



HAL
open science

Schéma d'ordre élevé basé sur le résidu pour la simulation numérique d'écoulements compressibles en maillages non-structurés

Xi Du

► **To cite this version:**

Xi Du. Schéma d'ordre élevé basé sur le résidu pour la simulation numérique d'écoulements compressibles en maillages non-structurés. Sciences de l'ingénieur [physics]. Arts et Métiers ParisTech, 2010. Français. NNT : 2010ENAM0003 . pastel-00005803

HAL Id: pastel-00005803

<https://pastel.hal.science/pastel-00005803>

Submitted on 16 Feb 2010

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

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

Ecole doctorale n° 432 : Science des Métiers de l'Ingénieur

Doctorat ParisTech

THÈSE

pour obtenir le grade de docteur délivré par

l'École Nationale Supérieure d'Arts et Métiers

Spécialité " Mécanique "

présentée et soutenue publiquement par

Xi DU

le 12 janvier 2010

**Schéma d'ordre élevé basé sur le résidu pour
la simulation numérique d'écoulements compressibles
en maillages non-structurés**

Directeur de thèse : **Alain LERAT**

Co-encadrement de la thèse : **Christophe CORRE**

Jury

M. Rémi ABGRALL, Professeur, MAB, Université de Bordeaux
M. Aldo BONFIGLIOLI, Professeur, D.I.F.A., Université de Basilicata
M. Emmanuel LERICHE, Professeur, LA MUSE, Université Jean Monnet
M. Christophe CORRE, Professeur, LEGI, INPG
M. Alain LERAT, Professeur, DynFluid, Arts et Métiers ParisTech
M. Ali REZGUI, Responsable R&D Mécanique numérique, MICHELIN

Président
Rapporteur
Rapporteur
Examineur
Examineur
Examineur

Remerciements

Tout d'abord je remercie sincèrement Alain Lerat, mon directeur de thèse, qui m'avait accueilli tout au début dans le laboratoire de SIMulation NUMérique en MEcaniques des Fluides (SINUMEF) faire un master de recherche. A cette occasion j'avais pu suivre beaucoup de cours intéressants dans le domaine aérodynamique et aéroacoustique et connaître l'ambiance du laboratoire, où mon travail de thèse s'est ensuite bien déroulé du octobre 2006 jusqu'à aujourd'hui. A travers ma thèse, grâce à ses connaissances étendues en méthodes numériques, M. Lerat m'a donné beaucoup de conseils en schémas numériques. Et plus il m'a beaucoup aidé dans la procédure administrative ce qui est un coté très important pour un étudiant étranger.

Ensuite je dois un grand remerciement à mon co-directeur de thèse, Christophe Corre, pour son soutien tout au long de cette thèse et le partage avec moi de sa riche expérience en analyse numérique et aérodynamique. J'ai beaucoup apprécié ses explications et écritures toujours précises. Et j'aimerais remercier Rémi Abgrall d'avoir accepté d'être le Président de mon jury, et d'avoir examiné mes travaux. Je tiens également mes remerciements aux deux rapporteurs du jury: Aldo Bonfiglioli, Emmanuel Leriche, pour leur lectures attentives de ma mémoire. Je remercie aussi Ali Rezgui pour son examination de mes travaux. Je pense qu'il est aussi important de remercier le projet européen ADIGMA qui a financé mes travaux de thèse, et m'a donné davantage d'occasions de communiquer avec les experts de mon domaine.

Je voudrais aussi remercier l'ensemble des personnes du laboratoire SINUMEF pour les échanges fructueux et l'ambiance conviviale qui m'a accompagné durant ces trois années de thèse, en particulier Paola Cinnela et mes anciens collègues Bertrand Michel et Fabrice Falissard pour toutes les discussions intéressantes au tour du schéma RBC, puis Patrick Kuszla, avec qui j'ai échangé des idées sur les aspects en maillages non-structurés, et Samuel Hercus pour son aide en anglais. Je n'oublierai pas mes co-bureaux: Xavier Merle(l'oiseau chantant), Florent Margnat(le guitariste) et Xavier Gloerfelt(le superman), et les autres collègues: Frédéric Alizard, Jean-Christophe Robinet, Virginie Daru, Farid Oukaci, et les anciens collègues: Mazen Zaki, Thomas Le Garrec, Julien Berland, David Alfano, Pascal Joubert de La Motte. Je souhaite une bonne continuation pour la suite du travail des doctorants: Stefania Cherubini, Song Ge, Valérie Bar, Rocio Marin Perez, Jérémie Chicheportiche, Guillaume Aubard et Grégory Dergham.

Enfin, je remercie mes parents et ma femme pour leur soutien tout au long de cette thèse.

Introduction	7
1 Some existing numerical methods for compressible flow simulation	15
1.1 Physical models for compressible flows	15
1.2 Original RBC scheme on structured grids	18
1.3 Classical numerical methods on unstructured grids	22
2 Design principles of 2nd and 3rd-order FV-RB schemes on unstructured grids	27
2.1 Formulation for 2D steady flows	29
2.1.1 Inviscid case	29
2.1.1.1 Design of the baseline FV-RB numerical flux	29
2.1.1.2 Boundary conditions	34
2.1.2 Viscous case	35
2.1.2.1 Extension of the RB numerical flux	35
2.1.2.2 Boundary conditions	37
2.1.3 Time integration	38
2.1.3.1 Explicit strategy	38
2.1.3.2 Implicit strategy	40
2.1.4 Limiting process	43
2.2 Formulation for 3D steady flows	46
2.2.1 Cell gradient computation	46
2.2.2 Quadrature points for flux integration on cell faces	47
2.2.3 Node solution and node gradient computation stencil	47
2.2.4 Shift cell for RB scheme	48
2.3 Formulation for 2D unsteady flows	48
2.3.1 Dual-time discretization method	49
2.3.2 RB numerical flux	50
2.3.3 New sensor for the RB scheme	51
2.3.4 Adaptation of the implicit strategy	52
3 Applications of FV-RB scheme	55
3.1 Steady inviscid smooth flows	56
3.1.1 2D circular advection problem	56
3.1.2 Subsonic inviscid flow over a NACA0012 airfoil	64
3.1.3 A 3D helicoidal advection problem	68
3.1.4 Subsonic inviscid flow over a 3D airfoil body	74
3.2 Steady inviscid flows with shocks	78
3.2.1 Transonic inviscid flow over a NACA0012 airfoil	78
3.2.2 Transonic inviscid flow over the ONERA M6 wing	80
3.3 Steady viscous flows	83
3.3.1 2D advection-diffusion problem	83
3.3.2 Subsonic laminar flow over a NACA0012 airfoil	88
3.4 Unsteady flows	93
3.4.1 Smooth flow: vortex propagation problem	94

3.4.2	Flow with shocks: double Mach reflection problem	97
3.5	Conclusion	101
4	Design principles for the SV-RB scheme	103
4.1	Introduction to the Spectral Volume method	104
4.2	Strategy for time-integration	108
4.2.1	Explicit time-advancement	108
4.2.2	Matrix-free implicit method for SV	109
4.3	Coupling the SV method and the RB numerical scheme	110
4.4	Numerical results	113
4.5	Conclusion	117
5	Extension of the SV-schemes to compressible flows	119
5.1	Extension of some classical upwind schemes	120
5.1.1	Rusanov scheme	120
5.1.2	HLLC scheme	120
5.2	Extension of the RB scheme	121
5.3	Boundary conditions	121
5.3.1	General strategy	121
5.3.2	Improved treatment for a curved wall-boundary	122
5.3.2.1	Interpolation points determination	123
5.3.2.2	Physical normal vector computation	124
5.3.2.3	Numerical flux computation on the curved wall	126
5.4	Numerical assessment	126
5.4.1	Influence of the numerical flux for SV method	126
5.4.2	Comparison between SV-RB & FV-RB schemes	132
5.5	Conclusion	136
	Conclusion	139
A	Structure of code NS^3_FV	145
B	Some important issues of code NS^3_SV	149

Nomenclature

FV	Finite Volume method
SV	Spectral Volume method
RBC	Residual Based Compact scheme
RB	Residual Based scheme
FV-RB	Residual Based scheme developed in Finite Volume framework
SV-RB	Residual Based scheme developed in Spectral Volume framework
DOF	Number of Degrees Of Freedom
CPIPD	CPU time Per Iteration Per DOF

a	Sound velocity
A	Jacobian matrix of flux in x direction: f
B	Jacobian matrix of flux in y direction: g
Bn	n^{th} order boundary condition
C_l	Lift coefficient
C_d	Drag coefficient
C_m	Moment coefficient
d	Numerical dissipation
E	Total energy of unit volume
e	Internal energy of unit volume
\mathcal{F}	Physical flux vector $[f \ g \ h]^T$
f	Vector component of physical flux vector in x direction
g	Vector component of physical flux vector in y direction
h	Vector component of physical flux vector in z direction
\mathcal{H}	Numerical flux
\mathbf{H}	Solution Hessian
H	Total enthalpy of unit volume
h	Characteristic mesh size
J	Jacobian matrix of flux
L^\perp	Distance projection in a normal direction
M	Mach number
\mathbf{n}	Unit normal vector $[n_x \ n_y \ n_z]^T$
$\mathcal{O}()$	order of precision
On	n^{th} order scheme
p	Pressure
Q	Vector of primitive variables: $[\rho \ p \ u \ v \ w]^T$
q	Primitive variable

r	Residual of a system of conservation law
\mathbf{r}	Position vector $[x \ y \ z]^T$
R	Integral of residual of a system of conservation law
S	Entropy
T	Temperature
t	Time
\mathbf{t}	Unit tangent vector $[t_x \ t_y \ t_z]^T$
U	Vector of conservative variables: $[\rho \ \rho u \ \rho v \ \rho w \ \rho H]^T$
u	Flow velocity in x direction
V	Flow velocity vector
v	Flow velocity in y direction
w	Flow velocity in z direction
W	Conservative variable
x,y,z	Cartesian coordinates components
α	Flow attack angle
γ	Heat capacity ratio
Γ	Face
ξ,η	Computational domain coordinates
κ	Thermal conductivity
λ	eigen value
μ	Dynamic viscosity
ν	Kinematic viscosity
ρ	Density
τ	Viscous stress tensor
Ω	Volume
ω	Quadrature weight
ρ	Spectral radius

Contexte de la thèse

La mécanique des fluides numérique (MFN) joue un rôle de plus en plus important dans le processus de conception aérodynamique. Comparé aux outils expérimentaux, le coût des simulations numériques est relativement moins élevé et il peut donner une prédiction sur les configurations des écoulements qui sont difficiles à étudier expérimentalement. Bien que la plupart des logiciels de MFN modernes peuvent traiter les écoulements visqueux turbulents autour de géométries complexes, leur efficacité de calcul est encore basse puisque les schémas numériques utilisés sont souvent seulement d'ordre deux en maillage nonstructuré, de sorte qu'un maillage très fin est nécessaire pour obtenir un résultat suffisamment convergé en maillage, c'est à dire indépendant du maillage.

Beaucoup de codes de calcul MFN ont été développés qui utilisent les schémas d'ordre élevé en maillage structuré, par exemple : les méthodes différences finies, méthodes spectrales ou les schémas compacts basés sur le résidu (RBC). Cependant, la génération d'un maillage structuré devient délicate et demande beaucoup de temps quand la géométrie est complexe. Les techniques spéciales comme overlapping sont donc introduites afin d'améliorer la flexibilité de la génération du maillage structuré. D'un autre côté, la génération d'un maillage non-structuré autour d'une géométrie complexe est plus facile. De plus, les maillages non-structurés s'adaptent facilement pour un raffinement local, qui réduit donc largement le nombre total des éléments du maillage, comparé au raffinement global utilisé souvent en maillage structuré. Ayant constaté les avantages des maillages non-structurés, des chercheurs ont commencé à développer des méthodes numériques compatibles avec ce type de maillage, avec l'accent récent sur la précision d'ordre élevé (supérieur à l'ordre deux). Les méthodes dites Galerkin discontinue, des volumes spectraux, des différences spectrales et les schémas à distribution de résidu sont en cours de développement à l'ordre élevé en maillages non-structurés. Ces méthodes d'ordre élevé ont besoin d'un nombre réduit de degré de liberté par rapport au schéma d'ordre deux pour avoir un niveau de précision comparable, ce qui donne finalement un coût de calcul global moins élevé.

Le projet européen ADIGMA a été proposé avec l'objectif de développer des méthodes numériques d'ordre élevé innovantes pour les équations d'Euler et de Navier-Stokes en aérodynamique compressible. Ce projet a débuté en septembre 2006 et s'est terminé en décembre 2009. Dans ce projet, 22 partenaires de l'industrie, des organismes de recherche et des universités ont travaillé sur les cas tests sélectionnés afin de donner une évaluation la plus concrète possible de l'avantage d'utiliser des méthodes numériques d'ordre élevé en maillage non-structuré. Arts et Métiers ParisTech(ENSAM) a participé à ce projet à travers son Laboratoire de Simulation Numérique en Mécanique des Fluides(SINUMEF) et la présente thèse.

Parmi les équipes de recherche impliqués dans le projet ADIGMA, il y a deux catégories: ceux qui développent les méthodes de Galerkin discontinue et ceux qui travaillent en les approches basées sur le résidu(les schémas à distribution de résidu et les schémas compacts basés sur le résidu). Le présent travail appartient à la deuxième catégorie puisqu'il se consacre à développer un schéma d'ordre élevé basé sur le résidu pour la simulation numérique d'écoulements compressibles en maillage non-structuré.

Le schéma compact basé sur le résidu (RBC) a été proposé par Lerat et Corre au début de 2000. Un schéma RBC d'ordre 3 a été d'abord conçu pour résoudre les équations d'Euler et de Navier-Stokes compressibles et appliqué notamment au calcul des écoulements transsoniques autour des profils d'aile en maillages curvilignes [30] [31]. Ce schéma est compact car il utilise $3 \times 3 \times 3$ points seulement pour les équations d'Euler en $3D$. Le principe de ce schéma n'est pas de traiter chaque dérivée spatiale séparément comme dans un schéma directionnel, mais d'approcher d'une manière globale le résidu, la somme des termes dans les équations du système. Un schéma basé sur le résidu peut être exprimé uniquement en termes du résidu, et sa dissipation numérique est exprimée en termes des dérivés du résidu. Ce point de vue conduit à des schémas compacts qui ne nécessitent pas de résoudre des systèmes algébriques et possèdent de bonnes propriétés de capture de choc, sans limiteurs et sans termes de correction. Grâce à sa compacité, le schéma RBC peut être facilement implicité et inconditionnellement stable. En un maillage structuré, il est exprimé dans une formulation volumes finis "cell-centered", avec un traitement spécial pour l'irrégularité du maillage [21][33]. Les premières applications du schéma RBC d'ordre 3 pour résoudre les équations de Navier-Stokes instationnaires ont été proposées dans [21][12]. Le résidu contient la dérivé en temps et le schéma est résolu en utilisant une approche à pas de temps dual. Dans le cadre de travail du Réseau français "Recherche Aéronautique sur le Supersonique", les schémas RBC d'ordre 2 et 3 ont été appliqués aux équations RANS et URANS avec une modélisation de turbulence RMS pour calculer les écoulements avec des chocs oscillants dans les prises d'air supersoniques [38]. Puis dans le programme DGAC AITEC, ces schémas ont été mis en œuvre dans le code de calcul de l'ONERA *elsA* et appliqués à des configurations réalistes, stationnaires et instationnaires, de turbomachine. Plus récemment, les schémas RBC ont été étendu à des précisions d'ordre supérieur à 3 (voir [32][11] [14]) par une construction des expressions compactes du résidu déduite de fractions de Padé particulières. Cette approche a conduit aux schémas d'ordre 5 et 7 à l'aide de $5 \times 5 \times 5$ points seulement pour les équations d'Euler en $3D$. Parmi les schémas RBC les plus compacts pour les équations de Navier-Stokes compressibles, on peut distinguer *RBC $E5V5$* (un schéma d'ordre 5 pour les termes d'Euler et les termes visqueux) et *RBC $E7V2$* (un schéma d'ordre 7 pour les termes d'Euler et d'ordre 2 pour les termes visqueux) qui donne une préférence à l'approximation des termes d'Euler et rend la méthode bien adaptée à des calculs aéroacoustiques. Les schémas RBC d'ordre élevé ont été appliqués aux problèmes acoustiques du projet européen TurboNoise-CFD, aux simulations RANS de chocs oscillants dans des tuyères et aux simulations Euler des ondes acoustiques tournantes dans les prises d'air des moteurs d'avion. Il a été confirmé que les schémas RBC d'ordre élevé ont besoin de beaucoup moins de points que le schéma RBC d'ordre 3 pour atteindre le même niveau de précision.

Vue le succès des schémas RBC en maillages structurés, l'idée de les étendre en maillages non-structurés s'est fait jour récemment. Un prototype du schéma d'ordre deux basé sur le résidu, développé à l'aide de la méthode volumes finis en maillages non-structurés généraux, a été proposé dans [42]. Comme cette formulation n'est pas nécessairement compacte pour les écoulements visqueux, elle est dite simplement le schéma basé sur le résidu (RB).

Objectif de la thèse

En accord avec l'objectif du projet ADIGMA, le but de cette thèse est de développer une version du schéma RB d'ordre élevé pour approcher les équations régissant les écoulements compressibles, d'intégrer ce schéma dans un code de calcul MFN et d'évaluer ensuite sa performance. Plus pré-

cisément, ce travail a mis l'accent sur le développement d'un schéma RB d'ordre trois en explorant deux stratégies :

- La première stratégie, celle la plus directe, s'appuie sur une extension directe du schéma RB proposé dans le cadre de la méthode volumes finis. On atteint une précision d'ordre élevé en étendant le stencil du schéma, ce qui donne une représentation précise des variables de l'écoulement. Comme il sera illustré avec une série des cas tests (les problèmes modèles et les cas proposé par le projet ADIGMA), cette stratégie atteint sa limite pour les calculs 3D, où les besoins de mémoire de stockage dus à l'extension du stencil de schéma deviennent extrêmement grands. Cette stratégie mérite d'être explorée en raison de l'intérêt du schéma d'ordre trois par rapport au schéma d'ordre deux, qui sera montré plus tard. Mais il est difficile de continuer sur ce chemin quand ces schémas RB de volumes finis(noté FV-RB) atteignent une précision d'ordre très élevé.
- Après une étude sur les méthodes d'ordre élevé effectuée récemment en maillages non-structurés, la méthode des volumes spectraux est retenue comme la base alternative du développement du schéma RB d'ordre élevé. Dans le présent travail, à cause de nécessité de calculer beaucoup de cas tests pour le projet ADIGMA avec les schémas FV-RB, notre attention est limitée au développement des schémas RB basés sur des volumes spectraux (noté SV-RB) d'ordre deux et trois. Il est intéressant de souligner que les schémas SV-RB proposés dans ce travail constituent une bonne base pour leur extension future en plus haute précision.

En résumé, on peut dire que l'objectif à court terme de ce travail est de démontrer l'intérêt de la version d'ordre trois du schéma FV-RB, tandis que l'objectif en long terme est d'explorer la faisabilité et le potentiel du schéma SV-RB, avec une validation limitée à l'ordre trois dans cette thèse.

Organisation du mémoire

Conformément aux idées qui viennent d'être exposées, ce mémoire est divisée en deux parties principales : la première partie est consacrée au développement du schéma FV-RB d'ordre deux et trois, tandis que la seconde partie décrit le développement du schéma SV-RB d'ordre deux et trois.

La première partie de ce mémoire est divisée en trois chapitres :

- Le chapitre 1 décrit brièvement les méthodes numériques disponibles au début de cette thèse. L'accent est mis sur les principes de conception du schéma RBC initialement développé en maillages structurés et la méthode des volumes finis classique pour atteindre la précision d'ordre deux et trois en maillages non-structurés avec des schémas décentré-amont.
- Le chapitre 2 explique comment les ingrédients précédents peuvent être combinés pour donner un schéma RB d'ordre deux et trois basé sur les volumes finis (noté par FV-RB O_2 et O_3) pour les équations d'Euler et de Navier-Stokes stationnaires. Des détails sont fournis sur les techniques d'accélération de convergence vers l'état stationnaire et sur les conditions aux limites utilisées dans les cas test montrés dans le chapitre 3. Enfin, ces schémas sont étendus aux problèmes instationnaires.
- Cette partie se termine avec le chapitre 3 qui présente les résultats numériques de certains problèmes modèles et des cas test proposés par le projet ADIGMA. La présentation suit un ordre qui va de problèmes simples à des problèmes complexes : d'abord des écoulements compressibles

stationnaires sans choc d'un fluide parfait en $2D$ et $3D$ sont calculés par le schéma FV-RB $O2$ et $O3$, puis des écoulements stationnaires avec choc d'un fluide parfait sont traités par ces schémas, ensuite ces schémas sont utilisés pour calculer des écoulements visqueux stationnaires sans choc, enfin des résultats sur les écoulements instationnaires non-visqueux sont obtenus. Pour l'ensemble de ces résultats, les avantages du schéma d'ordre trois par rapport à l'ordre deux sont discutés.

