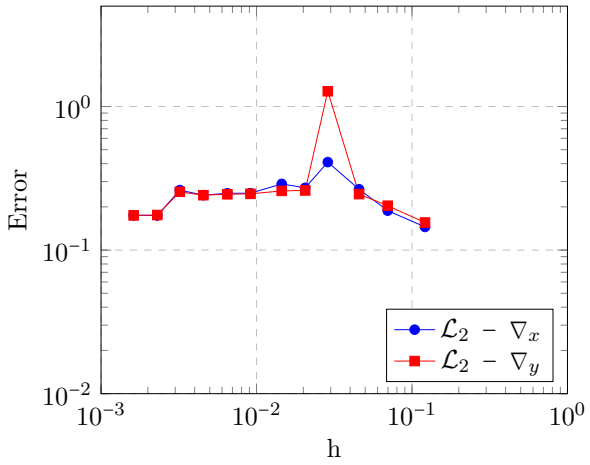


Figure 4.2: $\mathcal{L}_2$ error for finite element gradient on perturbed mesh with cell-averaged quantities.

The results show that the finite element gradient does not converge: the error between the P^1 finite element gradient and the exact solution does not decrease as the number of mesh elements increases. When calculating a finite element P^1 gradient, we use the value at the node of the mesh. However, in this case, we only know the averaged value per dual cell. In any mesh context, as in this case, the centre of gravity G_i of the dual cells and the nodes of the mesh i do not coincide, as can be seen in Figure (4.3). Under these conditions, assuming that the averaged value is equal to the value at the mesh node is a first-order hypothesis. This can be expressed mathematically as follows: $\bar{\mathbf{W}}_i = \mathbf{W}_i + O(h)$. A value at the node of order 1 leads to a gradient of order 0, which does not allow us to construct a numerical scheme of order 2. Therefore, to obtain a gradient of order 1, we must use values at the node of order 2 or higher. Additionally, the value at the nodes is used in MUSCL reconstruction and in calculating boundary conditions, as well as gradients. Therefore, it is necessary to calculate nodal values from the available averaged values. This process is known as deconvolution. The initial approach is set out in the 2002 article by Ollivier-Gooch and Van Altena [99].

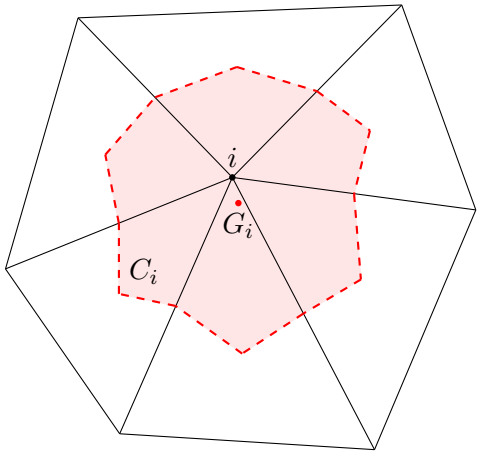


Figure 4.3: Center of gravity of the dual cell on any mesh.

4.1 Initial approach to deconvolution

In this section, the objective is to construct a vertex-centred finite volume scheme transporting the averaged value per dual cell. The numerical scheme under consideration is constructed from the V4 scheme. However, as has been demonstrated, the V4 scheme is not currently capable of transporting the averaged value per dual cell. Indeed, the V4 scheme requires information at mesh vertices and deconvolution enables transfering averared quantities into local polynomial approximation. It is evident that the key ingredient is how to link monomials of the polynomial approximation and the averaged quantities.

Ollivier-Gooch and Van Altena proposed an initial approach to deconvolution in 2002 [99]. The idea is to link the mean values with the monomials of the polynomial we are trying to construct. To do this, a Taylor series expansion of the reconstructed value is used:

$$\begin{aligned}\mathbf{W}_i^R(x-x_i,y-y_i) = \mathbf{W}|_i \quad &+ \quad \frac{\partial \mathbf{W}}{\partial x}|_i(x-x_i) + \frac{\partial \mathbf{W}}{\partial y}|_i(y-y_i) + \frac{\partial^2 \mathbf{W}}{\partial x^2}|_i\frac{(x-x_i)^2}{2} \\ &+ \quad \frac{\partial^2 \mathbf{W}}{\partial x \partial y}|_i(x-x_i)(y-y_i) + \frac{\partial^2 \mathbf{W}}{\partial y^2}|_i\frac{(y-y_i)^2}{2} + O(h^3),\end{aligned} \tag{4.2}$$

where $\mathbf{W}|_i$ is the value of the reconstructed solution and $\frac{\partial^{k+l}\mathbf{W}}{\partial x^k \partial y^l}|_i$ are its derivatives at the reference point (x_i, y_i). Conservation of the mean within a control volume requires that

$$\frac{1}{|C_i|}\int_{C_i} \mathbf{W}_i^R dV = \bar{\mathbf{W}}_i.$$

The mean value, over a single control volume C_i, is then

$$\frac{1}{|C_i|}\int_{C_i} \mathbf{W}_i^R = \mathbf{W}|_i + \frac{\partial \mathbf{W}}{\partial x}|_i\overline{x}_i + \frac{\partial \mathbf{W}}{\partial y}|_i\overline{y}_i + \frac{\partial^2 \mathbf{W}}{\partial x^2}|_i\frac{\overline{x^2}_i}{2} + \frac{\partial^2 \mathbf{W}}{\partial x \partial y}|_i\overline{xy} + \frac{\partial^2 \mathbf{W}}{\partial y^2}|_i\frac{\overline{y_i^2}}{2} + O(h^3), \tag{4.3}$$

where

$$\overline{x^n y^m}_i = \frac{1}{|C_i|}\int_{C_i}(x-x_i)^n(y-y_i)^m dv$$

are called moments associated with cell indexed i. Moreover, we have, for a cell $j \neq i$,

$$\begin{aligned}\bar{\mathbf{W}}_j = \frac{1}{|C_j|}\int_{C_j}\mathbf{W}_i^R = \mathbf{W}|_i \quad &+ \quad \frac{\partial \mathbf{W}}{\partial x}|_i\widehat{x}_{ij} + \frac{\partial \mathbf{W}}{\partial y}|_i\widehat{y}_{ij} + \frac{\partial^2 \mathbf{W}}{\partial x^2}|_i\frac{\widehat{x^2}_{ij}}{2} \\ &+ \quad \frac{\partial^2 \mathbf{W}}{\partial x \partial y}|_i\frac{\widehat{xy}_{ij}}{2} + \frac{\partial^2 \mathbf{W}}{\partial y^2}|_i\frac{\widehat{y^2}_{ij}}{2} + O(h^3),\end{aligned} \tag{4.4}$$

with

$$\widehat{x^n y^m}_{ij} = \sum_{l=0}^{m}\sum_{k=0}^{n}\begin{pmatrix} m \\ l \end{pmatrix}\begin{pmatrix} n \\ k \end{pmatrix}(x_j - x_i)^k \cdot (y_j - y_i)^l \cdot \overline{x^{n-k}y^{m-l}}_j.$$

These integrals are called moments of the cell j related to cell i. With the elements introduced, we can see that the reconstruction only depends on the metrics and derivatives of $\mathbf{W}$ at reference point i. For a given point i and the associated stencil (see figure 4.4), we seek to solve the following linear system, written in its general form:

$$\begin{bmatrix} 1 & \overline{x}_i & \overline{y}_i & \overline{x^2}_i & \overline{xy}_i & \overline{y^2}_i & \dots \\ \omega_{i1} & \omega_{i1}\widehat{x}_{i1} & \omega_{i1}\widehat{y}_{i1} & \omega_{i1}\widehat{x^2}_{i1} & \omega_{i1}\widehat{xy}_{i1} & \omega_{i1}\widehat{y^2}_{i1} & \dots \\ \omega_{i2} & \omega_{i2}\widehat{x}_{i2} & \omega_{i2}\widehat{y}_{i2} & \omega_{i2}\widehat{x^2}_{i2} & \omega_{i2}\widehat{xy}_{i2} & \omega_{i2}\widehat{y^2}_{i2} & \dots \\ \omega_{i3} & \omega_{i3}\widehat{x}_{i3} & \omega_{i3}\widehat{y}_{i3} & \omega_{i3}\widehat{x^2}_{i3} & \omega_{i3}\widehat{xy}_{i3} & \omega_{i3}\widehat{y^2}_{i3} & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \\ \omega_{iN} & \omega_{iN}\widehat{x}_{iN} & \omega_{iN}\widehat{y}_{iN} & \omega_{iN}\widehat{x^2}_{iN} & \omega_{iN}\widehat{xy}_{iN} & \omega_{iN}\widehat{y^2}_{iN} & \dots \end{bmatrix}\begin{pmatrix} \mathbf{W} \\ \frac{\partial \mathbf{W}}{\partial x} \\ \frac{\partial \mathbf{W}}{\partial y} \\ \frac{1}{2}\frac{\partial^2 \mathbf{W}}{\partial x^2} \\ \frac{\partial^2 \mathbf{W}}{\partial x \partial y} \\ \frac{1}{2}\frac{\partial^2 \mathbf{W}}{\partial y^2} \\ \vdots \end{pmatrix}_i = \begin{pmatrix} \bar{\mathbf{W}}_i \\ \omega_{i1}\bar{\mathbf{W}}_1 \\ \omega_{i2}\bar{\mathbf{W}}_2 \\ \omega_{i3}\bar{\mathbf{W}}_3 \\ \vdots \\ \omega_{iN}\bar{\mathbf{W}}_N \end{pmatrix},$$

where N denotes the number of control volumes in the stencil. In addition, the linear system is weighted to take into account the distance between the elements and the reference point [100]. The weights are defined as:

$$\omega_{ij} = \frac{1}{|\overrightarrow{x_j} - \overrightarrow{x_i}|^2}.$$

The matrix of the linear system is, by construction, rectangular. Several methods exist to resolve this type of problem. It is possible to solve the system using a singular value decomposition of the problem matrix, with LAPACK library for instance [101].

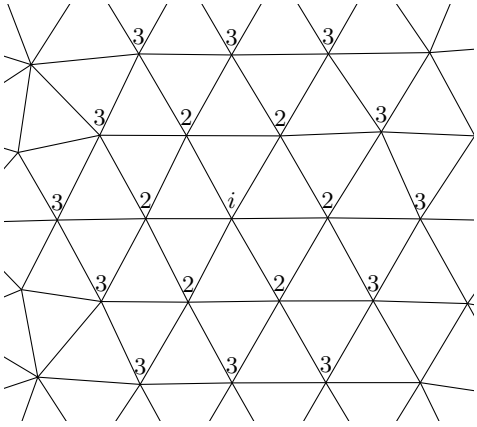


Figure 4.4: Standard stencil using edge connectivity from vertex i

The accuracy of such a linear problem solved by the least squares method depends on the number of neighbours considered in the stencil. This number must be at least equal to the number of derivatives we want to find. In order to have a good condition number of the left-hand side matrix, Bridel-Bertomeu [46] shows that, for a 2D case, this number should be at least equal to 15 for a second order reconstruction and to 25 for a third or fourth order reconstruction on a Cartesian grid, which corresponds to 2 rows of neighbours. To save computational time, we opt for another option introduced in [102], where we have to solve a smaller square matrix (3×3 for a second order reconstruction and 6×6 for a third order reconstruction).

4.2 A more compact approach to deconvolution

4.2.1 Introduction of the method

Bernard *et al.* [102] proposed a deconvolution approach based on the computation of a gradient and an Hessian with correction terms. This method leads to the solution of a small square system. We follow their idea in the following. We present here the method to achieve third order of accuracy.
Firstly, for a given cell i, for any cell $j \neq i$ and using the notations previously introduced, we can write the relationship 4.5. The first expression establishes a relationship between the averaged values, known quantities, and the values and successive derivatives at the vertices , *i.e.* the unknown quantities that we are trying to compute.

$$\begin{cases} \bar{\mathbf{W}}_i &= \mathbf{W}|_i + \dfrac{\partial \mathbf{W}}{\partial x}|_i \overline{x}_i + \dfrac{\partial \mathbf{W}}{\partial y}|_i \overline{y}_i + \dfrac{\partial^2 \mathbf{W}}{\partial x^2}|_i \dfrac{\overline{x^2}_i}{2} + \dfrac{\partial^2 \mathbf{W}}{\partial x \partial y}|_i \overline{xy}_i + \dfrac{\partial^2 \mathbf{W}}{\partial y^2}|_i \dfrac{\overline{y^2}_i}{2} + O(h^3) \\ \bar{\mathbf{W}}_j &= \mathbf{W}|_i + \dfrac{\partial \mathbf{W}}{\partial x}|_i \widehat{x}_{ij} + \dfrac{\partial \mathbf{W}}{\partial y}|_i \widehat{y}_{ij} + \dfrac{\partial^2 \mathbf{W}}{\partial x^2}|_i \dfrac{\widehat{x^2}_{ij}}{2} + \dfrac{\partial^2 \mathbf{W}}{\partial x \partial y}|_i \widehat{xy}_{ij} + \dfrac{\partial^2 \mathbf{W}}{\partial y^2}|_i \dfrac{\widehat{y^2}_{ij}}{2} + O(h^3). \end{cases} \tag{4.5}$$

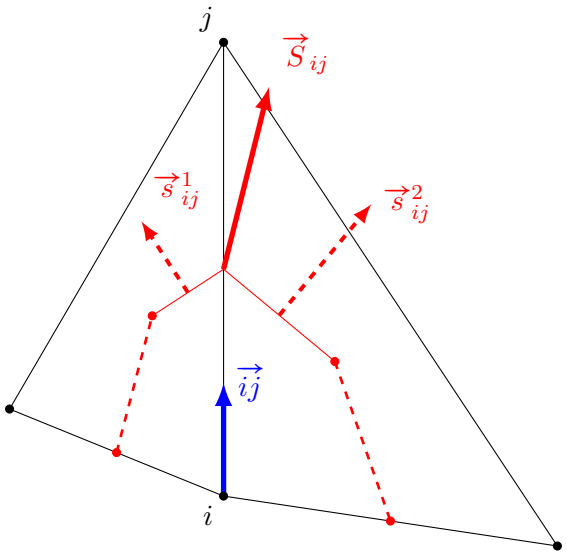


Figure 4.5: Notations for a 2D case - deconvolution

To reconstruct a third-order nodal quantity from averaged quantities, we first introduce an edge-based gradient based on the Green-Gauss theorem (4.6). As it will be shown after, the capability

of the operator to define a good approximation of the gradient is not a prerequisite. In practice, we choose:

$$\vec{R}_i = \sum_{j \in \mathcal{L}_i} \frac{\bar{\mathbf{W}}_i + \bar{\mathbf{W}}_j}{2} \vec{S}_{ij}. \tag{4.6}$$

The quantity $\frac{\bar{\mathbf{W}}_i + \bar{\mathbf{W}}_j}{2}$ assumes a linear reconstruction of the interface value $\mathbf{W}_f$. The approximation is accurate if the interface barycenter is located in the middle of centres of cells i and j. This situation occurs poorly for anisotropic adapted meshes, but it can be true for academic regular meshes.

We also introduce an edge-based Hessian, based on the Green-Gauss theorem (4.7). The interface gradient is approximated using a finite difference approach along the edge. This Hessian can be a poor approximation of the exact Hessian. These operators were first introduced in [100], although the Hessian operator was significantly modified to satisfy the mixed partial derivatives theorem. Indeed, by its construction, the Hessian operator is symmetric:

$$\bar{\bar{H}}_i = \sum_{j \in \mathcal{L}_i} \left(\frac{\bar{\mathbf{W}}_j - \bar{\mathbf{W}}_i}{\|IJ\|} \right) \frac{1}{2} (\vec{ij} \otimes \vec{S}_{ij} + \vec{S}_{ij} \otimes \vec{ij}). \tag{4.7}$$

We denote $\vec{ij}$ the unit vector from vertice i to vertice j and $\vec{S}_{ij}$ the outward normal from cell i to cell j. The norm of $\vec{S}_{ij}$ represents the area of the dual facet between cells i and j. The notations for both operators are illustrated in figure 4.5. The set $\mathcal{L}_i$ denotes the vertices that are adjacent to vertex i. An illustration of this neighbourhood is provided in Figure 4.6 for a Friedrich-Keller mesh. For points inside the domain, it is a simple matter to determine that each point always has six neighbours. This number of neighbours drops to four for certain vertices on the boundary of the domain. Finally, some points located in the corners have only two neighbours.

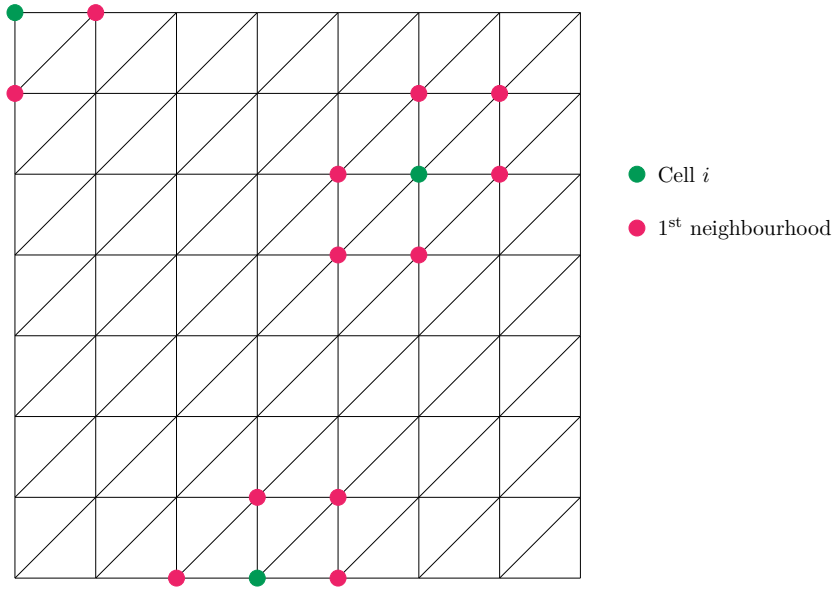


Figure 4.6: Stencil on a Friedrich-Keller

Following the introduction of the various operators, the procedure for constructing the nodal value $\mathbf{W}_i$ and the successive derivatives from the available averaged values will be outlined.

Firstly, the relations (4.5) are injected into the gradient and Hessian operators. This gives the following expression for the gradient:

$$\begin{aligned}\overrightarrow{R}_i(\bar{\mathbf{W}}_i) &= \sum_{j\in\mathcal{L}_i}\frac{\mathbf{W}|_i+\mathbf{W}|_i}{2}\overrightarrow{S}_{ij} + \frac{\partial\mathbf{W}}{\partial x}|_i\sum_{j\in\mathcal{L}_i}\frac{\overline{x}_i+\widehat{x}_{ij}}{2}\overrightarrow{S}_{ij}+\\ &\frac{\partial\mathbf{W}}{\partial y}|_i\sum_{j\in\mathcal{L}_i}\frac{\overline{y}_i+\widehat{y}_{ij}}{2}\overrightarrow{S}_{ij} + \frac{1}{2}\frac{\partial^2\mathbf{W}}{\partial x^2}|_i\sum_{j\in\mathcal{L}_i}\frac{\overline{x^2}_i+\widehat{x^2}_{ij}}{2}\overrightarrow{S}_{ij}+\\ &\frac{\partial^2\mathbf{W}}{\partial x\partial y}|_i\sum_{j\in\mathcal{L}_i}\frac{\overline{xy}_i+\widehat{xy}_{ij}}{2}\overrightarrow{S}_{ij} + \frac{1}{2}\frac{\partial^2\mathbf{W}}{\partial y^2}|_i\sum_{j\in\mathcal{L}_i}\frac{\overline{y^2}_i+\widehat{y^2}_{ij}}{2}\overrightarrow{S}_{ij}+O(h^3).\end{aligned}$$

By circulating along the edge of cell C_i, we have:

$$\sum_{j\in\nu_i}\frac{\mathbf{W}|_i+\mathbf{W}|_i}{2}\overrightarrow{S}_{ij}=0, \tag{4.8}$$

so it gives the following expression:

$$\begin{aligned}\overrightarrow{R}_i(\bar{\mathbf{W}}_i) &= \sum_{j\in\mathcal{L}_i}\Big(\frac{\partial\mathbf{W}}{\partial x}|_i\frac{\overline{x}_i+\widehat{x}_{ij}}{2}+\frac{\partial\mathbf{W}}{\partial y}|_i\frac{\overline{y}_i+\widehat{y}_{ij}}{2}+\frac{1}{2}\frac{\partial^2\mathbf{W}}{\partial x^2}|_i\frac{\overline{x^2}_i+\widehat{x^2}_{ij}}{2}+\\ &\frac{\partial^2\mathbf{W}}{\partial x\partial y}|_i\frac{\overline{xy}_i+\widehat{xy}_{ij}}{2}+\frac{1}{2}\frac{\partial^2\mathbf{W}}{\partial y^2}|_i\frac{\overline{y^2}_i+\widehat{y^2}_{ij}}{2}\Big)\overrightarrow{S}_{ij}+O(h^2),\end{aligned}$$

and the following expression for the Hessian:

$$\begin{aligned}\bar{\bar{H}}_i(\bar{\mathbf{W}}_i) &= \sum_{j\in\mathcal{L}_i}\Big(\frac{\mathbf{W}_i-\mathbf{W}_i}{\|IL\|}\Big)\frac{1}{2}(\overrightarrow{ij}\otimes\overrightarrow{S}_{ij}+\overrightarrow{S}_{ij}\otimes\overrightarrow{ij})+\\ &\frac{\partial\mathbf{W}}{\partial x}|_i\sum_{j\in\mathcal{L}_i}\Big(\frac{\widehat{x}_{ij}-\overline{x}_i}{\|IL\|}\Big)\frac{1}{2}(\overrightarrow{ij}\otimes\overrightarrow{S}_{ij}+\overrightarrow{S}_{ij}\otimes\overrightarrow{ij})+\\ &\frac{\partial\mathbf{W}}{\partial y}|_i\sum_{j\in\mathcal{L}_i}\Big(\frac{\widehat{y}_{ij}-\overline{y}_i}{\|IL\|}\Big)\frac{1}{2}(\overrightarrow{ij}\otimes\overrightarrow{S}_{ij}+\overrightarrow{S}_{ij}\otimes\overrightarrow{ij})+\\ &\frac{1}{2}\frac{\partial^2\mathbf{W}}{\partial x^2}|_i\sum_{j\in\mathcal{L}_i}\Big(\frac{\widehat{x^2}_{ij}-\overline{x^2}_i}{\|IL\|}\Big)\frac{1}{2}(\overrightarrow{ij}\otimes\overrightarrow{S}_{ij}+\overrightarrow{S}_{ij}\otimes\overrightarrow{ij})+\\ &\frac{\partial^2\mathbf{W}}{\partial x\partial y}|_i\sum_{j\in\mathcal{L}_i}\Big(\frac{\widehat{xy}_{ij}-\overline{xy}_i}{\|IL\|}\Big)\frac{1}{2}(\overrightarrow{ij}\otimes\overrightarrow{S}_{ij}+\overrightarrow{S}_{ij}\otimes\overrightarrow{ij})+\\ &\frac{1}{2}\frac{\partial^2\mathbf{W}}{\partial y^2}|_i\sum_{j\in\mathcal{L}_i}\Big(\frac{\widehat{y^2}_{ij}-\overline{y^2}_i}{\|IL\|}\Big)\frac{1}{2}(\overrightarrow{ij}\otimes\overrightarrow{S}_{ij}+\overrightarrow{S}_{ij}\otimes\overrightarrow{ij})+O(h).\end{aligned}$$

This expression can be rewritten as follows:

$$\begin{aligned}\bar{\bar{H}}_i(\bar{\mathbf{W}}_i) &= \sum_{j\in\mathcal{L}_i}\Big[\frac{\partial\mathbf{W}}{\partial x}|_i\Big(\frac{\widehat{x}_{ij}-\overline{x}_i}{\|IL\|}\Big)+\frac{\partial\mathbf{W}}{\partial y}|_i\Big(\frac{\widehat{y}_{ij}-\overline{y}_i}{\|IL\|}\Big)+\\ &\frac{1}{2}\frac{\partial^2\mathbf{W}}{\partial x^2}|_i\Big(\frac{\widehat{x^2}_{ij}-\overline{x^2}_i}{\|IL\|}\Big)+\frac{\partial^2\mathbf{W}}{\partial x\partial y}|_i\Big(\frac{\widehat{xy}_{ij}-\overline{xy}_i}{\|IL\|}\Big)+\\ &\frac{1}{2}\frac{\partial^2\mathbf{W}}{\partial y^2}|_i\Big(\frac{\widehat{y^2}_{ij}-\overline{y^2}_i}{\|IL\|}\Big)\Big]\frac{1}{2}(\overrightarrow{ij}\otimes\overrightarrow{S}_{ij}+\overrightarrow{S}_{ij}\otimes\overrightarrow{ij})+O(h).\end{aligned}$$

We can see that the expressions obtained for the gradient and the Hessian are independent of the nodal value $\mathbf{W}_i$. Thus, we can write the following linear system:

$$\begin{pmatrix} \overrightarrow{R}_i(\bar{\mathbf{W}}_i) \\ \overrightarrow{H}_i(\bar{\mathbf{W}}_i) \end{pmatrix} = \bar{\bar{\mathbf{M}}} \begin{pmatrix} \frac{\partial \mathbf{W}}{\partial x}|_i \\ \frac{\partial \mathbf{W}}{\partial y}|_i \\ \frac{\partial^2 \mathbf{W}}{\partial x^2}|_i \\ \frac{\partial^2 \mathbf{W}}{\partial xy}|_i \\ \frac{\partial^2 \mathbf{W}}{\partial y^2}|_i \end{pmatrix}. \tag{4.9}$$

In order to write the linear system (4.9), it is necessary to rewrite the Hessian operator in vector form. As it is symmetric by construction, this does not entail any particular difficulty.

Boundary closure The existing formulae do not permit cells located at the boundary of the domain to be closed. It is therefore proposed that the definition be extended to include cells located at the edge of the domain, with the objective of preserving property (4.8). Let C_i be a border cell. We can then write this new definition for the gradient operator.

$$\overrightarrow{R}_i(\bar{\mathbf{W}}_i) = \sum_{j \in \mathcal{L}_i} \frac{\bar{\mathbf{W}}_i + \bar{\mathbf{W}}_j}{2} \overrightarrow{S}_{ij} + \sum_{j \in \mathcal{L}_i^B} \mathbf{W}_{ij} \overrightarrow{S}_{ij}$$

where set $\mathcal{L}_i$ remains the same as before and $\mathcal{L}_i^B$ represents the set of boundary facets. We therefore seek to reconstruct an edge state $\mathbf{W}_{ij}$, located at point x_{ij}, the midpoint of the edge facet, as can be seen in Figure 4.7. To reconstruct the boundary state, we use the following Taylor expansions:

$$\begin{aligned} \mathbf{W}_{ij} &= \mathbf{W}_i + (x_i - x_{ij}) \frac{\partial \mathbf{W}}{\partial x}|_{ij} + (y_i - y_{ij}) \frac{\partial \mathbf{W}}{\partial y}|_{ij} + O(h^2), \\ \mathbf{W}_{ij} &= \mathbf{W}_j + (x_j - x_{ij}) \frac{\partial \mathbf{W}}{\partial x}|_{ij} + (y_j - y_{ij}) \frac{\partial \mathbf{W}}{\partial y}|_{ij} + O(h^2). \end{aligned}$$

It is evident that $x_i - x_{ij} = \frac{1}{4}(x_i - x_j)$ and $y_i - y_{ij} = \frac{3}{4}(x_j - x_i)$ As a consequence, the following reconstruction for the boundary state $\mathbf{W}_{ij}$ is proposed.

$$\mathbf{W}_{ij} = \frac{3\bar{\mathbf{W}}_i + \bar{\mathbf{W}}_j}{4} + O(h^2)$$

This proposed reconstruction is a 1-exact reconstruction. It is therefore only suitable for 1-exact deconvolution. To go up a further order, it is therefore necessary to increase the degree of accuracy of the value reconstructed on the edge of the domain.