La deuxième partie de ce mémoire se décompose en deux chapitres :

- Le chapitre 4 expose les principes de base de la méthode des volumes spectraux et les détails sur la façon de construire un schéma RB d'ordre deux et trois dans ce cadre. La convergence en maillage du schéma SV-RB est évaluée par un problème d'advection circulaire en $2D$ et une comparaison est faite entre les schémas RB développés à partir de deux approches différentes: FV-RB et SV-RB.
- Dans le cinquième et dernier chapitre, les schémas SV-RB sont étendus aux équations d'Euler $2D$. Comme il s'agit d'applications aérodynamiques, on a besoin de soins particuliers pour le traitement des parois solides. En particulier, l'approche SV s'appuyant sur une augmentation des degrés de liberté à l'intérieur d'une cellule du maillage, une représentation d'ordre élevé de la géométrie curviligne est nécessaire pour assurer la précision d'ordre élevé du schéma SV-RB en paroi. Une méthode pour représenter la paroi courbée à l'ordre élevé est donc mise en œuvre et évaluée. En outre, pour les applications complexes d'écoulement compressible, certaines techniques d'accélération de convergence ont été également utilisées pour l'intégration en temps associée à la discrétisation en espace. Une phase implicite dite sans matrice utilisée par le schéma FV-RB est ainsi adaptée pour les schémas SV-RB. On a effectué les comparaisons entre l'approche SV couplée avec le flux numérique d'un schéma décentré-amont standard et le schéma SV-RB, ainsi qu'entre le schéma FV-RB et SV-RB lors du calcul d'écoulements non-visqueux subsoniques autour d'un cylindre et d'un profil NACA0012.

Enfin ce mémoire s'achève par quelques conclusions et des perspectives à court et à long terme pour utiliser les apports originaux de ce travail.

Background of the thesis

Computational Fluid Dynamics(CFD) plays an increasingly important role in the aerodynamic design process. Compared to experimental tools, it is relatively less expensive and in some occasions it can yield a prediction on flow configurations which would be hard to study through experiments. Although most of up-to-date CFD softwares can deal fairly well with turbulent viscous flows around complex geometries, their computational efficiency is still low because the underlying numerical schemes are often second-order accurate only on unstructured grids, where the use of a very fine mesh is necessary to achieve grid-converged hence grid-independent results.

A number of CFD tools have been developed, which rely on high order schemes but remain limited to the use of structured grids, for instance : finite difference methods, spectral methods [28] or Residual based compact (RBC) schemes [30]. However, the structured mesh generation becomes particularly tricky and time-consuming when a complex geometry is immersed in the flow domain. Specific techniques such as overlapping grids for example were therefore introduced to improve the flexibility of structured grid generation. On the other hand, the mesh generation of unstructured grids around a complex body remains easier to handle. Moreover, such an unstructured grid can be easily adapted for the purpose of local refinement, which results in a large reduction of the total number of mesh elements, compared to the global mesh refinement which often takes place in the structured grid generation process. Having perceived the advantages of unstructured grids, researchers have started to develop numerical methods well fitted for such grids, with a recent emphasis on high order (higher than second-order) accuracy. Discontinuous Galerkin method [7], spectral volume method [47], spectral difference method [34] and residual distribution method[15] are all designed to yield high order numerical solutions on unstructured grids. These high order methods require a reduced number of degrees of freedom with respect to second-order schemes for achieving a comparable level of accuracy, which results in a reduction of the global computational cost.

The ADIGMA European project has been proposed with the main objective to develop innovative high order methods for compressible flow equations used in the context of aerodynamic applications and aircraft design. This projet started on Septembre 2006 and is completed on December 2009. In this project, 22 collaborators from industry, research organisms and universities have worked on selected test problems in order to produce an as much as possible objective assessment of the benefit expected from using high order numerical methods on unstructured grids. The Arts et Métiers ParisTech(ENSAM) was scientifically involved in the project through its Laboratory of Numerical Flow Simulation (SINUMEF) and the present thesis.

Among the research teams involved in the ADIGMA project, two main categories have emerged : those developing discontinuous Galerkin methods and those working on residual-based approaches (residual distribution methods, residual based compact schemes). Clearly, the present works belongs to this second family since it is devoted to the development of a high order residual-based scheme for the simulation of compressible flows on unstructured grids.

The Residual Based Compact (RBC) scheme was proposed by Lerat and Corre at the beginning of 2000. A 3rd order RBC scheme was firstly designed for solving the compressible Euler and Navier-

Stokes equations and applied notably to the computation of transonic flows around airfoils on the curvilinear meshes [30][31]. This scheme is compact since it involves $3 \times 3 \times 3$ points only for the 3D Euler equations. The design principle is not to treat each spatial derivative separately as in a directional scheme, but to approach the residual, i.e. the sum of the terms in the balance equations, in a global way. A residual based scheme can be expressed only in terms of the residual, and for the numerical dissipation, in terms of derivatives of the residual. This point of view leads to compact schemes which do not require the solution of algebraic systems and possess good shock capturing properties without limiter or correction terms. Owing to their compactness, RBC scheme can be easily made implicit and unconditionally stable. On a structured mesh, it is expressed in a cell-centered finite volume formulation, with a special treatment for mesh irregularity [21][33]. First applications of 3rd order RBC scheme to the solution of the unsteady Navier-Stokes equations were proposed in [21][12]. The residual contains the time derivative and the scheme is solved using a dual-time approach. In the frame work of the French Net "Recherche Aéronautique sur le Supersonique", 2nd and 3rd order RBC scheme have been applied to the RANS and URANS equations with a RSM turbulence modeling for computing flows with oscillating shocks in supersonic air intakes [38]. And then in the DGAC AITEC program, these schemes have been implemented in the ONERA *elsA* code and applied to a variety of realistic steady and unsteady turbomachinery configurations. More recently, the RBC schemes have been extended to accuracy orders higher than 3 (see [32][11] [14]) by constructing compact residual expressions deduced from particular Pade fractions. This approach has led to 5th and 7th order accurate schemes by using $5 \times 5 \times 5$ points only for the 3D Euler equations. Among the most compact RBC schemes for the compressible Navier-Stokes equations, one can distinguish *RBC $E5V5$* (a 5th order scheme for the Euler terms and also for the viscous terms) and *RBC $E7V2$* (a 7th order scheme for the Euler terms and 2nd order for the viscous terms), which gives a preference to the approximation of the Euler terms and makes the method well adapted to aeroacoustics computations. High order RBC schemes have been applied to acoustic problems of the TurboNoise-CFD European Program, RANS simulations of oscillating shocks in nozzles and Euler simulations of spinning acoustic waves in aircraft engine intakes. It has been confirmed that high order RBC schemes need much less points than 3rd order RBC scheme in order to reach the same accuracy level.

Having seen the successful development of RBC schemes on structured grids, the extension of this scheme to unstructured grids has been launched recently. A prototype of the second-order residual-based scheme developed in finite volume method for general unstructured grids has been proposed in [42]. Note that this formulation is not necessarily compact anymore for viscous flows, it will be denoted by RB (Residual Based) scheme.

Objective of the thesis

To be in line with the general purpose pursued by the ADIGMA project, the objective of this thesis is to propose a high order version of the RB scheme for approximating the compressible flow equations on unstructured grids, to implement this scheme in a CFD code and to assess its performances. In reality, this work has been focusing on the development of third-order RB schemes, where two strategies have been followed :

- The first strategy, the most straightforward one, relies on a direct extension of the RB scheme designed in the general framework of a finite-volume approach. Basically, high order accuracy

is achieved by extending the scheme stencil, thus allowing a more accurate representation of the flow variables. As will be clearly demonstrated on a variety of test-problems (model ones and flow problems retained as benchmark test-problems in the ADIGMA project), this strategy reaches its limits for $3D$ computations, where the memory requirements induced by the extension of the scheme's stencil tend to become unacceptably large. This strategy deserves to be explored regarding the interest of third-order scheme over second-order, which will be illustrated. But this path remains a dead-end when this finite volume based RB scheme (denoted by FV-RB schemes) goes to a very high order accuracy.

- After some studies on currently developed high order methods for unstructured grids, the spectral volume method has been retained as the alternative basis for the development of high order RB scheme. In the present work, because of the time-constraints induced by the computations of many test cases for the ADIGMA project by using the FV-RB schemes, our attention has been restricted to the development of second and third-order spectral volume based RB schemes (denoted by SV-RB schemes). It is necessary to emphasize however the spectral volume based schemes proposed in this work lays a good foundation for future higher-order extensions.

In summary, it could be said that the short-term objective of this work has been to demonstrate the interest of a third-order version of the FV-RB scheme while the long-term objective has been focused on exploring the feasibility and the potential of the SV-RB scheme, with an assessment limited to third-order in this work.

Organization of the thesis

Following the ideas which have just been exposed, the thesis is divided into two main parts : the first part is devoted to the development of second and third-order FV-RB schemes, while the second part deals with the development of second and third-order SV-RB schemes.

The first part of this thesis is divided into three chapters :

- Chapter 1 briefly reviews the numerical techniques available at the start of this thesis. The emphasis is put on the design principles of the RBC scheme initially developed on structured grids and the classical finite-volume method for achieving second or third-order accuracy on unstructured grids with general upwind schemes.
- Chapter 2 explains how the previous ingredients can be combined to yield a second and third-order finite volume based RB scheme (denoted by FV-RB $O2$ and $O3$) for steady Euler and Navier-Stokes equations. Details are provided on the convergence acceleration techniques to steady state and on boundary conditions used in test cases showed in chapter 3. And then these schemes are extended for unsteady problems.
- This part closes with chapter 3, which presents numerical results of some model problems and test cases proposed by the ADIGMA project. The presentation follows an order from simple problems to complex problems : at first $2D$ and $3D$ steady inviscid smooth compressible flow problems are computed by the FV-RB $O2$ and $O3$ schemes; then steady inviscid flow problems with shocks are treated by these schemes; next these schemes are used to deal with steady

viscous smooth flows; finally results on unsteady inviscid flows are obtained with these schemes. For all of these results, the advantages of the third-order scheme over the second-order one are discussed.

The second part of the thesis is organized in two chapters :

- Chapter 4 describes the basic principles of the spectral volume method and details on how a second and third-order RB scheme can be designed within this framework. The grid convergence of the developed SV-RB schemes is assessed on a $2D$ circular advection problem and a comparison is made between the RB schemes from two different approaches: FV-RB and SV-RB schemes.
- In the final chapter, Chapter 5, the SV-RB schemes are extended for the $2D$ Euler equations. Since the aeronautical applications of the proposed methods involve flows over solid bodies, special care must be given to the wall boundary treatment. In particular, because the SV approach relies on an increase of the degrees of freedom inside a grid cell, a high order representation of the curved geometry is necessary to ensure the high order accuracy of the SV-RB scheme on solid walls. A method to represent the curved boundary at high order is therefore implemented and tested. Moreover, for complex compressible flow applications, some convergence acceleration techniques also have to be given to the time integration associated with the space discretization. A matrix-free implicit scheme used for FV-RB scheme is thus adapted for the SV-RB schemes. Comparisons are performed between the SV approach coupled with a standard upwind numerical flux and the SV-RB scheme as well as between the FV-RB and SV-RB approaches when computing the $2D$ subsonic inviscid flow over a cylinder and a NACA0012 airfoil.

At last the thesis closes with some conclusions and perspectives on the short-term and long-term use of the findings in this work.

Some existing numerical methods for compressible flow simulation

French Ce chapitre préliminaire donne une présentation concise des méthodes existantes pour la simulation numérique des écoulements compressibles. L'accent est mis sur les ingrédients qui seront utilisées pour avoir le schéma FV-RB d'ordre deux et trois en maillages non-structurés généraux. Précisément, il s'agit de la description du schéma compact basé sur le résidu en maillages structurés et de la méthode des volumes finis classique d'ordre trois en maillages non-structurés. La façon de combiner ces ingrédients est l'objet du chapitre suivant.

English This introductory chapter gives a brief review of some methods already in use for compressible flow simulations. Our focus is concentrating on those ingredients which will be used in the next chapter to derive a second and third-order FV-RB scheme on general unstructured grids. Precisely it is on the description of the well-established residual-based compact scheme on structured grids and on the description of classical finite volume method for achieving third-order accuracy on unstructured grids. The way on how to combine these ingredients will be the topic of the next chapter.

1.1 Physical models for compressible flows

A general model for describing the movement of a compressible flow is the system of the Navier-Stokes equations, expressing the conservation of mass, momentum and total energy for a variable-density flow. If 3D Cartesian coordinates are used, this system of conservation laws can be put in the local form :

$$\frac{\partial U}{\partial t} + \frac{\partial f}{\partial x} + \frac{\partial g}{\partial y} + \frac{\partial h}{\partial z} = 0, \quad (1.1)$$

where U is the vector of conservative variables : $U = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho E \end{pmatrix}$ with ρ the density, (u, v, w) the

velocity components and E the specific total energy, $E = e + \frac{1}{2}(u^2 + v^2 + w^2)$, e being the specific internal energy. The physical flux vectors in the x , y and z directions can be decomposed into a convective part depending solely on U and a diffusive part depending on U and its gradient ∇U :

$$f(U, \nabla U) = f^E(U) - f^V(U, \nabla U), \quad g(U, \nabla U) = g^E(U) - g^V(U, \nabla U), \quad h(U, \nabla U) = h^E(U) - h^V(U, \nabla U),$$

where the inviscid (Euler) fluxes are given by :

$$f^E(U) = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ \rho uH \end{pmatrix}, g^E(U) = \begin{pmatrix} \rho v \\ \rho vu \\ \rho v^2 + p \\ \rho vw \\ \rho vH \end{pmatrix}, h^E(U) = \begin{pmatrix} \rho w \\ \rho wu \\ \rho wv \\ \rho w^2 + p \\ \rho wH \end{pmatrix}, \quad (1.2)$$

and the viscous fluxes are expressed as:

$$f^V(U, \nabla U) = \begin{pmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ \tau_{xz} \\ u\tau_{xx} + v\tau_{xy} + w\tau_{xz} + \kappa \frac{\partial T}{\partial x} \end{pmatrix}, g^V(U, \nabla U) = \begin{pmatrix} 0 \\ \tau_{yx} \\ \tau_{yy} \\ \tau_{yz} \\ u\tau_{yx} + v\tau_{yy} + w\tau_{yz} + \kappa \frac{\partial T}{\partial y} \end{pmatrix},$$

$$h^V(U, \nabla U) = \begin{pmatrix} 0 \\ \tau_{zx} \\ \tau_{zy} \\ \tau_{zz} \\ u\tau_{zx} + v\tau_{zy} + w\tau_{zz} + \kappa \frac{\partial T}{\partial z} \end{pmatrix}, \quad (1.3)$$

with p the pressure, T the temperature and H the total enthalpy, $H = E + p/\rho$. The system is closed with some thermodynamic equations of state relating p , T , ρ and e plus some constitutive laws for the thermal conductivity coefficient κ and the stress tensor components τ_{xx} , $\tau_{xy} = \tau_{yx}$, τ_{yy} . For a Newtonian fluid, the viscous stress tensor is such that :

$$\tau_{i,j} = \mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3} \delta_{i,j} \frac{\partial u_k}{\partial x_k} \right). \quad (1.4)$$

where $x_1 = x$, $x_2 = y$, $x_3 = z$, $u_1 = u$, $u_2 = v$, $u_3 = w$, $\delta_{i,j}$ is the Kronecker function; the fluid dynamic viscosity μ is determined by the Sutherland law. The thermal conductivity κ and μ are related through the (constant) Prandtl number : $Pr = \mu C_p / \kappa$ with C_p the (constant) specific heat coefficient at constant pressure.

In this thesis the perfect gas is chosen as fluid, the thermal and caloric equations of state used to describe its thermodynamic behavior are : $p = \rho r T$ and $e = C_v T$ with r the gas constant such that $r = C_p - C_v$ and $\gamma = C_p / C_v$ (C_v is the specific heat coefficient at constant volume). Gathering these relationships also yields $p = (\gamma - 1)\rho e$.

When the viscous effects can be neglected, the Navier-Stokes equations simplify into the Euler hyperbolic system of conservation laws :

$$\frac{\partial U}{\partial t} + \frac{\partial f^E(U)}{\partial x} + \frac{\partial g^E(U)}{\partial y} + \frac{\partial h^E(U)}{\partial z} = 0. \quad (1.5)$$

In order to perform the analysis of the error order for the numerical schemes developed in this work, it will be also convenient to deal with some simplified model problems such as :

- pure scalar advection :

$$W_t + a_1 W_x + a_2 W_y + a_3 W_z = 0, \quad (1.6)$$

with W the conservative variable and a_1, a_2, a_3 the advection coefficients in the x, y and z directions

- scalar advection-diffusion :

$$W_t + a_1 W_x + a_2 W_y + a_3 W_z = \nu_1 W_{xx} + \nu_2 W_{yy} + \nu_3 W_{zz} + \nu_{12} W_{xy} + \nu_{13} W_{xz} + \nu_{23} W_{yz}. \quad (1.7)$$

with ν_k and ν_{ij} viscosity or diffusion coefficients.

In the present work, we will systematically use the non-dimensional form of the Euler and Navier-Stokes equations. Since our interest will be focused on external flows over airfoils or wings, the typical situation will be that of uniform upstream far-field flow conditions, ρ_∞, U_∞ and p_∞ . The quantity V_∞ denotes the norm of the upstream far-field velocity; in 2D the upstream far-field flow direction is entirely defined by the angle of attack α . By using these conditions as a reference state to normalize the density, velocity and pressure, we introduce the non-dimensional quantities : $\bar{\rho} = \frac{\rho}{\rho_\infty}, \bar{u} = \frac{u}{V_\infty}, \bar{v} = \frac{v}{V_\infty}, \bar{p} = \frac{p}{p_\infty}$. By taking typically the airfoil chord c as a reference length (in 2D) and a reference time interval $t^* = c/U_\infty$, it is immediate to show that the non-dimensional Euler equations on the non-dimensional quantities $\bar{\rho}, \bar{u}, \bar{v}, \bar{p}$ are formally identical to (1.5). The sole difference lies in the far-field inflow condition which can be expressed using only the Mach number M_∞ based on far-field conditions and the angle of attack :

$$\bar{\rho}_\infty = 1 \quad , \quad \bar{u}_\infty = \cos(\alpha) \quad , \quad \bar{v}_\infty = \sin(\alpha) \quad , \quad p_\infty = \frac{1}{\gamma M_\infty^2}.$$

Note the system has been closed with the non-dimensional relationship between (non-dimensional) pressure, density and specific internal energy : $\bar{p} = (\gamma - 1)\bar{\rho}\bar{e}$; since $\bar{E} = \bar{e} + \frac{1}{2}(\bar{u}^2 + \bar{v}^2)$, this relationship can also be expressed as : $\bar{p}(U) = (\gamma - 1)\bar{\rho}(\bar{E} - \frac{1}{2}(\bar{u}^2 + \bar{v}^2))$.

When dealing with the (2D) Navier-Stokes equations, the non-dimensional form of the system of conservation laws involves the Reynolds number based on the far-field fluid properties ($\rho_\infty, U_\infty, \mu_\infty$) and the reference length (typically an airfoil chord for external aerodynamic flow problems)

$$Re_{\infty,c} = Re = \frac{\rho_\infty U_\infty c}{\mu_\infty} :$$

$$\frac{\partial U}{\partial t} + \frac{\partial(f^E(U) - f^V(U, \nabla U))}{\partial x} + \frac{\partial(g^E(U) - g^V(U, \nabla U))}{\partial y} = 0, \quad (1.8)$$

with

$$U = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho E \end{pmatrix}, \quad f^E(U) = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uH \end{pmatrix}, \quad g^E(U) = \begin{pmatrix} \rho v \\ \rho vu \\ \rho v^2 + p \\ \rho vH \end{pmatrix},$$

and

$$f^V(U, \nabla U) = \begin{pmatrix} 0 \\ \frac{1}{Re} \tau_{xx} \\ \frac{1}{Re} \tau_{xy} \\ u\tau_{xx} + v\tau_{xy} + \frac{\mu}{Re} \frac{\gamma}{Pr} \frac{\partial e}{\partial x} \end{pmatrix}, \quad g^V(U, \nabla U) = \begin{pmatrix} 0 \\ \frac{1}{Re} \tau_{yx} \\ \frac{1}{Re} \tau_{yy} \\ u\tau_{yx} + v\tau_{yy} + \frac{\mu}{Re} \frac{\gamma}{Pr} \frac{\partial e}{\partial y} \end{pmatrix}.$$

where the bar on top of the non-dimensional variables has been omitted to simplify the expression of this non-dimensional form of the Navier-Stokes equations.

1.2 Original RBC scheme on structured grids

The original residual based compact (RBC) scheme has been designed in a finite difference framework. This baseline formulation of the RBC scheme is recalled in this section, since it will be the starting point of our development on unstructured grids. Let us consider the following general two-dimensional system of conservation laws (which could be for instance the Euler equations) :

$$\frac{\partial U}{\partial t} + \frac{\partial f(U)}{\partial x} + \frac{\partial g(U)}{\partial y} = 0, \quad (1.9)$$

Using a general finite difference approach on a uniform Cartesian grid, this system is discretized at each grid point as :

$$\frac{\partial U_{i,j}}{\partial t} + \frac{\delta_1 \mathcal{H}_{i,j}^1}{\delta x} + \frac{\delta_2 \mathcal{H}_{i,j}^2}{\delta y} = 0, \quad (1.10)$$

where subscript 1 and 2 will systematically refer to the space directions x and y respectively; δ_1, δ_2 are difference operators over one grid cell, that is such that :

$$\begin{aligned} \delta_1 \mathcal{H}_{i,j}^1 &= \mathcal{H}_{i+1/2,j}^1 - \mathcal{H}_{i-1/2,j}^1, \\ \delta_2 \mathcal{H}_{i,j}^2 &= \mathcal{H}_{i,j+1/2}^2 - \mathcal{H}_{i,j-1/2}^2, \end{aligned} \quad (1.11)$$

with δx (resp. δy) the mesh size in the x (resp. y) space-direction. A conservative discretization scheme for (1.9) is uniquely defined by the expression of the numerical fluxes \mathcal{H}^1 and \mathcal{H}^2 on the respective interfaces $\Gamma_{i\pm\frac{1}{2},j}$ and $\Gamma_{i,j\pm\frac{1}{2}}$. Let us focus on the first grid direction and the interface $\Gamma_{i+\frac{1}{2},j}$ (see Figure 1.1); the RBC numerical flux on this face can be expressed as :

$$\mathcal{H}_{i+1/2,j}^1 = (\mathcal{H}^1)_{i+1/2,j}^c - d_{i+1/2,j}^1 = \mu_1 f_{i+1/2,j} - \frac{1}{2} \delta x \Phi_1 \tilde{r}_{i+1/2,j}^1, \quad (1.12)$$

where μ_1 is the arithmetic average operator over one grid cell in the x direction :

$$\mu_1 f_{i+1/2,j} = \frac{1}{2} (f_{i,j} + f_{i+1,j}), \quad (1.13)$$

Φ_1 is a $\mathcal{O}(1)$ dissipation coefficient which ensures the term $d_{i+1/2,j}^1$ to be dissipative to the numerical flux; \tilde{r}^1 is a centered approximation of the residual $r = f_x + g_y$ associated with system (1.9), computed on the face $\Gamma_{i+\frac{1}{2},j}$, which plays a key role for the accuracy and robustness of the RBC scheme.

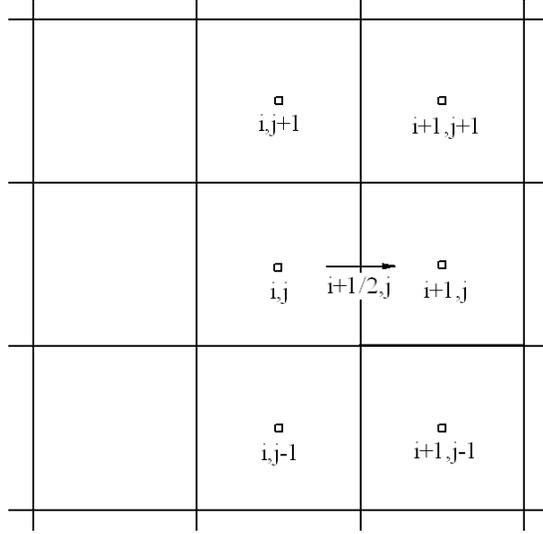


Figure 1.1: **The RBC numerical flux at face center $i + \frac{1}{2}, j$ is computed using the compact stencil made of the 6 grids points $(i, j - 1)$, (i, j) , $(i, j + 1)$, $(i + 1, j + 1)$, $(i + 1, j)$ and $(i + 1, j - 1)$.**

In the context of a structured grid discretization process, this residual approximation on the face $\Gamma_{i+1/2, j}$ is computed using the simply centered difference formula :

$$\tilde{r}_{i+\frac{1}{2}, j}^1 = \left(\frac{\delta_1 f}{\delta x} + \frac{\delta_2 \mu_1 \mu_2 g}{\delta y} \right)_{i+\frac{1}{2}, j}. \quad (1.14)$$

Similarly, the numerical flux on the interface $\Gamma_{i, j+1/2}$ is given by the formula :

$$\mathcal{H}_{i, j+1/2}^2 = (\mathcal{H}^2)_{i, j+1/2}^c - d_{i, j+1/2}^2 = \mu_2 g_{i, j+1/2} - \frac{1}{2} \delta y \Phi_2 \tilde{r}_{i, j+1/2}^2, \quad (1.15)$$

where the centered approximation of residual on the face $\Gamma_{i, j+\frac{1}{2}}$ is given by:

$$\tilde{r}_{i, j+1/2}^2 = \left(\frac{\delta_1 \mu_1 \mu_2 f}{\delta x} + \frac{\delta_2 g}{\delta y} \right)_{i, j+1/2}. \quad (1.16)$$

and Φ_2 is also a $\mathcal{O}(1)$ dissipation coefficient, playing the same role as Φ_1 . The way to properly design these dissipation coefficients has been initially proposed in [24], when building the so-called characteristic time-step scheme from the two-dimensional Roe and Lax-Wendroff schemes. Details of the derivation are not reproduced here and only the final formula for these coefficients is provided. If f and g are scalar fluxes with associated wavespeeds $A(U) = \frac{\partial f}{\partial U}$, $B(U) = \frac{\partial g}{\partial U}$, the dissipation coefficients are then computed as :

$$\begin{aligned} \Phi_1 &= \text{sign}(A)\varphi, \\ \Phi_2 &= \text{sign}(B)\psi, \end{aligned} \quad (1.17)$$

where φ and ψ are given by:

$$\begin{aligned} \varphi &= \min\left(1, \frac{1}{\alpha}\right), \\ \psi &= \min(1, \alpha), \end{aligned} \quad (1.18)$$

with a parameter α defined as the local advection direction with respect to the mesh :

$$\alpha = \frac{\delta x |B|}{\delta y |A|}. \quad (1.19)$$

Note that when the advection direction is aligned with the x direction, $|B| = 0$ and $\alpha = 0$ so that $\Phi_1 = 1$ and $\Phi_2 = 0$, which means that the numerical dissipation is added only in the x direction (similarly when $|A| = 0$, $\Phi_1 = 0$ and $\Phi_2 = 1$ yield a numerical dissipation in the y -direction only). In fact, for $1D$ problems, the RBC scheme becomes the Roe scheme, provided the Roe average is used to compute the quantities $sign(A)$, $sign(B)$, φ , ψ on each face in formula (1.17), (1.18). This choice of evaluation for the dissipation coefficients will be systematically used through this work.

To conclude this short description of the existing second-order RBC scheme, we briefly review the key points regarding the accuracy properties of this scheme. Straightforward Taylor developments allow to write (with $h = \delta x = \delta y$ if the Cartesian grid is supposed uniform) :

$$\frac{\delta_1 \mathcal{H}_{i,j}^1}{\delta x} = (f_x)_{i,j} + \mathcal{O}(h^2) - \frac{h}{2}(\Phi_1 r)_x + \mathcal{O}(h^3), \quad (1.20)$$

$$\frac{\delta_2 \mathcal{H}_{i,j}^2}{\delta y} = (g_y)_{i,j} + \mathcal{O}(h^2) - \frac{h}{2}(\Phi_2 r)_y + \mathcal{O}(h^3), \quad (1.21)$$

which leads to a total truncation error at point (i, j) (this index is omitted for the sake of simplicity) given by:

$$\epsilon^{RBC \ O(2)} = U_t + r + \mathcal{O}(h^2) - \frac{h}{2}(\Phi_1 r)_x - \frac{h}{2}(\Phi_2 r)_y + \mathcal{O}(h^3). \quad (1.22)$$

When computing a steady solution, $U_t \equiv 0$ and $r \equiv 0$ so that $\epsilon^{RBC \ O(2)} = \mathcal{O}(h^2)$ making the RBC scheme a second-order approximation to the system of conservation laws on a globally 3×3 -point stencil.

Note that in the same Cartesian grid, a conventional upwind scheme based on a MUSCL-type variable reconstruction would need a 5-point support in each space direction to achieve the second-order accuracy. The more compact support of the RBC scheme makes the treatment of boundary conditions simpler and also yields a better convergence rate to reach steady state when this explicit RBC scheme is coupled with a simple first-order upwind implicit scheme (see [13]).

On this same 3×3 -point stencil, a third-order RBC scheme can be derived [31] by using the residual vanishing at steady-state to get rid of the second-order error in (1.22). The basic idea for that extension is to consider as a starting point the second-order RBC scheme written as :

$$U_t + \tilde{r}_{i,j}^0 = \frac{1}{2} [\delta_1(\Phi_1 \tilde{r}^1) + \delta_2(\Phi_2 \tilde{r}^2)]_{i,j}, \quad (1.23)$$

Since the right-hand side of the above scheme has been shown to be $\mathcal{O}(h^3)$ at steady-state, third-order accuracy is achieved if an at least third-order approximation for $\tilde{r}_{i,j}^0$ is derived. On structured grids, the basic idea is to use the compact Pade formula in a first step and to take advantage of the

residual vanishing at steady-state in a second step. Since $r = f_x + g_y$, $\tilde{r}_{i,j}^0$ can be approximated at fourth-order by using usual Pade fractions :

$$(f_x + g_y)_{i,j} + \mathcal{O}(h^4) = \frac{I}{I + \frac{1}{6}\delta_1^2} \frac{\delta_1 \mu_1 f_{i,j}}{h} + \frac{I}{I + \frac{1}{6}\delta_2^2} \frac{\delta_2 \mu_2 g_{i,j}}{h}, \quad (1.24)$$

These formulas needs to solve a linear algebraic system in each space direction, which leads to a computation cost higher than an incompact scheme. Therefore an idea in [30] says that the focus should be on finding a small global error rather than discretizing each spatial derivative at high order. More precisely, f_x and g_y are approximated separately to second-order but the total error term can be expressed with a spatial derivative of the residual $r = f_x + g_y$, which will vanish at the steady-state. This idea can be expressed as :

$$\tilde{r}_{i,j}^0 = (I + \frac{1}{6}\delta_2^2) \frac{\delta_1 \mu_1 f_{i,j}}{h} + (I + \frac{1}{6}\delta_1^2) \frac{\delta_2 \mu_2 g_{i,j}}{h}, \quad (1.25)$$

where the computation of the residual $\tilde{r}_{i,j}^0$ at the cell center makes use of the 3×3 -point stencil previously introduced for computing the residual-based dissipation. It is to note that this compact formula is derived from the linear algebra usually associated with the use of Pade fractions. The numerical fluxes associated with this choice of $\tilde{r}_{i,j}^0$ (and the unchanged formulation of the residual-based dissipation) read :

$$\begin{aligned} \mathcal{H}_{i+1/2,j}^1 &= (\mathcal{H}^1)_{i+1/2,j}^c - d_{i+1/2,j}^1 = (I + \frac{1}{6}\delta_2^2) \mu_1 f_{i+1/2,j} - \frac{1}{2} \delta x \Phi_1 \tilde{r}_{i+1/2,j}^1, \\ \mathcal{H}_{i,j+1/2}^2 &= (\mathcal{H}^2)_{i,j+1/2}^c - d_{i,j+1/2}^2 = (I + \frac{1}{6}\delta_1^2) \mu_2 g_{i+1/2,j} - \frac{1}{2} \delta y \Phi_2 \tilde{r}_{i,j+1/2}^2, \end{aligned} \quad (1.26)$$

The resulting truncation error is given by :

$$\epsilon^{RBC \ O(3)} = U_t + r + \frac{h^2}{6} r_{xx} + \frac{h^2}{6} r_{yy} + \mathcal{O}(h^4) - \frac{h}{2} (\Phi_1 r)_x - \frac{h}{2} (\Phi_2 r)_y + \mathcal{O}(h^3). \quad (1.27)$$

so that at steady-state $\epsilon^{RBC \ O(3)} = \mathcal{O}(h^3)$. A usual upwind scheme based on purely directional approximation of f_x and g_y using MUSCL-type variable reconstruction would need a 5-point stencil in each grid direction to achieve the same third-order accuracy.

These design principles have been generalized in [32][11] to derive very high order (fifth, seventh) residual-based compact schemes on a 5×5 -point stencil. Extension to the compressible Navier-Stokes equations has also been performed in [32][14]; note that in that case care must be taken to derive specific Pade-type formula for first and second-derivatives that allow simplifications of the Pade fractions and avoid any linear algebra. Clearly, such a strategy to achieve high order accuracy crucially depends on the use of structured grids. Since the objective of the present work is to extend the RBC scheme on unstructured grids, we shall not venture further along this way. Instead, we will focus on the use of a finite-volume framework on general unstructured grids and try to insert the main features of our residual-based scheme inside this framework. To this effect, before performing this combination of residual-based approach and finite-volume technique in the next chapter, we briefly review in the next section some key ingredients of standard numerical methods on unstructured grids.

1.3 Classical numerical methods on unstructured grids

We restrict our discussion to the finite-volume method, which is based on the integral formulation of conservation laws on general unstructured grids. The finite-volume method can be either vertex-centered or cell-centered. In the vertex-centered approach, the control volume on which the system of equations is discretized is built around the vertices (nodes) of the grid while in the cell-centered approach, the grid cell itself plays the role of control volume (see for instance [4]). Only the cell-centered finite-volume formulation will be used in this work; this choice has some important consequences on the formulation of the wall boundary conditions in the case of aerodynamic flows around obstacles (airfoils, wings ...), which will be discussed later in this work.

Let us start our presentation of the finite-volume approximation of the Navier-Stokes equations from the compact two-dimensional formulation :

$$\frac{\partial U}{\partial t} + \nabla \cdot \mathcal{F}(U, \nabla U) = 0, \quad (1.28)$$

where \mathcal{F} is the flux vector with components f, g in the x, y directions respectively. On a general unstructured mesh, the cell-centered finite-volume discretization of (1.28) takes the form :

$$\frac{\partial U_i}{\partial t} + \frac{1}{|\Omega_i|} \int_{\Omega_i} \nabla \cdot \mathcal{F} d\Omega = 0, \quad (1.29)$$

where U_i is the cell average solution defined at the centroid of the mesh cell Ω_i , $|\Omega_i|$ denotes the surface of this grid cell. By using Green-Gauss theorem, this equation can be rewritten as :

$$\frac{\partial U_i}{\partial t} + \frac{1}{|\Omega_i|} \sum_{k \in \mathcal{I}(\Omega_i)} \int_{\Gamma_{i,k}} \mathcal{F} \cdot \mathbf{n} d\Gamma = 0, \quad (1.30)$$

with $\Gamma_{i,k}$ the k -th face of cell Ω_i , $|\Gamma_{i,k}|$ the length of this face and \mathbf{n} the face normal unit vector pointing outside the control cell. The set of faces forming the cell Ω_i is denoted by $\mathcal{I}(\Omega_i)$.

The integral of physical flux projection onto the face normal direction can be computed approximately by using a Gauss quadrature formula along each face of the control cell :

$$\int_{\Gamma_{i,k}} \mathcal{F} \cdot \mathbf{n} d\Gamma \simeq \sum_{g=1}^{N_g} \omega_g (\mathcal{H}_{i,k})_g |\Gamma_{i,k}|, \quad (1.31)$$

with g the index of the Gauss quadrature point on the face $\Gamma_{i,k}$, ω_g its associated quadrature weight; $N_g = \text{Integer}((p+1)/2)$ is the number of quadrature points along the face which is necessary to achieve an accuracy of order p - in particular, for a second order scheme ($p=2$) $N_g=1$ that is a single Gauss point on each face (midpoint of each face, see Figure 1.2) is sufficient to ensure second-order accuracy, as long as a polynomial of sufficient order (see below) is used to represent the solution in the cell; $(\mathcal{H}_{i,k})_g$ denotes the numerical flux at the Gauss point g , approximating the normal flux $\mathcal{F} \cdot \mathbf{n}$. In the general viscous case we choose to consider, this numerical flux includes two parts : $(\mathcal{H}_{i,k})_g = (\mathcal{H}_{i,k}^E)_g - (\mathcal{H}_{i,k}^V)_g$ where the inviscid numerical flux $(\mathcal{H}_{i,k}^E)_g$ approximating the Euler flux $\mathcal{F}^E(U) \cdot \mathbf{n} = (\mathcal{F}^E)^\perp$ is typically computed using an approximate Riemann solver, such as Roe, HLLC or AUSM+ schemes, while the viscous numerical flux $(\mathcal{H}_{i,k}^V)_g$ approximating the diffusive physical

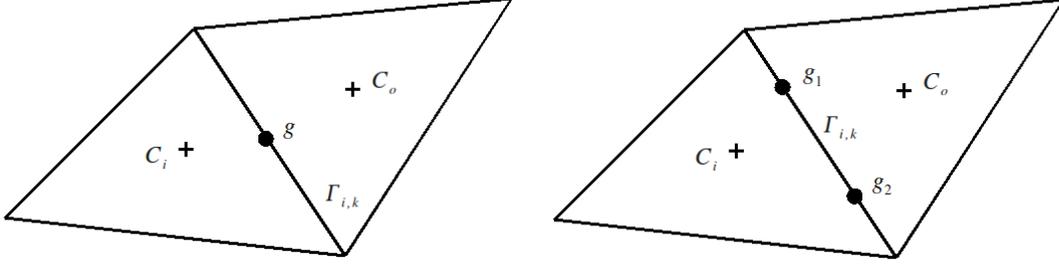


Figure 1.2: **Gauss quadrature point on a face of triangle cell 1** quadrature point, the face center g is used by the second-order scheme (left), 2 quadrature points (g_1 and g_2) are used by the third-order scheme (right).

flux $\mathcal{F}^V(U, \nabla U) \cdot \mathbf{n} = (\mathcal{F}^V)^\perp$ is computed by a simply centered approximation.

The inviscid numerical flux of a conventional upwind scheme typically reads:

$$(\mathcal{H}_{i,k}^E)_g = \mathcal{H}^E((U_{i,k}^L)_g, (U_{i,k}^R)_g; \mathbf{n}_{i,k}). \quad (1.32)$$

where the function \mathcal{H}^E depends on the specific scheme retained for approximating $(\mathcal{F}^E)_{i,k}^\perp$ (Roe, AUSM+ . . .), and $(U_{i,k}^{L/R})_g$ denote the reconstructed solutions at the Gauss-point g obtained by using the solution polynomial in the cell located on the left/right side of the face $\Gamma_{i,k}$ through which the numerical flux is computed; $\mathbf{n}_{i,k}$ is of course the unit normal vector to this face, pointing outward of cell Ω_i .

For a second-order scheme, solutions $U_{i,k}^{L/R}$ are computed at the single Gauss-point (the face center) with the following linear reconstruction polynomial:

$$(U_{i,k}^{L/R})_g = U_{i/o(i,k)} + (\mathbf{r}_g - \mathbf{r}_{i/o(i,k)}) \cdot \nabla U_{i/o(i,k)} \quad (1.33)$$

where \mathbf{r}_g is the position vector of the Gauss point, $[x_g \ y_g]^T$; $\mathbf{r}_{i/o(i,k)}$ is the position of the left or right cell centroid ($o(i, k)$ denoting the cell that shares the face $\Gamma_{i,k}$ with cell i); $\nabla U_{i/o(i,k)}$ is the gradient computed at the left or right cell centroid. These gradients can be computed using two distinct methods: a least-square approach or Green-Gauss formula. Since the least-square method is retained in this work, we shall focus on its description and we will not provide any details on the Green-Gauss approach (we refer the interested reader to [16] for instance). Let us mention the least-square approximation for the gradient (and, later on, the Hessian) computation is selected because of its greater ease of use.