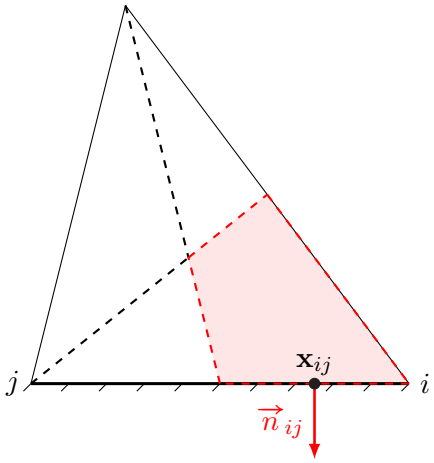


Figure 4.7: Notations for a boundary cell

Subsequently, the unknown $\mathbf{W}_i$ may be reintegrated into the system, resulting in the following linear system:

$$\begin{pmatrix} \bar{\mathbf{W}}_i \\ \vec{R}_i(\bar{\mathbf{W}}_i) \\ \vec{H}_i(\bar{\mathbf{W}}_i) \end{pmatrix} = \begin{pmatrix} 1 & \overline{x}_i & \overline{y}_i & \overline{x^2}_i & \frac{1}{2}\overline{xy}_i & \overline{y^2}_i \\ 0 & & & & & \\ 0 & & & & & \\ 0 & & & \bar{\bar{\mathbf{M}}} & & \\ 0 & & & & & \\ 0 & & & & & \end{pmatrix} \begin{pmatrix} \mathbf{W}|_i \\ \frac{\partial \mathbf{W}}{\partial x}|_i \\ \frac{\partial \mathbf{W}}{\partial y}|_i \\ \frac{\partial^2 \mathbf{W}}{\partial x^2}|_i \\ \frac{\partial^2 \mathbf{W}}{\partial xy}|_i \\ \frac{\partial^2 \mathbf{W}}{\partial y^2}|_i \end{pmatrix}. \tag{4.10}$$

The system matrix is called the convolution matrix. Since the matrix $\bar{\bar{\mathbf{M}}}$ is well conditioned and invertible, we can always write

$$\begin{pmatrix} \mathbf{W}|_i \\ \frac{\partial \mathbf{W}}{\partial x}|_i \\ \frac{\partial \mathbf{W}}{\partial y}|_i \\ \frac{\partial^2 \mathbf{W}}{\partial x^2}|_i \\ \frac{\partial^2 \mathbf{W}}{\partial xy}|_i \\ \frac{\partial^2 \mathbf{W}}{\partial y^2}|_i \end{pmatrix} = \begin{pmatrix} 1 & \overline{x}_i & \overline{y}_i & \overline{x^2}_i & \frac{1}{2}\overline{xy}_i & \overline{y^2}_i \\ 0 & & & & & \\ 0 & & & & & \\ 0 & & & \bar{\bar{\mathbf{M}}} & & \\ 0 & & & & & \\ 0 & & & & & \end{pmatrix}^{-1} \begin{pmatrix} \bar{\mathbf{W}}_i \\ \vec{R}_i(\bar{\mathbf{W}}_i) \\ \vec{H}_i(\bar{\mathbf{W}}_i) \end{pmatrix}. \tag{4.11}$$

The inverse matrix of this linear system is called the deconvolution matrix. This approach is assumed to be 2-exact. This means that from the averaged quantities we construct a third-order nodal quantity, a second-order nodal gradient and a first-order nodal Hessian. For a second order reconstruction, we have to solve the following linear system:

$$\begin{pmatrix} \bar{\mathbf{W}}_i \\ \vec{R}_i(\bar{\mathbf{W}}_i) \end{pmatrix} = \begin{pmatrix} 1 & \overline{x}_i & \overline{y}_i \\ 0 & & \\ 0 & & \bar{\bar{\mathbf{M}}} \end{pmatrix} \begin{pmatrix} \mathbf{W}|_i \\ \frac{\partial \mathbf{W}}{\partial x}|_i \\ \frac{\partial \mathbf{W}}{\partial y}|_i \end{pmatrix}. \tag{4.12}$$

It is also noteworthy that the gradient and Hessian operators introduced here are compact in the sense that, for a vertex i, they involve only its direct neighbours. The table 4.1 illustrates the linear system size that is to be solved, contingent on the problem's dimensions and the desired degree of precision. What emerges from the table 4.1 is that the linear system that is to be solved using this method is invariably a small square system in comparison to other methods. Indeed, for a third-order reconstruction in two dimensions, the linear system to be solved is of size 6×6. This is significantly smaller than the values attained in Ollivier-Gooch's approach, where the rectangular linear system can reach 6×25 for reasons pertaining to stability.

Size of the convolution matrix			
Order of reconstrution	**1D**	**2D**	**3D**
1	1	1	1
2	2	3	4
3	3	6	10
n	n	$\frac{n(n+1)}{2}$	$\frac{n(n+1)(n+2)}{6}$

Table 4.1: Size of the convolution matrix

To illustrate this point, consider the reconstruction of a value of order 3 in two dimensions. In such a scenario, it is necessary to construct and invert a convolution matrix of size 6×6. The system under consideration contains six unknowns. Indeed, there is one unknown for the value, two unknowns for the gradient, and three unknowns for the Hessian at the vertex, since the symmetry of the Hessian operator is taken into account. In order for the system to be well-posed and the convolution matrix to be invertible, it is necessary that vertex i has a minimum of five neighbours. As demonstrated in Figure 4.6, this is typically observed for points within the computational domain (although this is not universally applicable to all meshes). However, the number of neighbours is generally lower for vertices located on the boundary of the domain, as is the case, for example, with Friedrich-Keller meshes. In accordance with the prevailing definition of the gradient and Hessian operators, it is not feasible to construct a third order reconstruction for vertices that do not have a minimum of five neighbours. The section 4.3 will elaborate on an extension of this method that has been developed for the purpose of handling such cases.

Remark :
For a given cell i, if, $\forall j \in \nu_i$, $\overline{x}_i = 0$ and $\overline{y}_i = 0$, then it is then possible to prove the following results.

$$\begin{aligned} \bar{\mathbf{W}}_i &= \mathbf{W}|_i + O(h^2) \\ \overrightarrow{R}(\bar{\mathbf{W}}_i) &= \begin{pmatrix} \frac{\partial \mathbf{W}_i}{\partial x} \\ \frac{\partial \mathbf{W}_i}{\partial x} \end{pmatrix} + O(h^2) \end{aligned}$$

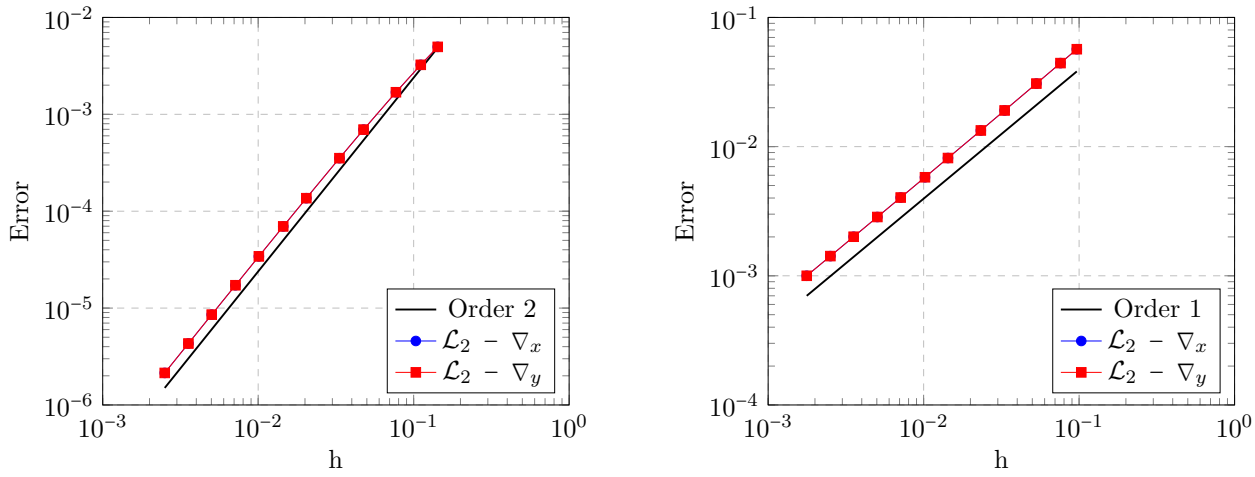
For this type of mesh (Friedrich-Keller for example),$\bar{\mathbf{W}}_i$ is an approximation of value $\mathbf{W}_i$ to the second order. In the case of any mesh, this is an approximation to the first order.

Following the introduction of the gradient and Hessian operators, the subsequent section will present a series of numerical results.

4.2.2 Numerical results

In this section, we present the results of the deconvolution method, as outlined in Section 4.2. To this end, we utilise the four types of meshes introduced in Section 3.2.1. For each mesh,

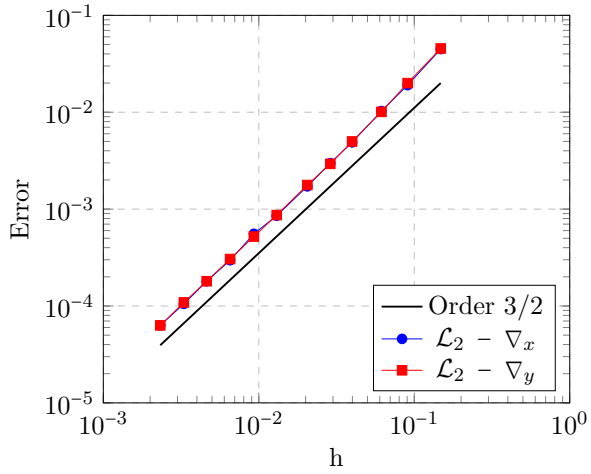
the sinusoidal function, as given in 3.6, is integrated over each dual volume. Firstly, a second-order deconvolution method is applied. Therefore, based on the averaged fields per dual cell, the objective is to reconstruct a second order nodal value and a first order nodal gradient. As illustrated in Figure 4.9, the $\mathcal{L}_2$ norm of the error for the gradient is shown as a function of mesh size for the four types of mesh. The error is calculated across all mesh vertices. The results demonstrate that for regular, Union Jack, and Friedrich-Keller meshes, the nodal gradient resulting from deconvolution converges at the hybrid order 3/2. The three types of mesh under consideration have the common feature of being regular, *i.e.* most interior points have the following property: for a given cell i, $\forall j \in \nu_i$, $\overline{x}_i = 0$ and $\overline{y}_i = 0$. It can thus be concluded that the nodal gradient converges to order 2 for interior points and to order 1 for edge vertices (see Figure 4.8). To plot the results on the interior points, only the regular points of the mesh are considered. The Friedrich-Keller meshes are used as an example in this figure, but the conclusion would be the same for the other two types of meshes. Finally, for perturbed meshes, the nodal gradient is first order. We find the expected results, namely that the gradient resulting from deconvolution is of order 1 on all types of mesh and that this gradient is of order 2 when the elements comprising the mesh are regular.



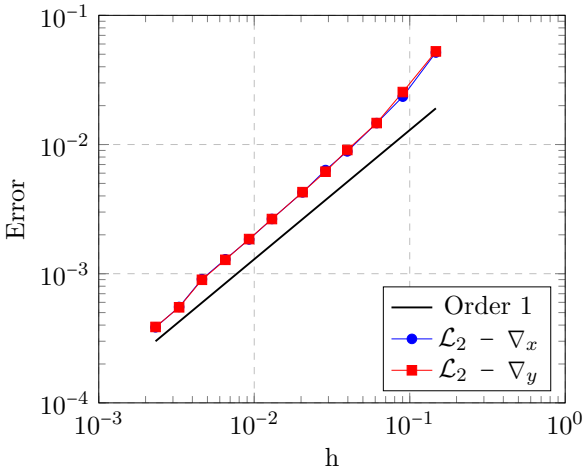
(a) Interior points

(b) Boundary points

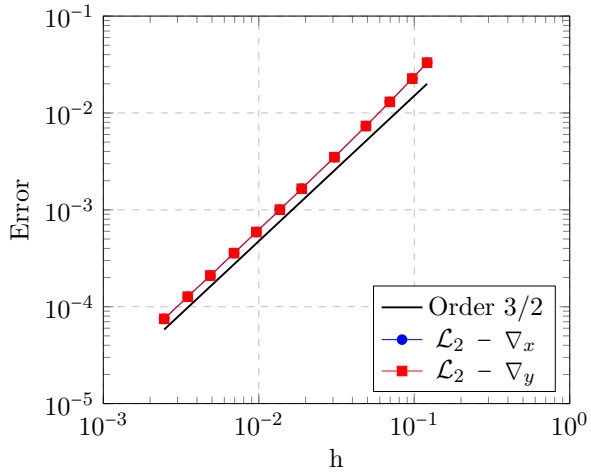
Figure 4.8: $\mathcal{L}_2$ error for the nodal gradient on Friedrich-Keller mesh for interior vertices (left) and boundary vertices (right) - O2 deconvolution



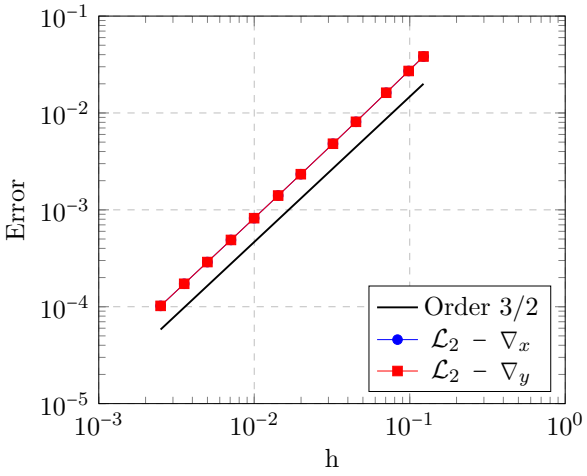
(a) Regular mesh



(b) Disturbed mesh



(c) Union jack mesh

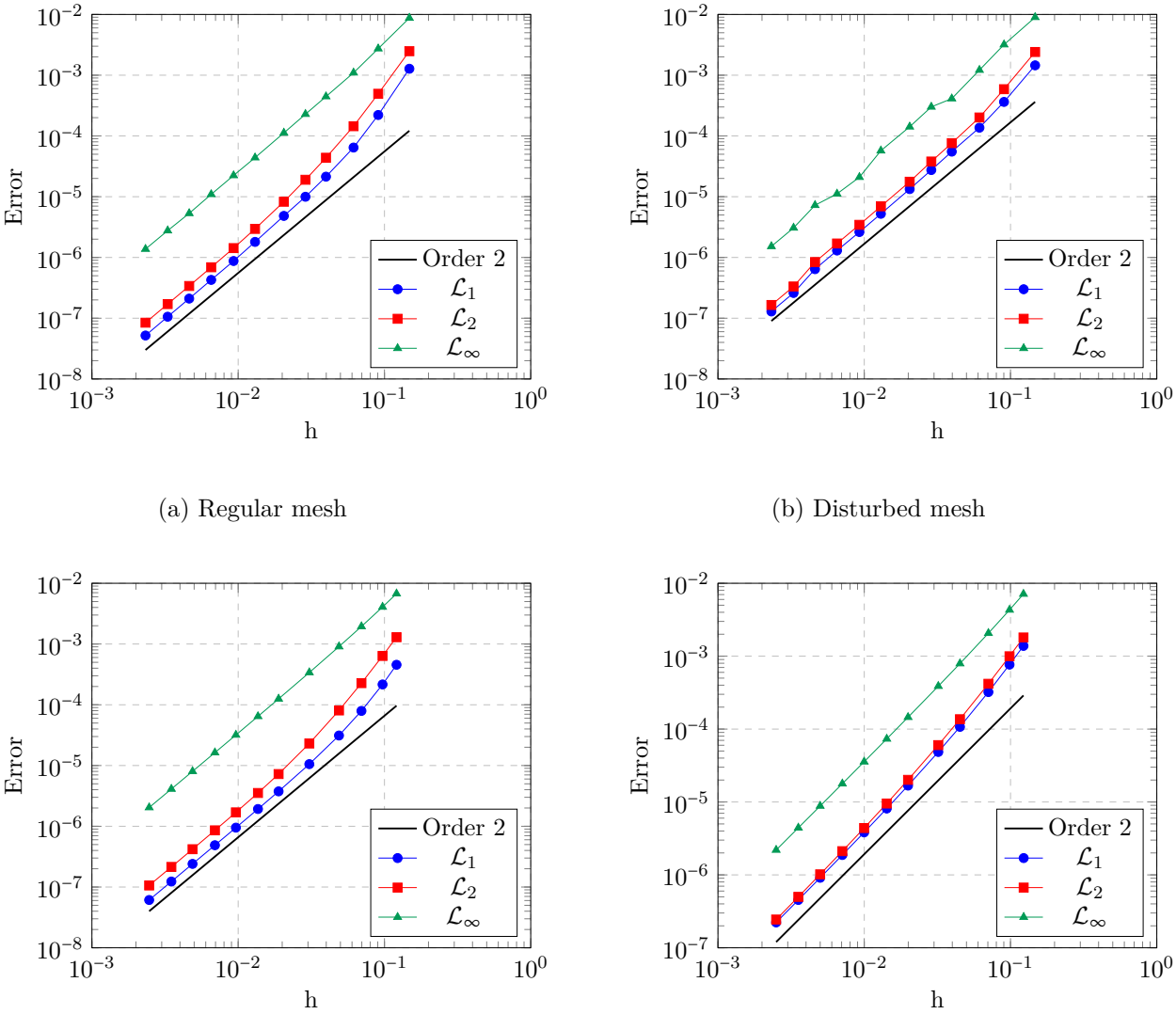


(d) Friedrich-Keller mesh

Figure 4.9: $\mathcal{L}_2$ error for the nodal gradient on regular, perturbed, Friedrich-Keller and Union Jack meshes, clockwise from top left - O2 deconvolution

Figure 4.10 shows the evolution of the $\mathcal{L}_2$ norm of the error between the value at the vertex reconstructed by deconvolution and the reference solution as a function of the mesh size. All points in the mesh are considered in the error calculation. The results show that, based on the averaged fields, the deconvolution operators can be used to construct a second-order nodal value on any type of mesh. This is an expected result.

We now wish to verify the results of third-order deconvolution. The operators, as currently defined, are not applicable to third order on all points of a given mesh. In fact, we can distinguish between two cases. If a vertex i has at least five neighbours, then we can apply the third-order deconvolution operators. Otherwise, it is not possible to apply the operators at order 3, and it is therefore necessary to downgrade the order of the deconvolution and apply the second-order operators, since it is always possible to construct and invert a second-order convolution matrix. A generalisation of third-order deconvolution operators to all points in the mesh is presented in the following section.



(a) Regular mesh

(b) Disturbed mesh

(c) Union jack mesh

(d) Friedrich-Keller mesh

Figure 4.10: $\mathcal{L}_2$ error for the corrected field on regular, perturbed, Friedrich-Keller and Union Jack meshes, clockwise from top left - O2 deconvolution

4.3 Extension to deconvolution

4.3.1 Presentation of the method

Our aim is to generalise third-order deconvolution for all points in the mesh. At present, the number of neighbours taken into account in the construction of the deconvolution matrix is the limiting factor. The current method is designed to consider only the direct neighbours of a given vertex i when constructing the convolution matrix. If this number of neighbours is insufficient, the convolution matrix is no longer invertible and it is then necessary to reduce the order of deconvolution. In this section, we will present a method for circumventing this problem. This method is a generalisation to unstructured meshes of the method presented by Mazaleyrat *et al.* in [103]. Since the limiting factor is the number of neighbouring vertices taken into account for the construction of the convolution matrix, we wish to extend the stencil considered during the construction of the correction matrix. However, we only want to extend this stencil when necessary, *i.e.* only for points that do not have enough neighbouring vertices for the convolution

matrix to be invertible. This number of vertices, for third-order deconvolution, is set, at least, at 6 in 2D and 10 in 3D. This means that if we want a stencil for vertex i containing n points, then we must take into account at least $n-1$ points in the neighbourhood, since vertex i counts as one. In order to ascertain the stencil $\mathcal{L}_i$ for each vertex of the mesh, the following procedure is followed. The first step in the process is to examine the first neighbourhood of each point. If the initial neighbourhood contains sufficient points, the search is halted. Otherwise, we take into account the points in the second neighbourhood and repeat the procedure until the desired number of points has been reached. If we represent our mesh as an undirected connected graph, then the search for this stencil can be handled using a breadth-first graph traversal algorithm [104]. An illustration of this stencil, for a Friedrich-Keller mesh, is given in Figure 4.11. For an interior vertex there is no need to use an extended stencil as a vertex always has 6 neighbours. For a boundary vertex we need to use the second neighbourhood to reach the target of 6 points. In addition, a corner vertex needs also the second neighbourhood to reach 6 points.

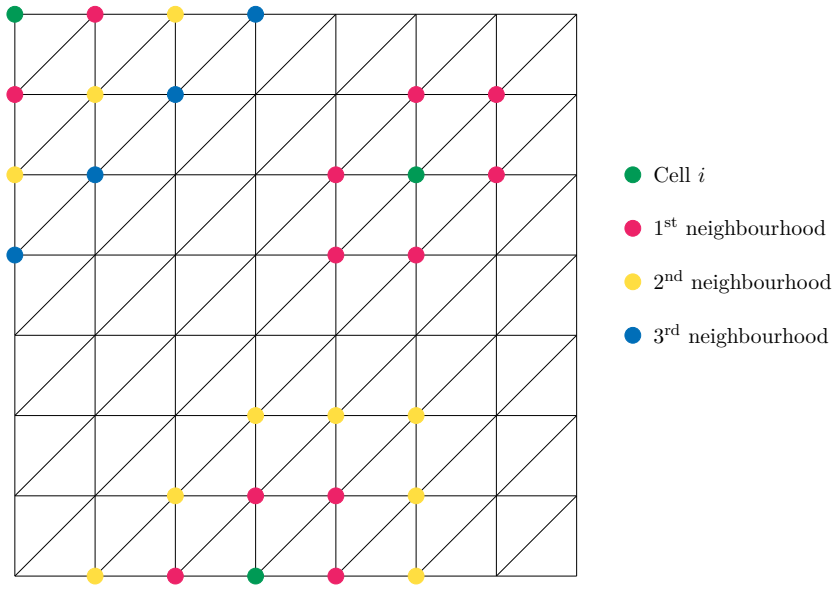


Figure 4.11: Extended stencil on a Friedrich-Keller mesh for gradient computation

Now that we are able to extend the stencil, it is necessary to modify the gradient and Hessian operators. A new definition of the gradient operator is given in 4.13:

$$\overrightarrow{R}_i(\mathbf{W}_i) = \sum_{j \in \mathcal{L}_i} \Big(\sum_{l \in \mathcal{L}_{ij}} \alpha_{il} \frac{\bar{\mathbf{W}}_i + \bar{\mathbf{W}}_l}{2} \overrightarrow{S}_{ij} \Big). \tag{4.13}$$

The operator introduced in this way differs from the one given in the literature. In fact, we work with an extended stencil when necessary. More precisely, we define $\mathcal{L}_i$ as the set of vertices neighbouring i, such that $\mathcal{L}_i$ contains a sufficient number of points. In addition, we introduce the set $\mathcal{L}_{ij}$, containing the set of vertices in $\mathcal{L}_i$ such that $\overrightarrow{ij} \cdot \overrightarrow{S}_{ij} > p$, where p is the proportion of the half-plane in the direction of edge $\overrightarrow{S}_{ij}$ that is taken into account. Generally, p is chosen from the interval $[0, 1[$, where 0 corresponds to the half-plane and $p \to 1$ involves considering only the immediate neighbour. In practical terms, this means that, for each face, we identify, thanks to the plane in the direction of edge $\overrightarrow{S}_{ij}$ a subset of the vertices of the stencil. Moreover, we weight our gradient and we choose to work with the following constraint:

$$\sum_{j\in\mathcal{L}_i}\Big(\sum_{l\in\mathcal{L}_{ij}}\alpha_{il}\frac{\mathbf{W}_i+\mathbf{W}_i}{2}\overrightarrow{n}_{ij}S_{ij}\Big)=0\Leftrightarrow\sum_{l\in\mathcal{L}_{ij}}\alpha_{il}=1.$$

So, we have the following definition of the weights :

$$\alpha_{il}=\begin{cases}0.9 \quad \text{if} \quad l=j,\\ \dfrac{0.1}{|\mathcal{L}_{ij}|-1} \quad \text{otherwise.}\end{cases}$$

This definition of weights gives greater importance to immediate neighbours and therefore reduces the influence, for a given vertex i, of points located far from it. Following the same methodology, we have developed a new definition of the Hessian operator:

$$\bar{\bar{H}}_i(\mathbf{W}_i)=\sum_{j\in\mathcal{L}_i}\Big(\sum_{l\in\mathcal{L}_{ij}}\alpha_{il}\Big(\frac{\bar{\mathbf{W}}_l-\bar{\mathbf{W}}_i}{\|IL\|}\Big)\frac{1}{2}(\overrightarrow{ij}\otimes\overrightarrow{S}_{ij}+\overrightarrow{S}_{ij}\otimes\overrightarrow{ij})\Big). \tag{4.14}$$

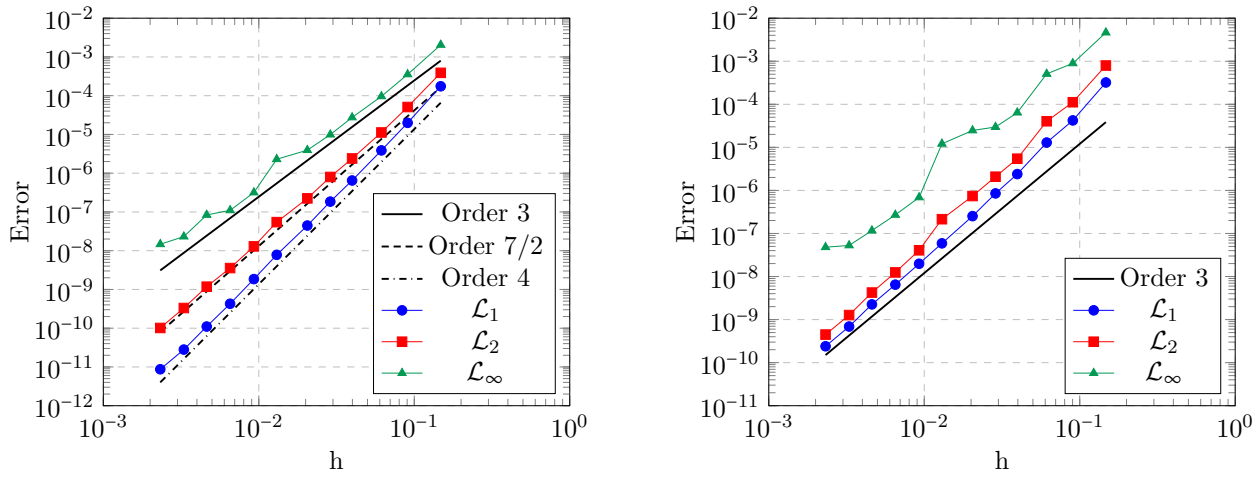
The weight definition for the gradient and Hessian operators remains unchanged. The convolution matrix is then constructed using the following procedure. Firstly, we determine the stencil for each vertex of the mesh. The relations 4.5 are then injected into the new operators, with the aim of constructing a new convolution matrix. The outcome of this process is a linear system that is comparable to 4.10. Our experience shows that this matrix is always well-conditioned and invertible for any point in a given mesh.
The development of these novel operators was undertaken with the objective of functioning with an expanded stencil for deconvolution. The following observations can be made about this novel approach. Firstly, although it is now possible to work with an extended stencil, this does not result in a change to the size of the linear system that is to be solved. The information given in Table 4.1 remains valid. The extension of the stencil merely effects a modification in the conditioning of the convolution matrix. This approach is distinct from the least squares method, in which the extension of the stencil results in the addition of columns to the convolution matrix. Secondly, given that the matrices in question are still of a small size, it is possible to increase the size of the stencil arbitrarily in order to improve the conditioning of the convolution matrix. From a numerical perspective, given that the dimensions of the convolution matrix remain constant, this does not result in any alterations to the data structure. Finaly, this definition of the stencil is such that it is possible to offset it locally if necessary, as may be required in the event of a shock, for example.
We proposed two parameters in order to enhance the quality of the numerical solution. These are the number n of neighbours to be taken into account in the stencil and the parameter p used to define the subset of points $\mathcal{L}_{ij}$. The ensuing section will present numerical results.

4.3.2 Numerical results

In this section, we present the numerical results of second-order deconvolution with stencil extension. To do this, we integrate the function 3.6 over each dual cell, then reconstruct a nodal value and the successive derivatives using deconvolution operators. To obtain these results, we set the parameter p to 0.9. Furthermore, for each point in the mesh, we aim to construct a stencil containing a minimum of six neighbours. As we add points by neighbourhoods iteratively, the stencil may contain more than six points. However, we do not intend to expand the stencil further. We therefore work with a minimal stencil for each point. Finally, we present numerical results for both regular and perturbed meshes. Union-Jack meshes are incompatible with the

Hessian operator because the convolution matrix is never invertible for certain points in the mesh. Furthermore, Friedrich-Keller meshes are a special case. In these cases, the regularity of the mesh at the boundary of the domain can cause the convolution matrix to become non-invertible.

Figure 4.12 shows the evolution of the norm of the error between the solution at the vertex reconstructed by deconvolution and the exact solution evaluated at the mesh vertex as a function of the mesh size. Figure 4.12-(a) shows the results on regular meshes. We observe that the error is equal to $O(h^{3+1/p})$ (based on the formulation by Tarsia-Morisco and Nishikawa [87]), where p is the norm under consideration. This shows us that the nodal solution converges to order 4 on the interior vertices and to order 3 for the vertices located on the boundary of the domain. This can be explained by the fact that, for this type of mesh, stencils for the interior vertices are regular. Furthermore, Figure 4.12-(b) shows the same results, but on perturbed meshes. We observe that the solution at the vertices converges to order 3, regardless of the norm considered. On any type of mesh, we are able to construct a nodal solution of order 3 from the available mean fields.



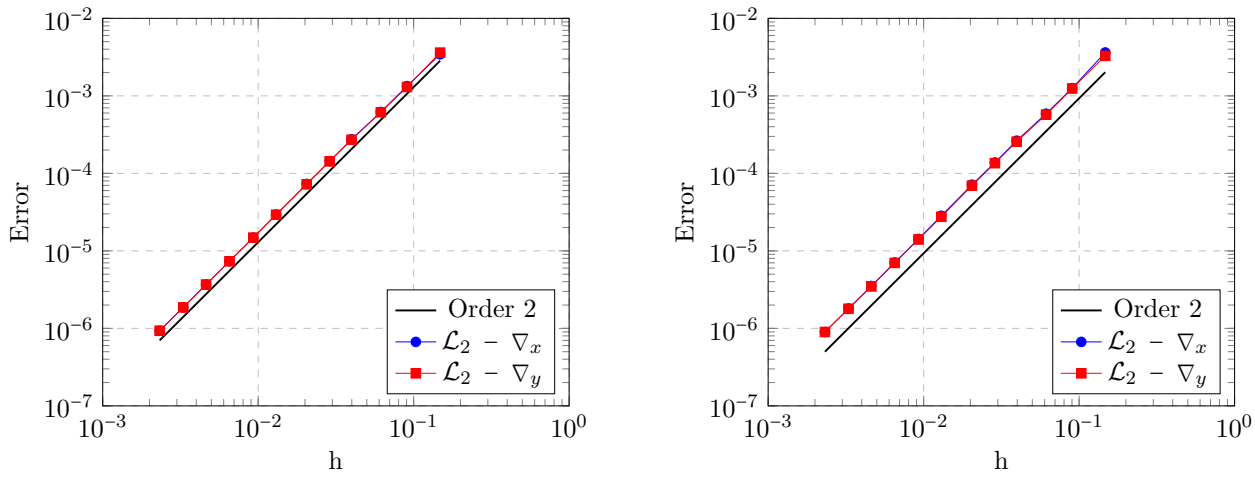
(a) Regular

(b) Disturbed

Figure 4.12: $\mathcal{L}_1$, $\mathcal{L}_2$ and $\mathcal{L}_\infty$ error for the corrected field on regular mesh (left) and disturbed mesh (right) - O3 Deconvolution

Figure 4.13 illustrates the evolution of the $\mathcal{L}_2$ norm of the error for the nodal gradient as a function of mesh size. The results are comparable for both types of mesh, and the conclusion is consistent, as a second-order gradient is constructed at any vertex for all types of mesh. Figure 4.14 also shows the $\mathcal{L}_2$ norm of the error for the Hessian at the vertices as a function of mesh size. The results demonstrate that the nodal Hessian is at least first order on all types of mesh. To be precise, it is of order one on any mesh and of order 1.5 on a regular mesh.