In order to apply the (linear) least-square method, a reconstruction stencil must be defined firstly. We recall the basic principle of the least-square approximation of the gradient at a given cell-center i is to write Taylor expansions around i on a set of cell centroids in the neighborhood of cell i ; this leads to an over-determined problem and the gradient components estimates are found as the values minimizing the global truncation error of these developments. In order to ensure the non-singularity

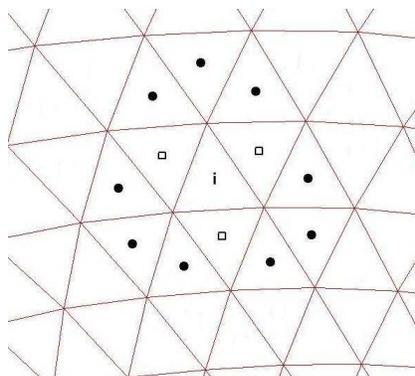


Figure 1.3: **Stencil cells for least-square reconstruction in 2D**

of the linear least-square reconstruction the minimum stencil number in 2D should include 3 centroids around i . A simple way ensuring this condition is to pick into the stencil all face-neighbors of i , that is all cells which share at least one face with cell i (see the cell centroids marked by a square in figure 1.3) for instance). For a cell interior to the grid, the minimum stencil number is guaranteed regardless of the grid cell type: triangle (3 neighbors in the least-square stencil) or quadrilateral (4 neighbors in the least-square stencil). Note however it is not necessarily the case for a boundary cell : for instance, a triangular cell on a boundary could have a single face-neighbor; in such situation, the node-neighbors (cells who share at least one node with the cell in which the gradient is to be computed) will be added into the least-square stencil.

For a given cell i , the system obtained by the linear least-square method takes the form :

$$\begin{bmatrix} \Delta x_1 & \Delta y_1 \\ \vdots & \vdots \\ \Delta x_n & \Delta y_n \end{bmatrix} \begin{pmatrix} \left(\frac{\partial U}{\partial x} \right)_i \\ \left(\frac{\partial U}{\partial y} \right)_i \end{pmatrix} = \begin{pmatrix} \Delta U_1 \\ \vdots \\ \Delta U_n \end{pmatrix}, \quad (1.34)$$

where $\Delta x_j = x_j - x_i$, $\Delta y_j = y_j - y_i$ are the distance differences between the cell i and a cell j in the stencil; $\Delta U_j = U_j - U_i$ denotes the solution difference between a cell j and the cell i ; n is the total number of neighboring cells which belong to the stencil of cell i . It is clear that this system can be solved analytically to yield the following expressions for the first-order estimates of the gradient components at the centroid of cell i :

$$\begin{aligned} \left(\frac{\partial U}{\partial x} \right)_i &= \frac{\sum_{j \in S(i)} (\Delta y_j)^2 \sum_{j \in S(i)} (\Delta U_j \Delta x_j) - \sum_{j \in S(i)} (\Delta x_j \Delta y_j) \sum_{j \in S(i)} (\Delta U_j \Delta y_j)}{\sum_{j \in S(i)} (\Delta x_j)^2 \sum_{j \in S(i)} (\Delta y_j)^2 - \left(\sum_{j \in S(i)} \Delta x_j \Delta y_j \right)^2}, \\ \left(\frac{\partial U}{\partial y} \right)_i &= \frac{- \sum_{j \in S(i)} (\Delta x_j \Delta y_j) \sum_{j \in S(i)} (\Delta U_j \Delta x_j) + \sum_{j \in S(i)} (\Delta x_j)^2 \sum_{j \in S(i)} (\Delta U_j \Delta y_j)}{\sum_{j \in S(i)} (\Delta x_j)^2 \sum_{j \in S(i)} (\Delta y_j)^2 - \left(\sum_{j \in S(i)} \Delta x_j \Delta y_j \right)^2}. \end{aligned} \quad (1.35)$$

where $S(i)$ denotes the set of cells j forming the stencil of the least-square formula applied in cell i . In summary, formula (1.35) yields first-order estimates of the cell-gradient which can be inserted

in the linear solution reconstruction formula (1.33); this formula is then applied in each cell to yield states $(U_{i,k}^{L/R})$ at the single Gauss point corresponding to the face center; these states are finally used in the numerical flux formula (1.32) along with a single-point Gauss quadrature formula (mid-point formula in the present case) to obtain the approximation of the physical (normal) flux along face $\Gamma_{i,j}$ of the control cell Ω_i . This process is very commonly used today for applying second-order schemes on general unstructured grids.

Let us now briefly review the steps that must be taken in order to upgrade this finite-volume strategy to third-order accuracy. First, two Gauss quadrature points are required along each face and the solution states $(U_{i,k}^{L/R})_g$ on the left and right of each of these points are computed using a quadratic polynomial :

$$\begin{aligned} (U_{i,k}^{L/R})_g = U_{i/o(i,k)} &+ (\mathbf{r}_g - \mathbf{r}_{i/o(i,k)}) \cdot \nabla U_{i/o(i,k)} \\ &+ \frac{1}{2} (\mathbf{r}_g - \mathbf{r}_{i/o(i,k)})^T \cdot \mathbf{H}_{i/o(i,k)} \cdot (\mathbf{r}_g - \mathbf{r}_{i/o(i,k)}). \end{aligned} \quad (1.36)$$

where $\mathbf{H}_{i/o(i,k)}$ is the Hessian of the solution at the left or right cell centroid. In a way similar to the second-order scheme, both the solution gradient and its Hessian are computed using a least-square method. This means the Taylor expansions written at cell centers j around the cell-center i are performed up to third-order (the cell-gradient is approximated at second-order while the Hessian is approximated at first-order only); this results again in an over-determined problem, the truncation error of which is computed and minimized with respect to the components of the gradient $(U_x)_i$, $(U_y)_i$ and the components of the symmetric Hessian matrix $(U_{xx})_i$, $(U_{xy})_i$, $(U_{yy})_i$. Naturally, since 5 unknown quantities have to be determined now, an extended stencil is required for these quadratic least-square formula with respect to the one used by the linear least-square formula. For example, this stencil should include a minimum number of 6 centers around i in 2D. Obviously, picking only face-neighbors is not enough to form an acceptable stencil for an interior cell whatever the cell type is; therefore node-neighbors (cells sharing at least one node with the cell i under consideration, marked by circles in figure 1.3) are added into the stencil list. The set of face-neighbors and node-neighbors is called first-level neighbors. In 2D, it is easy to check that there are 8 first-level neighbors for an interior quadrilateral cell and 12 first-level neighbors for an interior equilateral triangle cell, which is in both cases enough for the quadratic least-square reconstruction to be applied. Note again that for a boundary cell the first-level neighbors are often not enough to provide an acceptable stencil so that all face-neighbors of the first-level neighbors of the cell i in which the gradient must be computed at second-order will then be added into the stencil list to ensure the applicability of the quadratic least-square formula.

When applied in a given cell i , the quadratic least-square method yields the following system:

$$\begin{aligned} \begin{bmatrix} w_x^i \Delta x_1 & w_y^i \Delta y_1 & w_{xx}^i (\Delta x_1)^2 & w_{xy}^i \Delta x_1 \Delta y_1 & w_{yy}^i (\Delta y_1)^2 \\ \vdots & \vdots & & & \\ w_x^i \Delta x_n & w_y^i \Delta y_n & w_{xx}^i (\Delta x_n)^2 & w_{xy}^i \Delta x_n \Delta y_n & w_{yy}^i (\Delta y_n)^2 \end{bmatrix} \begin{pmatrix} (U_x)_i \\ (U_y)_i \\ (U_{xx})_i \\ (U_{xy})_i \\ (U_{yy})_i \end{pmatrix} \\ = \begin{pmatrix} \Delta U_1 \\ \vdots \\ \Delta U_n \end{pmatrix}, \end{aligned} \quad (1.37)$$

where, following [16], geometry-based weights have been introduced to rescale the system in order to ensure a proper conditioning for the linear system to solve. These weights are defined as : $w_x^i = 1/h_x^i$, $w_y^i = 1/h_y^i$, $w_{xx}^i = 1/(h_x^i)^2$, $w_{xy}^i = 1/(h_x^i h_y^i)$, $w_{yy}^i = 1/(h_y^i)^2$ where h_x^i and h_y^i are characteristic distances associated with cell i in the x and y directions, computed by $h_x^i = \max(|x_{k+1} - x_k|)$ and $h_y^i = \max(|y_{k+1} - y_k|)$ with indices k and $k + 1$ looping over all nodes of cell i . Such a normalization can of course be also performed for the linear reconstruction - but this latter is less sensitive to the grid scales than the quadratic reconstruction. In viscous computation, highly distorted meshes could be involved and this normalization will become very useful for both reconstruction [4]. Written in matrix form, the above quadratic least-square approach reads :

$$A_i \cdot U_i = B_i, \tag{1.39}$$

where it must be emphasized the matrix A_i is constant for a fixed-grid calculation (which will be systematically considered in the present work) while the right-hand-side B_i depends on the solutions in cell i and its neighbors, which varies during the computation.

After left-multiplication of the system by the transpose matrix of A_i :

$$M_i \cdot U_i = C_i, \tag{1.39}$$

with $M_i = A_i^T \cdot A_i$ a constant square matrix, $C_i = A_i^T \cdot B_i$ a varying solution vector. The solution of the system is obtained as :

$$U_i = M_i^{-1} \cdot C_i. \tag{1.40}$$

where the inverse matrix of M_i is computed once for all of cells at the initialization stage (it depends on purely geometrical data) and stored for later use at each iteration. Note this $5 \times N_i$ matrix (with N_i the number of cells in the stencil of cell i) M_i^{-1} in $2D$ ($9 \times N_i$ in $3D$) has to be stored for each grid cell, which may result in very large memory requirements for a fine $3D$ mesh.

This first chapter has been devoted to a brief review of the key ingredients for designing a residual-based scheme in a finite-difference framework on regular Cartesian grids on one hand and a second and third-order reconstruction-based upwind scheme in a finite-volume framework on unstructured grids on another hand. In the next chapter, we will explain how these ideas can be combined to derive a second and third-order FV-RB scheme on general unstructured grids.

2

Design principles of 2nd and 3rd-order FV-RB schemes on unstructured grids

French La description du schéma RB en maillages structurés dans le chapitre 1 est bref puisque les développements de ce schéma sont détaillés dans une série des papiers publiés précédemment et des thèses de doctorat [30], [31], [12], [11], [14], [21], [38].

Au contraire, le développement de la formulation FV du schéma RB en maillages non-structurés généraux était beaucoup moins avancé avant cette thèse. Le travail initial sur ce sujet était un projet de fin d'étude d'étudiants[42]. Le code développé dans ce projet a été étendu rapidement pour prendre en compte des équations d'état générales et a donné lieu à une série de travaux sur l'analyse d'écoulements de gaz denses[8] avec l'accent sur la physique d'écoulement plutôt que sur le code lui-même. En 2006, un schéma FV-RB d'ordre deux a été présenté à la conférence **ICCFD4**, où l'accent était mis sur une phase implicite économique dite sans matrice pré-conditionnée et couplée avec ce schéma[9]. Même si la version pré-conditionnée de ce schéma RB ne sera pas présentée dans ce mémoire, cette version du schéma FV-RB 2006 est le point de départ des développements du RB en FV présenté dans ce chapitre. Le schéma FV-RB d'ordre deux décrit au-delà est publié dans [10].

Dans ce travail le schéma FV-RB a été développé pour calculer les écoulements stationnaires/instationnaires, en $2D/3D$, du fluide parfait/visqueux. Au lieu de donner la formulation complète du schéma FV-RB pour les écoulements instationnaires tridimensionnels d'un fluide visqueux, on préfère présenter dans ce chapitre la conception du schéma, étape par étape, des problèmes simples aux problèmes complexes. De cette façon, on espère que les spécificités du schéma apparaissent clairement. Le schéma FV-RB d'ordre deux et trois pour les équations d'Euler $2D$ stationnaire est donc présenté d'abord. On donne ensuite des détails sur les conditions aux limites utilisées dans les applications. L'extension de ce schéma au cas visqueux est également détaillé et un élément important du schéma RB, la dissipation numérique basé sur le résidu, est modifié par les termes diffusifs introduits dans le système des lois de conservation. La description du solveur pour les problèmes visqueux stationnaires $2D$ est complétée avec quelques détails sur les conditions aux limites. La présentation du solveur stationnaire en $2D$ se termine par la description de la discrétisation temporelle et de la technique de limitation, qui est nécessaire pour calculer des écoulements avec des discontinuités. La formulation $3D$ du schéma FV-RB pour des problèmes stationnaires est donnée afin de compléter nos descriptions. Finalement, on décrit l'extension du schéma FV-RB aux problèmes instationnaires.

Il est important de souligner que notre objectif de développement du schéma RB en maillages non-structurés n'était pas limité à une précision d'ordre trois. Toutefois la méthode volumes finis montre

un besoin énorme en mémoire de stockage quand un schéma d'ordre élevé est utilisé, particulièrement pour les problèmes $3D$. C'est pour cette raison qu'on décide de se limiter à l'ordre trois pour le schéma basé sur la FV. On envisage plutôt une autre voie de recherche qui couple le flux numérique du schéma RB avec une nouvelle méthode de discrétisation en espace, dite la méthode Volumes Spectraux. Cette partie du travail sera présentée dans les chapitres 4 et 5, une fois que les schémas FV-RB d'ordre deux et trois aurait été évalués dans le chapitre 3 pour une grande gamme des problèmes.

English The description of the RB scheme for structured grid computations provided in chapter 1 has been brief on purpose, since these developments have been detailed in a series of previously published papers and PhD thesis [30], [31], [12], [11], [14], [21], [38].

On the contrary, the development of a FV formulation of the RB scheme on general unstructured grids has been much less documented before this thesis. The initial work on this topic was performed in 2004 in the framework of a students' project[42]; the solver developed in this work was soon extended to deal with general equations of state and gave rise to a line of work devoted to the analysis of dense gas flows [8] with a focus on the flow physics rather than on the numerical solver. In 2006, a second-order FV-RB scheme was presented at the **ICCFD4** Conference with an emphasis on the low-cost preconditioned matrix-free implicit stage with which it was coupled [9]. Even though the preconditioned version of the RB scheme will not be presented in this memoir, this 2006 FV-RB solver was the starting point of the FV developments performed during this thesis and presented in this chapter. The second-order version of the FV-RB scheme described hereafter was published in [10].

The FV-RB scheme has been developed in this work to deal with steady / unsteady, inviscid / viscous flows for $2D$ and $3D$ configurations. Rather than directly providing the full formulation of the scheme for $3D$ unsteady viscous flows, we prefer to present the RB scheme design step by step from simple problems to more complex problems in this chapter. In this way, hopefully the specificities of the RB formulation could be clarified. Thus the FV-RB scheme for the $2D$ steady Euler equations will be first presented in its second-order and third-order formulation. And some details will be provided on the boundary conditions used in the application cases presented in the next chapter. Then the extension of the scheme to the viscous case will be then detailed, where one important element of the RB scheme, the residual-based numerical dissipation is modified by the diffusive terms introduced in the system of conservation laws to be solved. The description of the $2D$ steady viscous solver will be completed with some details on the boundary treatment. The presentation of the $2D$ steady FV-RB solvers will be closed with the description of the time discretization strategy, and the description of some limiting techniques, which are necessary to perform flow computations involving discontinuities. The $3D$ formulation of the FV-RB scheme for steady flows will be provided for the sake of completeness. Finally, the extension of the FV RB strategy to unsteady flows will be described.

It is important to point out that our original objective of the RB scheme development on unstructured grids was not limited to third-order accuracy. However the FV strategy displays memory requirements that tend to become exceedingly huge when high order scheme is used, in particular for $3D$ computations. This is the reason why it was decided to push no further than third-order accuracy along the FV path. Instead, another line of research was explored, consisting of coupling the RB numerical flux with a new spatial discretization method, called Spectral Volume method. This part of our work will be displayed in chapter 4 and 5, after the second and third-order FV-RB schemes are assessed in chapter 3 for a wide range of problems.

2.1 Formulation for 2D steady flows

2.1.1 Inviscid case

2.1.1.1 Design of the baseline FV-RB numerical flux

As previously explained in chapter 1, the FV-discretization of the conservation law (1.30), with the physical flux integral approximated by the Gauss quadrature rule (1.31), takes the form :

$$\frac{\partial U_i}{\partial t} + \frac{1}{|\Omega_i|} \sum_{k \in \mathcal{I}(\Omega_i)} \sum_{g=1}^{N_g} \omega_g (\mathcal{H}_{i,k})_g |\Gamma_{i,k}| = 0, \quad (2.1)$$

where, in the inviscid case, the numerical flux $(\mathcal{H}_{i,k})_g = (\mathcal{H}_{i,k}^E)_g$ approximates the inviscid normal flux $\mathcal{F}^E(U) \cdot \mathbf{n} = (\mathcal{F}^E)^\perp$. When this inviscid numerical flux is computed by the residual-based (RB) scheme, it takes the following form:

$$(\mathcal{H}_{i,k}^E)_g = (\mathcal{H}_{i,k}^{RB})_g = (\mathcal{H}_{i,k}^c)_g - (d_{i,k})_g, \quad (2.2)$$

where

- $(\mathcal{H}_{i,k}^c)_g$ is a purely centered (non-dissipative) approximation of the physical normal flux vector $(\mathcal{F}^E)^\perp$ computed at the Gauss-point $(\cdot)_g$ of face $\Gamma_{i,k}$. This non-dissipative flux is computed from some evaluation of the physical fluxes f, g at the control cell face.
- $(d_{i,k})_g$ is a dissipation flux, which is based on the residual and thus depends on the gradients of the physical fluxes f, g .

A first strategy considered in the course of this work was to rely on a least-square estimate of the physical fluxes f, g , and their first- and second-derivatives on the face to build the RB numerical flux. The interest of this strategy lies in the fact that it allows to retain the compactness of the original numerical flux. Unfortunately, this strategy did not provide a stable scheme, for reasons that remain unclear.

The alternative method that was then followed is now described. First, the non-dissipative part of the numerical flux is computed using reconstructed values at the face :

$$\mathcal{H}^c((U_{i,k}^L)_g, (U_{i,k}^R)_g) = \frac{1}{2}((\mathcal{F}_{i,k,g}^E)_L^\perp + (\mathcal{F}_{i,k,g}^E)_R^\perp), \quad (2.3)$$

If the general upwind numerical flux (1.32) can be also decomposed as the sum of a non-dissipative and a dissipative flux (this is easily done for the Roe numerical flux for instance), it is clear the sole difference between the RB scheme and this classical upwind solver lies on the evaluation of the dissipation term $(d_{i,k})_g$. For the RB scheme that we have seen in chapter 1, this dissipation flux vector is based on the approximation of the residual associated with system (1.29) computed on the interface.