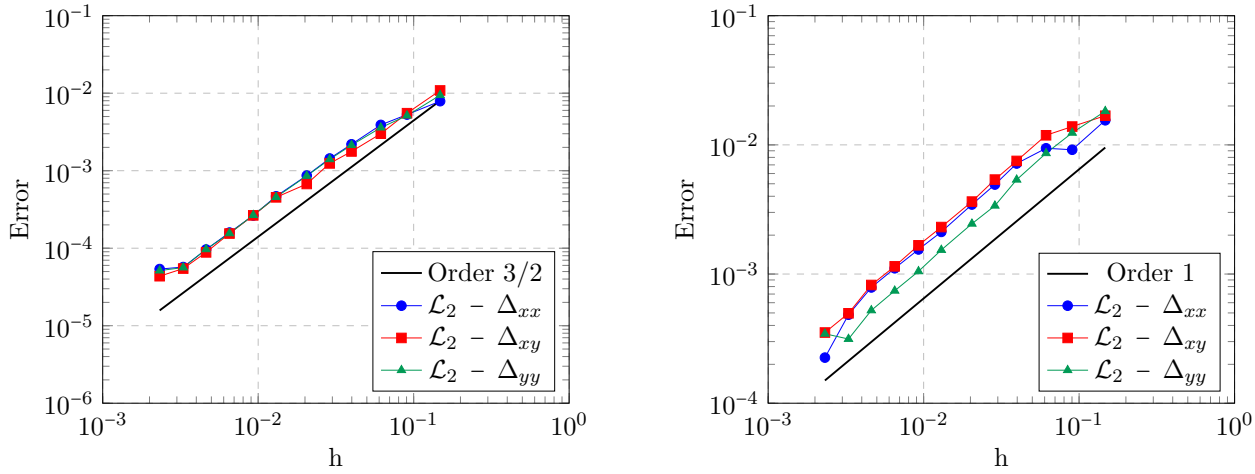
Finally, given that the stencil extension was largely motivated by boundary vertices, the quality of the gradient and Hessian on these elements is examined. Figure 4.15 shows the gradient results, while Figure 4.16 illustrates the results obtained for the Hessian. The nodal gradient is of order 2 and the Hessian is of order 1 for both types of meshes considered. These results provide validation for the stencil extension method for the nodes located on the boundary of the computational domain.



(a) Regular

(b) Disturbed

Figure 4.13: $\mathcal{L}_2$ error for the nodal gradient on regular mesh (left) and disturbed mesh (right) - O3 deconvolution



(a) Regular

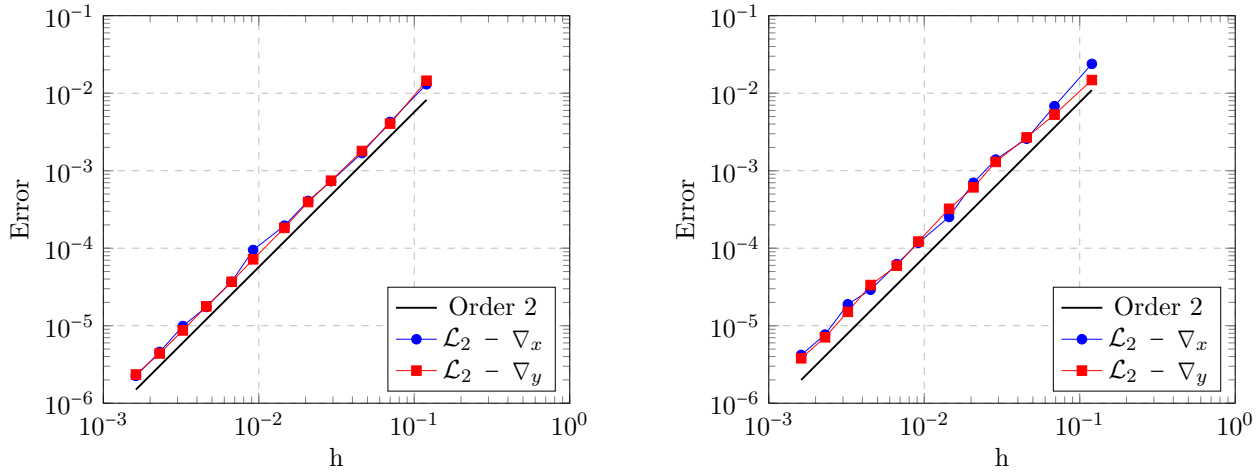
(b) Disturbed

Figure 4.14: $\mathcal{L}_2$ error for the nodal hessian on regular mesh (left) and disturbed mesh (right) - O3 deconvolution

In this section, we have presented a method for generalising third-order deconvolution for all points in the mesh. To do this, we have modified the deconvolution operators so that they can work on an extended stencil. This new stencil is determined, for a given vertex, using its successive neighbourhoods. As a result, we seek to construct the most compact stencil possible and therefore always work with the smallest possible stencil, the number of points to be considered being given by the number of unknowns in the system to be solved. This approach also allows the stencil to be shifted off-centre if necessary, as may be the case when there is a shock, for example. The results demonstrate the capability to build values at the third order, gradient at second order, and Hessian at first-order at the vertices, utilising the available averaged values.

The subsequent objective is to use these deconvolution results to construct a vertex-centred

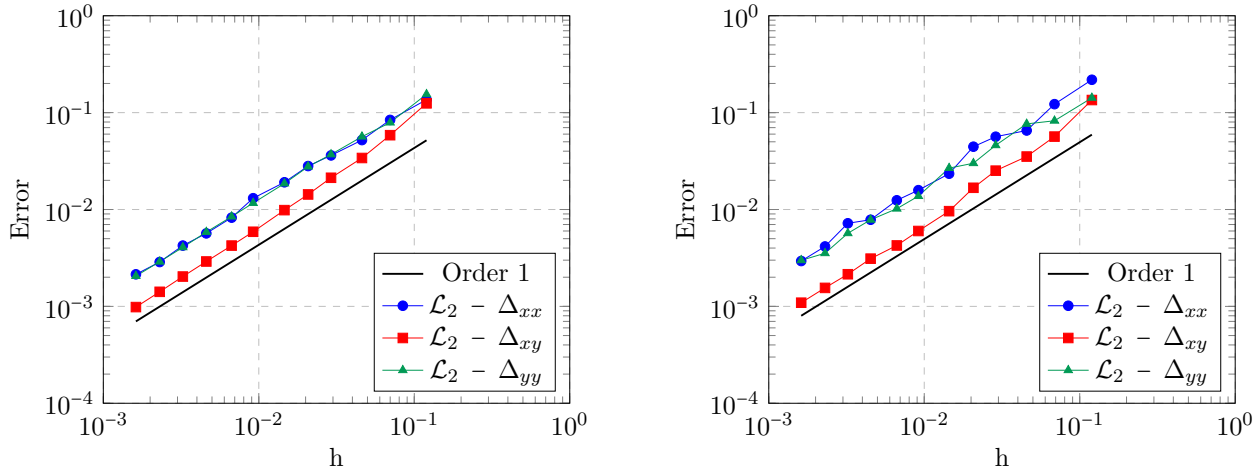
finite volume scheme, based on the V4 scheme, capable of carrying the mean value per dual cell.



(a) Regular

(b) Disturbed

Figure 4.15: $\mathcal{L}_2$ error on boundary vertices for the nodal gradient on regular mesh (left) and disturbed mesh (right) - O3 deconvolution



(a) Regular

(b) Disturbed

Figure 4.16: $\mathcal{L}_2$ error on boundary vertices for the nodal hessian on regular mesh (left) and disturbed mesh (right) - O3 deconvolution

4.4 Application of the deconvolution

In this section, we detail the strategy for constructing a vertex-centred finite volume scheme for transporting the mean value. To do this, we use the V4 scheme as a basis, but another numerical scheme could also be used. The objective is to construct a scheme of order 2, given that the V4 scheme is of order 2 on any mesh when the value at a node is transported. Consequently, we can rely exclusively on first-order deconvolution, as it enables us to derive a second-order nodal value and a first-order nodal gradient from the available mean values. First, we will present the corrections required to construct the V4 scheme. Then, we will present the treatment of boundary conditions.

4.4.1 MUSCL reconstruction

The V4 scheme is a numerical scheme based on a MUSCL reconstruction in which the gradient is defined as a blending gradient between upwind and centred contributions. MUSCL reconstruction allows the calculation of an extrapolated value at point M_{ij}, which is the midpoint of edge ij, from the value $\mathbf{W}_i$ (or $\mathbf{W}_j$) and a blending gradient. The calculation of this high-order gradient is derived from the nodal values. However, in this particular instance, only the averaged values $\bar{\mathbf{W}}_i$ are known. We then rewrite our operators with the available averaged values. The extrapolated values, calculated from the available averaged values, can be written as follows:

$$\begin{cases} \bar{\mathbf{W}}_{ij} = \bar{\mathbf{W}}_i + \frac{1}{2}(\overrightarrow{r_{ij}} \cdot \overrightarrow{\nabla \bar{\mathbf{W}}_i^{\beta}}), \\ \bar{\mathbf{W}}_{ji} = \bar{\mathbf{W}}_j - \frac{1}{2}(\overrightarrow{r_{ij}} \cdot \overrightarrow{\nabla \bar{\mathbf{W}}_j^{\beta}}), \end{cases} \tag{4.15}$$

where $\overrightarrow{\nabla \bar{\mathbf{W}}_i^{\beta}}$ (and $\overrightarrow{\nabla \bar{\mathbf{W}}_j^{\beta}}$) are defined based on the available averaged values. The notation $\bar{\mathbf{W}}_{ij}$ serves to highlight the fact that the extrapolated values are calculated from averaged values, yet are mathematically meaningless. The high-order gradient is a linear combination of centred and upwind/downwind gradients, which can be written as follows.

$$\overrightarrow{\nabla \bar{\mathbf{W}}_{ij}^{C}} \cdot \overrightarrow{ij} = \bar{\mathbf{W}}_j - \bar{\mathbf{W}}_i \quad \text{and} \quad (\nabla \mathbf{W})|_K = \sum_{P_k \in K} \bar{\mathbf{W}}_k \nabla \phi_k|_K.$$

As shown in Figure 4.2, the finite element gradient thus defined is of order 0. It is therefore not possible to use the operators defined in this way. The idea is to use the results of the deconvolution to correct the various operators. So we can write:

$$\overrightarrow{\nabla \bar{\mathbf{W}}_{ij}^{C,\text{corr}}} \cdot \overrightarrow{ij} = \mathbf{W}_j - \mathbf{W}_i \quad \text{and} \quad (\nabla \mathbf{W})|_K^{\text{corr}} = \sum_{P_k \in K} \mathbf{W}_k \nabla \phi_k|_K.$$

Finally, the MUSCL reconstruction can be corrected by calculating the reconstructed state $\mathbf{W}_{ij}$ from the nodal values.

$$\begin{cases} \mathbf{W}_{ij} = \mathbf{W}_i + \frac{1}{2}(\overrightarrow{r_{ij}} \cdot \overrightarrow{\nabla \mathbf{W}_i^{\text{corr}}}), \\ \mathbf{W}_{ji} = \mathbf{W}_j - \frac{1}{2}(\overrightarrow{r_{ij}} \cdot \overrightarrow{\nabla \mathbf{W}_j^{\text{corr}}}). \end{cases} \tag{4.16}$$

4.4.2 Boundary state reconstruction

The results of the deconvolution process are then utilised to address boundary conditions. On the one hand, based on the available averaged fields, the high-order gradient at the edge of the domain can be defined as follows:

$$\overrightarrow{\nabla \bar{\mathbf{W}}_{ij}^{U}} \approx 2\overrightarrow{\nabla \bar{\mathbf{W}}_{P_i}} - \overrightarrow{\nabla \bar{\mathbf{W}}_{ij}^{C}}. \tag{4.17}$$

As with the other operators, the gradient thus defined is of order 0. Similarly, the gradient is corrected using the results of the deconvolution. Consequently, the following operator can be formulated:

$$\overrightarrow{\nabla \mathbf{W}_{ij}^{U}} \approx 2\overrightarrow{\nabla \mathbf{W}_{P_i}} - \overrightarrow{\nabla \mathbf{W}_{ij}^{C}}. \tag{4.18}$$

On the other hand, we use the results of the deconvolution to improve the processing of boundary conditions. As we saw in Chapter 3, boundary conditions are processed weakly. We therefore calculate an analytical flux per facet, at point x_{ij}, as shown in Figure 4.7. For the V4 scheme

carrying the point value, the value at node $\mathbf{W}_i$ is used, since we do not know the value at point $\mathbf{x}_{ij}$. Using the results of the deconvolution, we can extrapolate a value of order 2 in the middle of the facet. More precisely, we can write the following expression:

$$\mathbf{W}(\mathbf{x}_{ij}) = \mathbf{W}_i + \frac{\partial \mathbf{W}}{\partial x}|_i (x_{ij} - x_i) + \frac{\partial \mathbf{W}}{\partial y}|_i (y_{ij} - y_i) + O(h^2).$$

Based on the results of the deconvolution, we can, on the one hand, construct the high-order gradient and thus the MUSCL reconstruction defining the V4 scheme. On the other hand, we can improve the handling of boundary conditions by extrapolating a value at the centre of each facet. With the V4 scheme carrying the averaged quantity now fully defined, we present the numerical results for the advection equation.

4.5 Numerical results : 2D scalar transport test case

To plot the convergence results, we used again the following definition of the mesh size:

$$h_{eff} = \frac{1}{N} \sum_i |C_i|^{\frac{1}{d}}, \tag{4.19}$$

with N the number of vertices, $|C_i|$ the measure of the dual cell C_i and d the dimension of the problem. In our case, d is fixed to 2. The V4 scheme, as we define it, carries the averaged value. Therefore, we use the integral definition to calculate the $\mathcal{L}_p$ error. The expressions for the $\mathcal{L}_p$ error are given in (4.20):

$$\|w - w^{\text{ref}}\|_{\mathcal{L}_p} = \Big(\int_\Omega (w - w^{\text{ref}})^p dV \Big)^{\frac{1}{p}} \approx \Big(\sum_i |C_i| (w_i - w_i^{\text{ref}})^p \Big)^{\frac{1}{p}}; \quad \|w - w^{\text{ref}}\|_{\mathcal{L}_\infty} = \max_i |w_i - w_i^{\text{ref}}|, \tag{4.20}$$

with the reference value defined as follows:

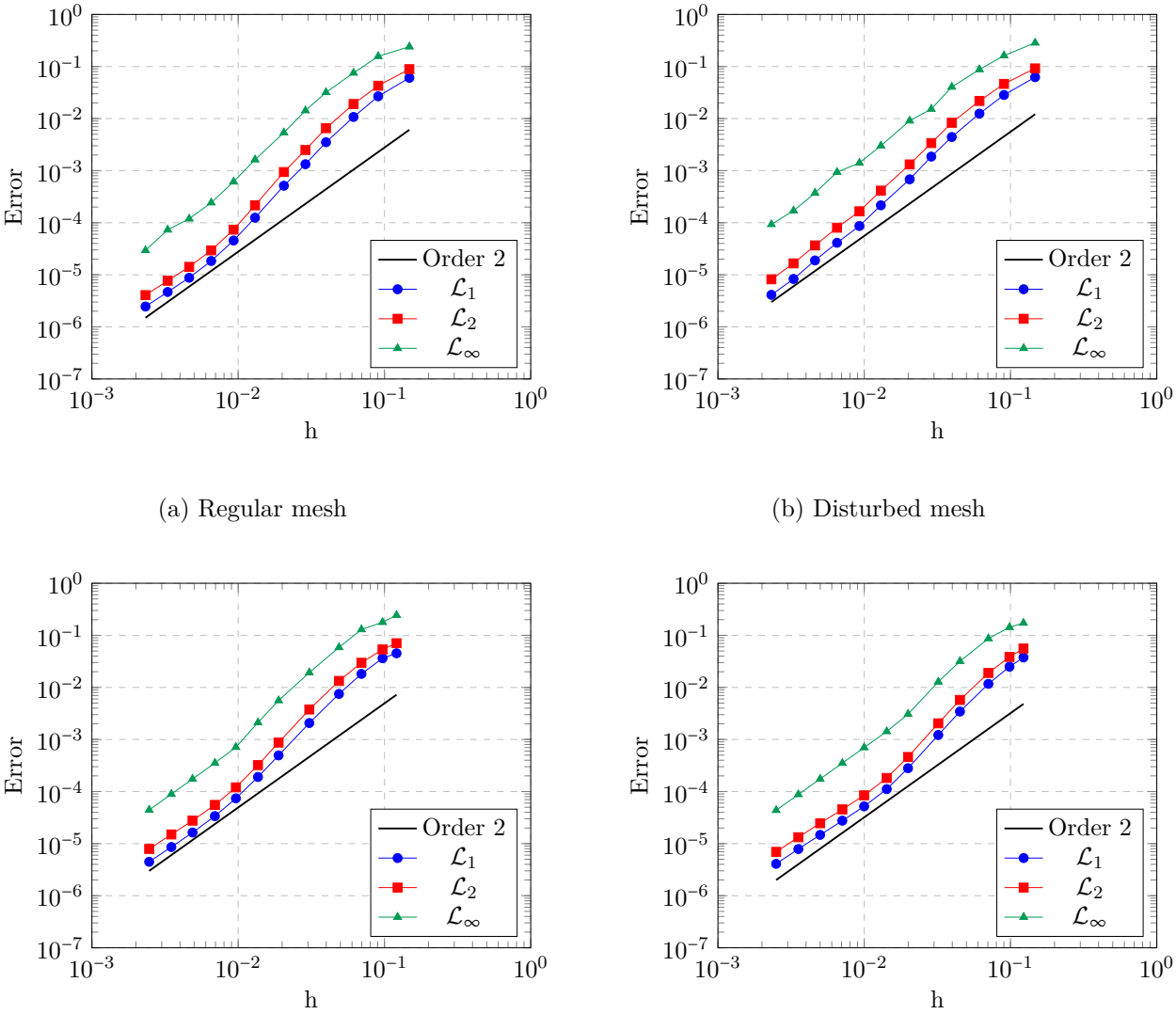
$$w_i^{\text{ref}} = \frac{1}{|C_i|} \int_{C_i} w_{\text{exact}}(x, y) d\Omega, \quad \forall i \in \{1, ..., N\}.$$

Remark :

This change in the choice of the norm is of strong importance. In this chapter, we switch to standard norms for integral formulations.

4.5.1 Results on standard meshes

To obtain the results presented in this section, we still use an explicit time scheme, namely the RK3 scheme of Shu and Osher [64]. Figure 4.17 shows the evolution of the error norm as a function of mesh size. The numerical solution is compared with the averaged value per dual cell for all points in the mesh. Results show that, in the sense of finite volumes, the scheme is of order 2, *i.e.* in comparison with the averaged value per dual cell. It can be seen that all the considered norms converge to order 2. Therefore, we can conclude that the scheme is indeed of order 2 for all cells in the mesh, meaning that the boundary conditions are well posed.



(a) Regular mesh

(b) Disturbed mesh

(c) Union jack mesh

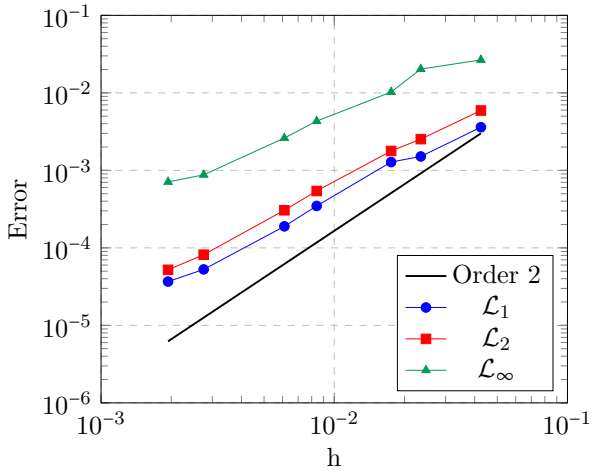
(d) Friedrich-Keller mesh

Figure 4.17: $\mathcal{L}_1$, $\mathcal{L}_2$ and $\mathcal{L}_\infty$ error on regular, perturbed, Friedrich-Keller and Union Jack meshes, clockwise from top left - finite volume V4.

In this section, we have seen that the V4 scheme with deconvolution is a second-order scheme in the sense of finite volumes. To validate this, we used four types of mesh, including randomly perturbed meshes. However, none of these meshes contained strongly anisotropic elements. To evaluate the behaviour of the V4 scheme on a strongly anisotropic mesh, we employed the adaptive mesh hierarchy presented in Section 3.2.3.

4.5.2 Results on adapted meshes

In the following analysis, the focus is on the results obtained from the finite volume V4 scheme when used on adapted meshes. As illustrated in Figure 4.18, the discretisation error norm evolves as a function of mesh size, for all mesh cells. The results show that the discretisation error norm converges to an order of magnitude slightly lower than 2, which is similar to the results obtained for the V4 edge-based scheme where a decrease in the order of convergence was also observed. By comparing the results obtained for the V4 finite volume scheme with those obtained for the V4 edge-based scheme, we can conclude that the V4 scheme retains its precision and robustness properties on highly anisotropic meshes in its finite volume formulation.



(a) Adapted mesh

Figure 4.18: $\mathcal{L}_1$, $\mathcal{L}_2$ and $\mathcal{L}_\infty$ discretization error on adapted mesh - finite volume V4.

So far, we have studied the global error of our numerical scheme. For each cell, we calculate the error relative to the exact solution and then calculate the error norm for all the cells that make up the study domain. While this approach enables us to determine the global order of our numerical scheme, it does not allow us to measure the reconstruction order locally. In the following section, we therefore present a study of the local error of our numerical scheme.

4.6 Error analysis

The objective of this study is to construct a second-order numerical scheme that can effectively transport the mean value. Following a detailed exposition of the construction, it has been demonstrated that the scheme, as defined, is a second-order scheme in terms of discretization error. The objective of this study is to demonstrate that the aforementioned scheme is locally second-order. This study is based on the approach presented by Tarsia-Morisco *et al.* in [105]. To do this, we start again from the integrated Taylor series expansions on each cell, the expression of which is given by 4.21:

$$\begin{cases} \bar{\mathbf{W}}_i &= \mathbf{W}|_i + \dfrac{\partial \mathbf{W}}{\partial x}|_i \overline{x}_i + \dfrac{\partial \mathbf{W}}{\partial y}|_i \overline{y}_i + \dfrac{\partial^2 \mathbf{W}}{\partial x^2}|_i \dfrac{\overline{x^2}_i}{2} + \dfrac{\partial^2 \mathbf{W}}{\partial x \partial y}|_i \overline{xy}_i + \dfrac{\partial^2 \mathbf{W}}{\partial y^2}|_i \dfrac{\overline{y^2}_i}{2} + O(h^3), \\ \bar{\mathbf{W}}_j &= \mathbf{W}|_i + \dfrac{\partial \mathbf{W}}{\partial x}|_i \widehat{x}_{ij} + \dfrac{\partial \mathbf{W}}{\partial y}|_i \widehat{y}_{ij} + \dfrac{\partial^2 \mathbf{W}}{\partial x^2}|_i \dfrac{\widehat{x^2}_{ij}}{2} + \dfrac{\partial^2 \mathbf{W}}{\partial x \partial y}|_i \widehat{xy}_{ij} + \dfrac{\partial^2 \mathbf{W}}{\partial y^2}|_i \dfrac{\widehat{y^2}_{ij}}{2} + O(h^3). \end{cases} \tag{4.21}$$

In order to measure the order of our numerical scheme locally, Taylor expansions are injected into the numerical scheme. In the case of a Cartesian and uniform mesh, this study can be conducted manually with relative ease. In the context of any 2D mesh, the undertaking of this study can be arduous, and in some cases, unfeasible. However, an alternative strategy involves the numerical implementation of the aforementioned exercise. Firstly, the method and the results for a numerical scheme of order 1 will be presented.

First order analysis. In this work, the convective flux is calculated using an upwind scheme.

In practice, for two cells C_i and C_j, we have:

$$\int_{f_{ij}} F(\mathbf{W}) \cdot \overrightarrow{n}_{f_{ij}} ds \approx \mathcal{A}_{ij}(\bar{\mathbf{W}}_i, \bar{\mathbf{W}}_j) = S_{ij} \begin{cases} \mathbf{a}_{ij} \cdot \overrightarrow{n}_{ij} \bar{\mathbf{W}}_i & \text{if} \quad \mathbf{a}_{ij} \cdot \overrightarrow{n}_{ij} > 0, \\ \mathbf{a}_{ij} \cdot \overrightarrow{n}_{ij} \bar{\mathbf{W}}_j & \text{otherwise,} \end{cases} \tag{4.22}$$

where S_{ij} is the area of f_{ij}, $\mathbf{a}_{ij}$ is the velocity at the centre of edge ij and $\bar{\mathbf{W}}_i$ is the averaged quantitie over the control volumes. The flux balance, for a given cell i can the be written as:

$$\int_{\partial C_i} F(\mathbf{W}) \cdot \overrightarrow{n}_{C_i} ds \approx \mathcal{A}_i = \sum_{f_j \in \nu_i} \mathcal{A}_{ij}(\bar{\mathbf{W}}_i, \bar{\mathbf{W}}_j).$$

The first step is to analyse the flux balance error for a given dual cell. By injecting the Taylor series expansion (4.21) into the operator (4.22), we can easily compute the different terms of the approximation.

- For the constant term:

$$\mathcal{A}_i = \sum_{f_j \in \nu_i} \mathcal{A}_{ij}(\mathbf{W}_i, \mathbf{W}_i)$$

- For the first order derivative in x:

$$\mathcal{A}_i = \sum_{f_j \in \nu_i} \mathcal{A}_{ij}(\overline{x}_i, \widehat{x}_{ij})$$

- For the first order derivative in y:

$$\mathcal{A}_i = \sum_{f_j \in \nu_i} \mathcal{A}_{ij}(\overline{y}_i, \widehat{y}_{ij})$$

- We can then obtain the successive derivative terms using $\overline{x^n y^m}_i$ and $\widehat{x^n y^m}_{ij}$.

For any cell i in the domain, we can directly state that the constant term is zero by ciculation on the edge of the cell. We can therefore state directly that the flux balance is, a least, of order 1 in this case. For the rest, it is necessary to look at the numerical results. We first look at the numerical results for the Friedrich-Keller meshes. Figure 4.20 shows us that the upwind scheme for the advection equation is, unsurprisingly, of order 1 for all points in the mesh. Figure 4.21 shows us the results for the first-order derivative. We can say that a point i is regular if all neighbours j of i satisfy $\overline{x}_j = 0$. We observe that for regular points on the mesh, the upwind scheme is of order 2. For points that are not regular, the scheme is only of order 1. Figure 4.22 shows us that the upwind scheme is perfect to compute the constant term of the truncation error on a perturbed mesh. However, Figure 4.26 shows us that this scheme is never of order 2 on any mesh. The results show us that the upwind scheme is of order 1 on any type of mesh.

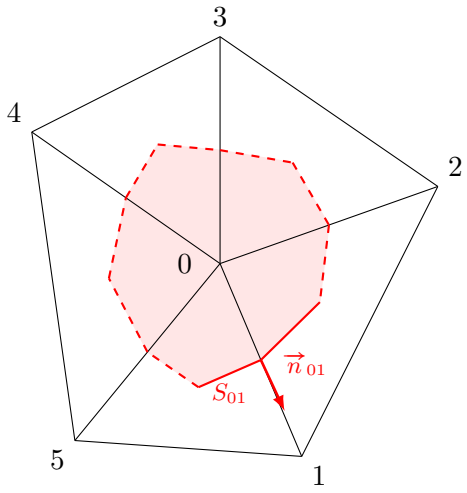


Figure 4.19: Dual cell - first order analysis

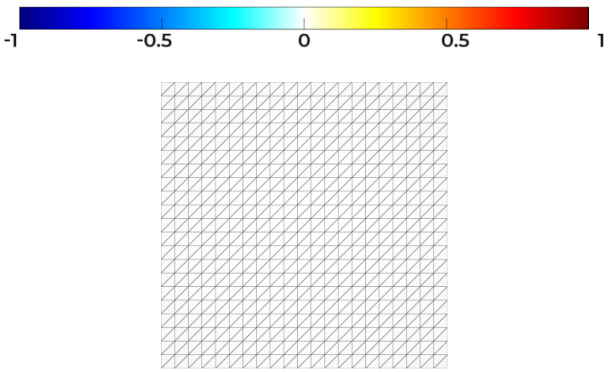


Figure 4.20: Friedrich-Keller mesh - constant term - no correction

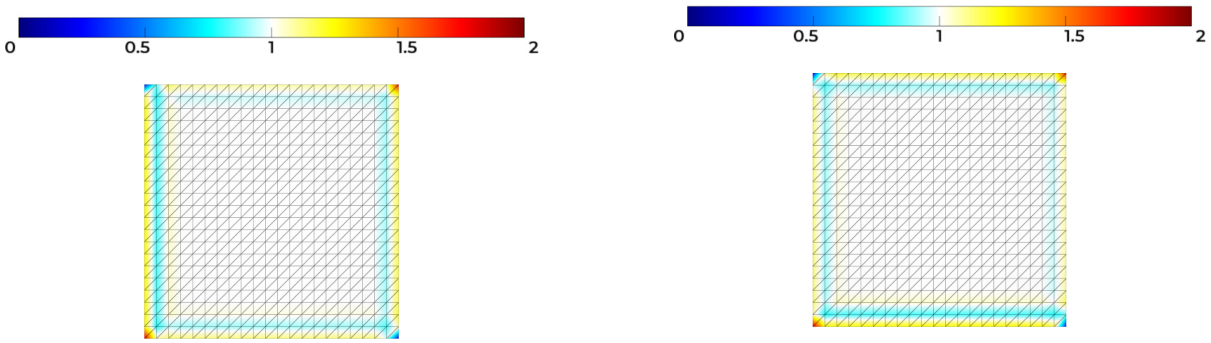


Figure 4.21: Friedrich-Keller mesh - first order derivatives term. $|C_i|\partial_x w$ field (left), $|C_i|\partial_y w$ field (right) - no correction

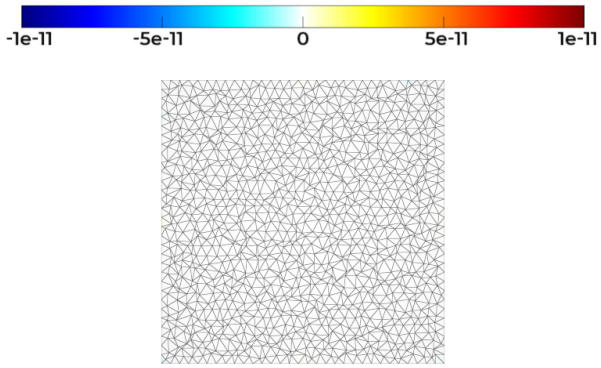


Figure 4.22: Disturbed mesh - constant term - no correction

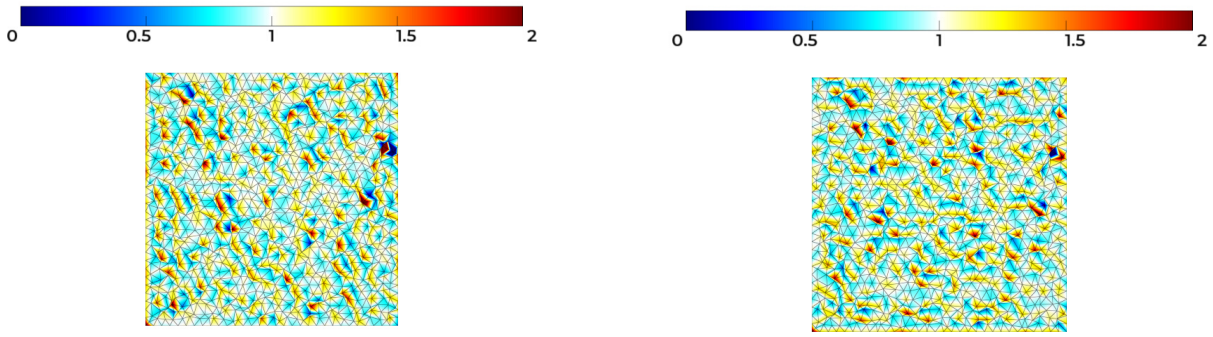


Figure 4.23: Disturbed mesh - first order derivatives term. $|C_i|\partial_x w$ field (left), $|C_i|\partial_y w$ field (right) - no correction

In the same way as for the first order, we carry out an order study on the second order approximation.

Second order analysis. The MUSCL approach consists of constructing interpolated values from the mean values available. The aim is to increase the accuracy of the numerical flux and obtain a second-order numerical scheme. We use the V4 scheme with a finite volume definition, *i.e.* the different operators are calculated using the results of the deconvolution.
To calculate the numerical flux between cells i and j, the mean value $\bar{\mathbf{W}}_i$ and $\bar{\mathbf{W}}_j$ are replaced by the extrapolated value $\mathbf{W}_{ij}$ or $\mathbf{W}_{ji}$. The notations used are given in figure 4.24. We construct the extrapolated value $\mathbf{W}_{ij}$ as follows:

$$\begin{aligned}
\mathbf{W}_{ij} &= \bar{\mathbf{W}}_i + \frac{1}{2}\nabla\mathbf{W}_i^{\beta}\cdot\overrightarrow{ij} \\
&= \bar{\mathbf{W}}_i + \tfrac{1}{2}\big[(1-\beta)(\bar{\mathbf{W}}_j - \bar{\mathbf{W}}_i) + \beta\sum_{a\in T_{ij}}\bar{\mathbf{W}}_a\nabla\phi_a\cdot\overrightarrow{ij}\big] \\
&= \bar{\mathbf{W}}_i\big[\frac{1}{2} + \frac{\beta}{2}(1+\overrightarrow{\nabla\phi}_i\cdot\overrightarrow{ij})\big] + \bar{\mathbf{W}}_j(\frac{1-\beta}{2}) + \bar{\mathbf{W}}_b(\frac{\beta}{2}\overrightarrow{\nabla\phi}_b\cdot\overrightarrow{ij}) + \bar{\mathbf{W}}_c(\frac{\beta}{2}\overrightarrow{\nabla\phi}_c\cdot\overrightarrow{ij}),
\end{aligned} \tag{4.23}$$

where ϕ_i, ϕ_a, and ϕ_b are the following shape functions:

$$\begin{cases} \phi_i(x,y) = \dfrac{(x-x_c)(y_b-y_c)-(y-y_c)(x_b-x_c)}{2|T_{ij}|}, \\ \phi_b(x,y) = \dfrac{(x-x_c)(y_c-y_i)-(y-y_c)(x_c-x_i)}{2|T_{ij}|}, \\ \phi_c(x,y) = \dfrac{(x-x_i)(y_i-y_b)-(y-y_i)(x_i-x_b)}{2|T_{ij}|}, \end{cases}$$

with $|T_{ij}|$ the measure of triangle T_{ij}. Following the same idea, we construct the extrapolated value $\mathbf{W}_{ji}$:

$$\mathbf{W}_{ji} = \bar{\mathbf{W}}_j\Big[\frac{1}{2}+\frac{\beta}{2}(1-\overrightarrow{\nabla\phi_j}\cdot\overrightarrow{ij})\Big]+\bar{\mathbf{W}}_i(\frac{1-\beta}{2})-\bar{\mathbf{W}}_k(\frac{\beta}{2}\overrightarrow{\nabla\phi_k}\cdot\overrightarrow{ij})-\bar{\mathbf{W}}_l(\frac{\beta}{2}\overrightarrow{\nabla\phi_l}\cdot\overrightarrow{ij}). \quad (4.24)$$

We then define the following operators for the second-order convective flux:

$$\int_{f_j} F(\mathbf{W})\cdot\vec{n}_{f_j}ds \approx \mathcal{A}_{ij}(\bar{\mathbf{W}}_i,\bar{\mathbf{W}}_j,\bar{\mathbf{W}}_b,\bar{\mathbf{W}}_c,\bar{\mathbf{W}}_k,\bar{\mathbf{W}}_l) = S_{ij}\begin{cases} \mathbf{a}_{ij}\cdot\vec{n}_{ij}\mathbf{W}_{ij} & \text{if } \mathbf{a}_{ij}\cdot\vec{n}_{ij}>0 \\ \mathbf{a}_{ij}\cdot\vec{n}_{ij}\mathbf{W}_{ji} & \text{otherwise.}\end{cases} \quad (4.25)$$

As can be seen in equation (4.23), we can write the extrapolated values $\mathbf{W}_{ij}$ and $\mathbf{W}_{ji}$ as a linear combination of the averaged states of the problem. Since the transport equation is a linear equation, the numerical scheme depends linearly on the states of the problem.

As for order 1, we inject the Taylor series expansion 4.5 into the operator 4.25 and we can therefore easily compute the different terms of the approximation.

- For the constant term:

$$\mathcal{A}_i = \sum_{f_j\in\nu_i}\mathcal{A}_{ij}(\mathbf{W}_i,\mathbf{W}_i,\mathbf{W}_i,\mathbf{W}_i,\mathbf{W}_i,\mathbf{W}_i)$$

- For the first order derivative in x:

$$\mathcal{A}_i = \sum_{f_j\in\nu_i}\mathcal{A}_{ij}(\overline{x}_i,\widehat{x}_{ij},\widehat{x}_{ib},\widehat{x}_{ic},\widehat{x}_{ik},\widehat{x}_{il})$$

- For the first order derivative in y:

$$\mathcal{A}_i = \sum_{f_j\in\nu_i}\mathcal{A}_{ij}(\overline{y}_i,\widehat{y}_{ij},\widehat{y}_{ib},\widehat{y}_{ic},\widehat{y}_{ik},\widehat{y}_{il})$$

- We can then obtain the successive derivative terms using $\overline{x^n y^m}_i$ and $\widehat{x^n y^m}_{i\alpha}$, for $\alpha\in\{b,c,k,l\}$.

where b, c, k, and l are the indices of the vertices of the upwind and downwind triangles of the egde ij.

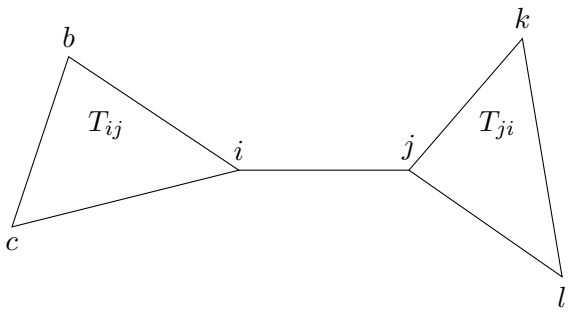


Figure 4.24: V4 scheme stencil and notations for advection

Following the definition of the various operators for order 2, the subsequent analysis will focus on the numerical results. The focus of this study is the results for first-order derivative terms. As demonstrated in Figure 4.25, the V4 scheme is of order 2 on the Friedrich-Keller mesh. Indeed, we are able to reproduce the theoretical results for both gradient terms. The conclusion remains consistent for perturbed meshes, as Figure 4.26 demonstrates that the V4 scheme is also of order 2 on this type of mesh. The computations also highlight a default of the scheme for corner nodes, associated with degenerated dual cell. It is important to note this flawness and keep it in mind for the next chapter.

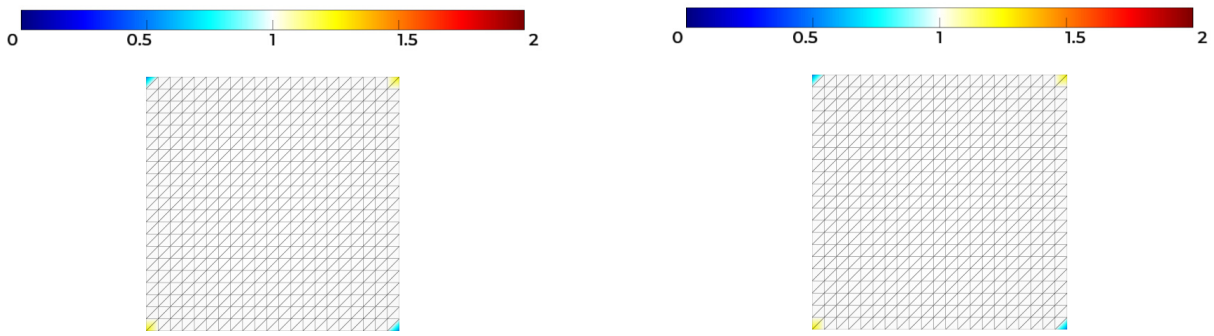


Figure 4.25: Friedrich-Keller mesh - first order derivatives term. $|C_i|\partial_x w$ field (left), $|C_i|\partial_y w$ field (right) - O2 deconvolution

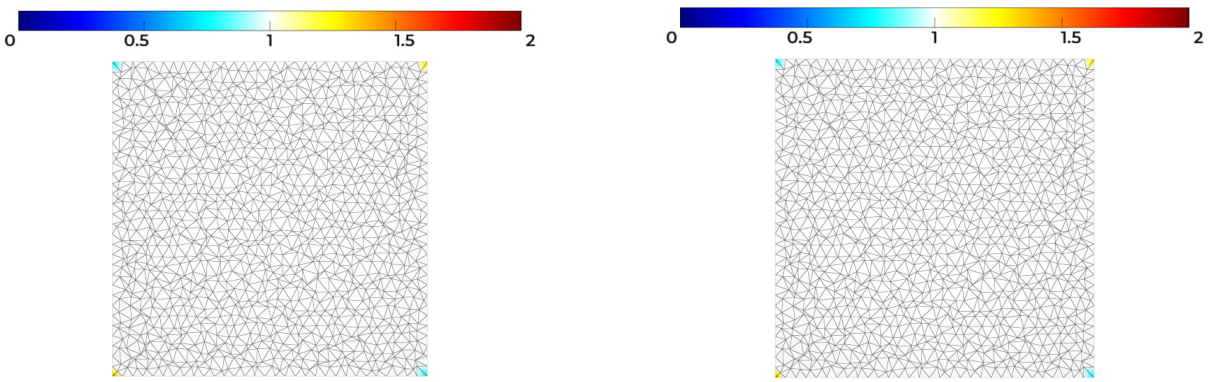


Figure 4.26: Disturbed mesh - first order derivatives term. $|C_i|\partial_x w$ field (left), $|C_i|\partial_y w$ field (right) - O2 deconvolution

In this section, we conducted a local analysis of the error in our numerical scheme. We showed

that the upwind scheme is of order 1 on all types of meshes. Furthermore, we have shown that V4 reconstruction coupled with deconvolution allows us to construct a second-order numerical scheme that carries the mean value. This local analysis supports the conclusions of the discretization error analysis, namely that the scheme thus defined is second-order on all types of mesh.

4.7 Extension to Euler equation

In this section, we extend the previously introduced method to Euler equations. So far, we have dealt with the scalar advection equation. This equation is linear. Thus, we transport the mean value and then, using deconvolution operators, we reconstruct a nodal solution from the available mean values. For Euler equations, it is the conservative quantities $\mathbf{W}_i = (\rho, \rho\mathbf{u}, \rho E)^\top$ that are transported. Or, for reasons of stability of the numerical scheme, the MUSCL reconstruction or the computation of the numerical flux is done from the primitive variables $\mathbf{U}_i = (\rho, \mathbf{u}, P \text{ or } T)^\top$. It is therefore necessary, from the conservative variables, to determine the primitive variables. For the case of the V4 scheme transporting the nodal value, we can easily compute the primitive variables from the conservative variables without loss of precision. For the scheme carrying the mean value, in order to ensure stability, it is necessary to reconstruct the nodal values of the primitive variables from the mean values of the primitive variables. However, it should be noted that only the mean values of the conservative variables are known. In order to apply deconvolution operators to primitive variables, it is first necessary to calculate an approximation of the mean values of said variables. In order to achieve this, we introduce the vector $\tilde{\mathbf{U}}$, which is defined as follows.

$$\tilde{\mathbf{U}} = \begin{pmatrix} \tilde{\rho} \\ \tilde{\mathbf{u}} \\ \tilde{P} \text{ or } \tilde{T} \end{pmatrix}$$

where :

$$\tilde{\mathbf{U}} = \begin{cases} \tilde{\rho} &= \overline{\rho} \\ \tilde{\mathbf{u}} &= \dfrac{\overline{\rho\mathbf{u}}}{\overline{\rho}} \\ \tilde{P} &= (\gamma - 1)(\overline{\rho E} - \frac{1}{2}\dfrac{|\overline{\rho\mathbf{u}}|^2}{\overline{\rho}}) \\ \tilde{T} &= \dfrac{\tilde{P}}{R\overline{\rho}} \end{cases} \tag{4.26}$$

The vector $\tilde{\mathbf{U}}$ thus defined is a first-order approximation of the mean value of the primitive variables. However, for deconvolution, we want to determine, for a given degree of precision n, the mean value of the primitive variables in each dual cell. To do this, we need to evaluate $\Delta\mathbf{U}$ such that.

$$\bar{\mathbf{U}}_i = \tilde{\mathbf{U}}_i + \Delta\mathbf{U}_i + O(h^n)$$

The approach presented here is largely inspired by the k-exact scheme studied by Pont *et al.* in [34]. However, it should be noted that their work is part of a cell-centred finite volume formalism. Consequently, the first-order volume moments are invariably zero, and the vector $\tilde{\mathbf{U}}$ is, at least, of order two. In this instance, the first-order volume moments are not necessarily zero. Consequently, the formulas proposed in this study and those of Pont differ.

Since we seek to reconstruct the mean values of the primitive variables for any degree of precision, we introduce new notations. On the one hand, for any quantity q, we denote by $D_q^{(m)}|_i$ the tensor of the derivatives of order k of at point $\mathbf{x}_i$. In addition, we give in (4.27) a tensor formulation of the volume moments.

$$\mathcal{M}_i^{(m)} = \frac{1}{|C_i|}\int_{C_i}(\mathbf{x}-\mathbf{x}_i)^{\otimes m}d\Omega \tag{4.27}$$

where, for any vector $\mathbf{v}$:

$$\mathbf{v}^{\otimes m} = \underbrace{\mathbf{v}\otimes\mathbf{v}\otimes\ldots\otimes\mathbf{v}}_{\text{m times}}$$

Thus, using the different notations introduced, we can generalize the link between the mean value and the averaged value for any quantity q.

$$\bar{q}_i - q_i = \sum_{m=1}^{n-1}\frac{1}{m!}\mathcal{M}_i^{(m)} : D_q^{(m)}|_i + O(h^n) \tag{4.28}$$

Now that the generalities have been introduced, we detail, for each primitive variable, the construction of an average quantity of arbitrary degree n. Since density is both a primitive and conservative quantity, there is no special treatment to be provided. We therefore detail the cases of speed, pressure and temperature.

4.7.1 Velocity

In this section, the objective is to determine the mean value of arbitrary degree n of the velocity vector on each dual cell. First, we can write the first order approximation.

$$\tilde{\mathbf{u}}_i = \frac{\overline{\rho\mathbf{u}}_i}{\overline{\rho}_i} = \mathbf{u}_i + O(h). \tag{4.29}$$

By introducing $\mathbf{u}_i$ into the previous relationship and after some development, we have

$$\begin{aligned}\tilde{\mathbf{u}}_i &= \mathbf{u}_i + \frac{\overline{\rho\mathbf{u}}_i - \overline{\rho}_i\mathbf{u}_i}{\overline{\rho}_i}\\ &= \frac{\rho_i(\overline{\mathbf{u}}_i-\mathbf{u}_i)}{\overline{\rho}_i} + \mathbf{u}_i + \frac{1}{\overline{\rho}_i}\sum_{m=2}^{n-1}\frac{1}{m!}\mathcal{M}_i^{(m)} : \Big(\sum_{l=1}^{m-1}\begin{pmatrix} m \\ l\end{pmatrix} D_\rho^{(m-l)}|_i \otimes D_{\mathbf{u}}^{(l)}|_i\Big) + O(h^n).\end{aligned} \tag{4.30}$$

Finally, we have the following correction term for calculating the averaged velocity per dual cell:

$$\Delta\mathbf{u}_i = \tfrac{1}{\overline{\rho}}\textstyle\sum_{m=2}^{n-1}\frac{1}{m!}\Big[\Big(\mathcal{M}_i^{(m-1)}\Big)^2 - \mathcal{M}_i^{(m)}\Big]\sum_{l=1}^{m-1}\begin{pmatrix} m \\ l\end{pmatrix} D_\rho^{(m-l)}|_i \otimes D_{\mathbf{u}}^{(l)}|_i + O(h^n). \tag{4.31}$$

At order 2, expression (4.31) gives $\Delta\mathbf{u}_i = 0 + O(h^2)$. Therefore, it is not necessary to calculate an additional term to construct the averaged value of the velocity per dual cell to order 2. Consequently, we are able to write.

$$\overline{\mathbf{u}}_i = \frac{\overline{\rho\mathbf{u}}_i}{\overline{\rho}_i} + \Delta\mathbf{u}_i = \tilde{\mathbf{u}}_i + O(h^2). \tag{4.32}$$

4.7.2 Pressure

As with velocity, we seek to construct the averaged value of pressure per dual cell. We can directly write the first-order approximation:

$$\tilde{P}_i = (\gamma-1)\Big(\overline{\rho E}_i - \frac{1}{2}\frac{|\overline{\rho u}_i|^2}{\overline{\rho}_i}\Big) = P_i + O(h). \tag{4.33}$$

As for velocity, we introduce P_i in the previous expression. This gives the following development:

$$\begin{aligned}
\tilde{P}_i &= P_i + (\gamma-1)(\overline{\rho E} - \rho E) - (\gamma-1)\tfrac{1}{2}\Big(\tfrac{|\overline{\rho u}_i|^2}{\overline{\rho}_i} - \tfrac{|\rho u_i|^2}{\rho_i}\Big) \\
&= P_i + (\gamma-1)\frac{\rho_i|\mathbf{u}_i|^2}{2\overline{\rho}_i}\sum_{m=1}^{n-1}\frac{1}{m!}\mathcal{M}_i^{(m)} : D_\rho^{(m)}|_i \\
&+ (\gamma-1)\sum_{m=1}^{n-1}\frac{1}{m!}\mathcal{M}_i^{(m)} : \Big(D_{\rho E}^{(m)}|_i - \frac{1}{2\overline{\rho}_i}D_{|\rho u|^2}^{(m)}|_i\Big) + O(h^n).
\end{aligned} \tag{4.34}$$

On the one hand, expression 4.34 can be simplified by the following equality.

$$\begin{aligned}
\textstyle\sum_{m=1}^{n-1}\frac{1}{m!}\mathcal{M}_i^{(m)} : D_{|\rho u|^2}^{(m)}|_i &= \rho_i\sum_{m=1}^{n-1}\frac{1}{m!}\mathcal{M}_i^{(m)} : D_{\rho|u|^2}^{(m)}|_i + \rho_i|\mathbf{u}_i|^2\sum_{m=1}^{n-1}\frac{1}{m!}\mathcal{M}_i^{(m)} : D_\rho^{(m)}|_i \\
&+ \textstyle\sum_{m=2}^{n-1}\frac{1}{m!}\mathcal{M}_i^{(m)} : \sum_{l=1}^{m-1}\begin{pmatrix} m \\ l \end{pmatrix} D_\rho^{(m-l)}|_i \otimes D_{\rho|u|^2}^{(l)}|_i + O(h^n).
\end{aligned}$$

On the other hand, using the equation of states $P = \rho RT$, the relation $C_v(\gamma-1) = R$ and after some developments, we obtain the following formulation:

$$-(\gamma-1)D_{\rho E}^{(m)} + \frac{(\gamma-1)}{2\overline{\rho}}\rho D_{\rho|\mathbf{u}|^2}^{(m)} = -D_P^{(m)} - \frac{(\gamma-1)}{2\overline{\rho}}\Big[\sum_{l=0}^{m}\begin{pmatrix} m \\ l \end{pmatrix} D_\rho^{(m-l)} \otimes D_{|\mathbf{u}|^2}^{(l)}\Big]\Big[\sum_{m=1}^{n-1}\frac{1}{m!}\mathcal{M}^{(m)} : D_\rho^{(m)}|\Big].$$

Finally, we have the following correction term for computing the averaged pressure per dual cell:

$$\Delta P_i = -\frac{(\gamma-1)}{2\overline{\rho}}\sum_{m=2}^{n-1}\frac{1}{m!}\Big[\Big(\mathcal{M}_i^{(m-1)}\Big)^2 - \mathcal{M}_i^{(m)}\Big]\sum_{l=1}^{m-1}\begin{pmatrix} m \\ l \end{pmatrix} D_\rho^{(m-l)}|_i \otimes D_{\rho|\mathbf{u}|^2}^{(l)}|_i + O(h^n). \tag{4.35}$$

The expression (4.35) provides the correction term for the calculation of the mean value for pressure. With regard to velocity, a zero correction term is identified for a reconstruction to order 2. More precisely, we have $\Delta P_i = 0 + O(h^2)$. Consequently, it is not necessary to calculate an additional correction term, of order 2, for constructing the mean value of pressure per dual cell. Consequently, we are able to write:

$$\overline{P}_i = (\gamma-1)\Big(\overline{\rho E} - \frac{1}{2}\frac{|\overline{\rho u}_i|^2}{\overline{\rho}_i}\Big) + \Delta P_i = \tilde{P}_i + O(h^2). \tag{4.36}$$

4.7.3 Temperature

The objective of this section is to determine the mean temperature value for each dual cell. Firstly, the first-order approximation can be written as:

$$\tilde{T}_i = \frac{\tilde{P}}{R\overline{\rho}} = T_i + O(h). \tag{4.37}$$

By introducing T_i in the first order approximation and after some developments, we have the following expression:

$$\tilde{T}_i = T_i + \frac{1}{C_v\overline{\rho}_i}(\overline{\rho E}_i - \overline{\rho}_i E_i) + \frac{1}{2C_v}\Big(- \frac{|\overline{\rho\mathbf{u}}_i|^2}{\overline{\rho}_i^2} + \frac{|\rho\mathbf{u}_i|^2}{\rho^2}\Big). \tag{4.38}$$

After some development, we can write the following expression for $\tilde{T}_i$:

$$\begin{aligned}\tilde{T}_i &= T_i + \frac{1}{C_v\overline{\rho}_i}\Big[\rho_i \sum_{m=1}^{n-1}\frac{1}{m!}\mathcal{M}_i^{(m)} : (C_v D_T^{(m)}|_i + \frac{1}{2}D_{|\mathbf{u}|^2}^{(m)}|_i)\\ &+ \sum_{m=2}^{n-1}\tfrac{1}{m!}\mathcal{M}_i^{(m)} : \sum_{l=1}^{m-1}\begin{pmatrix} m \\ l \end{pmatrix} D_\rho^{(m-l)}|_i \otimes (C_v D_T^{(l)}|_i + \tfrac{1}{2}D_{|\mathbf{u}|^2}^{(l)}|_i)\\ &- \frac{1}{2\overline{\rho}_i}\sum_{m=1}^{n-1}\frac{1}{m!}\mathcal{M}_i^{(m)} : \sum_{l=1}^{m}\begin{pmatrix} m \\ l \end{pmatrix} D_{\rho^2}^{(m-l)}|_i \otimes D_{|\mathbf{u}|^2}^{(l)}|_i\Big] + O(h^n).\end{aligned} \tag{4.39}$$

Finally, and after some algebra, we have the expression of ΔT:

$$\begin{aligned}\Delta T_i &= \frac{1}{\overline{\rho}_i}\sum_{m=2}^{n-1}\frac{1}{m!}\Big[\Big(\mathcal{M}_i^{(m-1)}\Big)^2 - \mathcal{M}_i^{(m)}\Big] : \sum_{l=1}^{m-1}\begin{pmatrix} m \\ l \end{pmatrix} D_\rho^{(m-l)}|_i \otimes D_T^{(l)}|_i\\ &- \frac{\rho_i}{2C_v\overline{\rho}_i}\sum_{m=1}^{n-1}\frac{1}{m!}\mathcal{M}_i^{(m)} : D_{|\mathbf{u}|^2}^{(m)}|_i - \frac{1}{2C_v\overline{\rho}_i}\sum_{m=2}^{n-1}\frac{1}{m!}\mathcal{M}_i^{(m)} : \sum_{l=1}^{m-1}\begin{pmatrix} m \\ l \end{pmatrix} D_\rho^{(m-l)}|_i \otimes D_{|\mathbf{u}|^2}^{(l)}|_i\\ &+ \frac{1}{2C_v\overline{\rho}_i^2}\sum_{m=1}^{n-1}\frac{1}{m!}\mathcal{M}_i^{(m)} : \sum_{l=1}^{m}\begin{pmatrix} m \\ l \end{pmatrix} D_{\rho^2}^{(m-l)}|_i \otimes D_{|\mathbf{u}|^2}^{(l)}|_i + O(h^n)\end{aligned} \tag{4.40}$$

The previous expressions give us the correction terms for calculating the mean value per dual cell for pressure. As for velocity and pressure, the correction term is zero for a second-order reconstruction, which translates to $\Delta T_i + O(h^2)$. It is therefore not necessary to calculate an additional correction term to construct, at order 2, the averaged value of the temperature per dual cell. Finally, we have the following expression.

$$\bar{T}_i = \frac{\tilde{P}}{R\overline{\rho}} + \Delta T_i = \tilde{T}_i + O(h^2) \tag{4.41}$$

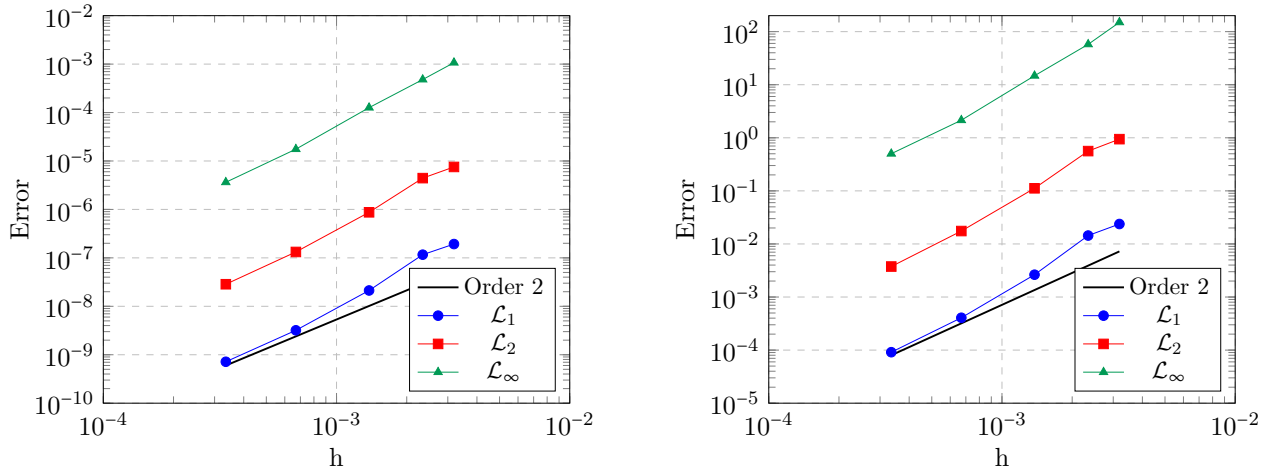
In this section, an attempt was made to construct, from the mean values per dual cell of the conservative variables, the mean variables per dual cell of the primitive variables. It was demonstrated that the correction terms are indeed zero for a second-order reconstruction, and this is observed for all primitive variables. Finally, based on the averaged values of the conservative variables, we are now able to construct the averaged values, to order 2, of the primitive variables. Consequently, the deconvolution operators can be applied to the primitive variables. As a consequence, the numerical scheme can be evaluated on Euler's equations.