The proper computation of the residual-based dissipation $(d_{i,k})_g$ is crucial for ensuring the accuracy and robustness of the RB scheme. It is important at this stage to point out the difference between the RB scheme and a conventional upwind scheme on this term :

- the built-in high-order dissipation of a conventional upwind scheme is obtained from the use of a high-order reconstruction of the flow variables (going for instance from linear to quadratic least square formula as described in the previous chapter) which are inserted in the upwind numerical flux formula (be it a Flux Vector Splitting such as Van Leer, Flux Difference Splitting such as Roe or Hybrid scheme such as AUSM+).
- the high-order dissipation of the RB scheme is explicitly added to a purely centered high-order formula. Depending on the way to compute the residual on the considered face (at first or second order as will be explained below), the dissipative flux will be $\mathcal{O}(h^2)$ or $\mathcal{O}(h^3)$. In order to reduce the cost of the RB scheme, this second or third-order dissipation will be uniquely computed once on a face and shared by all of RB fluxes at the Gauss-points located on the same $\Gamma_{i,k}$ face :

$$(d_{i,k})_g = d_{i,k} = \frac{1}{2} L^\perp \Phi_{i,k} \mathcal{R}_{i,k}, \quad (2.4)$$

where L^\perp is the projection onto the face normal direction of the distance between two cell centroids (C_i and C_o in Figure (2.1)) on the left and right side of this face :

$$L^\perp = \Delta \mathbf{r}_{C_i, C_o} \cdot \mathbf{n}_{i,k}, \quad (2.5)$$

Obviously this distance has the order of the mesh size $\mathcal{O}(h)$; besides, $\Phi_{i,k}$ is a matrix coefficient of order $\mathcal{O}(1)$ designed so as to ensure the dissipation of the scheme; basically, with respect to the description provided for a Cartesian grid in the previous chapter, the dissipation coefficient on a face will be built from the wavespeeds or eigenvalues associated with the "normal" and "tangential" Jacobian, that is the Jacobian of the flux $\mathcal{F}^E \cdot \mathbf{n}$, $\mathcal{F}^E \cdot \mathbf{t}$ where \mathbf{n} (resp. \mathbf{t}) denotes the unit normal (resp. tangent) vector to the face through which the numerical flux is computed. The quantity $\mathcal{R}_{i,k}$ is an approximation of the system residual $R_{i,k}$ defined as :

$$R_{i,k} = \frac{1}{|\Omega_{i,k}|} \int_{\Omega_{i,k}} r \, d\Omega. \quad (2.6)$$

The integral form $R_{i,k}$ of the residual r associated with system (1.28) is computed in a shift cell $\Omega_{i,k}$, enclosing the face $\Gamma_{i,k}$ and formed by the nodes of face $\Gamma_{i,k}$ and the two cell centroids C_i , C_o on each side of this face (see figure 2.1). Note that, in this way, the quantity $\mathcal{R}_{i,k}$ in the dissipation flux (2.4) adapts itself to the problem under study through the definition of r . For the Euler case considered in this first part of the chapter, the residual r is given by $r = \nabla \cdot \mathcal{F}^E$. Obviously, a second-order dissipation is obtained with a simple first-order estimate of $R_{i,k}$ - since this estimate is then multiplied by the first-order term L^\perp to yield the dissipative flux. In what follows, we will rather build a third-order dissipation which will be systematically added to the non-dissipative contribution to the flux, yielding a second or third-order RB scheme, depending on the linear or quadratic reconstruction used for the flow variables in the centered formula (2.3).

In the following let us detail the computation of the dissipative flux. For inviscid flows, the flux integral appearing in (2.6) can be expressed as :

$$R_{i,k} = \frac{1}{|\Omega_{i,k}|} \sum_{l \in \mathcal{I}(\Omega_{i,k})} \int_{\Gamma_l} \mathcal{F}^E \cdot \mathbf{n} \, d\Gamma, \quad (2.7)$$

Using a single quadrature point on each face (face center) of the shift cell, a second-order approximation of this integral reads :

$$\mathcal{R}_{i,k} = \frac{1}{|\Omega_{i,k}|} \sum_{l \in \mathcal{I}(\Omega_{i,k})} (\mathcal{H}_c^E)_l |\Gamma_l|, \quad (2.8)$$

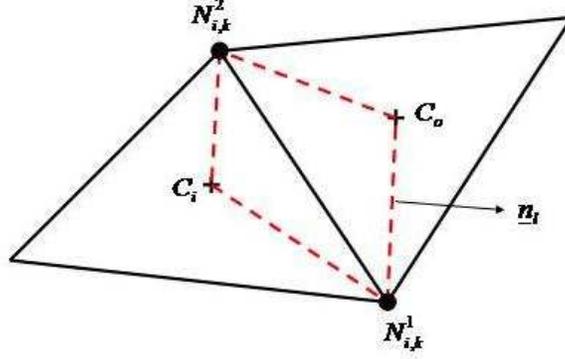


Figure 2.1: Shift cell used for the dissipation calculation in 2D

where the numerical flux $(\mathcal{H}_c^E)_l$ is an approximation of the normal physical flux $(\mathcal{F}^E)_l^\perp$ at the center of face Γ_l of the shifted cell $\Omega_{i,k}$. Since a second-order approximation of the face integral is sought for, this numerical flux can be computed by the trapezoidal rule. In the 2D case, on a face Γ_l with vertices $(N_{i,k}^1, C_o)$ for example (see Figure 2.1), the flux $(\mathcal{H}_c^E)_l$ is computed as :

$$(\mathcal{H}_c^E)_l = \frac{1}{2} \left(\mathcal{F}^E(U_{N_{i,k}^1}) + \mathcal{F}^E(U_{C_o}) \right) \cdot \mathbf{n}_l. \quad (2.9)$$

where \mathbf{n}_l denotes the unit normal vector to the face Γ_l pointing outwards the shift cell $\Omega_{i,k}$.

In order to provide this second-order accurate estimation of the flux integral along a face of the shift cell, the order of precision of the node value is of course a key element. A simple choice for computing the node value U_N at a node N of the computational grid is an inverse-distance weighted average of the cell-centered values in the cells sharing the node; this strategy was used by Frink [19] in the context of viscous flux evaluation, where node values may also be needed, depending on the quadrature formula used for the viscous flux estimate (this point will be detailed later on, when dealing with the viscous extension of the RB scheme). However, this method of evaluation for the grid node values turns out to be less than second order accurate, which is not sufficient to derive a second-order RB scheme. Therefore, another averaging method proposed by Holmes and Connell [23] has been used : it is a bit more complex hence expensive but yields second-order accurate estimates for the node values.

The general form of a node estimate from neighboring cell-center values can be formulated as :

$$U_N = \frac{\sum_{j \in \mathcal{I}(N)} \omega_j U_j}{\sum_{j \in \mathcal{I}(N)} \omega_j}, \quad (2.10)$$

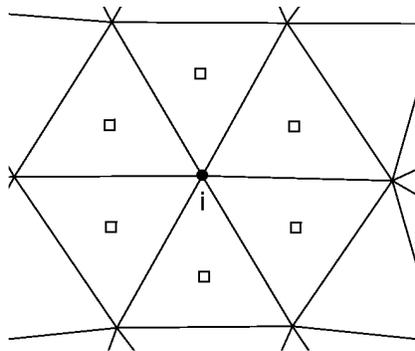


Figure 2.2: **Cells sharing a node** stencil used for computing the solution and solution gradient on node i .

where $\mathcal{I}(N)$ denotes the set of cells Ω_j sharing the node N , which is the stencil associated with the node estimate formula (see Figure 2.2); and the expression of the weight factor ω_j depends on the specific method retained for node value evaluation. For the inverse-distance weighted average, $\omega_j = 1/L_j$, where L_j is the distance between the node N and the centroid of the j -th stencil cell Ω_j .

In the method proposed by Holmes and Connell, for 2D case, the weight factor ω_j is given by:

$$\omega_j = 1 + \lambda_x (x_j - x_N) + \lambda_y (y_j - y_N), \quad (2.11)$$

where λ_x and λ_y are Lagrange multipliers constructed from purely geometrical data:

$$\begin{aligned} \lambda_x &= \frac{I_{xy} \cdot (1, \Delta y) - I_{yy} \cdot (1, \Delta x)}{D}, \\ \lambda_y &= \frac{I_{xy} \cdot (1, \Delta x) - I_{xx} \cdot (1, \Delta y)}{D}, \end{aligned} \quad (2.12)$$

with the quantity $(\Delta\phi, \Delta\psi)$ defined by :

$$(\Delta\phi, \Delta\psi) = \sum_{j \in \mathcal{I}(N)} (\phi_j - \phi_N) (\psi_j - \psi_N). \quad (2.13)$$

Quantities I_{xx} , I_{yy} , I_{xy} and D are defined by : $I_{xx} = (\Delta x, \Delta x)$, $I_{yy} = (\Delta y, \Delta y)$ and $I_{xy} = (\Delta x, \Delta y)$; the denominator D is given by $D = I_{xx} I_{yy} - I_{xy}^2$. On some severely distorted grids, the weight ω_j can differ significantly from unity; it is therefore bounded in the range $[0, 2]$.

Yet another strategy for node value computation has been considered. Since the solution gradient and Hessian are available in each cell, they can be used as following to estimate the node value :

$$U_N = \frac{\sum_{j \in \mathcal{I}(N)} (U_j + \Delta \mathbf{r}_{j,N} \cdot \nabla U_j)}{N_s}, \quad (2.14)$$

where $\mathcal{I}(N)$ still represents the set of cells sharing the node N , N_s is the number of cells in this stencil, *i.e.* the cardinal of $\mathcal{I}(N)$; $\Delta \mathbf{r}_{j,N} = \mathbf{r}_N - \mathbf{r}_j$ is the position difference vector. Note the node

estimate (2.14) makes use of (1.33) assuming a linear reconstruction is used in the non-dissipative flux evaluation; if a quadratic reconstruction is used in this same non-dissipative flux, the node estimate would rely on the average of values reconstructed in each cell from formula (1.36) :

$$U_N = \frac{\sum_{j \in \mathcal{I}(N)} \left(U_j + \Delta \mathbf{r}_{j,N} \cdot \nabla U_j + \frac{1}{2} (\Delta \mathbf{r}_{j,N})^T \cdot \mathbf{H}_j \cdot \Delta \mathbf{r}_{j,N} \right)}{N_s}. \quad (2.15)$$

This method for computing the node values will be referred to as gradient-based extrapolation method. In our computations (see next chapter) it has been found that the RB scheme with this gradient-based extrapolation method for node value computation yields more accurate results with respect to the inverse-distance or Holmes and Connell averaging. It should be noted however that this extrapolation method is likely to produce a more oscillatory solution for cases with shocks and may also cause some positivity problems in the viscous case. In these situations, the Holmes and Connell method will be preferred.

The remaining key ingredient in the dissipation term (2.4) of the RB numerical flux is the dissipation matrix coefficient $\Phi_{i,k}$. If A^E , B^E denote respectively the Jacobian matrix of the inviscid fluxes f^E and g^E for a 2D problem, let us define as following the *normal* and *tangential* Jacobian matrix with respect to the face $\Gamma_{i,k}$:

$$\begin{aligned} (J_{\perp})_{i,k} &= A_{i,k}^E(n_x)_{i,k} + B_{i,k}^E(n_y)_{i,k}, \\ (J_{\parallel})_{i,k} &= A_{i,k}^E(t_x)_{i,k} + B_{i,k}^E(t_y)_{i,k}, \end{aligned} \quad (2.16)$$

where $\mathbf{n}_{i,k}(n_x, n_y)$ (resp. $\mathbf{t}_{i,k}(t_x, t_y)$) is the unit vector normal (resp. tangent) to the face $\Gamma_{i,k}$. Since the Euler equations form a hyperbolic system of conservation laws, these normal and tangential Jacobian matrices can also be expressed as :

$$\begin{aligned} (J_{\perp})_{i,k} &= (T_{\perp})_{i,k} \cdot \text{Diag}((\lambda_{\perp}^{(l)})_{i,k}) \cdot (T_{\perp}^{-1})_{i,k}, \\ (J_{\parallel})_{i,k} &= (T_{\parallel})_{i,k} \cdot \text{Diag}((\lambda_{\parallel}^{(l)})_{i,k}) \cdot (T_{\parallel}^{-1})_{i,k}, \end{aligned} \quad (2.17)$$

where $\text{Diag}(\cdot)$ represents a diagonal matrix; the quantity $\lambda_{\perp}^{(l)}$ (resp. $\lambda_{\parallel}^{(l)}$) denotes the l^{th} eigenvalue of the matrix J_{\perp} (resp. J_{\parallel}), and $(T_{\perp})_{i,k}$ (resp. $(T_{\parallel})_{i,k}$) is the matrix the columns of which are the eigenvectors associated with the eigenvalues $\lambda_{\perp}^{(l)}$ (resp. $\lambda_{\parallel}^{(l)}$). The dissipation matrix $\Phi_{i,k}$ is built by assuming its eigenvectors to be those of $(J_{\perp})_{i,k}$, which yields :

$$\Phi_{i,k} = (T_{\perp})_{i,k} \cdot \text{Diag}(\phi_{i,k}^{(l)}) \cdot (T_{\perp}^{-1})_{i,k}, \quad (2.18)$$

with the eigenvalues $\phi_{i,k}^{(l)}$ computed from the eigenvalues of J_{\perp} and J_{\parallel} :

$$\phi_{i,k}^{(l)} = \text{sign}((\lambda_{\perp}^{(l)})_{i,k}) \min \left(1, \frac{|\Gamma_{i,k}| |(\lambda_{\perp}^{(l)})_{i,k}|}{|L^{\perp}| m(J_{\parallel})_{i,k}} \right). \quad (2.19)$$

where $m(J_{\parallel}) = \min_l(|\lambda_{\parallel}^{(l)}|)$ is the minimum eigenvalue related to the face tangent direction. The eigenvalues and eigenvectors on the face $\Gamma_{i,k}$ are computed using the Roe-average of the solutions at the centroids of the cells sharing this face. As will be seen in the following paragraphs, the definition of

the dissipation matrix by (2.18)- (2.19) will be left unchanged when computing viscous or unsteady problems. Only the residual r , hence $R_{i,k}$ in the expression of the dissipation flux (2.4), will be adapted to the system of conservation laws to be solved; leaving the dissipation matrix unchanged has of course no impact on the scheme's accuracy, since the dissipation matrix remains $\mathcal{O}(1)$, but it could affect nonetheless the stability or efficiency properties of the method : since no such flaw has been previously detected in the context of structured grid calculations, the same simple choice was adopted for the present unstructured grid computations.

2.1.1.2 Boundary conditions

We briefly review the specificities of boundary conditions when the RB scheme is used for Euler equations.

- Far-field boundary condition** Note first that when computing external flows around a body, the far-field boundary of the flow domain will be systematically set far enough from the object in order to reduce the effect of the far-field boundary condition on the flow field near the body. The physical state on the far-field boundary face is computed with a characteristic-based non-reflecting boundary condition, which makes use of the interior solution and the prescribed far-field state. The flux on the boundary face is then computed by using the physical flux formula applied with the computed physical state. Clearly, this boundary treatment is not dependent on the numerical flux used for the interior cell faces. Note however that the RB numerical flux on faces having one node located on the far-field boundary requires the estimation of this boundary node value : this estimation is actually provided using the averaging in use for all the interior cells with the sole difference that the stencil $\mathcal{I}(N)$ on which the averaging is applied displays some directionality (see Figure 2.3).

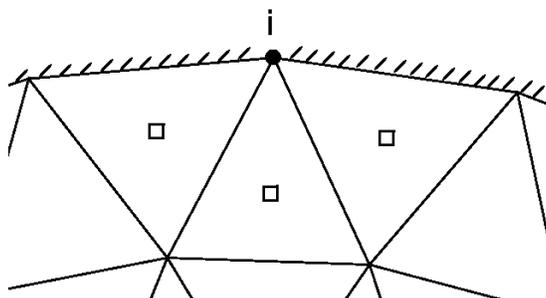


Figure 2.3: **Cells sharing a boundary node** stencil used for computation of the solution and solution gradient on the boundary node i .

- Slip wall condition** For inviscid flows, the slip wall condition implies that the velocity in the direction normal to the wall is zero on the wall face. This condition can be introduced into the physical flux in the wall normal direction and it therefore reduces to :

$$\mathcal{F}_w = [0 \quad p_w n_x \quad p_w n_y \quad 0]^T. \quad (2.20)$$

where n_x and n_y are the components of the unit normal vector to the wall face, p_w is the pressure on the wall face. This pressure is estimated using the solution polynomial in the

interior cell adjacent to the wall face under consideration. Here again, the boundary treatment does not depend on the numerical flux. However, an estimation of the solution at the wall node is necessary for computing the RB numerical flux through an interior face with a node on the wall. Therefore, the slip boundary condition has to be imposed at the node; this condition is enforced as follows :

- a first estimate \mathbf{V}_N^* of the velocity vector at a wall node (with components u_N^* , v_N^*) is computed by equation (2.10) or (2.14) applied to the velocity vector components, u and v .
- next, the normal velocity at the wall node is computed from $V_N^\perp = u_N^*(n_x)_N + v_N^*(n_y)_N$ where $(n_x)_N$, $(n_y)_N$ are the components of the unit wall normal vector at node N . This vector is computed as the average of the unit normal vectors associated with the wall faces sharing node N . Since the wall faces in our computations vary smoothly, such an approximation is sufficient to provide accurate results.
- finally, a wall velocity with zero-component in the normal direction is computed as $\mathbf{V}_N = \mathbf{V}_N^* - V_N^\perp$.

2.1.2 Viscous case

2.1.2.1 Extension of the RB numerical flux

The baseline RB numerical flux derived in the inviscid case is made of two contributions : a purely centered flux and a residual-based dissipation term. In the viscous case (typically when solving the Navier-Stokes equations or, in a simplified version, advection-diffusion problems), the viscous contribution to the physical flux must be included both into the purely centered flux and in the residual on which the dissipation flux is based.

Purely centered flux Firstly, the centered numerical flux becomes :

$$\mathcal{H}^c = \mathcal{H}^c((U_{i,k})_g, (\nabla U_{i,k})_g), \quad (2.21)$$

which means that not only the solution but also the solution gradient needs to be computed at each Gauss quadrature point on each interface. The solution computation at each quadrature point has been previously described and remains unchanged; thus, the gradient computation is now the main focus.

For the second-order RB scheme, a first-order accurate solution gradient is computed at the only quadrature point on the face, *i.e.* the face center M (Figure (2.4)). There are several ways to compute this gradient [23][16]. In our computation, it is computed by a simple average of gradients on all of nodes belonging to this face :

$$(\nabla U_{i,k})_g = \nabla U_M = \frac{1}{2}(\nabla U_{N_{i,k}^1} + \nabla U_{N_{i,k}^2}). \quad (2.22)$$

This approximation gives only first-order precision because the gradient at each mesh node is computed to first-order accuracy. As the same node gradient is also necessary in the RB numerical dissipation computation, its evaluation will be explained later on, when describing the estimate of

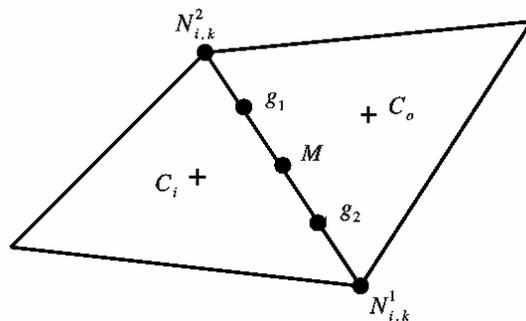


Figure 2.4: **Gauss quadrature points for RB numerical flux computation** face center M is the only quadrature point used for second-order RB scheme; two quadrature points g_1 and g_2 are used for third-order RB scheme.

this RB dissipation.

For the third-order RB scheme, the computation of the solution gradient with second-order accuracy at each quadrature point on the face is required : a reconstruction method using the cell gradient and cell Hessian is adopted - for an interior face, the solution gradient and Hessian are available in both neighbor cells. The gradient estimate at the quadrature point g is computed as the arithmetic average of the reconstructed gradient in each neighboring (left/right) cell :

$$(\check{\nabla}U_{i,k})_g = \frac{1}{2}((\check{\nabla}U_g)^L + (\check{\nabla}U_g)^R), \quad (2.23)$$

where the reconstructed left (resp. right) gradient is computed from the gradient and Hessian estimated at centroid of cell i (resp. o) :

$$\begin{aligned} (\check{\nabla}U_g)^L &= \check{\nabla}U_i + \mathbf{H}_i \cdot \Delta\mathbf{r}_{i,g}, \\ (\check{\nabla}U_g)^R &= \check{\nabla}U_o + \mathbf{H}_o \cdot \Delta\mathbf{r}_{o,g}. \end{aligned} \quad (2.24)$$

where the position difference vectors are defined as $\Delta\mathbf{r}_{i,g} = \mathbf{r}_g - \mathbf{r}_{C_i}$, $\Delta\mathbf{r}_{o,g} = \mathbf{r}_g - \mathbf{r}_{C_o}$.

Dissipation flux According to equation (2.6) and extending formula (2.7) to the viscous case in a straightforward manner, the approximation of the residual associated with the viscous system becomes :

$$\mathcal{R}_{i,k} = \frac{1}{|\Omega_{i,k}|} \sum_{l \in \mathcal{I}(\Omega_{i,k})} (\mathcal{H}_c^E - \mathcal{H}_c^V)_l |\Gamma_l|, \quad (2.25)$$

where the way to compute the numerical viscous flux \mathcal{H}_c^V at each face center of the shift cell needs to be specified. Using the trapezoidal rule along the face l of the shift cell (see (2.1), the viscous flux is calculated by :

$$(\mathcal{H}_c^V)_l = \frac{1}{2} \left(F^V(U_{N_{i,k}^1}, \nabla U_{N_{i,k}^1}) + F^V(U_{C_o}, \nabla U_{C_o}) \right) \cdot \underline{n}_l. \quad (2.26)$$

Hence, both gradients at cell centroids and cell nodes are needed. Since the gradient at cell centroids is already available - as well as the solution at cell nodes -, the extra-computation introduced by the

viscous formulation is the gradient on mesh nodes.