4.8 Numerical results : 2D Euler test case

In this section, we present the results obtained for Euler's equations. The two test cases, isentropic vortex advection and Ringleb's case, remain the same as in Chapter 3.

4.8.1 Advection of an isentropic vortex

Here we present the results for the isentropic vortex advection test case. The initial solution of the conservative states, as described in 3.14, is integrated over each dual volume using the method given in 3. The calculation is thus initialised using the mean field per dual cell. The convective flux is calculated using a Roe solver, with a V4 reconstruction of the states at the face. To do this, we use the results of the deconvolution as described in section 4.4. Moreover, we use an RK4 scheme for the time scheme. For all the results presented in this section, we compare the solution from the CFD solver to the averaged solution per dual cell, so the error is calculated using definition (4.20). Vortex advection, as defined here, does not involve the boundary of the domain. Thus, the discretisation error only involves the cells inside the domain.

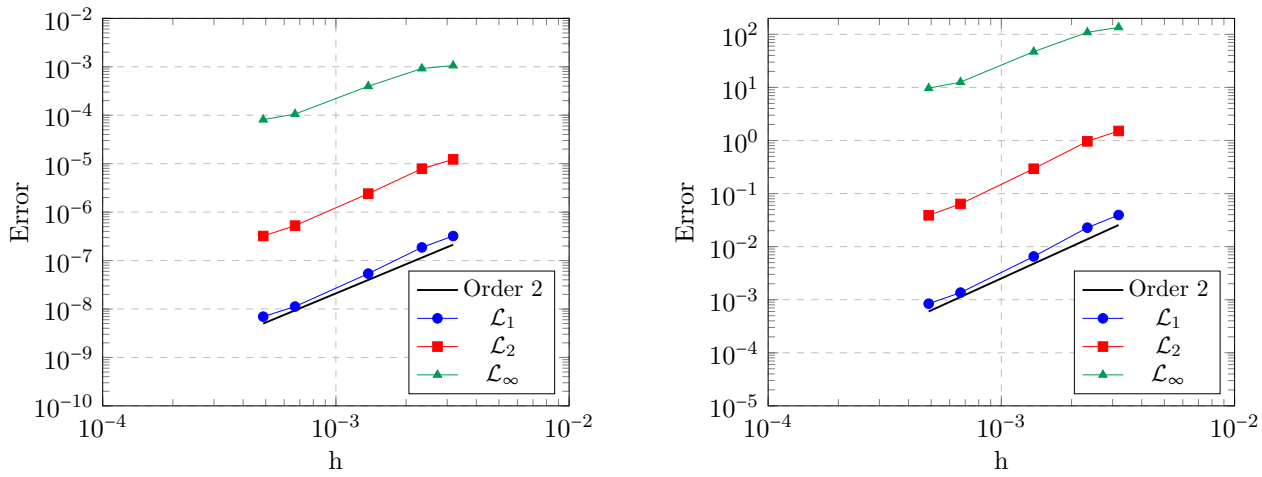


(a) Density

(b) Pressure

Figure 4.27: $\mathcal{L}_2$ error for density (left) and pressure (right) on regular mesh - Finite volume V4

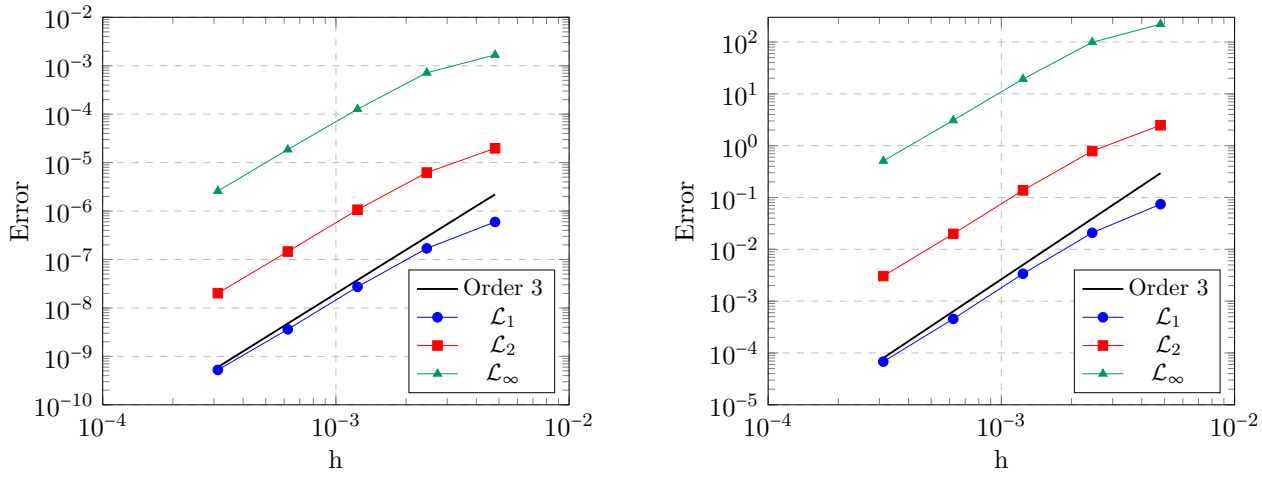
Figure 4.27 shows how the norm of the pressure and density errors evolve with mesh size for regular meshes. The results demonstrate that the discretisation error converges to order 2, and this is true for all the norms considered. Thus, the results show that the V4 scheme is of order 2 on regular meshes in its finite volume formulation. To confirm these results, we examine the same test case on perturbed meshes.



(a) Density (b) Pressure

Figure 4.28: $\mathcal{L}_2$ error for density (left) and pressure (right) on disturbed mesh - Finite volume V4

Figure 4.28 shows the results on perturbed meshes, which illustrate the evolution of the norm of the error for pressure and density as a function of mesh size. These results show us that the V4 scheme in its finite volume formulation is of order 2 in the sense of finite volumes on any mesh, since the norm of the discretization error converges to order 2. These results are similar to those obtained on a regular mesh.



(a) Density (b) Pressure

Figure 4.29: $\mathcal{L}_2$ error for density (left) and pressure (right) on Friedrich-Keller mesh - Finite volume V4

Finally, Figure 4.29 illustrates the results of the advection test case on Friedrich-Keller meshes. Results are provided for both density and pressure. We observe that the error norm converges to order 3 for this type of mesh. These results therefore show that the V4 scheme is at least of order 2 on all types of mesh and even seems to be of order 3 on Friedrich-Keller mesh for the advection test of an isentropic vortex.

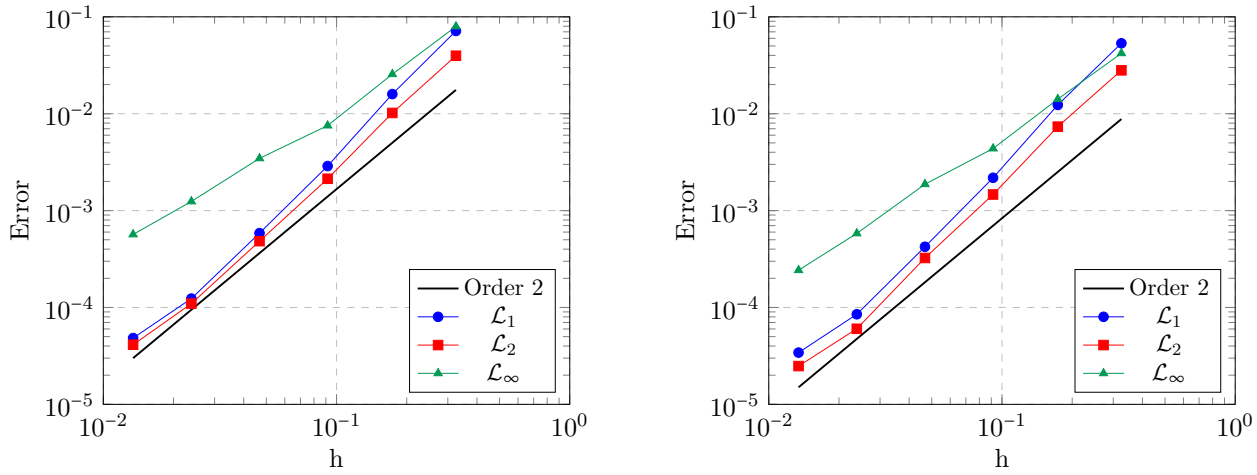
The various results obtained show that, when considering the mean value as the reference solu-

tion, the V4 scheme with deconvolution correction is at least second-order on all types of meshes. These results complement those previously obtained for the scalar advection equation. We will now present the results obtained for the Ringleb test case. The Ringleb test case represents an additional difficulty, since it involves boundary edges.

4.8.2 Ringleb

In this section, we present the results for the Ringleb test case. As with the advection test case, the initial solution, given in 3.3.2, is integrated over each dual volume using formulation described in 3. The initial solution is therefore the averaged value per dual cell. The solutions produced by the CFD solver are compared with the exact solution, which, for this test case, is the averaged value per dual cell. This enables us to evaluate the order of convergence of our numerical scheme in comparison with the averaged value per dual cell. We use the same mesh hierarchy as in Chapter 3. Two types of mesh are used: regular and perturbed. For all meshes, the error is calculated by considering all the cells that make up the mesh. Here, we use the V4 scheme, which is defined according to the finite volume formalism and a second-order deconvolution operator. Additionally, we use an implicit LUSSOR pseudo-time scheme. Finally, for boundary conditions, an analytical flux is calculated for each dual facet. To do this, the value at the centre of the facet is reconstructed using the nodal value and the nodal gradient obtained from the deconvolution results. This reconstruction is explained in section 4.4.2. Finally, this extrapolated value is used to calculate the boundary condition.

Figure 4.30 illustrates how the discretisation error norm evolves with mesh size for density and pressure. These results were obtained using regular meshes. The results are similar for both fields. Indeed, the discretisation error converges asymptotically to order 2 for all the considered norms. These initial results demonstrate that the numerical scheme defined is of order 2 compared to the averaged value per dual cell. These results are supplemented by results on perturbed meshes.



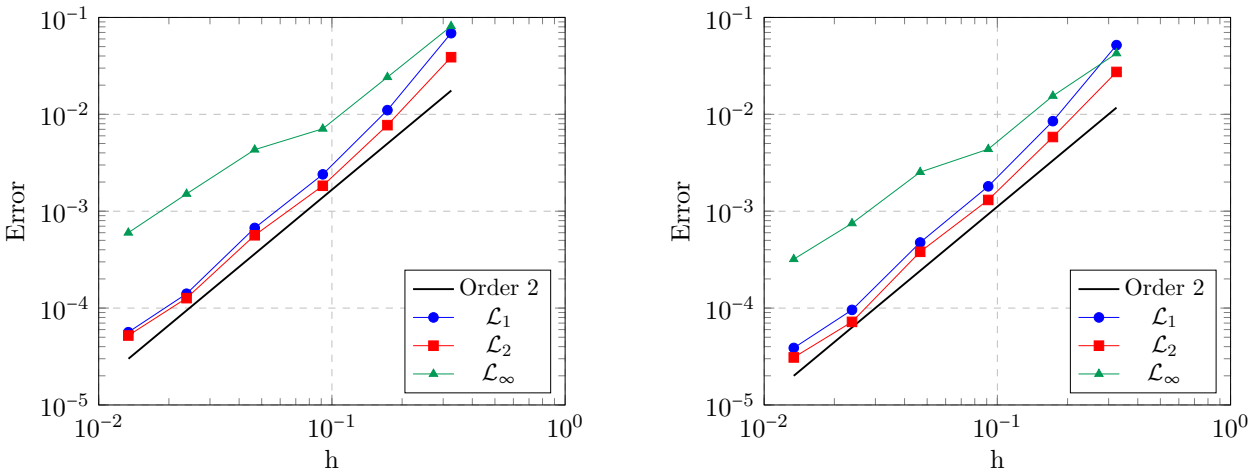
(a) Density

(b) Pressure

Figure 4.30: $\mathcal{L}_2$ error for density (left) and pressure (right) on regular mesh - Finite volume V4

Figure 4.31 illustrates how the discretisation error norm evolves with mesh size. As with regular meshes, the results are presented for density and pressure. These results were also obtained for perturbed meshes. As can be seen, the results are similar for both fields considered. The discretisation error converges to the order 2 for all norms considered. The conclusions regarding this type of mesh are consistent with those regarding regular meshes. The V4 scheme, which

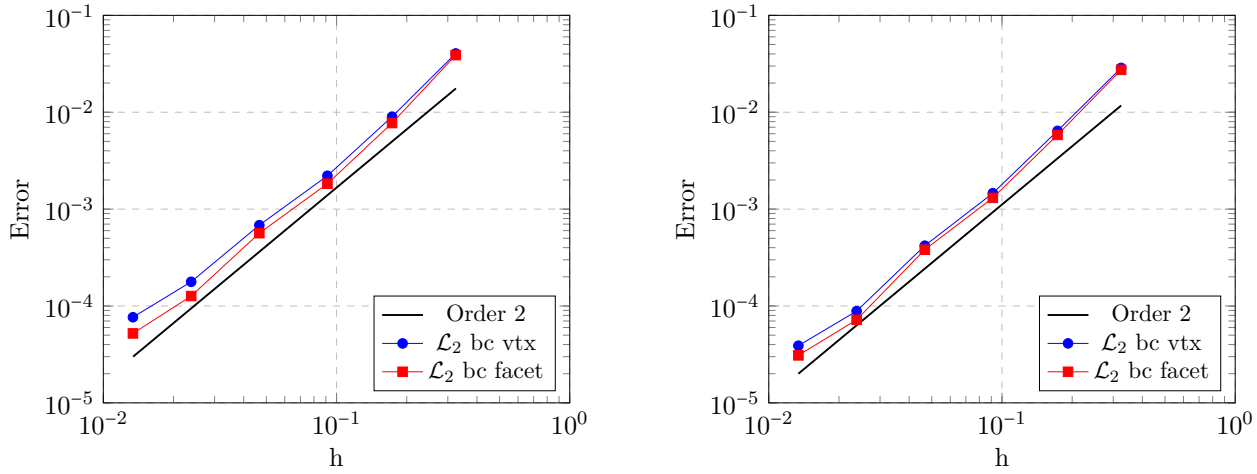
is defined according to a finite volume formalism with a second-order deconvolution operator, enables a second-order scheme to be constructed on any mesh type in comparison with the dual averaged value per cell. Figure 4.32 compares the results obtained with different boundary condition treatments. The blue curve shows the results obtained for a boundary condition treatment using the vertex value. More specifically, a vertex value is calculated from the available averaged values using a deconvolution method. The boundary conditions are then evaluated by facet based on this nodal value. This boundary condition treatment is precisely the one used in the V4 scheme carrying the vertex value. The results in red show the results for the boundary condition treatment using the value extrapolated to the facet. More precisely, based on the deconvolution results, a value is extrapolated at the centre of each facet. This extrapolated value is then used in the calculation of the boundary conditions. The results show that the discretisation error is lower when using a value extrapolated to the centre of the facet than when using a value at the vertex. However, this has no effect on the order of convergence of the error.



(a) Density

(b) Pressure

Figure 4.31: $\mathcal{L}_2$ error for density (left) and pressure (right) on disturbed mesh - Finite volume V4

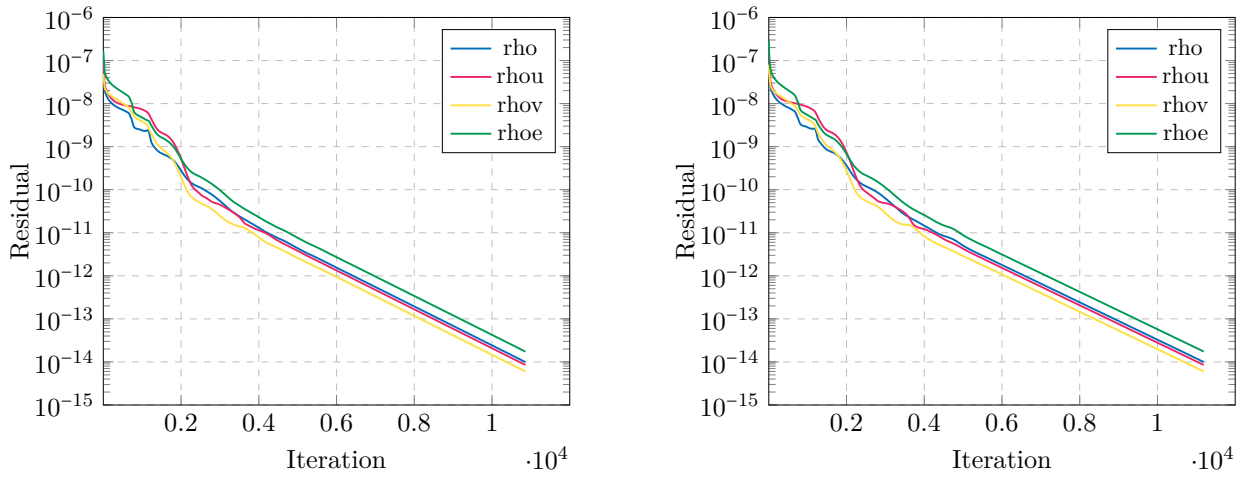


(a) Density

(b) Pressure

Figure 4.32: $\mathcal{L}_2$ error for density (left) and pressure (right) on disturbed mesh. Use of vertex value (blue) and facet value (red) for boundary condition - Finite volume V4

As for the edge-based scheme, Figure 4.33 shows the convergence of the residuals for the most refined meshes. For the Ringleb case, the V4 scheme, in its finite volume formulation, converges to zero machine, which is assumed here to be equal to 10^{-14}. It should also be noted that perturbed meshes require more iterations than regular meshes to achieve the same level of convergence. These results demonstrate that poorer quality meshes require more iterations to converge to zero machine.



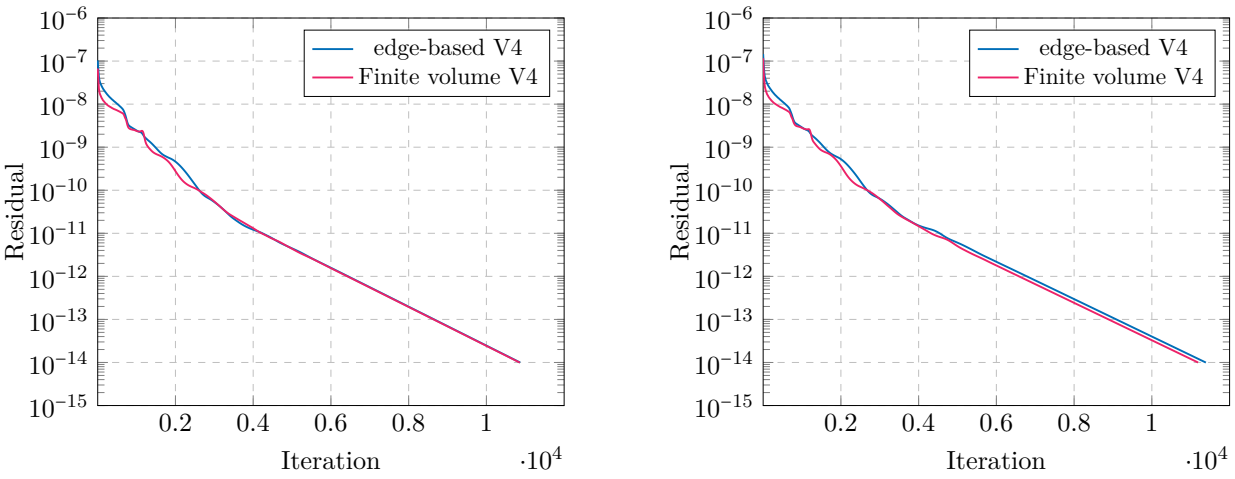
(a) Regular mesh

(b) Disturbed mesh

Figure 4.33: Convergence of residuals on regular mesh (left) and disturbed mesh (right) - Finite volume V4

It would be interesting to compare the edge-based and finite volume schemes. However, comparing error levels seems difficult since the error is not calculated in the same way. One piece of data that could be used for comparison is the convergence speed towards the stationary solution. Figure 4.34 shows how quickly the two schemes converge towards the stationary solution. For regular meshes, the number of iterations is the same for both schemes. For perturbed meshes,

however, the finite volume formalism converges slightly faster.



(a) Regular mesh

(b) Disturbed mesh

Figure 4.34: Convergence of density on regular mesh (left) and disturbed mesh (right) using edge-based V4 and finite volume V4

4.9 Conclusion

In this chapter, we have demonstrated how to develop a vertex-centred finite volume scheme based on the V4 scheme. In a finite volume formalism, the unknowns of the problem were the mean values per dual cell. However, the V4 scheme requires values at the mesh vertices for the calculation of convective fluxes and the treatment of boundary conditions. In the general framework, however, the centre of gravity of the dual cells and the mesh nodes are not the same. Therefore, it was necessary to reconstruct the vertex values from the available mean values, a process known as deconvolution. First, we presented Ollivier-Gooch's historical approach [99]. This approach is based on a least-squares method and can lead to the resolution of large linear systems. However, we opted for the approach of Bernard *et al.* [102], which is based on compact operators and enables a small, square linear system to be solved. In the framework of 1-exact deconvolution, we proposed a closure for the boundary cells. Moreover, we extended the definition of the gradient and Hessian operators to all mesh cells in the context of 2-exact deconvolution. We then proposed a method of using the deconvolution results to construct the V4 scheme. We also used the deconvolution results to treat boundary conditions. As with the edge-based formulation, we tested our numerical scheme on cases of increasing complexity. The results for the scalar advection equation demonstrated that the scheme is second-order accurate in the finite volume sense on any type of mesh, *i.e.* compared to the averaged value per dual cell. Results obtained on an anisotropic, adapted mesh demonstrated for the considered cases that the finite volume scheme retains the robustness of the edge-based scheme on this mesh type. We also exploited the linearity of the scalar advection equation to perform a local order analysis of our numerical scheme. The results demonstrated that the scheme, when defined according to a finite volume formalism, is indeed locally of order 2. We then extended the definition of our numerical scheme to the Euler equations. We demonstrated that there is no need to calculate additional correction terms to construct a finite volume scheme of order 2. We then evaluated our numerical scheme using the isentropic vortex advection test case. This test allowed us to validate the scheme at the current point. The results demonstrated that the scheme is of order 2 in the finite volume sense for all considered meshes. We concluded the study with the Ringleb case, which enabled us to incorporate boundary conditions. For this test case,

we also observed that the scheme is of order 2 on any type of mesh in the finite volume sense. Finally, we compared the convergence speeds of the edge-based and finite volume schemes. The results showed that the finite volume scheme converges slightly faster to the stationary solution for perturbed meshes. The results presented in this chapter were obtained with a convective flux calculated at order 1. In the next chapter, we will demonstrate how to utilise the finite volume formulation to calculate a convective flux at order 2.

Chapter 5

Flux correction

Contents

5.1 Geometric error **121**
5.2 Scalar advection equation **124**
5.2.1 Numerical results on standard meshes 126
5.2.2 Numerical results on adapted meshes 127
5.3 Euler equations **127**
5.3.1 Numerical results: the convection of a vortex 128
5.3.2 Numerical results: Ringleb test case 130
5.4 Error analysis **133**
5.5 Conclusion **134**

Scheme V4 was originally proposed in an edge-based formalism. The discretization of the conservation law system leads to the following discrete system:

$$\frac{d\mathbf{W}_i}{dt} + \frac{1}{|C_i|}\sum_{j\in\nu_i} \Phi_{ij}\,\overrightarrow{n}_{ij} = 0. \tag{5.1}$$

This discretisation enables convective flux to be calculated using a Riemann solver at point M_{ij}. By construction of the dual cell, this point is the midpoint of edge ij. As Nishikawa demonstrated in [36], this formula enables us to compute a convective flux of order three at most for meshes made up of simplexes. Therefore, it seems impossible to construct an edge-based scheme of order higher than three from the discrete system (5.1). Conversely, the finite volume formulation allows the convective flux to be calculated using a higher-order Gaussian quadrature thanks to its integral formulation. In this chapter, we propose a method for calculating the convective flux to order 2. This approach only requires the convective flux to be calculated once per edge and leads to a reduction in discretisation error.

5.1 Geometric error

This section details the calculation of convective flux. Discretising the conservation law equation system in a finite volume formalism yields to the expression (5.2) for calculating the convective flux:

$$\int_{\partial C_i} F(\mathbf{W})\cdot\overrightarrow{n}_{C_i}\,d\Gamma = \sum_{f_j\in\nu_i}\int_{f_j} F(\mathbf{W})\cdot\overrightarrow{n}_{ij}\,d\Gamma. \tag{5.2}$$

To compute convective flux, the border ∂C_i is decomposed into many faces. For a given dual cell C_i, ν_i denotes all the faces of C_i.

To calculate the first-order convective flux, a first-order quadrature method is used. In practice, the convective flux is calculated at point M_{ij}. Note that this formulation leads to the calculation of the convective flux in the same way as for the edge-based formulation. Calculating the convective flux to order 2 will not increase the order of our numerical scheme. In fact, in order to have a numerical scheme of order 3, it is necessary to calculate the convective flux to order 3. However, by calculating the convective flux to order 2, we seek to reduce the discretization error. It is well-known that a second-order flux computation requires at the same time a piecewise linear extrapolation and a second-order approximation of the integral. For the last term, the integration rule consists of applying Gauss approximation. A standard second-order formula needs the computation of the flux in the face center because first order flux derivative terms cancel due to surface moments. Indeed, for a second-order accurate flux computation, the numerical integration method must be of order 2 and therefore exact for affine polynomials. The mid-point method is a second-order numerical method. Indeed, for $f : (x, y) \mapsto f(x, y) = a + bx + cy$, with $(a, b, c) \in \mathbb{R}$, we always have:

$$\int_{S_j} f(x, y)\, d\Gamma = f(x_{G_j}, y_{G_j})|S_j|,$$

where (x_{G_j}, y_{G_j}) denotes the barycenter of S_j and $|S_j|$ its measure. This method assumes a constant normal vector to the face. To have a second order approximation of (5.3), we have to evaluate $F(\mathbf{W}) \cdot \vec{n}$ at the barycenter of $\partial C_i \cap \partial C_j$. In the vertex-centred approximation, the convective flux is estimated at the edge mid-point M_{ij} for a face composed of all facets that share the mid-point. For any facet, the integral is never at second order. And in addition, the integration point is never the center of gravity of the dual face. A second-order vertex-centred approach needs the computation of the flux

$$\int_{\partial C_i \cap \partial C_j} F(\mathbf{W}) \cdot \vec{n}\, d\Gamma \tag{5.3}$$

using a second-oder quadrature using the unknowns computed at the facet centers denoted G^1_{ij} and G^2_{ij} on Fig. 5.1.