It is seen that the actual order of scheme degraded by one order because of the involvement of the physical viscous flux, i.e. the solution gradients, in the pure centered flux. This is also the case for the RB dissipation flux.

For the second-order RB scheme, the node gradient is computed to first-order by a linear least square reconstruction applied around the node; the stencil used for this least-square estimate includes all of cells sharing the node (also referred to as the first-level neighbors). For a mesh with pure triangle cells, the first-level neighbors of a node can ensure the good least-square reconstruction. But these neighbors could be not enough when quadrilateral cells are involved in the mesh, for example, node i has only 2 first-level neighbors (see Figure 2.5), while 3 is the minimum stencil number to ensure a successful least-square reconstruction. In this case, face neighbors of first-level neighbors are added into the stencil list.

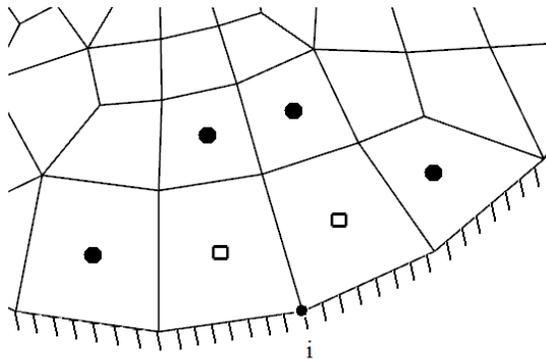


Figure 2.5: **Special stencil for the solution gradient computation with linear least-square reconstruction on the boundary node in a mesh with quadrilateral cells.** cells marked with square are the first-level neighbors, marked with dot are face neighbors of the first-level neighbors.

For the third-order RB scheme, a second-order accurate dissipation flux is enough because the whole scheme is degraded to second-order by the pure centered flux. The same first-order residual approximation used by the second-order RB scheme is still adopted. The node gradient is therefore still computed by the linear least-square reconstruction.

2.1.2.2 Boundary conditions

A brief description of the boundary conditions used by the RB scheme for computing viscous flows is presented here.

- **Far-field boundary condition** The far-field boundary treatment described in the inviscid case is also applied in the viscous case. The sole difference lies in the fact that the inviscid flux obtained with the solution computed on the boundary cell by using the theory of characteristics must be completed with the estimate of the viscous physical flux. For the second-order scheme, equation (2.22) is used for the first-order gradient computation as on internal faces. Solution and its gradient are therefore necessary on the boundary node. Node solution computation is

the same as the inviscid case. Node gradient is computed to first order as stated in the RB numerical dissipation flux computation. For the third-order scheme, equation (2.23) and (2.24) used for the second-order gradient computation are not successfully used on the boundary faces, the reason is possibly that there are no cells either on the left or on the right side of the boundary faces, which makes this kind of upwind gradient approximation unstable. A linear weighted approximation using node gradients is used instead specially for the gradient approximation on boundary faces:

$$(\nabla w_{i,k})_g = (1 - \alpha_{N_{i,k}^1, g}) \nabla w_{N_{i,k}^1} + \alpha_{N_{i,k}^1, g} \nabla w_{N_{i,k}^2}. \quad (2.27)$$

with the weighting coefficient $\alpha_{N_{i,k}^1, g} = \frac{|\Delta \mathbf{r}_{N_{i,k}^1, g}|}{|\Delta \mathbf{r}_{N_{i,k}^1, N_{i,k}^2}|}$. It is easy to see that this equation will become the equation (2.22) when the second-order RB scheme is used, where the Gauss quadrature point is the face center. Here the node gradient is always first-order accurate as mentioned in the RB numerical dissipation computation.

- **No-Slip wall condition** The no-slip boundary condition on a wall face is introduced in the physical flux estimate through this face (the velocities are set to zero in the physical flux formula on the solid wall). Like on the far-field boundary, equation (2.27) is used in the face gradient approximation for both the second and third-order RB scheme. In a way similar to the inviscid case, the no-slip boundary condition is also imposed in the wall-boundary node solution computation.

2.1.3 Time integration

We briefly recall the well-known limitations of a simple Euler-explicit time-integration, regardless of the numerical flux under consideration, and proceed to describe a simple matrix-free implicit stage which has been used throughout this work to speed up the convergence of the computations to a steady-state (including the convergence to a pseudo-steady state in the case of dual-time computations applied to unsteady flows).

2.1.3.1 Explicit strategy

Let us consider the 2D Navier-Stokes equations (1.8) :

$$\frac{\partial U}{\partial t} + \frac{\partial (f^E(U) - f^V(U, \nabla U))}{\partial x} + \frac{\partial (g^E(U) - g^V(U, \nabla U))}{\partial y} = 0,$$

and introduce the viscous Jacobian matrices : $A_0^V = \frac{\partial f^V}{\partial U}$, $A_1^V = \frac{\partial f^V}{\partial U_x}$, $A_2^V = \frac{\partial f^V}{\partial U_y}$ and similarly

$B_0^V = \frac{\partial g^V}{\partial U}$, $B_1^V = \frac{\partial g^V}{\partial U_x}$, $B_2^V = \frac{\partial g^V}{\partial U_y}$. Let us recall the Jacobian matrices of the inviscid fluxes

are $A^E = \frac{df^E}{dU}$, $B^E = \frac{dg^E}{dU}$. The Navier-Stokes equations can also be expressed in the following quasi-linear form :

$$U_t + A^E U_x + B^E U_y = A_0^V U_x + A_1^V U_{xx} + A_2^V U_{xy} + B_0^V U_y + B_1^V U_{xy} + B_2^V U_{yy}. \quad (2.28)$$

The simplest time-integration strategy associated with the space-discretization described up to now would be to use an Euler-explicit scheme, which takes the general form :

$$\frac{U_i^{n+1} - U_i^n}{\Delta t} + \mathcal{E}(U^n) = 0, \quad (2.29)$$

where U^n denotes the solution at time $n\Delta t$ and the explicit stage \mathcal{E} depends only on discrete values of the solution vector at time-level n . When the Navier-Stokes equations in their conservative or quasi-linear formulation are solved by the finite volume method on a general unstructured grid, the explicit stage takes the following form :

$$\mathcal{E}(U^n) = \frac{1}{|\Omega_i|} \sum_{k \in \mathcal{I}(\Omega_i)} \sum_{g=1}^{N_g} \omega_g (\mathcal{H}_{i,k}^n)_g |\Gamma_{i,k}|. \quad (2.30)$$

where the numerical flux \mathcal{H}^n is computed from known solutions at time-level n . The time-step Δt appearing in the explicit scheme (2.29) must be chosen so as to satisfy some numerical stability requirements. If (2.28) reduces to the 1D Euler equation ($B^E = A_0^V = A_1^V = A_2^V = B_0^V = B_1^V = B_2^V = 0$) a well-known solution for choosing Δt is the so-called *CFL*-condition :

$$\Delta t_i = CFL \frac{\Delta x}{\rho(A^E)_i} \text{ with } CFL \leq 1, \quad (2.31)$$

where $\rho(A^E)$ is the spectral radius of the Jacobian matrix A^E (that is the maximum absolute values of the eigenvalues of A^E). For the 1D Euler equations, this spectral radius computed in the cell i is given by $\rho(A^E)_i = |u_i| + a_i$ where a_i denotes the speed of sound computed in the cell i .

In the case of the 2D Euler equations solved on a general unstructured grid, this criterion of choice for Δt is extended as follows :

$$\Delta t_i = CFL \frac{h_i}{\rho_i^C} \text{ with } CFL \leq 1, \quad (2.32)$$

where h_i is a characteristic length of the cell i and ρ_i^C is the characteristic convective wave speed in the cell. In order to ensure the stability of the explicit time-integration, the quantity h_i is defined so as to be minimal while the quantity ρ_i^C is defined so as to be maximal. More precisely, in our computations, the characteristic length is computed by taking the smallest distance between the cell centroid and the center of the faces belonging to this cell. For a triangle for example (see Figure 2.6), one will have : $h_i = \min(|\overrightarrow{C_i C_k}|), k = 1, 3$. As for the characteristic wave speed, it is computed as : $\rho_i^C = |V_i| + a_i$.

In the viscous case, a time-step Δt_i^V related to the diffusive effects has also to be taken into account, along with the convective characteristic time-step given by $\Delta t_i^C = \frac{h_i}{\rho_i^C}$. A straightforward extension of a 1D stability analysis for advection-diffusion problems leads to the final choice of time-step :

$$\Delta t_i = CFL \min(\Delta t_i^C, \Delta t_i^V), \quad (2.33)$$

where the *CFL* number is taken smaller than unity to ensure the stability of the Euler-explicit time-integration and :

$$\Delta t_i^V = \frac{h_i^2}{2\rho_i^V}, \quad (2.34)$$

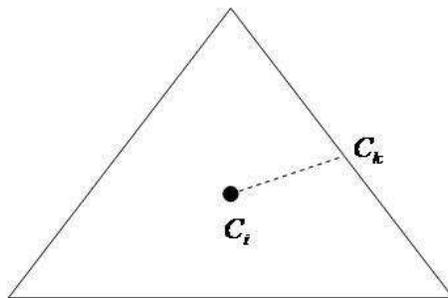


Figure 2.6: **Characteristic size of a 2D triangle cell**

with ρ_i^V the maximum eigenvalue of the viscous Jacobian matrices. This viscous spectral radius is precisely given by :

$$\rho_i^V = \frac{1}{\rho_i} \cdot \max\left(\frac{4}{3}\mu_i, \kappa_i\right). \quad (2.35)$$

with μ_i the fluid viscosity computed in cell i and κ_i the fluid thermal conductivity.

2.1.3.2 Implicit strategy

A way to get rid of the previous restrictive time-step limitations is to make use of an Euler-implicit time-integration, that is to discretize the system of conservation laws under consideration with the following scheme :

$$\frac{U_i^{n+1} - U_i^n}{\Delta t_i} + \mathcal{E}(U^{n+1}) = 0, \quad (2.36)$$

In the linear case, it can be easily proved such fully implicit strategy yields unconditional stability : this means the time-step Δt_i can be chosen as large as possible; moreover the convergence speed to steady-state is increased when the time-step is increased. In practice, for non-linear problems, limitations can be encountered in the choice of Δt_i but the maximum allowable time-step remains much larger than the time-step associated with the Euler-explicit strategy. However, the price to pay for such an improved stability and efficiency is the need to solve the non-linear problem (in the unknown U^{n+1}) (2.36) at each time-step. When dealing with the Euler or Navier-Stokes equations on unstructured grids, U^{n+1} is the solution of :

$$\frac{U_i^{n+1} - U_i^n}{\Delta t_i} = -\frac{1}{|\Omega_i|} \sum_{k \in \mathcal{I}(\Omega_i)} \sum_{g=1}^{N_g} \omega_g (\mathcal{H}_{i,k}^{n+1})_g |\Gamma_{i,k}| = -\mathcal{E}(U^{n+1}), \quad (2.37)$$

Rather than solving this non-linear problem, which might be very expensive, one may take advantage of the fact that only the steady-state solution of this problem is of interest in this work (this steady-state solution satisfies $\mathcal{E}(U) = 0$). Following this idea, the implicit solution update is built in the following form :

$$\frac{\Delta U_i^n}{\Delta t_i} + \frac{1}{|\Omega_i|} \sum_{k \in \mathcal{I}(\Omega_i)} (\Delta \mathcal{H}_{i,k}^{imp})^n |\Gamma_{i,k}| = -\mathcal{E}(U^n), \quad (2.38)$$

where $\Delta U_i^n = U_i^{n+1} - U_i^n$ is the unknown time-increment of the solution and $(\Delta \mathcal{H}^{imp})^n = (\mathcal{H}^{imp})^{n+1} - (\mathcal{H}^{imp})^n$ is a time-increment on the numerical flux \mathcal{H}^{imp} retained in the so-called implicit stage (left-hand-side of formula (2.38)); a key point is that this implicit numerical flux formula is not necessarily the same as the one used in the explicit stage $\mathcal{E}(U^n)$. The only requirement on the choice of \mathcal{H}^{imp} is that, when coupled with the explicit numerical flux \mathcal{H} appearing in \mathcal{E} , it drives the explicit stage to zero - corresponding to the target steady-state in a stable and efficient way. Formula (2.38) can also be expressed as :

$$\frac{\Delta U_i^n}{\Delta t_i} + \mathcal{I}(\Delta U^n) = -\mathcal{E}(U^n). \quad (2.39)$$

where the couple implicit stage \mathcal{I} / explicit stage \mathcal{E} determines the convergence rate of the scheme to a steady-state $\Delta U = 0$, the accuracy of which is solely defined by the explicit stage \mathcal{E} . The computations to be presented in the next chapter of this report make use of various numerical flux formula \mathcal{H} to define the explicit flux balance (second or third-order Roe scheme, second or third-order RB scheme in particular). However, a single numerical flux formula will be used for \mathcal{H}^{imp} ; this formula is only first-order accurate in space but this is of no consequence on the result, because the implicit flux balance vanishes at steady-state. More importantly this formula is designed so as to ensure a stable formulation for large values of Δt_i when coupled with various explicit formulas; its simplicity allows to minimize the cost of solving the linear algebraic system associated with the implicit stage, thus provides a globally efficient time-integration. The implicit numerical flux is decomposed as $\mathcal{H}^{imp} = \mathcal{H}^{E(imp)} - \mathcal{H}^{V(imp)}$ where $\mathcal{H}^{E(imp)}$ is the inviscid or Euler implicit numerical flux and $\mathcal{H}^{V(imp)}$ denotes the viscous implicit numerical flux.

The inviscid numerical flux increment on a cell face used in the implicit stage throughout this work reads :

$$\left(\Delta \mathcal{H}_{i,k}^{E(imp)}\right)^n = \frac{1}{2} [(\Delta \mathcal{F}_i^E)^n \cdot \mathbf{n}_{i,k} + (\Delta \mathcal{F}_{o(i,k)}^E)^n \cdot \mathbf{n}_{i,k} - (\rho_\perp^E)_{i,k}^n (\Delta U_{o(i,k)}^n - \Delta U_i^n)]. \quad (2.40)$$

where ρ_\perp^E is the spectral radius of the Jacobian matrix J_\perp (see also (2.16)). Formula (2.40) derives from ideas originally proposed in the work of Jameson and Turkel [26], evolving through [27] and [44] to yield the present matrix-free implicit stage, used in particular by Lohner and his co-workers for solving the unsteady Navier-Stokes equations on unstructured grids [35] [36]. This flux increment corresponds in fact to the choice of the first-order Rusanov scheme to build the implicit numerical flux formula.

For a 2D system, the time-increment of the physical viscous flux projected onto the face normal \mathbf{n} takes the form :

$$(\Delta(\mathcal{F}^V)^\perp)^n = (\Delta f^V)^n n_x + (\Delta g^V)^n n_y, \quad (2.41)$$

making use of the viscous Jacobian matrices previously introduced, the physical flux increment $(\Delta f^V)^n$ can be expanded as :

$$(\Delta f^V)^n = \frac{\partial f^V}{\partial U} \Delta U^n + \frac{\partial f^V}{\partial U_x} (\Delta U_x)^n + \frac{\partial f^V}{\partial U_y} (\Delta U_y)^n = A_0^V \Delta U^n + A_1^V (\Delta U_x)^n + A_2^V (\Delta U_y)^n, \quad (2.42)$$

and similarly for $(\Delta g^V)^n$. Introducing the partial derivatives U_\perp and U_\parallel with respect to the normal and tangent directions to the cell face (such that $U_x = U_\perp n_x - U_\parallel n_y$ and $U_y = U_\perp n_y + U_\parallel n_x$), inserting these quantities into the expansion of $(\Delta f^V)^n$ and $(\Delta g^V)^n$ and combining both developments yields :

$$(\Delta(\mathcal{F}^V)^\perp)^n = (J_0^V)^n \Delta U^n + (J_\perp^V)^n \Delta U_\perp^n + (J_\parallel^V)^n \Delta U_\parallel^n, \quad (2.43)$$

where $(J_0^V)^n = A_0^V n_x + B_0^V n_y$, $(J_\perp^V)^n = A_1^V n_x^2 + B_2^V n_y^2 + (A_2^V + B_1^V) n_x n_y$, $(J_\parallel^V)^n = A_2^V n_x^2 - B_1^V n_y^2 + (B_2^V - A_1^V) n_x n_y$. If the time-increment of the implicit numerical viscous flux $\Delta\mathcal{H}^{V(imp)}$ is built as the centered discretization of (2.43), the resulting scheme will be expensive to solve, with in particular blocks containing J_0^V , (J_\perp^V) , (J_\parallel^V) to invert. If only contributions involving a positive-definite matrix coefficient are retained, the calculation cost will be dramatically reduced because this full matrix will be simply replaced by its spectral radius, without compromising the scheme stability. This is why the numerical viscous flux eventually retained in the implicit stage corresponds to the following simplified version of (2.43) :

$$(\Delta\mathcal{H}^{V(imp)})^n = (J^V)^n \Delta U_\perp^n, \quad (2.44)$$

with $(J^V)^n = A_1^V n_x^2 + B_2^V n_y^2$ a positive-definite matrix (A_1^V and B_2^V are positive definite matrices), with spectral radius $(\rho_\perp^V)^n$. The normal derivative of U in (2.44) is computed with a simple centered approximation by using the values at the centroid of the cells sharing the face $\Gamma_{i,k}$:

$$\Delta(U_\perp^n)_{i,k} = \frac{\Delta U_{o(i,k)}^n - \Delta U_i^n}{|\Delta\mathbf{r}_{i,o} \cdot \mathbf{n}_{i,k}|}, \quad (2.45)$$

where $\Delta\mathbf{r}_{i,o}$ is the position difference vector from the centroid of cell i to the centroid of cell o (short for $o(i,k)$ when there is no ambiguity). The numerical viscous flux in the implicit stage is finally given by :

$$(\Delta\mathcal{H}_{i,k}^{V(imp)})^n = \frac{(\rho_\perp^V)_{i,k}^n}{|\Delta\mathbf{r}_{i,o} \cdot \mathbf{n}_{i,k}|} (\Delta U_{o(i,k)}^n - \Delta U_i^n). \quad (2.46)$$

Inserting (2.40) and (2.46) into (2.38) and taking into account $\sum_{k \in \mathcal{I}(\Omega_i)} \mathbf{n}_{i,k} = 0$ yields :

$$\frac{\Delta U_i^n}{\Delta t_i} + \frac{1}{|\Omega_i|} \sum_{k \in \mathcal{I}(\Omega_i)} \left[\frac{1}{2} (\Delta\mathcal{F}_{o(i,k)}^E)^n \cdot \mathbf{n}_{i,k} - \left(\frac{1}{2} (\rho_\perp^E)_{i,k}^n - \frac{(\rho_\perp^V)_{i,k}^n}{|\Delta\mathbf{r}_{i,o(i,k)} \cdot \mathbf{n}_{i,k}|} \right) (\Delta U_{o(i,k)}^n - \Delta U_i^n) \right] |\Gamma_{i,k}| = -\mathcal{E}(U^n),$$

after introducing the scalar coefficients $C_{i,k}^n$ and D_i^n defined by:

$$\begin{cases} C_{i,k}^n = (\rho_\perp^E)_{i,k}^n + \frac{2(\rho_\perp^V)_{i,k}^n}{|\Delta\mathbf{r}_{i,o} \cdot \mathbf{n}_{i,k}|}, \\ D_i^n = \frac{1}{\Delta t_i} + \frac{1}{2|\Omega_i|} \sum_{k \in \mathcal{I}(\Omega_i)} C_{i,k}^n |\Gamma_{i,k}|, \end{cases} \quad (2.47)$$

the simplified implicit stage used in this work can be expressed as :

$$D_i^n \Delta U_i^n - \frac{1}{2|\Omega_i|} \sum_{k \in \mathcal{I}(\Omega_i)} C_{i,k}^n \Delta U_{o(i,k)}^n |\Gamma_{i,k}| = -\mathcal{E}(U^n) - \frac{1}{2|\Omega_i|} \sum_{k \in \mathcal{I}(\Omega_i)} (\Delta\mathcal{F}_{i,k}^E)^n \cdot \mathbf{n}_{i,k} |\Gamma_{i,k}|, \quad (2.48)$$

The implicit treatment (2.48) is said to be matrix-free so as to emphasize $C_{i,k}^n$ and D_i^n are scalar coefficients which are inexpensive to compute. In this work, (2.48) will be simply solved using a Point-Jacobi relaxation technique, which is retained for its very low cost per iteration and very modest memory requirements, which makes up for its rather low intrinsic efficiency; in practice, the number of iterations to reach a steady-state is of course larger than the number of iterations offered

by a more sophisticated solution method but each iteration is extremely cheap. The time-integration from level n to $n + 1$ takes the form :

$$\begin{cases} \Delta U_i^{(0)} = 0, \\ l = 1, L \\ D_i^n \Delta U_i^{(l+1)} = -\mathcal{E}(U^n) - \frac{1}{2|\Omega_i|} \sum_{k \in \mathcal{I}(\Omega_i)} \left((\Delta \mathcal{F}_{i,k}^E)^{(l)} \cdot \mathbf{n}_{i,k} - C_{i,k}^m \Delta U_{o(i,k)}^{(l)} \right) |\Gamma_{i,k}|, \\ \Delta U_i^n = \Delta U_i^{(L)}. \end{cases} \quad (2.49)$$

where $(\Delta \mathcal{F}_{i,k}^E)^{(l)} = (\mathcal{F}_{i,k}^E)^{(l)} - (\mathcal{F}_{i,k}^E)^n$. The typical value for the number of sub-iterations is $L = 10$. Note that the convergence rate of (2.49) to a steady-state depends on the use of a second or third-order conventional upwind or RB numerical flux in the explicit stage $\mathcal{E}(U^n)$ (specific comments on this point will be provided in the next chapter, when analyzing the numerical test-cases); however, this implicit treatment has been systematically used with these various explicit stages and shows good stability properties at large CFL numbers (typically, (2.49) has been used with $CFL = 10^6$ in the time-step formula (2.33)).