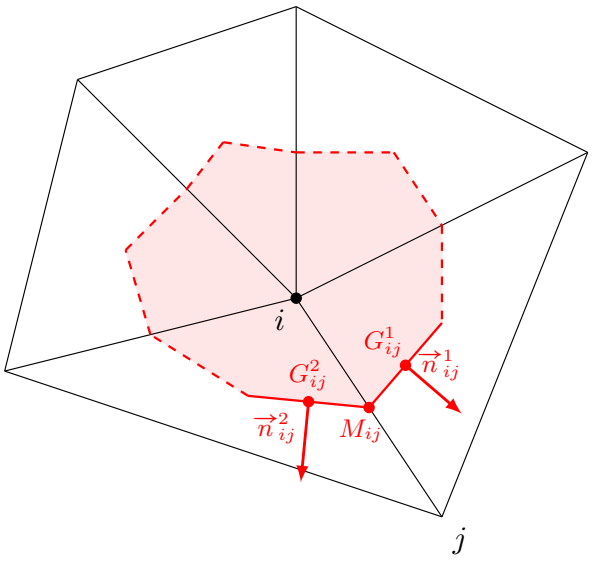


Figure 5.1: Geometric error for a 2D case

For each facet, the convective flux at its barycentre is evaluated. We therefore employ a quadrature of order 2, resulting in the following expression.

$$\int_{S_{ij}^1} F(\mathbf{W})\,\vec{n}\;d\Gamma = F(\mathbf{W})|_{G_{ij}^1}\,\vec{n}_{ij}|S_{ij}^1| + O(h^2)$$

where h represents the mesh size. Here, we present a second-order method. However, it is possible to calculate the convective flux to a higher order by increasing the number of quadrature points. This approach is commonly used in the literature, whether in the cell-centred [99, 106, 107, 108] or vertex-centred formalism [99, 109], to calculate the convective flow at multiple quadrature points. However, this method can be difficult to implement in an existing solver. Furthermore, the number of quadrature points can be substantial, particularly in 3D. Calculating the convective flux via an approximate Riemann solver can incur significant computational costs. Multiplying the number of Gaussian quadrature points also multiplies the number of calls to an approximate Riemann solver, which can make the computational cost of using the method prohibitive. Furthermore, evaluating the convective flux at a given point necessitates the reconstruction, to a certain degree of accuracy, of the state values at that point. This process requires reconstructing as many states as there are integration points, which can also represent a significant computational cost. For these reasons, this study takes a different approach, integrating the convective flux at a single point. Pont *et al.* [34] and Menasria *et al.* [110] have used this approach in a cell-centred formalism, and Stezwein [29] has used it in a vertex-centred formalism. In our case, the integration point for the convective flux is point M_{ij}. The idea is to use a Taylor development to return to the integration point of the convective flux. For each facet, we can derive a Taylor series expansion development between the centre of facet G_{ij} and point M_{ij}. To do so, introduce the coordinates $(x_{M_{ij}}, y_{M_{ij}})$ (resp. $(x_{G_{ij}^1}, y_{G_{ij}^1})$ and $(x_{G_{ij}^2}, y_{G_{ij}^2})$) of M_{ij} (resp. G_{ij}^1 and G_{ij}^2). Thus, for each facet, we can write.

$$\begin{aligned} F(\mathbf{W})|_{G_{ij}^1}\,\vec{n}_{ij}^1|S_{ij}^1| = F(\mathbf{W})|_{M_{ij}}\,\vec{n}_{ij}^1|S_{ij}^1| \;&+\; \frac{\partial F}{\partial x}|_{M_{ij}}(x_{G_{ij}^1} - x_{M_{ij}})\,\vec{n}_{ij}^1|S_{ij}^1| \\ &+\; \frac{\partial F}{\partial y}|_{M_{ij}}(y_{G_{ij}^1} - y_{M_{ij}})\,\vec{n}_{ij}^1|S_{ij}^1| + O(h^2). \end{aligned} \tag{5.4}$$

Finally, the results from each of the facets (S_{ij}^1 and S_{ij}^2 in our case) are added together to obtain the expression of the convective flux for each face S_{ij}.

$$\begin{aligned} F(\mathbf{W})|_{G_{ij}^1}\,\vec{n}_{ij}^1|S_{ij}^1| + F(\mathbf{W})|_{G_{ij}^2}\,\vec{n}_{ij}^2|S_{ij}^2| \;&=\; \\ \underbrace{F(\mathbf{W})|_{M_{ij}}(\vec{n}_{ij}^1|S_{ij}^1| + \vec{n}_{ij}^2|S_{ij}^2|)}_{\text{First-order term}} \;&+\; \frac{\partial F}{\partial x}|_{M_{ij}}\Big[(x_{G_{ij}^1} - x_{M_{ij}})\,\vec{n}_{ij}^1|S_{ij}^1| + (x_{G_{ij}^2} - x_{M_{ij}})\,\vec{n}_{ij}^2|S_{ij}^2|\Big] \\ &+\; \frac{\partial F}{\partial y}|_{M_{ij}}\Big[(y_{G_{ij}^1} - y_{M_{ij}})\,\vec{n}_{ij1}|S_{ij}^1| + (y_{G_{ij}^2} - y_{M_{ij}})\,\vec{n}_{ij}^2|S_{ij}^2|\Big]. \end{aligned}$$

In the current cell-vertex approach, we only compute the first-order term. Terms for second-order accuracy are therefore missing. In order to evaluate these terms, we must evaluate $\vec{\nabla} F|_{M_{ij}}$. This expression then depends on the system of equations considered.

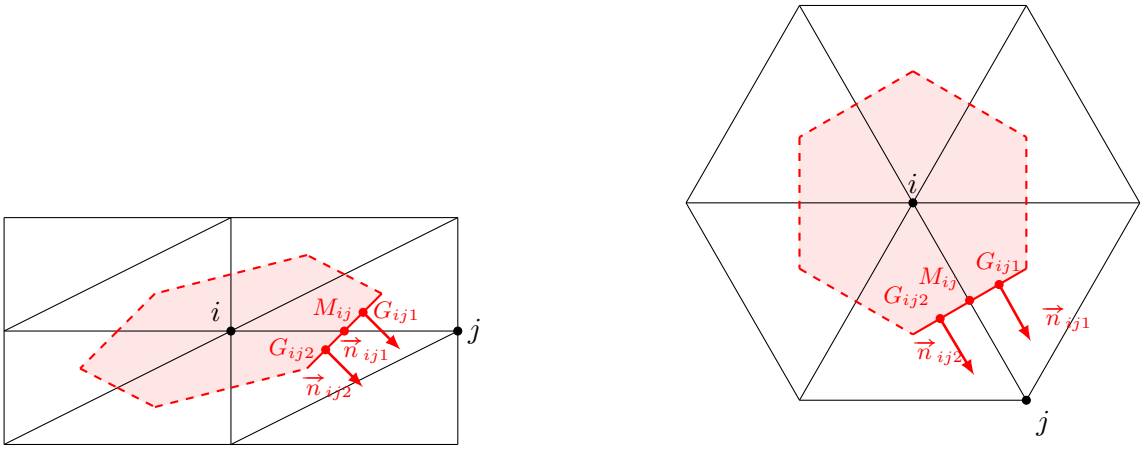
Remark :

As demonstrated in Figure 5.2 for Friedrich-Keller or regular meshes, and for cells within the domain, faces are flat and we have $|S_{ij1}| = |S_{ij2}|$ and $\vec{n}_{ij1} = \vec{n}_{ij2}$. As a consequence, we have the following relationship:

$$x_{M_{ij}} = \frac{x_{ij1} + x_{ij2}}{2} \quad \text{and} \quad y_{M_{ij}} = \frac{y_{ij1} + y_{ij2}}{2}.$$

As a consequence,we can write:

$$\begin{aligned} F(\mathbf{W})|_{G_{ij1}} \overrightarrow{n}_{ij1}|S_{ij1}| + F(\mathbf{W})|_{G_{ij2}} \overrightarrow{n}_{ij2}|S_{ij2}| &= \\ F(\mathbf{W})|_{M_{ij}} (\overrightarrow{n}_{ij1}|S_{ij1}| + \overrightarrow{n}_{ij2}|S_{ij2}|) &+ \frac{\partial F}{\partial x}|_{M_{ij}} \Big[((x_{ij1} + x_{ij2} - 2x_{M_{ij}})) + \overrightarrow{n}_{ij2}|S_{ij2}| \Big] \\ &+ \frac{\partial F}{\partial y}|_{M_{ij}} \Big[((y_{ij1} + y_{ij2} - 2y_{M_{ij}}))\, \overrightarrow{n}_{ij1}|S_{ij1}| \Big]. \end{aligned}$$



(a) Friedrich-Keller mesh

(b) Regular mesh

Figure 5.2: Correction term for convective flux on Friedrich-Keller mesh (left) and regular mesh (right)

As a result, we can state the following equality:

$$F(\mathbf{W})|_{G_{ij1}} \overrightarrow{n}_{ij1}|S_{ij1}| + F(\mathbf{W})|_{G_{ij2}} \overrightarrow{n}_{ij2}|S_{ij2}| = F(\mathbf{W})|_{M_{ij}} \overrightarrow{n}_{ij}|S_{ij}|.$$

In the event of the faces being planar, orthogonal to the edge, and with the same length, point M_{ij} is the barycentre of the face between cells i and j. As a consequence, the convective flux thus calculated is of order 2. However, this is a special case and in the case of any mesh, the flux thus calculated is never of order 2. It is therefore necessary to provide additional correction terms to ensure a flux of order 2 for any type of mesh.

5.2 Scalar advection equation

For the transport equation, the derivative of the convective flux is given in 5.5:

$$\frac{\partial F}{\partial x}|_{M_{ij}} = \frac{\partial \mathbf{W}}{\partial x}|_{M_{ij}} \mathbf{a}_{M_{ij}} + \mathbf{W}_{M_{ij}} \frac{\partial \mathbf{a}}{\partial x}|_{M_{ij}} \tag{5.5}$$

The derivatives of the velocity vector appear in the derivative of the convective flux. To make the notations simpler, the Jacobian matrix of the advection velocity is introduced:

$$J_{M_{ij}} = \begin{bmatrix} \frac{\partial a_1}{\partial x}|_{M_{ij}} & \frac{\partial a_1}{\partial y}|_{M_{ij}} \\ \frac{\partial a_2}{\partial x}|_{M_{ij}} & \frac{\partial a_2}{\partial y}|_{M_{ij}} \end{bmatrix}$$

We can then write the second-order convective flux in the case of the scalar transport equation:

$$\begin{aligned} F(\mathbf{W})|_{G^1_{ij}} \vec{n}^1_{ij}|S^1_{ij}| + F(\mathbf{W})|_{G^2_{ij}} \vec{n}^2_{ij}|S^2_{ij}| \quad &= \\ \underbrace{F(\mathbf{W})|_{M_{ij}}(\vec{n}^1_{ij}|S^1_{ij}| + \vec{n}^2_{ij}|S^2_{ij}|)}_{\text{First-order term}} \quad &+ \quad \mathbf{W}_{M_{ij}}\Big[\quad |S^1_{ij}|J_{M_{ij}} \begin{pmatrix} (x_{G^1_{ij}} - x_{M_{ij}}) \\ (y_{G^1_{ij}} - y_{M_{ij}}) \end{pmatrix} \vec{n}^1_{ij} + \\ & \qquad\qquad |S^2_{ij}|J_{M_{ij}} \begin{pmatrix} (x_{G^2_{ij}} - x_{M_{ij}}) \\ (y_{G^2_{ij}} - y_{M_{ij}}) \end{pmatrix} \vec{n}^2_{ij}\Big] \\ &+ \quad \frac{\partial \mathbf{W}}{\partial x}|_{M_{ij}}\Big[\quad (x_{G^1_{ij}} - x_{M_{ij}})|S^1_{ij}|\mathbf{a}_{M_{ij}} \cdot \vec{n}^1_{ij} + \\ & \qquad\qquad (x_{G^2_{ij}} - x_{M_{ij}})|S^2_{ij}|\mathbf{a}_{M_{ij}} \cdot \vec{n}^2_{ij}\Big] \\ &+ \quad \frac{\partial \mathbf{W}}{\partial y}|_{M_{ij}}\Big[\quad (y_{G^1_{ij}} - y_{M_{ij}})|S^1_{ij}|\mathbf{a}_{M_{ij}} \cdot \vec{n}^1_{ij} + \\ & \qquad\qquad (y_{G^2_{ij}} - y_{M_{ij}})|S^2_{ij}|\mathbf{a}_{M_{ij}} \cdot \vec{n}^2_{ij}\Big] \\ &+ \quad O(h^2). \end{aligned} \tag{5.6}$$

As shown by expression 5.6, the calculation of the second-order convective flux depends, on the one hand, on the terms given by the geometry of the mesh. These elements are straightforward to calculate and remain constant for a given mesh. On the other hand, the correction terms require the values of the states at point M_{ij} to be known. As these terms have already been calculated by MUSCL reconstruction, they can be used directly in the convective flux calculation. Finally, calculating a second-order flux requires the gradient of the states at point M_{ij}. This term is not known in advance, so it must be calculated. To achieve this, we propose basing our approach on the results of the deconvolution. As we demonstrated in Chapter 4, the deconvolution process enables us to calculate a nodal value at the second-order and a nodal gradient at the first-order from the available average values. We propose reconstructing the gradient of the states at point M_{ij} from the gradients at the nodes resulting from the deconvolution. For a given edge ij, the gradient at point M_{ij} is calculated using the expression 5.7:

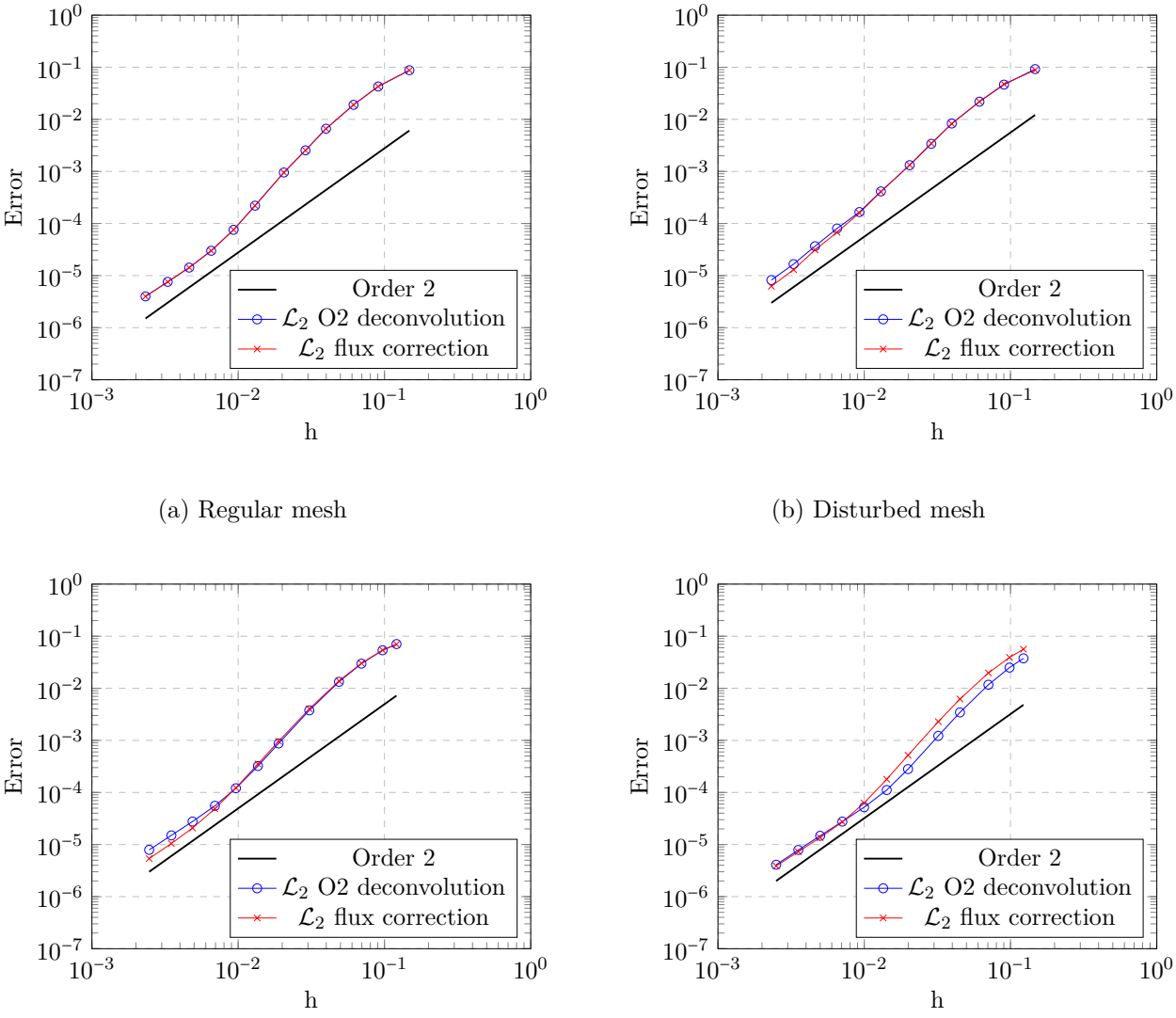
$$\frac{\partial \mathbf{W}}{\partial x}|M_{ij} = \frac{1}{2}\left(\frac{\partial \mathbf{W}}{\partial x}|_i + \frac{\partial \mathbf{W}}{\partial x}|_j\right). \tag{5.7}$$

Finally, the calculation of the second-order convective flux is based on surface moments, which can be pre-calculated. It is also dependent on the states at the midpoint of each mesh edge, as determined by MUSCL reconstruction. The calculation of the second-order flux is also based on the gradients of the states in the middle of each mesh edge, which can be easily calculated from the deconvolution results. Therefore, the calculation of the second-order flux can be easily derived from previously calculated terms. Consequently, it does not incur a high computational cost and is straightforward to implement.

In order to evaluate the contribution of this correction term, we return to the Gaussian circular advection test case.

5.2.1 Numerical results on standard meshes

In this section, we present the results obtained for the case of circular advection of a Gaussian as presented in 3.8. We consider the V4 scheme transporting the mean value per dual cell. We also integrate a correction term into the calculation of the convective flux. This correction term, whose formula is given in 5.6, is calculated using the nodal gradient from the deconvolution and based on formulation 5.7.



(a) Regular mesh

(b) Disturbed mesh

(c) Union jack mesh

(d) Friedrich-Keller mesh

Figure 5.3: $\mathcal{L}_2$ error on regular, perturbed, Friedrich-Keller and Union Jack meshes, clockwise from top left - Finite volume V4 and flux correction

The numerical results are presented in Figure 5.3. For each type of mesh, we compare the solution obtained with and without this geometric correction term. For regular meshes, the two solutions are similar. As we have seen with this type of mesh, the correction term cancels out for internal cells. It is therefore not surprising that there is no reduction in the discretisation error for this type of mesh. The conclusion is the same for Friedrich-Keller meshes, where the

solutions are similar for the finest meshes, although this is not exactly the case for the coarsest meshes. We know that on this type of mesh, the correction term is also zero for cells inside the domain. It is therefore consistent to see similar results. For perturbed and Union Jack meshes, we observe only a very slight decrease in discretisation error for the finest meshes. The results show that flux correction does not significantly reduce discretisation error. However, the four types of meshes considered here show little or no anisotropy. If the mesh is more anisotropic, the correction term will be greater. We will now look at the results obtained on an adapted mesh, where highly anisotropic elements appear.

5.2.2 Numerical results on adapted meshes

We will now focus on the results obtained using adapted meshes, where highly anisotropic elements may appear. As the correction term is proportional to the size of the dual facets, it follows that the greater the anisotropy of the cells, the greater the correction term. The results are presented in Figure 5.4, and as with the previous meshes, we compare the results with and without convective flux correction. To achieve this, we compare the $\mathcal{L}_2$ norm of the discretisation error. The results demonstrate that the discretisation error decreases when the convective flux is calculated at order 2. This is true for all the meshes considered. However, we observe no difference in the order of convergence of the numerical scheme. It is therefore evident that the convective flux correction term is a relevant factor in the context of anisotropic adaptive meshes. Indeed, for the scalar advection equation, the calculation of the convective flux to order 2 only allows a significant reduction in the discretisation error for highly anisotropic meshes. In order to complete our study, we will now turn our attention to test cases for Euler's equations.

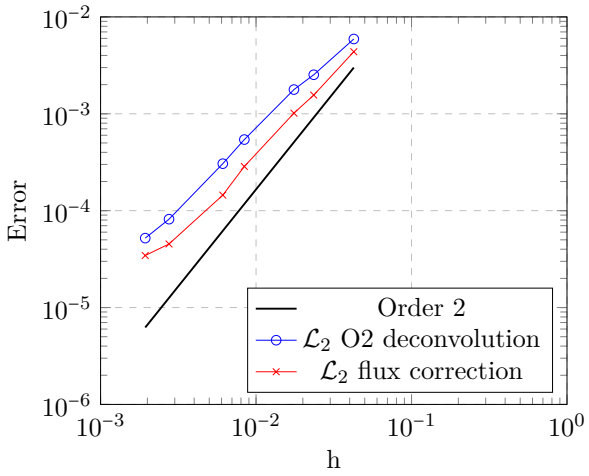


Figure 5.4: $\mathcal{L}_2$ error on adapted mesh - Finite volume V4 and flux correction.

5.3 Euler equations

We now present the convective flux correction for the Euler equations. The first step is to calculate the convective flux gradient. We present the results for the first spatial coordinate in

5.8, but the results are similar for the other components:

$$\begin{cases} \dfrac{\partial \rho \mathbf{u}}{\partial x} &= \dfrac{\partial \rho}{\partial x}\mathbf{u} &+ \rho\dfrac{\partial \mathbf{u}}{\partial x}, \\ \dfrac{\partial (\rho \mathbf{u}\otimes\mathbf{u} + p\mathbf{I}_d)}{\partial x} &= \dfrac{\partial \rho}{\partial x}\mathbf{u}\otimes\mathbf{u} &+ \rho(\dfrac{\partial \mathbf{u}}{\partial x}\otimes\mathbf{u} + \mathbf{u}\otimes\dfrac{\partial \mathbf{u}}{\partial x}) + \dfrac{\partial p}{\partial x}\mathbb{I}, \\ \dfrac{\partial (\mathbf{u}(\rho E + p))}{\partial x} &= \dfrac{1}{2}\dfrac{\partial \rho}{\partial x}\|\mathbf{u}\|^2\mathbf{u} &+ \dfrac{\partial \mathbf{u}}{\partial x}(\dfrac{3}{2}\rho\|\mathbf{u}\|^2 + \dfrac{\gamma}{\gamma-1}p) + \dfrac{\gamma}{\gamma-1}\dfrac{\partial p}{\partial x}\mathbf{u}. \end{cases} \tag{5.8}$$

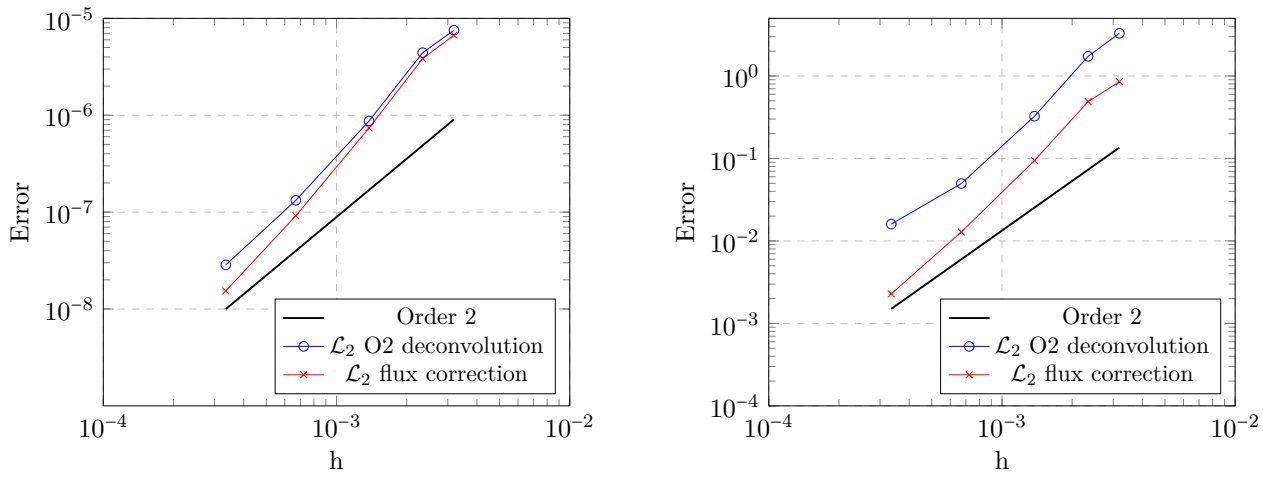
The third equality is obtained by using the fact that

$$\rho E = \frac{1}{2}\rho\|\mathbf{u}\|^2 + \frac{p}{\gamma-1}.$$

To calculate the convective flux, we need to know the derivative of the flux at point M_{ij}, the midpoint of edge ij. Note that the derivative of the convective flux depends on the states at point M_{ij} and on the gradient of the states at point M_{ij}. The states at point M_{ij} are given by the MUSCL reconstruction. To calculate the gradient of the states at point M_{ij}, we use the nodal gradients from the deconvolution. Finally, we use definition (5.7) to calculate the gradients of the states at point M_{ij}. The calculation of the gradient of the convective flux can be easily evaluated from the results of the deconvolution and the MUSCL reconstruction. This correction can therefore be easily implemented in a calculation code and does not represent a significant computational cost. To measure the contribution of this correction in terms of reducing discretisation error and increasing the order of convergence, we return to the vortex advection and Ringleb test cases.

5.3.1 Numerical results: the convection of a vortex

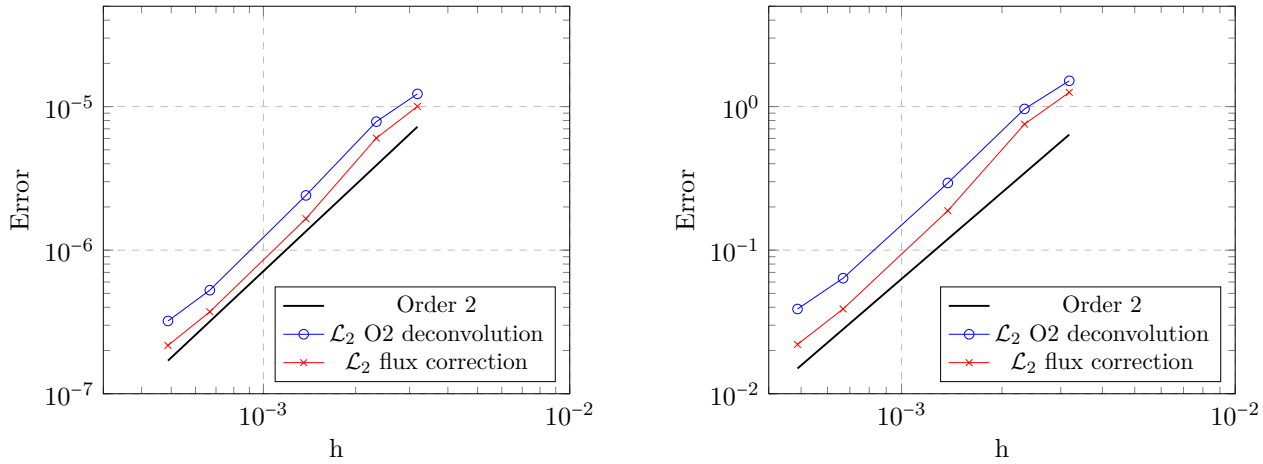
In this section, we present the results of the vortex advection test case, as described in Section 3.14, which is initialised with the averaged value of the conservative quantities per dual cell. Here, we use the V4 scheme and a second-order deconvolution method, with the numerical parameters given in Table 3.2. The time scheme is the explicit Runge-Kutta method with 4 steps. We use three types of mesh: regular, perturbed and Friedrich-Keller. For each mesh type, we compare the results obtained using the V4 scheme and second-order deconvolution with the results obtained using the V4 scheme, second-order deconvolution and convective flux correction.



(a) Density (b) Pressure

Figure 5.5: $\mathcal{L}_2$ error for density (left) and pressure (right) on regular mesh - Finite volume V4 and flux correction

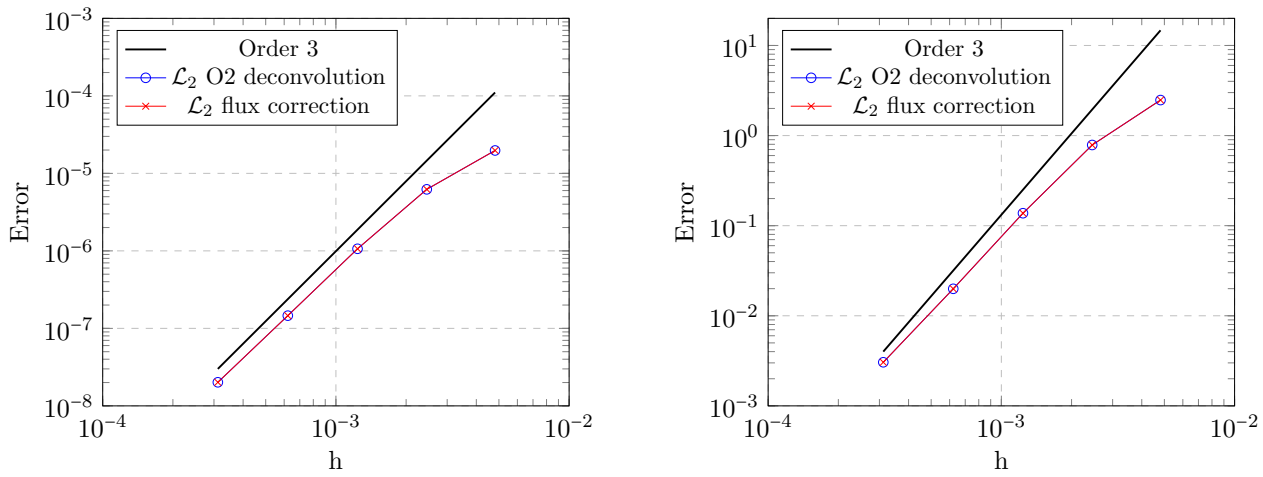
Figure 5.5 illustrates how the $\mathcal{L}_2$ norm of the discretisation error evolves on regular meshes in relation to mesh size. Here, we present the results for density and pressure. In both cases, the discretisation error decreases when the flux correction is used. This decrease is more pronounced for pressure. However, the convective flux correction does not appear to affect the order of the numerical scheme, as the error norm appears to converge asymptotically to order 2.



(a) Density (b) Pressure

Figure 5.6: $\mathcal{L}_2$ error for density (left) and pressure (right) on disturbed mesh - Finite volume V4 and flux correction

Figure 5.6 also illustrates how the $\mathcal{L}_2$ norm of the discretisation error evolves with mesh size for perturbed meshes. The results are presented for both density and pressure. In both cases, the discretisation error decreases when the convective flux correction is used. These results are consistent with those obtained on regular meshes. Second-order convergence is still observed for our numerical scheme.



(a) Density

(b) Pressure

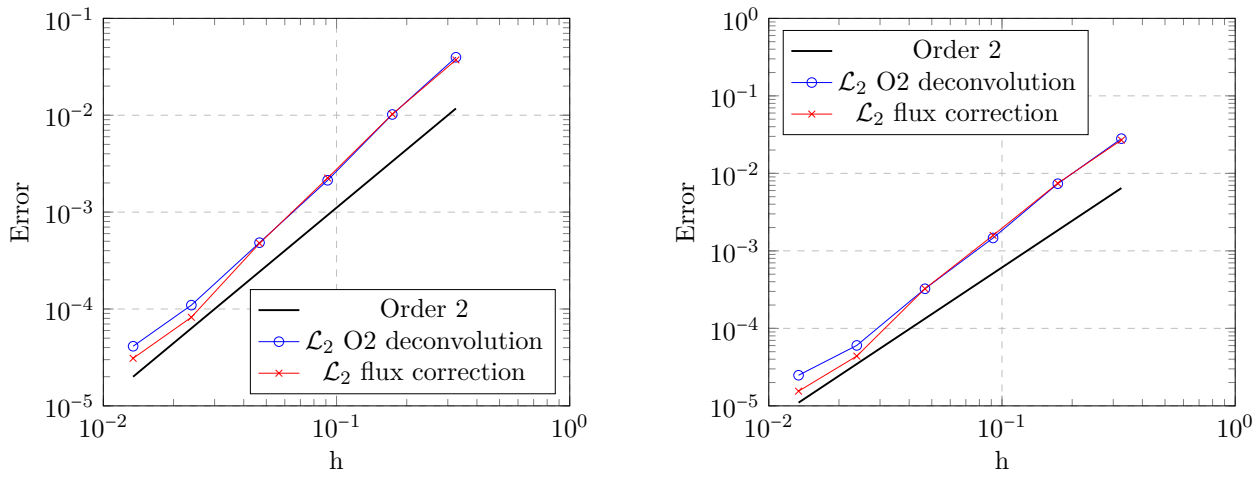
Figure 5.7: $\mathcal{L}_2$ error for density (left) and pressure (right) on Friedrich-Keller mesh - Finite volume V4 and flux correction.

Finally, Figure 5.7 illustrates how the $\mathcal{L}_2$ norm of the discretisation error evolves with mesh size for Friedrich-Keller meshes. Results are provided for both density and pressure. It is clear that the results obtained with or without convective flux correction are identical. As we have previously demonstrated, the correction term is zero for Friedrich-Keller meshes. Since vortex advection does not involve boundary cells, these results are entirely consistent.

The results of the vortex advection test case demonstrate a reduction in discretisation error when the corrective flux term is employed, with the exception of Friedrich-Keller meshes, where this term is zero. A greater reduction in discretisation error is observed for the Euler equations than for the scalar transport equation. Indeed, we observe a significant reduction in the discretization error for all the meshes considered for the vortex advection test case. In the next section, we present the results for the Ringleb test case.

5.3.2 Numerical results: Ringleb test case

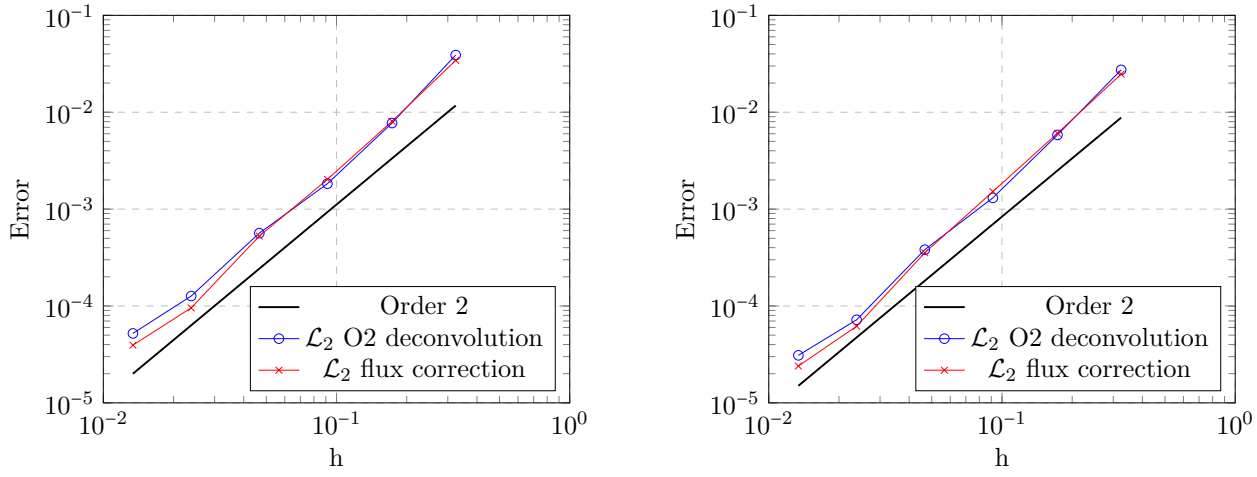
In this section, we present the results for the Ringleb test case, as described in Section 3.3.2, which is initialised using the averaged value of the conservative quantities per dual cell. The numerical parameters for the test case are provided in Table 3.5, and as with the vortex advection test case, the V4 finite volume scheme is employed here. We employ a second-order deconvolution method and calculate the convective flux by integrating the corrections presented in this chapter. We use an implicit pseudo-time scheme of the LUSSOR type. Two types of mesh are used: regular and perturbed. For each type, the results obtained using the V4 scheme, second-order deconvolution and first-order convective flux are compared with those obtained using the V4 scheme, second-order deconvolution and second-order convective flux.



(a) Density (b) Pressure

Figure 5.8: $\mathcal{L}_2$ error for density (left) and pressure (right) on regular mesh - Finite volume V4 and flux correction.

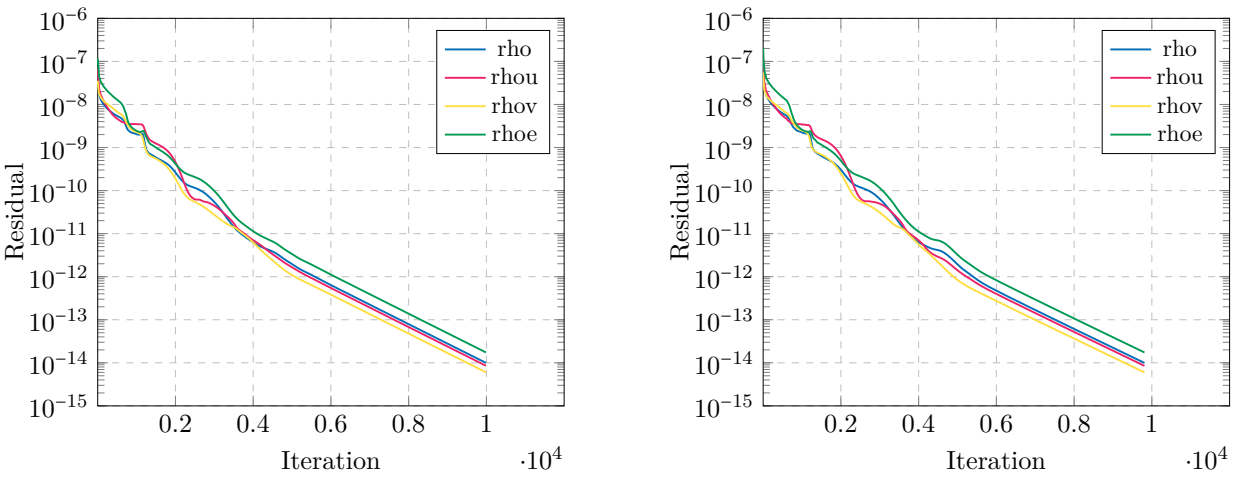
Figure 5.8 illustrates how the $\mathcal{L}_2$ norm of the discretisation error evolves on a regular mesh, depending on the mesh size. Here, we present the results for density and pressure. We observe a decrease in the discretisation error when using a convective flux of order 2, but only for the most refined meshes, for both density and pressure. However, the order of convergence remains the same. Compared to the vortex advection test case, the decrease in the discretisation error is smaller.



(a) Density (b) Pressure

Figure 5.9: $\mathcal{L}_2$ error for density (left) and pressure (right) on disturbed mesh - Finite volume V4 and flux correction.

Figure 5.9 illustrates how the $\mathcal{L}_2$ norm of the discretisation error evolves on perturbed meshes in relation to mesh size. As with regular meshes, we present the density and pressure results here. These results are very similar to those obtained for regular meshes. Indeed, we observe a decrease in the discretisation error when calculating a convective flux of order 2 for the most refined meshes only. These results are valid for both density and pressure. Furthermore, the numerical scheme defined in this way is always of order 2 in the sense of finite volumes.

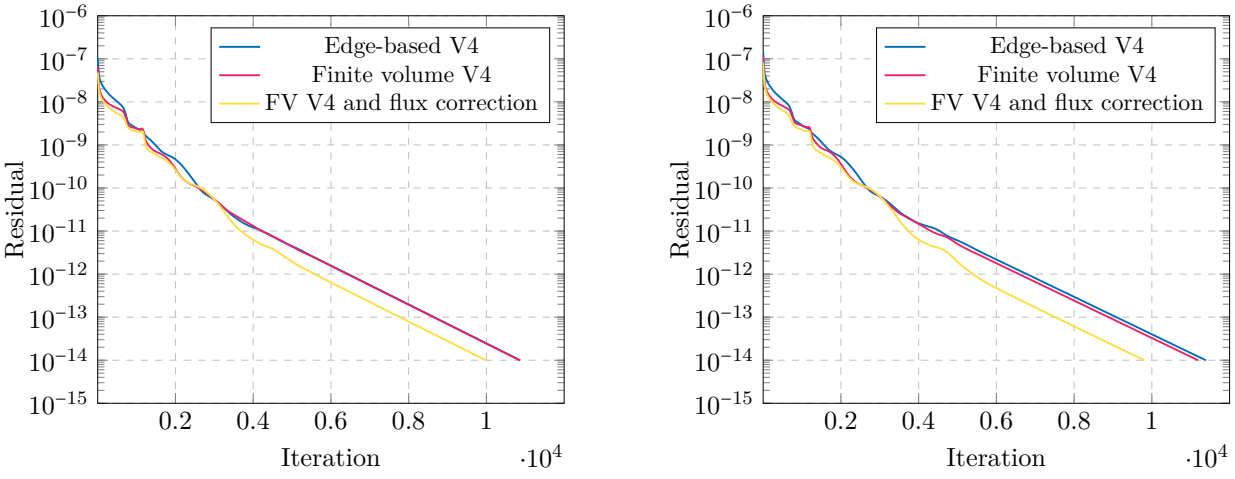


(a) Regular mesh

(b) Disturbed mesh

Figure 5.10: Convergence of residuals on regular mesh (left) and disturbed mesh (right) - Finite volume V4 and flux correction.

Figure 5.10 shows how the residuals converge for the Ringleb case and the most refined meshes. It illustrates the convergence of the residuals for all conservative variables. The results demonstrate that the residuals converge to 10^{-14} for both mesh types. Furthermore, we observe that the convergence of the residuals in the case of convective flux calculated at order 2 requires fewer iterations on perturbed meshes than on regular meshes. This was not the case when the convective flux was calculated at order 1, where perturbed meshes required more iterations than regular meshes to converge the residuals. To compare the edge-based and finite volume schemes, we examine the convergence of the density residual.



(a) Regular mesh

(b) Disturbed mesh

Figure 5.11: Convergence of density on regular mesh (left) and disturbed mesh (right) using edge-based V4 and finite volume V4.

Figure 5.11 shows the results for each mesh type, comparing the edge-based scheme with the finite volume schemes featuring convective fluxes of orders 1 and 2. There is a clear reduction in the number of iterations for the finite volume scheme with a convective flux calculated at order 2 compared to the edge-based scheme and the finite volume scheme with a convective flux

calculated at order 1. This decrease is more pronounced on perturbed meshes than on regular meshes. On regular meshes, the correction term is zero for most internal cells. On a perturbed mesh, however, the correction term is involved for all the mesh's cells. Therefore, it is not surprising to observe a greater contribution on perturbed meshes. The results demonstrate that calculating the convective flux to order 2 not only reduces the number of iterations required for convergence to machine zero, but also minimises the discretisation error.

5.4 Error analysis

In this section, we conclude our analysis of local errors from Chapter 4. Using the linearity of the scalar advection equation, we can measure the order of the numerical scheme for each dual cell. The results showed in Chapter 4 that scheme V4, defined using the finite volume method, allowed us to construct a second-order scheme on all cells of the mesh except those defined around the four corners. We conduct the same analysis here, this time using a second-order convective flux. The results are presented in Figures 5.12 and 5.13 for Friedrich-Keller and perturbed meshes, respectively. These figures show that calculating the second-order convective flux enables us to build a locally second-order scheme for the cells around the mesh corners. Therefore, for the V4 scheme to be of order 2 on all the cells of a given mesh when defined in terms of finite volumes, it is necessary to use a deconvolution formula of order 2, as well as calculate the convective flux to order 2.

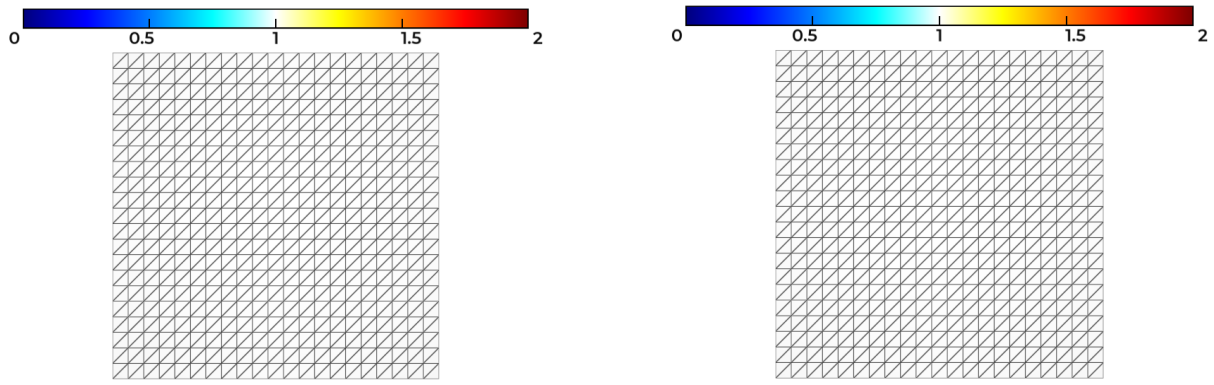


Figure 5.12: Friedrichs-Keller mesh - first order derivatives term. $|C_i|\partial_x w$ field (left), $|C_i|\partial_y w$ field (right) - O2 deconvolution and flux correction

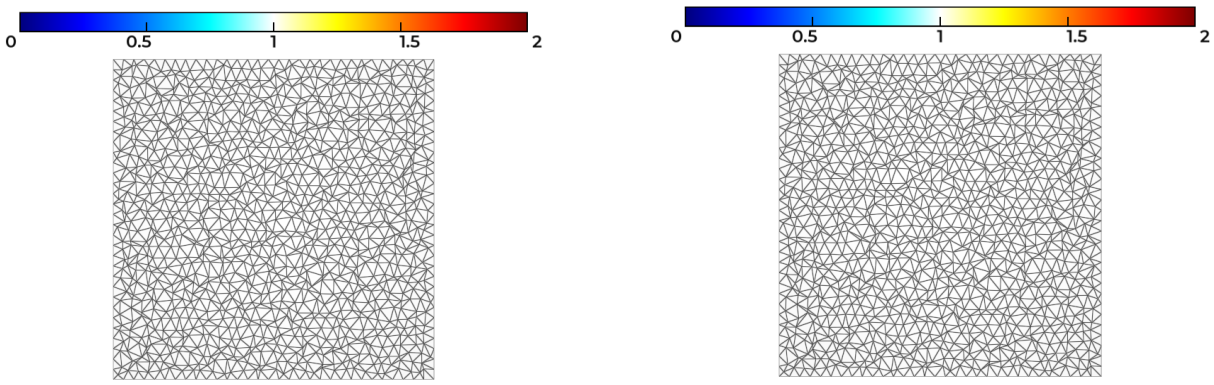


Figure 5.13: Disturbed mesh - first order derivatives term. $|C_i|\partial_x w$ field (left), $|C_i|\partial_y w$ field (right) - O2 deconvolution and flux correction

5.5 Conclusion

In this chapter, we presented an approach for calculating second-order convective flux within a vertex-centred finite volume formulation. To achieve this, we should increase the number of quadrature points used in the convective flux calculation. In order to limit the computational cost of the approach and simplify its implementation, we selected a single integration point for the convective flux calculation. This necessitates a Taylor series expansion between the quadrature points and the convective flux integration point. In our work, we restrict ourselves to second-order convective flux. The quadrature points are then the barycentres of the dual facets. This approach enables us to calculate a correction term depending on the surface moments, as well as the problem states and their derivatives at the convective flux integration point. The states at the convective flux integration point are provided by the MUSCL reconstruction. The state derivatives are calculated from the deconvolution results. To test this new approach, we used several tests of increasing complexity. First, we tested our scheme on a scalar advection case. Initial results on certain meshes without anisotropy suggest a slight reduction in discretisation error for the most refined meshes. Results obtained using an adapted mesh, however, show a more significant reduction in discretisation error. We then tested the scheme on the Euler equations. These results also showed a more marked decrease in discretisation error for the most refined meshes. In addition, calculating the second-order convective flux significantly reduced the number of iterations required for the residuals to converge. Finally, we demonstrated that calculating the convective flux to second order made it possible to define a second-order numerical scheme for all cells in a given mesh.

Chapter 6

Conclusion and perspectives

The V4 scheme, which was originally proposed by A. Dervieux [16], is based on an edge-based discretisation of the hyperbolic problem equations. It is now commonly used in mesh adaptation and has demonstrated great robustness on highly anisotropic meshes. However, we were not convinced by the analysis published in the literature, essentially based on a finite-difference error computation, while the scheme was defined in a finite-volume environment.

During this work, we presented a complete study of the V4 scheme. After detailing its construction, we demonstrated that the scheme is at least of order 2 on any mesh type if the vertex value is assumed to be the reference solution. This was demonstrated through the analysis of both the truncation and discretisation errors. We also demonstrated that a specific reconstruction was required for the boundary vertices to ensure that the scheme is of order 2 at all mesh vertices, including those located at the boundary. We studied the V4 scheme in cases of increasing complexity, ranging from scalar equation to Euler equations. Nevertheless, we identified several limitations of the V4 scheme. Indeed, the edge-based discretisation of the problem equations does not permit the construction of an edge-based scheme of order greater than three. Additionally, the edge-based formulation necessitates the treatment of boundary conditions at each node, which can introduce errors when cells share multiple boundary conditions. For these reasons, we sought to construct a vertex-centred finite volume scheme of order 2 in the sense of finite volumes on any type of mesh. Finite volume schemes allow us to circumvent the limitations of the edge-based formulation.

To keep the precision and robustness of the V4 scheme on an anisotropic mesh, we based our finite volume on the V4 formulation. We then demonstrated that, to develop such a scheme, it was necessary to reconstruct nodal values using the available mean values. This process is known as deconvolution. To achieve this, we replaced the standard ideas of the k-exact reconstruction, mainly introduced by Ollivier-Gooch, by a recent approach proposed by Bernard *et al.* [102]. The idea is to compute a badly-accurate gradient and Hessian, and to correct them by inverting a small square system of equations. The matrix to invert does only depend on the mesh itself and can be computed during a preprocessing step, which can strongly limit the CPU overhead. We suggested an extension of these operators up to order 3, using extended stencils. Finally, we demonstrated how to apply the deconvolution results to the V4 scheme. First, we tested this approach on the scalar advection equation. The results showed that the resulting scheme is of order 2 on any type of mesh. Additionally, the scheme retains its robustness on an anisotropic, adapted mesh. A local error analysis also enabled us to demonstrate that the resulting scheme is of order 2. We then demonstrated how to apply these corrections to the Euler equations. We implemented the approach in the SoNICS code and validated it on increasingly complex cases. The results for the studied test cases demonstrate that the scheme is also of order 2 in the sense of finite volumes on any type of mesh. Finally, we proposed an approach to computing the convective flux to order 2. This approach is enabled by the finite volume formulation of our numerical scheme. It is based on a second-order quadrature of the convective flux and involves

calculating a correction term based on surface moments and the derivative of the convective flux. This derivative is calculated using extrapolated states at faces, as well as the deconvolution results. However, the results for the scalar advection equation do not demonstrate a notable reduction in discretisation error for isotropic meshes. On the other hand, for anisotropic meshes, we observe a reduction in the discretisation error. This correction has also been implemented in the SoNICS code to enable its application to the Euler equations. Unlike the scalar advection equation, the results show a decrease in discretisation error with isotropic meshes. Even if the convective flux is calculated to order 1 (the flux is computed at the edge midpoint and not at the facet centers), our finite volume V4 scheme is always at order 2 for all mesh types in the sense of finite volumes, and that it preserves the properties of the V4 scheme for anisotropic meshes.

Our last effort focused on the correction of the flux computation at the edge midpoint. A theoretical analysis reveals that this approach is only first-order accurate. We proposed a correction for the flux using the available gradients at mesh nodes computed with deconvolution. The proposed correction comes from a geometrical correction between facet centers and the midpoint. The last improvement is validated for Euler's equations and shows a significant decrease in the number of iterations for our applications. In addition, the discretisation error is also reduced. We have demonstrated that calculating the convective flux to second order makes it possible to define a second-order numerical scheme on all cells of a given mesh. This corrects the defects observed in certain cells, particularly those located at the corners. Overall, the results demonstrate that our numerical scheme is of order 2 for all mesh types in the sense of finite volumes, and that it preserves the properties of the V4 scheme for anisotropic meshes.

This study enables us to consider various perspectives for the future of this work.

Firstly, no study on anisotropic adapted meshes for the Euler equations has been conducted in this work. This is due to technical limitations and time constraints, rather than a defect in the method itself. Several applications of interest for turbomachinery and external aerodynamics should be considered. This seems mandatory to confirm that deconvolution is not associated with a lack of robustness.

Additionally, the test cases examined in this study did not enable us to demonstrate the superiority of the finite volume approach over the edge-based approach in handling boundary conditions. Therefore, it would be interesting to compare the two approaches in this respect and see if one stands out. Such a point is also of strong importance for non-linear implicit solvers (implicit time integration) or linear implicit solvers (adjoint equations for goal-oriented mesh adaptation, for instance). Indeed, it could drastically change the way boundary conditions are enforced. It is well-known that weakly imposed boundary conditions enable a faster convergence to the solution [111].

This study was also limited to the Euler equations, so it does not address the viscous part of the Navier-Stokes equations. However, the MEV (mixed element volume) formulation already enables the Navier-Stokes equations to be solved using an edge-based discretisation. Based on the results of deconvolution, we could consider treating the viscous part in our formulation. Here, the main requirement for the viscous flux should be the integration of a third-order deconvolution, because the derivation is performed twice and requires a more accurate solution reconstruction. Deconvolution gives access to the nodal solution and gradients, and in addition to the nodal Hessian. We plan to try to combine them to define a new diffusion scheme.

The work presented in the thesis corresponds to a strong demand on reviewing schemes for steady solutions, typically for Reynolds-Averaged Navier-Stokes equations. We can imagine extending the deconvolution to deal with Large Eddy Simulation, where a third-order accuracy should be positive in the control of scheme dissipation and dispersion.

Finally, the proposed work should be extended to account for strong discontinuities. Applications should be twofold. First, mesh adaptation is today analysed to deal with hypersonic

flows, focusing on the wall heat-flux distribution [77]. We could ask ourselves how to extend deconvolution to discontinuities. In the same topic, anisotropic mesh adaptation is applied successfully to electric arc [112], where the arc is associated with a strong discontinuity and requires robustness.

Appendix A

Initialisation of the Ringleb test case

Algorithm 1 Initialisaton of the Ringleb test case

Require: (x, y)
Ensure: (ρ, u, v, p)
  if $x < -1.2$ **then**
    $a = 0.8$
  else
    $a = 0.001$
  end if
  $b = 0.999$
  while $b - a > 10^{-12}$ **do**
    $c = 0.5(a + b)$
    $J_a = \frac{1}{a} + \frac{1}{3a^3} + \frac{1}{5a^5} - 0.5\log(\frac{1+a}{1-a})$; $V_a = \sqrt{\frac{2(1-a^2)}{\gamma-1}}$; $\rho_a = a^{\frac{2}{\gamma-1}}$; $f_a = (x+0.5J_a)^2 + y^2 - \frac{1}{4\rho_a^2 V_a^4}$
    $J_c = \frac{1}{c} + \frac{1}{3c^3} + \frac{1}{5c^5} - 0.5\log(\frac{1+c}{1-c})$; $V_c = \sqrt{\frac{2(1-c^2)}{\gamma-1}}$; $\rho_c = a^{\frac{2}{\gamma-1}}$; $f_c = (x+0.5J_c)^2 + y^2 - \frac{1}{4\rho_c^2 V_c^4}$
    if $f_a f_c \leq 0$ **then**
      $b = c$
    else
      $a = c$
    end if
  end while
  $J = \frac{1}{c} + \frac{1}{3c^3} + \frac{1}{5c^5} - 0.5\log(\frac{1+c}{1-c})$; $V = \sqrt{\frac{2(1-c^2)}{\gamma-1}}$; $\psi = \sqrt{\frac{1}{2V^2} + y(x + \frac{J}{2})}$
  $\rho = c^{\frac{2}{\gamma-1}}$
  $p = \frac{1}{\gamma} c^{\frac{2\gamma}{\gamma-1}}$
  if $|V\psi - 1| < 10^{-9}$ **then**
    $\theta = \frac{\pi}{2}$
  else
    $\theta = \arcsin(V\psi)$
  end if
  if $y > 0$ **then**
    $u = V\cos\theta$
  else
    $u = -V\cos\theta$
  end if
  $v = -V^2\psi$

We provide a detailed description of the initialisation of the Ringleb test case. For each (x, y) coordinate of the mesh, we apply Algorithm 1. This enables us to calculate the density, velocity and pressure.

Appendix B

CFD software

During the course of this thesis, two CFD solvers were used. The first is the DeepDive code, which was developed specifically for this thesis. The second is the SoNICS code, which is a next-generation CFD code that is currently being developed jointly by ONERA and Safran. We provide a description of both solvers below.

B.1 SoNICS

The SoNICS code [19] is a new-generation computational fluid dynamics (CFD) code that is currently under development by ONERA and Safran. It is intended to replace the elsA code, which was also developed by ONERA, Safran and Airbus, and is widely used by the Safran Group for industrial purposes. A full description of the architecture in use for the SoNICS solver can be found in the publication by Lienhardt and Michel which has been submitted to CRAS [113]. The initial scope is also detailed in the article by Gueyffier, Plot and Soismier [19]. The SoNICS code relies on an explicit workflow comprising three distinct techniques.

- Its many configuration options are handled using the Software Product Lines (SPL) formalism [114].
- The computation is modelled as an acyclic graph of operators. This enables SoNICS to perform a wide range of analyses and optimisations, particularly thanks to its highly efficient distribution and communication scheme.
- SoNICS implements four graph transformation algorithms. These algorithms manage memory allocation and optimisation, establish communication between different memory nodes and implement automatic differentiation [115].

The solver is designed for both CPU and GPU architectures. It enables aerodynamic calculations in ideal or multi-species gases, solving the Euler and Navier-Stokes equations with or without a turbulence model. Its applications range from subsonic to hypersonic configurations related to the aeronautical industry, such as aircraft, compressors, turbines and nozzles. It also relies on the CGNS (CFD General Notation System) [116] standard for describing the topology and associated data.

Before the start of this thesis, the V4 edge-based scheme had already been implemented in the SoNICS code, in both 2D and 3D. Several studies have been conducted using this formulation and the SoNICS code. Notable examples include the work of Lugin *et al.* [76] and the work of Grenson and Manueco [117]. During this work, we implemented the 2D finite volume V4 scheme in the SoNICS code. This required several developments, which are detailed here. All of these developments concern the 2D part of the code.

- Implementation of first-order volume moments.
- Implementation of first-order surface moments.
- Construction of the convolution matrix and calculation of its inverse.
- Calculation of an edge-based gradient for deconvolution.
- Calculation of the nodal field from the mean fields by second-order deconvolution for all primitive fields.
- Calculation of the nodal gradient from the mean fields by second-order deconvolution for all primitive fields.
- Calculation of a finite element gradient from the fields at the verticess.
- Calculation of Clément's gradient from the fields at the vertices.
- Calculation of the MUSCL reconstruction from the fields at the vertices.
- Calculation of the correction term for the convective flux.
- Implementation of a second-order convective flux.
- Implementation of far-field boundary conditions for the Ringleb case.

B.2 DeepDive

DeepDive is a prototype of CFD code, developed with the only objective to provide a very simple and potentially inefficient code. It is designed to enable scheme development and validation. It is written in C and coupled with the libMeshb library provided by Inria for managing mesh and solution with vizir4. The development of DeepDive began before the PhD thesis. It was widely extended during the thesis.

Bibliography

[1] J. Amilhat, Fly the green deal, europe's vision for sustainable aviation, Advisory Council for Aviation Research and Innovation in Europe (ACARE) (2022).

[2] Safran, Lancement de rise, https://www.safran-group.com/fr/groupe/histoire-patrimoine/timeline/2021-lancement-rise#:~:text=GE%20et%20Safran%20ont%20lanc%C3%A9,par%20rapport%20aux%20moteurs%20actuels. [Consulted on : (22/10/2025)] (2021).

[3] L.-W. Chen, R. Pichler, R. Sandberg, Compressible DNS of a low pressure turbine subjected to inlet disturbances, in: Direct and Large-Eddy Simulation IX, Springer International Publishing, Cham, 2015, pp. 383–388.

[4] J. Marty, Numerical investigations of separation-induced transition on high-lift low-pressure turbine using RANS and LES methods, Proceedings of the Institution of Mechanical Engineers, Part A: Journal of Power and Energy (2014) 39 pages.

[5] L. Castillon, N. Gourdain, X. Ottavy, Multiple-frequency phase-lagged unsteady simulations of experimental axial compressor, Journal of Propulsion and Power 31 (2015) 444–455.

[6] G. Mouret, Adaptation of phase-lagged boundary conditions to large-eddy simulation in turbomachinery configuration, Theses, Institut National Polytechnique de Toulouse - INPT (Jun. 2016).