2.1.4 Limiting process

Since the interest of this work is on the accurate computation of compressible flows, we are bound to encounter flow configurations displaying discontinuous solution fields. In the next chapter, we will address in particular the computation of 2D and 3D steady transonic flows as well as the computation of a 2D unsteady supersonic flow. When dealing with such flows, it is well known that the solution reconstruction used for the $U^{L/R}$ states appearing in the numerical flux of a conventional upwind scheme but also in the non-dissipative part of the RB numerical flux must be completed by a limitation process in order to avoid the occurrence of oscillations in the computed solution that could lead to the failure of the computation.

Linear reconstruction with limiting For the second-order FV scheme, the Barth limiter modified according to Venkatakrishnan's proposal [46] (denoted VK limiter from now on) is introduced into the linear solution reconstruction (1.33). The reconstructed state U_g^r at a point g in cell i (g can be of course located on a boundary of cell i and coincide with a Gauss point) is given by :

$$U_g^r = U_i + \phi_i \Delta \mathbf{r}_{i,g} \cdot \nabla U_i. \quad (2.50)$$

where $\Delta \mathbf{r}_{i,g}$ is the position difference vector between the point g under consideration and the centroid of cell i where the solution gradient is made available, $\Delta \mathbf{r}_{i,g} = \mathbf{r}_g - \mathbf{r}_i$. The VK limiter is denoted by ϕ_i and is computed by the following formula for each cell i :

$$\phi_i = \frac{(\Delta_+)^2 + 2\Delta_+ \Delta_- + \varepsilon}{(\Delta_+)^2 + \Delta_+ \Delta_- + 2(\Delta_-)^2 + \varepsilon}, \quad (2.51)$$

where

$$\begin{cases} \text{if } \Delta \mathbf{r}_{i,g} \cdot \nabla U_i > 0 : \Delta_+ = U_j^{max} - U_i, \Delta_- = \Delta \mathbf{r}_{i,g} \cdot \nabla U_i, \\ \text{if } \Delta \mathbf{r}_{i,g} \cdot \nabla U_i < 0 : \Delta_+ = U_j^{min} - U_i, \Delta_- = \Delta \mathbf{r}_{i,g} \cdot \nabla U_i, \\ \text{if } \Delta \mathbf{r}_{i,g} \cdot \nabla U_i = 0 : \phi_i = 1. \end{cases}$$

with U_j^{max} and U_j^{min} the maximum and minimum values of the solution computed for all the face-neighbors of cell Ω_i . And the parameter ε is chosen to be $(Kh_i)^3$, with the characteristic length h_i of cell i , and the user-defined limiting parameter K , which is case-dependent and its value will be provided in the test cases presented in the next chapter.

Quadratic reconstruction with limiting and troubled cell detection When the quadratic solution reconstruction (1.36) is used for the third-order FV scheme, we follow the strategy initially proposed by Delanaye [16] : the VK limiter is still applied onto the linear part of the reconstruction while another switch, σ_i , is used for the quadratic part as follows :

$$U_g^r = U_i + ((1 - \sigma_i)\phi_i + \sigma_i)\Delta\mathbf{r}_{i,g} \cdot \nabla U_i + \frac{1}{2}\sigma_i(\Delta\mathbf{r}_{i,g})^T \cdot \mathbf{H}_i \cdot \Delta\mathbf{r}_{i,g}, \quad (2.52)$$

The quantity σ_i allows the reconstruction to switch from "quadratic without limiting" to "linear with limiting" when the local flow goes from "smooth" to "discontinuous" or at least "with strong local variations". The function σ_i will be called a sensor and is designed so as to allow a smooth transition between the linear reconstruction with limiter to the quadratic reconstruction when going from high-gradient to low-gradient regions. It must be emphasized the sensor σ_i used in this work is the one proposed in [39]. It is computed as follows :

$$\sigma_i = \frac{1 - \tanh(S(\epsilon_i - \beta))}{2}. \quad (2.53)$$

where ϵ_i is the troubled cell indicator; the parameter β is a threshold to be determined for each problem. For a given β , $\epsilon_i > \beta$ in a cell i means that this cell is located in a high-gradient region, which leads to a σ_i close to 0 : the quadratic part is then removed from the solution reconstruction and the linear part is limited by the VK limiter. Reversely, if $\epsilon_i < \beta$, the solution in cell i is considered to be smooth enough, which yields a value of σ_i close to 1 : the full quadratic solution reconstruction is recovered. The parameter S in (2.53) is a user-defined constant controlling the stiffness of the hyperbolic tangent function : a small value for S will make the hyperbolic tangent switch smoother. In our computations it is found that the value of ϵ is normally between 0 and 0.2. Therefore the variation of the sensor σ for a ϵ in this value range is showed in Figure 2.7. The possible choices for the parameters β and S will be detailed in the numerical test cases presented in the next chapter.

Based on the truncation error indicator initially designed by Löhner[37], the troubled cell indicator is derived in [16] and used in our computation, which takes the form :

$$\epsilon_i = \frac{\sum_{j \in \mathcal{I}(i)} |\Delta\mathbf{r}_{i,j} \cdot (\nabla q_j - \nabla q_i)|}{\sum_{j \in \mathcal{I}(i)} (|\Delta\mathbf{r}_{i,j} \cdot \nabla q_j| + |\Delta\mathbf{r}_{i,j} \cdot \nabla q_i|) + \alpha_i \bar{q}}, \quad (2.54)$$

where $\mathcal{I}(i)$ is the set of cells belonging to the stencil used for the gradient computation in cell i , $\Delta\mathbf{r}_{i,j}$ is still a position difference vector connecting the centroid of cell i and the centroid of the cell j in the set $\mathcal{I}(i)$, ∇q is the gradient of an indicator variable q , that will be specified below. The average state \bar{q} is computed by :

$$\bar{q} = \sum_{j \in \mathcal{I}(i)} (|q_j| + |q_i|), \quad (2.55)$$

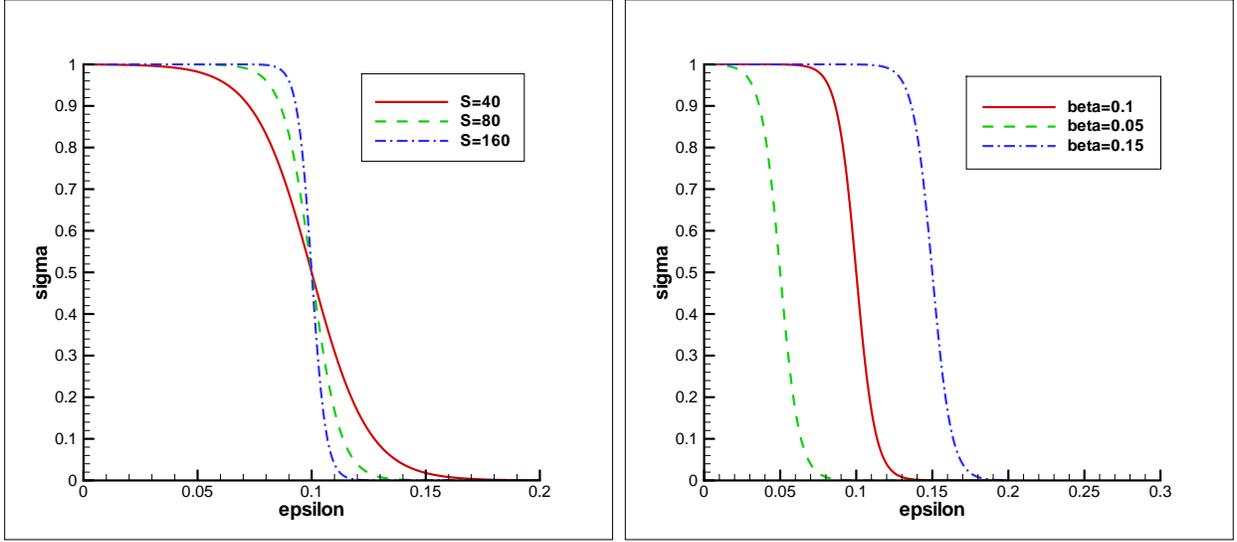


Figure 2.7: **The variation of the sensor** σ_i the effect of S with $\beta = 0.1$ (left), effect of β with $S = 80$ (right).

Basically, formula (2.54) is a multidimensional estimation of the ratio between second-order and first-order variations of the solution vector in cell i . The term $\alpha_i \bar{q}$ prevents formula from (2.54) becoming undefined in smooth-flow regions where all the terms involving gradients will become close to zero. In high-gradient regions, this same term is not needed. To account for this behavior, the parameter α is designed as the following :

$$\alpha_i = \frac{\gamma \bar{q}}{\sum_{j \in \mathcal{I}(i)} |\Delta \mathbf{r}_{i,j} \cdot (\nabla q_j - \nabla q_i)| + \sum_{j \in \mathcal{I}(i)} (|\Delta \mathbf{r}_{i,j} \cdot \nabla q_j| + |\Delta \mathbf{r}_{i,j} \cdot \nabla q_i|) + \bar{q}}. \quad (2.56)$$

In regions characterized by high density gradients, α is close to 0, while in quasi-uniform flow regions α tends to γ , which is chosen as $\gamma = 0.5$.

The indicator variable q could be density, pressure, velocity or other flow variables. In our computation, density has been systematically used and judged to yield satisfactory results. Further refinement of the sensor could be gained by using a combination of variables but this path has not been investigated. In Figure 2.8, the distribution of the troubled cell indicator ϵ_i and its parameter α_i for a transonic flow with inlet condition $Mach = 0.8$, attack angle 1.25° over a NACA0012 airfoil is showed for an exemple. It is found that α is close to $\gamma = 0.5$ in the smooth flow region, and smaller than this value in regions where the gradient is large, the minimal value is in the shock region where the gradient is the maximum in the whole flow field. But the value is not close to 0, possibly because the gradient is not big enough to make it be. The variation of ϵ_i shows well the strong shock on the suction side, while the weak shock on the pressure side is not detected because the density gradient variation is small.

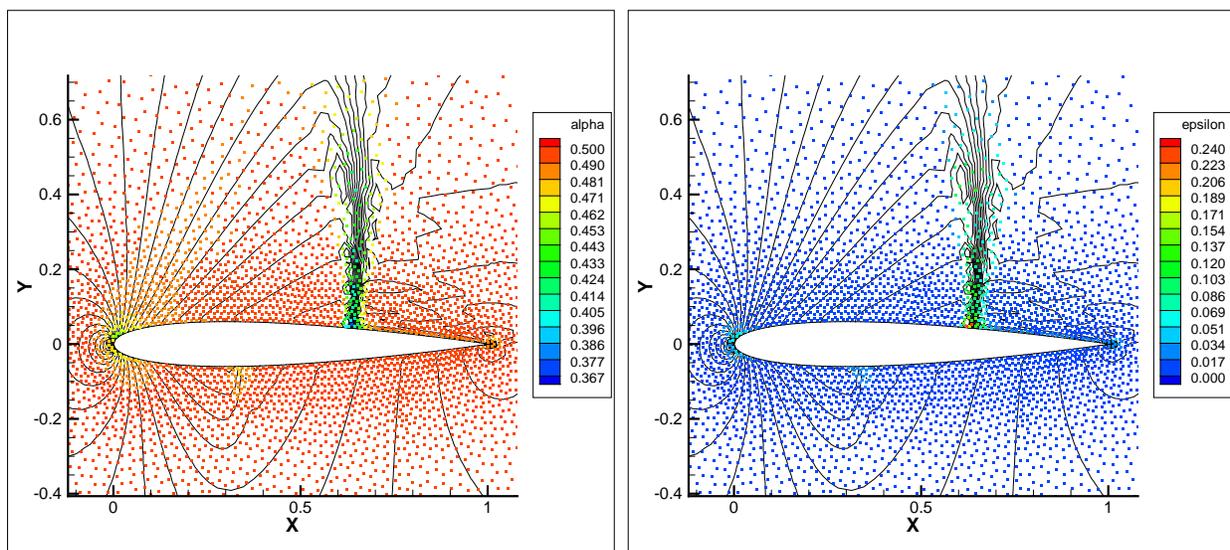


Figure 2.8: **Distribution of the troubled cell indicator ϵ_i and its parameter α_i for a transonic flow over the NACA0012 airfoil α_i distribution (left), and ϵ_i distribution (right), Mach number iso-contour (black lines in both figures).**

2.2 Formulation for 3D steady flows

The design principles of the finite volume method applied in 3D are similar to what has been described up to now in 2D. Therefore we will simply review in this section the salient features of the FV conventional upwind or RB schemes and point out the main quantitative differences that appear when going from 2D to 3D, in particular in terms of stencil used for gradient computation in each cell, and the shape of shift cell used in RB dissipation flux computation.

2.2.1 Cell gradient computation

For a conventional upwind scheme but also for the non-dissipative part of the RB numerical flux in 3D, the solution at any point in a cell is still reconstructed by equation (1.33) to second-order, and by equation (1.36) to third-order, where the solution gradient and Hessian (for third-order scheme) at a cell centroid are also computed by the Least-square method. The difference with the 2D scheme is the number of cells in the stencil used for the least-square reconstruction (see Table 2.1). For the second-order scheme, there are 4 degrees of freedom necessary to be determined in the solution polynomial in 3D, therefore at least 4 cells in the stencil are necessary for the linear least-square reconstruction. For an internal hexahedron element, its 6 face-neighbors are enough for this demand; and a tetrahedron element has exactly 4 face-neighbors, which sometimes can result in a singular reconstruction when the grid cell is really distorted, therefore edge neighbors of the cell are added into the stencil. For cells located on the boundary, it is also the case because of insufficient numbers of their face-neighbors. For the third-order scheme, the solution polynomial needs 10 degrees of freedom in 3D, so a stencil with at least 10 cells is necessary. For an internal hexahedron cell, the sum of its face-neighbor and edge-neighbor is 14, which can satisfy this requirement; this sum of an internal regular tetrahedron is 16, which is also enough. But in actual computation, it is found that the minimum stencil number 18 is necessary to ensure the robustness of the reconstruction.

Reconstruction	2D minimum/maximum	3D minimum/maximum
Linear	3/10	6/16
Quadratic	8/20	18/36

Table 2.1: **Minimal number of cells in the stencil used by least-square reconstruction.**

In Table 2.1, it is found that the minimal and possible maximal number of cells in the stencil needed for the least-square reconstruction. The 3D reconstruction needs about 1.6 to 2.2 times more cells in the stencil than the reconstruction in 2D, which leads to store a very large matrix M_i^{-1} (see equation (1.40)), therefore a huge memory requirement for 3D computation.

2.2.2 Quadrature points for flux integration on cell faces

In 2D the number of quadrature points used for physical flux integration on faces of a certain cell is related to the precision order of the scheme, while this number depends not only on the scheme order but also the shape of the considered face. For the second-order scheme, only 1 quadrature point is necessary which is the center of the face regardless of its shape; but for the third-order scheme, 3 quadrature points are needed on a triangle face and 4 quadrature points are necessary on a quadrilateral face (see Figure 2.9).

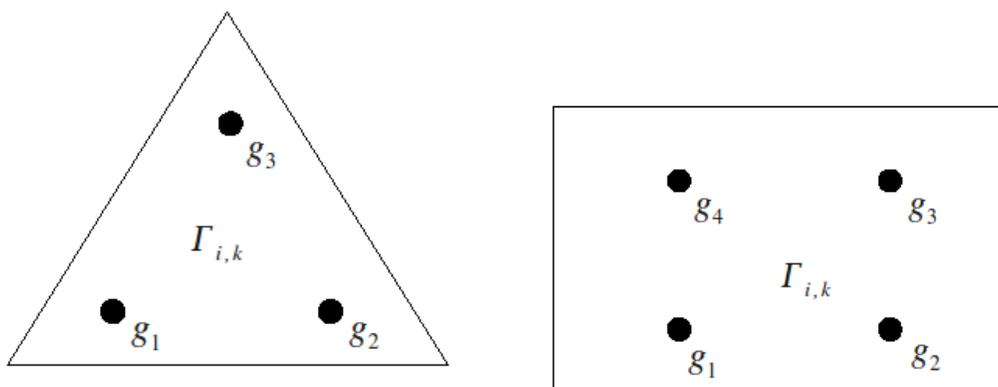


Figure 2.9: **Gauss quadrature points used for third-order FV scheme** 3 points used by a triangle face (left), 4 points used by a quadrilateral face (right).

2.2.3 Node solution and node gradient computation stencil

For the 2D case, the stencil needed for computation of the solution and solution gradient is the same for a mesh node inside the computation domain (internal node), while the stencil used for node value computation could be not enough for the node gradient computation for a node located on the computational domain boundary (boundary node), and in this case, face-neighbors of cells in the node-value-stencil have to be added into the node-gradient-stencil. It is also the situation in 3D. Although 4 is the minimal number of cells in the node-gradient-stencil for the linear least-square

reconstruction in 3D, 6 cells are guaranteed in the stencil to avoid the instability of the reconstruction. Table 2.2 shows that the minimum and possible maximum stencil used for node solution and gradient computation (for viscous case only). It is found that the stencil size for the node solution and gradient computation in 3D can be 4.9 times more than the one in 2D, the much higher computation cost of the 3D case is obvious.

Computation	2D minimum/maximum	3D minimum/maximum
Node solution	2/9	4/44
Node gradient	3/9	6/44

Table 2.2: Number of cells in the stencil used for solution and its gradient computation on a mesh node.

2.2.4 Shift cell for RB scheme

As we have seen in 2D, the shift cell $\Omega_{i,k}$ constructed for the residual computation in RB scheme numerical dissipation is always a quadrilateral regardless of the shape of the cell which it crossed. But in 3D case, this shift cell changes its shape with the type of elements on which it is based. There are two possible types of shape showed in Figure (2.10) (red dotted line). In order to compute the residual in the shift cell with equation (2.8), the numerical flux at the center of a face Γ_l with vertices $(N_{i,k}^1, N_{i,k}^2, C_o)$ is computed by:

$$(\mathcal{H}_c^E)_l = \frac{1}{3} \left(F^E(U_{N_{i,k}^1}) + F^E(U_{N_{i,k}^2}) + F^E(U_{C_o}) \right) \cdot \mathbf{n}_l. \quad (2.57)$$

for viscous case, the numerical viscous flux at each face center is computed in the same way.