[7] W. Riéra, J. Marty, L. Castillon, S. Deck, Zonal Detached-Eddy Simulation Applied to the Tip-Clearance Flow in an Axial Compressor, AIAA Journal 54 (8) (2016) 2377–2391.

[8] U. B. Mehta, Guide to credible computer simulations of fluid flows, Journal of Propulsion and Power 12 (5) (1996) 940–948.

[9] M. W. Blake, NASA geometry data exchange specification for computational fluid dynamics (NASA IGES), Vol. 1338, National Aeronautics and Space Administration, Ames Research Center, 1994.

[10] A. Jameson, L. Martinelli, Mesh refinement and modeling errors in flow simulation, AIAA Journal 36 (5) (1998) 676–686.

[11] W. C. Tyson, C. J. Roy, A higher-order error estimation framework for finite-volume CFD, Journal of Computational Physics 394 (2019) 632–657.

[12] A. Remigi, M. Philit, F. Alauzet, E. Guilbert, Evaluation of tandem transonic compressor performances with mixing plane and metric-based mesh adaptation, in: AIAA SCITECH 2024 Forum, AIAA Paper 2024-0523, 8-12 January, Orlando, FL, 2024.

[13] A. Remigi, E. Parente, F. Alauzet, Feature based and goal oriented mesh adaptation for high pressure turbines applications, in: AIAA SCITECH 2025 Forum, AIAA Paper 2025-1109, 6-10 January 2025 Orlando, FL, 2025.

[14] P. Frey, F. Alauzet, Anisotropic mesh adaptation for cfd computations, Computer Methods in Applied Mechanics and Engineering 194 (48) (2005) 5068–5082.

[15] C. Morisco, E. Guilbert, M. Maunoury, F. Alauzet, Practical use of metric-based anisotropic mesh adaptation for industrial applications, Advances in Applied Mechanics (2025) 1–55.

[16] A. Dervieux, Steady euler simulations using unstructured meshes, in: Partial Differential Equations of Hyperbolic Type and Applications, World Scientific, 1985, pp. 33–111.

[17] B. van Leer, Towards the ultimate conservative difference scheme. iii. upstream-centered finite difference schemes for ideal compressible flow, Journal of Computational Physics 23 (3) (1977) 263–275.

[18] F. Alauzet, L. Frazza, Feature-based and goal-oriented anisotropic mesh adaptation for rans applications in aeronautics and aerospace, Journal of Computational Physics 439 (2021) 110340.

[19] D. Gueyffier, S. Plot, M. Soismier, SoNICS: a new generation CFD software for satisfying industrial users needs, in: OTAN/STO/Workshop AVT-366, NATO Science and Technology Organization, Online, France, 2022, pp. 10–1,10–16.

[20] Y. Gorschka, G. Puigt, A. Remigi, C. Content, B. Maugars, C. T. Morisco, Reviewing the vertex-centered v4 scheme for more accurate mesh adaptation, in: AIAA SCITECH 2025 Forum, AIAA Paper 2025-1108, 6-10 January, Orlando, FL, 2025.

[21] W. Sutherland, LII. The viscosity of gases and molecular force, Philosophical Magazine Series 5 36 (223) (1893) 507–531.

[22] J. Fourier, Théorie analytique de la chaleur, Landmarks of Science, Didot, 1822.

[23] V. Auffray, Etude comparative de schémas numériques pour la mod/'elisation de phénomènes diffusifs sur maillages multiéléments, Ph.D. thesis, Institut National Polytechnique de Toulouse (2007).

[24] T. Hughes, T. Tezduyar, Finite element methods for first-order hyperbolic systems with particular emphasis on the compressible euler equations, Computer Methods in Applied Mechanics and Engineering 45 (1) (1984) 217–284.

[25] P. Batten, N. Clarke, C. Lambert, D. M. Causon, On the choice of wavespeeds for the HLLC Riemann solver, SIAM J. Sci. Comput. 18 (1997) 1553–1570.

[26] P. Roe, Approximate riemann solvers, parameter vectors, and difference schemes, Journal of Computational Physics 43 (2) (1981) 357–372.

[27] E. Toro, Riemann Solvers and Numerical Methods for Fluid Dynamics: A Practical Introduction, Springer Berlin Heidelberg, 2009.

[28] F. Setzwein, P. Ess, P. Gerlinger, An implicit high-order k-exact finite-volume approach on vertex-centered unstructured grids for incompressible flows, Journal of Computational Physics 446 (2021) 110629.

[29] F. Setzwein, P. Ess, P. Gerlinger, An unstructured high-order finite-volume scheme for the simulation of reactive multi-species flows, Journal of Computational Physics 545 (2026) 11449.

[30] F. Setzwein, P. Ess, P. Gerlinger, High-order k-exact finite volume scheme for vertex-centered unstructured grids, in: AIAA Scitech 2020 Forum, AIAA Paper 2020-1785, 6-10 January, Orlando, FL, 2020.

[31] F. Setzwein, M. Spraul, P. Ess, P. M. Gerlinger, On the structure of correction matrices for a k-exact high-order finite-volume scheme on vertex-centered unstructured grids, in: AIAA Scitech 2021 Forum, AIAA Paper 2021-1548, 11–15 and 19–21 January 2021, 2021.

[32] T. Barth, Recent developments in high order k-exact reconstruction on unstructured meshes, in: 31st Aerospace Sciences Meeting, AIAA Paper 93-0668, 11-14 January, Reno, NV, U.S.A., 1993.

[33] F. Haider, P. Brenner, B. Courbet, J.-P. Croisille, Efficient implementation of high order reconstruction in finite volume methods, Springer Proceedings in Mathematics 4 (01 2011).

[34] G. Pont, P. Brenner, P. Cinnella, B. Maugars, J.-C. Robinet, Multiple-correction hybrid k-exact schemes for high-order compressible RANS-LES simulations on fully unstructured grids, Journal of Computational Physics 350 (2017) 45–83.

[35] G. Pont, Self adaptive turbulence models for unsteady compressible flows modèles de turbulence auto-adaptatifs pour la simulation des écoulements compressibles instationnaires, Ph.D. thesis, Ecole Nationale Supérieure d'Arts et Métiers (2015).

[36] H. Nishikawa, Accuracy-preserving boundary flux quadrature for finite-volume discretization on unstructured grids, Journal of Computational Physics 281 (2015) 518–555.

[37] C. Debiez, A. Dervieux, Mixed-element-volume MUSCL methods with weak viscosity for steady and unsteady flow calculations, Computers & Fluids 29 (1) (2000) 89–118.

[38] H. Nishikawa, An efficient implementation of edge-based discretization without forming dual control volumes, Advances in Aerodynamics 7 (1) (2025).

[39] B. van Leer, Towards the ultimate conservative difference scheme. iv. a new approach to numerical convection, Journal of Computational Physics 23 (3) (1977) 276–299.

[40] B. van Leer, Towards the ultimate conservative difference scheme. v. a second-order sequel to Godunov's method, Journal of Computational Physics 32 (1) (1979) 101–136.

[41] H. Nishikawa, The QUICK scheme is a third-order finite-volume scheme with point-valued numerical solutions, International Journal for Numerical Methods in Fluids 93 (7) (2021) 2311–2338.

[42] H. Nishikawa, On false accuracy verification of UMUSCL scheme, Communications in Computational Physics 30 (4) (2021) 1037–1060.

[43] H. Nishikawa, Uses of zero and negative volume elements for node-centered edge-based discretization, in: 23rd AIAA Computational Fluid Dynamics Conference, AIAA Paper 2017-4295, 5-9 June, Denver, Colorado, 2017.

[44] B. van Leer, H. Nishikawa, Towards the ultimate understanding of MUSCL: Pitfalls in achieving third-order accuracy, Journal of Computational Physics 446 (2021) 110640.

[45] D. Mavriplis, Revisiting the least-squares procedure for gradient reconstruction on unstructured meshes, in: 16th AIAA Computational Fluid Dynamics Conference, AIAA Paper 2003-3986, 23 - 26 June 2003, Orlando, Florida, 2003.

[46] T. Bridel-Bertomeu, Immersed boundary conditions for hypersonic flows using ENO-like least-square reconstruction, Computers & Fluids 215 (2020) 104794.

[47] H. Nishikawa, Beyond interface gradient: A general principle for constructing diffusion schemes, in: 40th Fluid Dynamics Conference and Exhibit, AIAA Paper 2010-5093, 28 June - 01 July, Chicago, Illinois, USA, 2010.

[48] B. Diskin, J. Thomas, Accuracy analysis for mixed-element finite-volume discretization schemes, Tech. rep., National Institute of Aerospace, NIA Report No. 2007-08 (2007).

[49] H. Nishikawa, Y. Nakashima, N. Watanabe, Effects of high-frequency damping on iterative convergence of implicit viscous solver, Journal of Computational Physics 348 (2017) 66–81.

[50] H. Nishikawa, A face-area-weighted 'centroid' formula for finite-volume method that improves skewness and convergence on triangular grids, Journal of Computational Physics 401 (2020) 109001.

[51] C. T. Morisco, H. Nishikawa, Applications of implicit edge-based gradients to third-order edge-based scheme for adaptive tetrahedral grids, in: AIAA SCITECH 2025 Forum, AIAA Paper 2025-0302, 6-10 January, Orlando, FL, 2025.

[52] H. Nishikawa, First, Second, and Third Order Finite-Volume Schemes for Navier-Stokes Equations, in: 7th AIAA Theoretical Fluid Mechanics Conference, AIAA Paper 2014-2091, 16-20 June, Atlanta, GA, 2014.

[53] H. Nishikawa, Updates to implicit edge-based gradient methods, in: AIAA SCITECH 2024 Forum, AIAA Paper 2024-1748, 8-12 January, Orlando, FL, 2024.

[54] A. Katz, V. Sankaran, An efficient correction method to obtain a formally third-order accurate flow solver for node-centered unstructured grids, Journal of Scientific Computing 51 (05 2012).

[55] A. Katz, V. Sankaran, Mesh quality effects on the accuracy of CFD solutions on unstructured meshes, Journal of Compututational Physics 230 (2011) 7670–7686.

[56] R. Carpentier, Approximation et analyse numérique d'écoulements instationnaires : application à des instabilités tourbillonnaires, Ph.D. thesis, Université de Nice (1995).

[57] J.-A. Desideri, A. Goudjo, V. Selmin, Third-order numerical schemes for hyperbolic problems, Research Report RR-0607, INRIA (1987).

[58] C. Tarsia Morisco, L.-M. Tenkès, F. Alauzet, Vertex-centered mixed finite elementfinite volume scheme for 2d anisotropic hybrid mesh adaptation, Computer Methods in Applied Mechanics and Engineering 419 (2024) 116638.

[59] L.-M. Tenkes, Methods for generating quad-dominant adaptive meshes for cfd simulations, Ph.D. thesis, Institut polytechnique de Paris (2022).

[60] J. Carlson, L. R. Center, Inflow/outflow Boundary Conditions with Application to FUN3D, NASA technical memorandum, National Aeronautics and Space Administration, Langley Research Center, 2011.

[61] R. J. Spiteri, S. J. Ruuth, A new class of optimal high-order strong-stability-preserving time discretization methods, SIAM Journal on Numerical Analysis 40 (2) (2002) 469–491.

[62] C.-W. Shu, Total-variation-diminishing time discretizations, SIAM Journal on Scientific and Statistical Computing 9 (6) (1988) 1073–1084.

[63] J. Butcher, A history of Runge-Kutta methods, Applied Numerical Mathematics 20 (3) (1996) 247–260.

[64] C.-W. Shu, S. Osher, Efficient implementation of essentially non-oscillatory shock-capturing schemes, Journal of Computational Physics 77 (2) (1988) 439–471.

[65] S. Gottlieb, C.-W. Shu, Total variation diminishing Runge-Kutta schemes., Mathematics of Computation 67 (1996).

[66] R. J. LeVeque, R. J. Leveque, Numerical methods for conservation laws, Vol. 132, Springer, 1992.

[67] S. Yoon, A. Jameson, A multigrid LU-SSOR scheme for approximate newton iteration applied to the euler equations, Tech. rep., NASA, NASA Contractor Report 179524 (1986).

[68] M. Gazaix, A. Jollès, M. Lazareff, The elsA object-oriented computational tool for industrial applications, in: 23RD INTERNATIONAL CONGRESS OF AERONAUTICAL SCIENCES, 8-13 September, Toronto, Canada, 2002, p. 19.

[69] G. Wang, A. Schwöppe, R. Heinrich, Comparison and evaluation of cell-centered and cell-vertex discretization in the unstructured tau-code for turbulent viscous flows, 5th European Conference on Computational Fluid Dynamics, European Community on Computational Methods in Applied Scienc Es CFD (01 2010).

[70] B. Diskin, J. Thomas, Comparison of node-centered and cell-centered unstructured finite volume discretizations: Inviscid fluxes, AIAA Journal 49 (4) (2011) 836–854.

[71] B. Diskin, J. Thomas, E. Nielsen, H. Nishikawa, J. White, Comparison of node-centered and cell-centered unstructured finite-volume discretizations: Viscous fluxes, AIAA Journal 48 (7) (2010) 1326–1338.

[72] G. Yu, B. Yu, Y. Zhao, J. Wei, Comparative studies on accuracy and convergence rate between the cell-centered scheme and the cell-vertex scheme for triangular grids, International Journal of Heat and Mass Transfer 55 (2012) 80518060.

[73] A. Syrakos, S. Varchanis, Y. Dimakopoulos, A. Goulas, J. Tsamopoulos, A critical analysis of some popular methods for the discretisation of the gradient operator in finite volume methods, Physics of Fluids 29 (12) (2017).

[74] P. Brenner, Unsteady flows about bodies in relative motion, in: First AFOSR Conference on Dynamic Motion CFD, New Brunswick, New Jersey, June 3 - 5, 1996, pp. 401–412.

[75] F. Alauzet, L. Frazza, D. Papadogiannis, Periodic adjoints and anisotropic mesh adaptation in rotating frame for high-fidelity RANS turbomachinery applications, Journal of Computational Physics 450 (2022) 110814.

[76] M. Lugrin, B. Isnard, B. Maugars, C. Content, Validation of an automated toolset for mesh adapted navier-stokes simulation of high speed vehicles using DNS and experiment, in: HiSST 2025 - HIGH-SPEED VEHICLE SCIENCE & TECHNOLOGY conference, Tours, September 22-29, 2025.

[77] G. Puigt, V. Golliet, F. Alauzet, Metric-based mesh adaptation for hypersonic flows: capturing wall heat flux with anisotropic triangles and tetrahedrons, in: HiSST 2025 - HIGH-SPEED VEHICLE SCIENCE & TECHNOLOGY conference, Tours, September 22-29, 2025.

[78] G. Barreau, Application d'une méthode semi-implicite couplée avec des outils d'adaptation de maillage anisotropique pour modéliser des arcs continus, Ph.D. thesis, Université de Toulouse (2024).

[79] G. Barreau, F. Pechereau, B. Khiar, P. Lalande, G. Puigt, J. Vanharen, F. Alauzet, Semi-implicit scheme for modelling the fluid dynamics part of a DC-powered arc, in: Aerospace Europe Conference 2023 – 10th EUCASS – 9th CEAS, 2023.

[80] V. Manet, La Méthode des Éléments Finis: Vulgarisation des aspects mathématiques, Illustration des capacités de la méthode (2012).

[81] C. Geuzaine, J.-F. Remacle, Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities., International Journal for Numerical Methods in Engineering 79 (2009) 1309–1331.

[82] D. J. Mavriplis, An advancing front Delaunay triangulation algorithm designed for robustness, Journal of Computational Physics 117 (1) (1995) 90–101.

[83] A. Harten, High resolution schemes for hyperbolic conservation laws, Journal of Computational Physics 49 (3) (1983) 357–393.

[84] A. Jameson, Analysis and design of numerical schemes for gas dynamics, 1: artificial diffusion, upwind biasing, limiters and their effect on accuracy and multigrid convergence, International Journal of Computational Fluid Dynamics 4 (3-4) (1995) 171–218.

[85] S. Piperno, S. Depeyre, Criteria for the design of limiters yielding efficient high resolution TVD schemes, Computers & Fluids 27 (2) (1998) 183–197.

[86] Clément, Ph., Approximation by finite element functions using local regularization, R.A.I.R.O. Analyse Numérique 9 (1975) 77–84.

[87] C. Tarsia Morisco, H. Nishikawa, On correctly evaluating orders of truncation error for node-centered edge-based schemes, Computers & Fluids 285 (2024) 106470.

[88] B. Maugars, Méthodes de volumes finis d'ordre élevé en maillages non coïncidents pour les écoulements dans les turbomachines, Ph.D. thesis, Ecole Nationale Supérieure d'Arts et Métiers (2016).

[89] F. Alauzet, A. Loseille, Metrix User Guide. Error Estimates and Mesh Control for Anisotropic Mesh Adaptation, Technical Report RT-0363, INRIA (2009).

[90] A. Loseille, R. Lohner, Cavity-based operators for mesh adaptation, in: 51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition, AIAA Paper 2013-0152, 07 - 10 January, Grapevine (Dallas/Ft. Worth Region), Texas, 2013.

[91] A. Loseille, F. Alauzet, V. Menier, Unique cavity-based operator and hierarchical domain partitioning for fast parallel generation of anisotropic meshes, Computer-Aided Design 85 (2017) 53–67, 24th International Meshing Roundtable Special Issue: Advances in Mesh Generation.

[92] Z. Wang, K. Fidkowski, R. Abgrall, F. Bassi, D. Caraeni, A. Cary, H. Deconinck, R. Hartmann, K. Hillewaert, H. Huynh, N. Kroll, G. May, P.-O. Persson, B. van Leer, M. Visbal, High-order cfd methods: current status and perspective, International Journal for Numerical Methods in Fluids 72 (8) (2013) 811–845.

[93] J. Vanharen, High-order numerical methods for unsteady flows around complex geometries, Ph.D. thesis, Institut National Polytechnique de Toulouse (2017).

[94] F. Ringleb, Exakte lösungen der differentialgleichungen einer adiabatischen gasströmung, ZAMM - Journal of Applied Mathematics and Mechanics / Zeitschrift für Angewandte Mathematik und Mechanik 20 (4) (1940) 185–198.

[95] G. Chiocchia, Chapter 4: Exact solutions to transonic and supersonic flows, in: Test cases for insviscid flow field methods, AGARD AR-211 report, 1984, pp. 4–1, 4–14.

[96] K. Förster, G. für Angewandte Mathematik und Mechanik, Boundary Algorithms for Multidimensional Inviscid Hyperbolic Flows: a GAMM-Workshop, Notes on Numerical Fluid Mechanics and Multidisciplinary Design, Vieweg+Teubner Verlag, 1978.

[97] J. Vila-Pérez, M. Giacomini, R. Sevilla, A. Huerta, Hybridisable discontinuous Galerkin formulation of compressible flows, Archives of Computational Methods in Engineering 28 (2) (2020) 753–784.

[98] A. Quarteroni, R. Sacco, F. Saleri, Numerical Mathematics, Vol. 37, Springer, 2007.

[99] C. Ollivier-Gooch, M. Altena, A high-order-accurate unstructured mesh finite-volume scheme for the advection–diffusion equation, Journal of Computational Physics 181 (2002) 729–752.

[100] T. J. Barth, Aspects of unstructured grids and finite-volume solvers for the euler and navier-stokes equations, AGARD, special course on unstructured grid methods for advection dominated flows (1992).

[101] Anderson, Bai, Bischof, Blackford, Demmel, Dongarra, D. Croz, Greenbaum, Hammarling, McKenney, Sorensen, LAPACK Users' Guide, 3rd Edition, Society for Industrial and Applied Mathematics, 1999.

[102] M. Bernard, G. Lartigue, G. Balarac, V. Moureau, G. Puigt, A framework to perform high-order deconvolution for finite-volume method on simplicial meshes, International Journal for Numerical Methods in Fluids 92 (11) (2020) 1551–1583.

[103] J. Mazaleyrat, D. Zuzio, J.-L. Estivales, G. Puigt, An original high-ordre accurate and polydirectional scheme with finite volume method for fluid-interface problems using the level-set method., Tech. rep., Onera (2023).

[104] E. F. Moore, The shortest path through a maze, in: Proc. International Symposium on the Theory of Switching, Part II, Harvard University Press, 1959, pp. 285–292.

[105] C. Tarsia Morisco, F. Alauzet, G. Puigt, From simplex to mixed element: Extension of a vertex-centered discretization, focus on accuracy analysis and 3d rans applications, Computers & Fluids 288 (2025) 106526.

[106] Q. Wang, Y.-X. Ren, W. Li, Compact high order finite volume method on unstructured grids ii: Extension to two-dimensional Euler equations, Journal of Computational Physics 314 (03 2016).

[107] W. Li, Y.-X. Ren, High-order k-exact WENO finite volume schemes for solving gas dynamic euler equations on unstructured grids, International Journal for Numerical Methods in Fluids 70 (2012) 742–763.

[108] M. Charest, C. Groth, P. Gauthier, A high-order central ENO finite-volume scheme for three-dimensional low-speed viscous flows on unstructured mesh, Communications in Computational Physics 17 (2015) 615–656.

[109] M. R. Charest, T. R. Canfield, N. R. Morgan, J. Waltz, J. G. Wohlbier, A high-order vertex-based central eno finite-volume scheme for three-dimensional compressible flows, Computers & Fluids 114 (2015) 172–192.

[110] A. Menasria, P. Brenner, P. Cinnella, G. Pont, Toward an improved wall treatment for multiple-correction k-exact schemes, in: 2018 Fluid Dynamics Conference, AIAA Paper 2018-4164, June 25-29, Atlanta, Georgia, 2018.

[111] C. Tarsia Morisco, H. Nishikawa, A parameter-free weak boundary condition procedure for P^1 Galerkin viscous discretization for tetrahedral meshe, Journal of Computational Physics under review (2025).

[112] F. Pechereau, F. Tholin, J. Vanharen, G. Puigt, F. Alauzet, P. Lalande, 3D modeling of an arc in interaction with a composite panel, in: Aerospace Europe Conference 2023 - 10th EUCASS - 9th CEAS, Lausane, Switzerland, July 9-13, 2023.

[113] M. Lienhardt, B. Michel, SoNICS: design and workflow of a new CFD software, Compte Rendu de l'Academie des Sciences (2025).

[114] P. Clements, L. Northrop, Software product lines: practices and patterns, Addison-Wesley Longman Publishing Co., Inc., USA, 2001.

[115] L. Hascoet, V. Pascual, The Tapenade automatic differentiation tool: Principles, model, and specification, ACM Transactions on Mathematical Software (TOMS) 39 (3) (2013) 1–43.

[116] D. Poirier, S. Allmaras, D. McCarthy, M. Smith, F. Enomoto, The CGNS system, in: 29th AIAA, Fluid Dynamics Conference, AIAA Paper 98-3007, 15 - 18 June, Albuquerque, NM, U.S.A., 1998.

[117] L. Manueco, P. Grenson, Comparative analysis of porous bleed simulation techniques: From porous wall boundary condition to automatic mesh refinement, in: AIAA AVIATION FORUM AND ASCEND, AIAA Paper 2025-3509, 21 - 25 July, Las Vegas, Nevada, 2025.

RÉPUBLIQUE FRANÇAISE
Liberté
Égalité
Fraternité

Université de Toulouse

Titre : Approche volume fini centrée au noeud : reformulation du schéma V4 original pour l'adaptation de maillage.

Mots clés : Volumes finis, Nœud centré, Adaptation de maillage, Turbomachine

Résumé : La mécanique des fluides numériques, généralement désignée par le terme anglais Computational Fluid Dynamics (CFD), permet de représenter les écoulements par simulation numérique. C'est la stratégie qui permet aujourd'hui de dimensionner les avions ou les moteurs. Pour cela, le domaine d'intérêt est recouvert de cellules qui partagent des nœuds et on parle alors de maillage. Sans connaître a priori l'écoulement, le domaine d'étude est maillé par des cellules isotropes. Avec cette stratégie, le nombre de degré de liberté peut devenir très important, notamment si l'on souhaite suivre des phénomènes anisotropes, comme des chocs ou des tourbillons. Un nombre très important de degré de liberté rend les simulations numériques impossibles en pratique. L'adaptation de maillage anisotrope consiste à faire converger dans un seul processus le couple (maillage, solution) en s'appuyant sur un critère mathématique de mesure de l'erreur numérique. L'adaptation sur maillage tétraédrique anisotrope permet d'aligner les mailles tétraédriques avec les directions préférentielles de l'écoulement tout en minimisant le nombre de mailles et cette approche démontre tout son intérêt pour des applications en milieu industriel. En conséquence, cette approche nécessite des schémas numériques robustes, puisque l'anisotropie dégrade la précision des schémas numériques. L'approche généralement utilisée pour l'adaptation de maillage est l'approche centrée au nœud (ou node-centered) dans laquelle les volumes de contrôle sont construits explicitement autour des nœuds du maillage initial. Jusqu'à présent, l'approche exclusivement utilisée est une approche dite edge-based, dans laquelle les inconnues du problème sont localisées aux nœuds du maillage. Le schéma V4, proposé à l'origine par Alain Dervieux, est une extension de l'approche MUSCL de van Leer à ce formalisme edge-based pour les flux convectifs. Cette approche permet de construire un schéma d'ordre 2 au sens de la valeur au nœud sur tout type de maillage. De plus, les résultats récents pour ce schéma numérique montrent une grande robustesse sur des maillages présentant une anisotropie de l'ordre du million. Or, il n'existe pas aujourd'hui de schéma numérique d'ordre 2 au sens des volumes finis (c'est-à-dire où les inconnues du problème sont les valeurs moyennes par cellule duale) pour l'approche centrée au nœud pour les écoulements compressibles. Dans ce travail de thèse, ce schéma numérique est construit à partir du schéma V4. Bien que les inconnues du problème soient les valeurs moyennes par cellules duales, la construction du schéma V4 se fait à partir des valeurs aux nœuds. Le principe de déconvolution, proposé à l'origine par Ollivier-Gooch, permet de construire la valeur au nœud à partir des valeurs moyennes disponibles. Dans ce travail de thèse, nous avons utilisé des opérateurs proposés par Bernard et al, qui conduisent à la résolution d'un système carré de petite taille. Nous avons d'abord travaillé avec l'équation de transport scalaire. Nous sommes parvenus à montrer que le schéma V4 ainsi défini est bien d'ordre 2 sur tout type de maillage au sens des volumes finis. Ce schéma conserve également ses propriétés de robustesse sur maillage adapté anisotrope. Nous avons également étendu la définition des opérateurs de déconvolution, ce qui permet une reconstruction de la valeur au nœud à l'ordre 3 sur tout type de maillage. Nous avons ensuite traité les équations d'Euler. Les résultats montrent également que le schéma numérique est d'ordre 2 sur maillage quelconque au sens des volumes finis. En se basant sur les résultats de la déconvolution, nous avons également proposé une formulation plus précise des conditions aux limites. Enfin, la formulation volume fini nous permet d'utiliser une quadrature d'ordre 2 pour les flux convectifs, ce qui conduit à une diminution de l'erreur dans le cas de maillage adapté anisotrope.

Title: Node-centred finite volume approach: reformulation of the original V4 scheme for mesh adaptation.

Key words: Finite volume, Node-centred, Mesh adaptation, Turbomachinery

Abstract: Computational fluid dynamics (CFD) is used to represent fluid flows through numerical simulation. It is the strategy currently used to design aircraft and engines. To do this, the area of interest is covered with cells that share nodes, which is referred to as meshing. Without prior knowledge of the flow, the domain of study is meshed with isotropic cells. With this strategy, the number of degrees of freedom can become very large, especially if one wishes to track anisotropic phenomena such as shocks or vortices. A very large number of degrees of freedom makes numerical simulations impossible in practice. Anisotropic mesh adaptation consists of converging the pair (mesh, solution) in a single process based on a mathematical criterion for measuring numerical error. Adaptation on an anisotropic tetrahedral mesh allows the tetrahedral meshes to be aligned with the preferred flow directions while minimising the number of elements, and this approach is particularly useful for industrial applications. Consequently, this approach requires robust numerical schemes, since anisotropy degrades the accuracy of numerical schemes. The approach generally used for mesh adaptation is the node-centred approach, in which control volumes are constructed explicitly around the nodes of the initial mesh. Until now, the approach used exclusively has been an edge-based approach, in which the unknowns of the problem are located at the nodes of the mesh. The V4 scheme, originally proposed by Alain Dervieux, is an extension of van Leer's MUSCL approach to this edge-based formalism for convective flux. This approach allows the construction of a second-order scheme in terms of node value on any type of mesh. Furthermore, recent results for this numerical scheme show great robustness on meshes with anisotropy of the order of one million. However, there is currently no second-order numerical scheme in the finite volume sense (i.e., where the unknowns of the problem are the average values per dual cell) for the node-centred approach for compressible flows. In this thesis, this numerical scheme is constructed from the V4 scheme. Although the unknowns of the problem are the average values per dual cell, the V4 scheme is constructed from the values at the nodes. The principle of deconvolution, originally proposed by Ollivier-Gooch, allows the value at the node to be constructed from the available average values. In this thesis, we used operators proposed by Bernard et al., which lead to the resolution of a small square system. We first worked with the scalar advection equation. We were able to show that the V4 scheme thus defined is indeed order 2 on any type of mesh in the finite volume sense. This scheme also preserves its robustness properties on adapted anisotropic meshes. We also extended the definition of deconvolution operators, which allows for a third-order reconstruction of the value at the node on any type of mesh. We then dealt with Euler's equations. The results also show that the numerical scheme is second-order on any mesh in the finite volume sense. Based on the deconvolution results, we also proposed a more precise formulation of the boundary conditions. Finally, the finite volume formulation allows us to use second-order quadrature for convective fluxes, which leads to a reduction in error in the case of anisotropic adapted meshes.