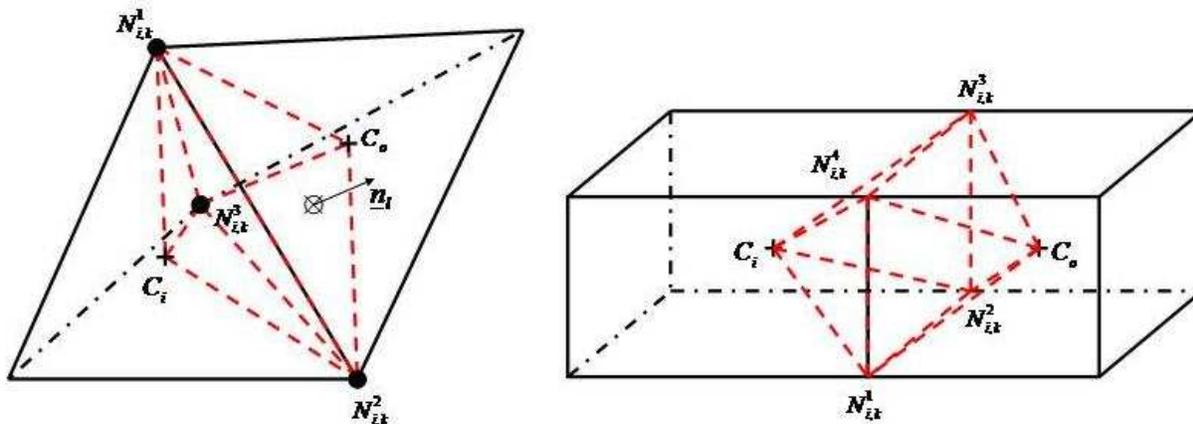


Figure 2.10: Possible shift cell for RB dissipation calculation in 3D

2.3 Formulation for 2D unsteady flows

We conclude our description of the FV RB scheme for computations on unstructured grids with the adaptation of the steady solver to unsteady flows. This adaptation is performed in the framework of a dual-time formulation [25] with a time-accuracy which remains limited to second-order.

2.3.1 Dual-time discretization method

The basic idea of a dual-time strategy can be formulated as follows : the solution of the unsteady equations $U_t + \nabla \cdot \mathcal{F} = 0$ is found as the steady solution with respect to the so-called dual or fictitious time τ of the evolution problem $U_\tau + U_t + \nabla \cdot \mathcal{F} = 0$, which can also be recast in the form $U_\tau + \nabla \cdot \mathcal{F} = -U_t$. Since the steady solution of $U_t + \nabla \cdot \mathcal{F} = 0$ is such that $\nabla \cdot \mathcal{F} = 0$, it can also be seen as the steady solution with respect to a dual-time τ of the evolution problem $U_\tau + \nabla \cdot \mathcal{F} = 0$. Thus, starting from a steady solver, the development of an unsteady solver simply requires to insert the existing (dual) time loop within a physical time loop and to account for a source term $S = -U_t$ in the discretization. In the case of the RB scheme, the extension to unsteady flows follows the same general lines but specific developments are needed when building the RB dissipation since it relies on a residual r that includes now the physical time derivative.

A dual-time discretization applied to the FV formulation of the 2D Euler equations reads :

$$\frac{\partial U_i}{\partial \tau} + \frac{\partial U_i}{\partial t} + \frac{1}{|\Omega_i|} \sum_{k \in \mathcal{I}(\Omega_i)} \int_{\Gamma_{i,k}} \mathcal{F}^E \cdot \mathbf{n} d\Gamma = 0, \quad (2.58)$$

The dual-time finite-volume approach considered in the present study drives (2.58) to a steady-state with respect to τ by using :

- a first-order (Euler implicit) approximation for the dual time-derivative, which will vanish at steady-state anyway,
- a second-order (three-level) implicit approximation for the physical time-derivative,
- a Gauss-quadrature formula for approximating the flux integral on the face $\Gamma_{i,k}$.

The resulting discretization of equation (2.58) reads:

$$\begin{cases} \frac{\Delta U_i^{n,m}}{\Delta \tau_i^{n,m}} + \mathcal{R}_i^{n,m} = 0, \\ \mathcal{R}_i^{n,m} = \mathcal{T}(U_i^{n,m}, U_i^n, U_i^{n-1}) + \frac{1}{|\Omega_i|} \sum_{k \in \mathcal{I}(\Omega_i)} \sum_g \omega_g (\mathcal{H}_{i,k}^E)^{n,m} |\Gamma_{i,k}|, \end{cases} \quad (2.59)$$

where m is the pseudo-time (or dual-time) iteration counter, n is the physical time iteration counter, $\Delta U_i^{n,m} = U_i^{n,m+1} - U_i^{n,m}$ with $U_i^{n,0} = U_i^n$. The time-discretization operator \mathcal{T} is chosen as :

$$\mathcal{T}(U_i^{n,m}, U_i^n, U_i^{n-1}) = \frac{\frac{3}{2}(U_i^{n,m} - U_i^n) - \frac{1}{2}\Delta U_i^{n-1}}{\Delta t_i}, \quad (2.60)$$

with $\Delta U_i^{n-1} = U_i^n - U_i^{n-1}$; it is such that $\mathcal{T}(U_i^{n+1}, U_i^n, U_i^{n-1}) = (U_t)_i^{n+1} + \mathcal{O}(\Delta t^2)$. Besides, the numerical flux balance approximates the physical flux balance at order $p = 2$ or $p = 3$. Therefore, when the pseudo-time marching reaches a steady solution $U^{n+1} = U^{n,m+1} = U^{n,m}$, scheme (2.59) yields :

$$(U_t)_i^{n+1} + \mathcal{O}(\Delta t^2) + \frac{1}{|\Omega_i|} \sum_{k \in \mathcal{I}(\Omega_i)} \int_{\Gamma_{i,k}} (\mathcal{F}^E)^{n+1} \cdot \mathbf{n} d\Gamma + \mathcal{O}(h^p) = 0. \quad (2.61)$$

which is an approximation of the unsteady flow solution at order 2 in time and p in space.

In practice, scheme (2.59) is made implicit to speed up the convergence to the pseudo steady-state. The slight modification brought by the dual-time approach to the matrix-free implicit stage previously presented for steady flow computations will be detailed below. At first, we wish to emphasize the difference between a conventional scheme and the RB scheme when extended to the calculation of unsteady flows using the dual-time strategy. A conventional scheme is based on decoupled approximations for the various terms appearing in the system of conservation laws to which it is applied. Let us assume that a conventional scheme is applied to the discretization of the unsteady Navier-Stokes equations (for more generality) with a dual-time framework :

$$\underbrace{\frac{\partial U_i}{\partial \tau} + \frac{1}{|\Omega_i|} \sum_{k \in \mathcal{I}(\Omega_i)} \int_{\Gamma_{i,k}} \mathcal{F}^E \cdot \mathbf{n} d\Gamma}_{\text{physical inviscid flux balance } \mathcal{B}_{\mathcal{F}^E}} = \underbrace{\frac{1}{|\Omega_i|} \sum_{k \in \mathcal{I}(\Omega_i)} \int_{\Gamma_{i,k}} \mathcal{F}^V \cdot \mathbf{n} d\Gamma}_{\text{physical viscous flux balance } \mathcal{B}_{\mathcal{F}^V}} \underbrace{- \frac{\partial U_i}{\partial t}}_{\text{physical time derivative}}, \quad (2.62)$$

when the conventional scheme is used, the approximate solution of the above system is computed by this equation :

$$\frac{\partial U_i}{\partial \tau} + \mathcal{B}_{\mathcal{H}^E} = \mathcal{B}_{\mathcal{H}^V} - S. \quad (2.63)$$

where

- the numerical inviscid flux balance $\mathcal{B}_{\mathcal{H}^E}$ depends only on the physical inviscid fluxes,
- the numerical viscous flux balance $\mathcal{B}_{\mathcal{H}^V}$ depends only on the physical viscous fluxes,
- the numerical approximation for the source term S depends only on the physical time-step and the solution vector U .

Now, when the RB scheme is applied to this same system (2.62), it can be put under the same generic form (2.63) but the key difference now is that the RB numerical inviscid flux depends on the residual of the full system (2.62) for building its dissipative flux, hence the numerical inviscid flux balance $\mathcal{B}_{\mathcal{H}^E}$ depends not only on the physical inviscid fluxes but also on the physical viscous fluxes as well the physical time-step and the solution vector U . In the next section the design details of the RB numerical flux for unsteady inviscid flows will be given.

2.3.2 RB numerical flux

The RB numerical flux remains formally given by (2.2), which is decomposed as a purely centered flux contribution on one hand and a dissipative contribution on the other hand. For unsteady problems, the purely centered flux is computed in the same way as for the steady case while the residual approximation (2.2) in the dissipation flux (2.4) must take into account the physical time-derivative approximation. Let us recall the residual integral in a shift cell (see Figure 2.1) related to the considered face $\Gamma_{i,k}$:

$$R_{i,k} = \frac{1}{|\Omega_{i,k}|} \int_{\Omega_{i,k}} r \, d\Omega, \quad (2.64)$$

where the residual r at steady-state is now associated with the unsteady system (2.58), that is $r = U_t + \nabla \cdot \mathcal{F}^E$. The integral in (2.64) can then be expressed as:

$$R_{i,k} = \frac{1}{|\Omega_{i,k}|} \int_{\Omega_{i,k}} U_t \, d\Omega + \frac{1}{|\Omega_{i,k}|} \int_{\partial\Omega_{i,k}} \mathcal{F}^E \cdot \mathbf{n} \, d\Gamma, \quad (2.65)$$

and is discretized as $\mathcal{R}_{i,k} = \mathcal{R}_{i,k}^E + \mathcal{R}_{i,k}^t$ where $\mathcal{R}_{i,k}^E$ approximates the inviscid flux balance over the shift cell $\Omega_{i,k}$ and $\mathcal{R}_{i,k}^t$ approximates the physical time-derivative over the shift cell. The evaluation of $\mathcal{R}_{i,k}^E$ remains unchanged with respect to the steady case - note only the solution used when computing the numerical flux balance over the shift cell is $U^{n,m}$ instead of U^n in the steady case - :

$$\mathcal{R}_{i,k}^E = \frac{1}{|\Omega_{i,k}|} \sum_{l \in \mathcal{I}(\Omega_{i,k})} (\mathcal{H}^E)_l^{n,m} |\Gamma_l|. \quad (2.66)$$

where the numerical flux $(\mathcal{H}^E)_l^{n,m}$ is computed by using (2.9).

In order to build $\mathcal{R}_{i,k}^t$, the physical time-derivative is approximated at face center $M_{i,k}$ and the time discretization operator (2.60) is used to yield :

$$\mathcal{R}_{i,k}^t = \frac{1}{2} \left(\mathcal{T}(U_i^{n,m}, U_i^n, U_i^{n-1}) + \mathcal{T}(U_{o(i,k)}^{n,m}, U_{o(i,k)}^n, U_{o(i,k)}^{n-1}) \right). \quad (2.67)$$

2.3.3 New sensor for the RB scheme

If the RB scheme is applied to unsteady problems involving discontinuities, limited values of the reconstructed states $U^{L/R}$ are used in the numerical flux. Although this strategy seems sufficient to ensure the robustness of the computations for a large panel of applications, it was observed in the course of the ADIGMA project that the RB scheme failed to compute problems such as the interaction of blast waves in 1D or the double Mach reflection problem in 2D, both involving very strong shocks around which the RB scheme developed fatal oscillations. Some numerical experiments allowed to identify that this lack of robustness in the vicinity of discontinuities is related to the estimation of the physical time-derivative in the residual used to compute the RB numerical dissipation. When the term $\mathcal{R}_{i,k}^t$ was taken out from (2.65), an excellent robustness was observed for the RB scheme, but at the expense of accuracy since the scheme's accuracy drops down to first order when the discrete residual is no longer consistent with the exact residual (which includes the physical time-derivative). It was thus proposed to compute the residual approximation on the interface as follows, in the unsteady case :

$$\mathcal{R}_{i,k} = \sigma_{i,k}^t \frac{1}{2} (\mathcal{T}(U_i^{n,m}, U_i^n, U_i^{n-1}) + \mathcal{T}(U_{o(i,k)}^{n,m}, U_{o(i,k)}^n, U_{o(i,k)}^{n-1})) + \frac{1}{|\Omega_{i,k}|} \sum_{l \in \mathcal{I}(\Omega_{i,k})} (\mathcal{H}^E)_l^{n,m} |\Gamma_l|, \quad (2.68)$$

The switch $\sigma_{i,k}^t$ is analogous to the switch σ_i used for the quadratic solution reconstruction. In regions dominated by flow discontinuities, the robustness of the scheme is locally improved by canceling the physical time-derivative estimate ($\sigma_{i,k}^t \rightarrow 0$), thus increasing the RB dissipation (from a third-order term to a first-order term); third-order accuracy is recovered away from these discontinuities with $\sigma_{i,k}^t \rightarrow 1$. For the sake of simplicity, or let us say because we were more interested by a demonstration of feasibility than by fine-tuning the proposed switch, a binary definition of $\sigma_{i,k}$ was retained :

$$\begin{aligned} \sigma_{i,k} &= 1 & \text{if } \beta_{i,k} < \beta_{RB}, \\ \sigma_{i,k} &= 0 & \text{if } \beta_{i,k} \geq \beta_{RB}, \end{aligned} \quad (2.69)$$

where the parameter β_{RB} is a threshold value which is not necessary equal to the threshold β used for the quadratic solution reconstruction. The parameter $\beta_{i,k}$ is directly linked with the troubled-cell indicator computed in each grid cell :

$$\beta_{i,k} = \frac{1}{2}(\epsilon_i + \epsilon_{o(i,k)}). \quad (2.70)$$

Note this strategy was not necessary for computing unsteady compressible flows with the RB scheme (on structured grids) in previous works [12], [11]; let us emphasize once again it was the occurrence of very strong shocks for some ADIGMA test-cases (Interacting Blast Waves and Double Mach Reflection) which specifically motivated the development of this switch in the unsteady version of the RB dissipative flux.

2.3.4 Adaptation of the implicit strategy

Since the robustness of the solver applied to the computation of flows involving shocks depends not only on the space-discretization procedure but also on the time-integration (both in dual and physical time), we deem to be important to provide some information on the implicit stage which is used to efficiently drive the second- or third-order FV-RB scheme to a pseudo steady-state. For unsteady flows computed with a dual-time approach, the steady solution with respect to the pseudo-time τ is obtained after a reduced number of pseudo-iterations by solving:

$$\frac{\Delta U_i^{n,m}}{\Delta \tau_i^{n,m}} + \frac{3}{2} \frac{\Delta U_i^{n,m}}{\Delta t_i} + \frac{1}{|\Omega_i|} \sum_{k \in \mathcal{I}(\Omega_i)} \left(\Delta \mathcal{H}_{i,k}^{E(imp)} \right)^{n,m} |\Gamma_{i,k}| = -\mathcal{R}_i^{n,m}, \quad (2.71)$$

where $\Delta(\mathcal{H}^{E(imp)})^{n,m} = (\mathcal{H}^{E(imp)})^{n,m+1} - (\mathcal{H}^{E(imp)})^{n,m}$ and $\mathcal{H}^{E(imp)}$ denotes the numerical flux formula retained in the implicit stage. The choice of a numerical flux $\mathcal{H}^{E(imp)}$ is the same as the one previously used in the steady case. The implicit stage is therefore formally unchanged with respect to (2.48) and given by :

$$D_i^{n,m} \Delta U_i^{n,m} - \frac{1}{2|\Omega_i|} \sum_{k \in \mathcal{I}(\Omega_i)} C_{i,k}^{n,m} \Delta U_{o(i,k)}^{n,m} |\Gamma_{i,k}| = -\mathcal{R}_i^{n,m} - \frac{1}{2|\Omega_i|} \sum_{k \in \mathcal{I}(\Omega_i)} (\Delta \mathcal{F}_{i,k}^E)^{n,m} \cdot \mathbf{n}_{i,k} |\Gamma_{i,k}|, \quad (2.72)$$

but the scalar diagonal coefficient also includes a contribution coming from the physical time derivative implicitation :

$$\begin{cases} C_{i,k}^{n,m} = (\rho_{\perp}^E)_{i,k}^{n,m}, \\ D_i^{n,m} = \frac{1}{\Delta \tau_i^{n,m}} + \frac{3}{2\Delta t_i} + \frac{1}{2|\Omega_i|} \sum_{k \in \mathcal{I}(\Omega_i)} C_{i,k}^{n,m} |\Gamma_{i,k}|. \end{cases} \quad (2.73)$$

The dual-time integration from sub-level (or inner-level) n, m to $n, m + 1$ takes the form :

$$\begin{cases} \Delta U_i^{(0)} = 0, \\ l = 0, L - 1 \\ D_i^{n,m} \Delta U_i^{(l+1)} = -\mathcal{R}_i^{n,m} - \frac{1}{2|\Omega_i|} \sum_{k \in \mathcal{I}(\Omega_i)} \left((\Delta \mathcal{F}_{i,k}^E)^{(l)} \cdot \mathbf{n}_{i,k} - C_{i,k}^{n,m} \Delta U_{o(i,k)}^{(l)} \right) |\Gamma_{i,k}|, \\ \Delta U_i^{n,m+1} = \Delta U_i^{(L)}. \end{cases} \quad (2.74)$$

Note that when Δt_i is chosen small enough so that the physical-time discretization in $\mathcal{R}_i^{n,m}$ is dominant with respect to the other contributions, and since this physical-time discretization is made fully implicit (contribution of the coefficient $\frac{3}{2\Delta t}$ in the diagonal coefficient $D_i^{n,m}$), the above implicit treatment converges to a steady-state after a few iterations only.

3

Applications of FV-RB scheme

French Les schémas RB d'ordre deux et d'ordre trois basés sur la méthode des volumes finis (notés FV-RB $O2$ et FV-RB $O3$ respectivement) décrits dans le dernier chapitre sont appliqués maintenant à une série de cas tests afin d'évaluer leur précision, efficacité et robustesse. Nos analyses se concentrent sur la comparaison entre les schémas FV-RB $O2$ et $O3$, ainsi que sur la comparaison entre les schémas FV-RB et un schéma classique décentré-amont d'ordre deux ou trois (typiquement le schéma de Roe avec une reconstruction de solution linéaire ou quadratique). La comparaison est menée dans l'ordre suivant :

- Premièrement on considère des écoulements stationnaires sans choc de fluide parfait en $2D$ et en $3D$ afin de ne pas avoir d'influences du limiteur de la solution. Pour chaque cas (en $2D$ ou en $3D$), une série des calculs sur un problème modèle dont la solution exacte est connue sont effectués pour évaluer l'ordre de précision effectif des schémas mis en oeuvre dans notre code de calcul de volumes finis en maillages non-structurés. Ensuite la version de ces schémas pour les équations d'Euler est utilisée pour calculer un cas classique bidimensionnel, l'écoulement subsonique de fluide parfait autour d'un profil NACA0012 et aussi un écoulement subsonique de fluide parfait autour d'un profil tridimensionnel proposé dans le projet ADIGMA. Les solutions exactes ne sont pas disponibles pour ces problèmes, mais des résultats de référence existent pour le problème $2D$. Cependant on emploie une stratégie d'évaluation de la solution proposée par le projet ADIGMA, au lieu de comparer le résultat avec ceux qui ont été publiés, car cette comparaison dépend souvent du schéma numérique utilisé, du choix de maillage et aussi du traitement des conditions aux limites. La stratégie utilisée ici est une étude de convergence en maillage avec un critère approprié. Cette étude est menée pour les schémas FV-RB développés ici et elle donne des éléments pour une comparaison claire entre le schéma FV-RB $O2$ et $O3$.
- Ensuite, l'analyse des schémas appliqué aux équations d'Euler est réalisée pour des écoulements avec chocs en $2D$ et en $3D$ afin d'évaluer la performance des limiteurs décrits dans le chapitre précédent.
- Pour le cas d'un écoulement visqueux, on reprend la méthode d'analyse utilisée en fluide parfait : un problème d'advection-diffusion, dont la solution exacte est connue, est calculé afin d'estimer l'ordre de précision effectif des schémas FV développés. Les versions de ces schémas pour les équations de Navier-Stokes sont ensuite appliquées au calcul d'écoulement laminaire autour d'un profil NACA0012, où l'étude de convergence en maillage est encore une fois menée.
- Enfin la dernière partie de ce chapitre traite de l'évaluation des schémas FV-RB pour les calculs d'écoulement instationnaire, dans le cas sans choc de la propagation d'une tourbillon et aussi dans un cas avec choc beaucoup plus difficile, la double réflexion de Mach.

English The second and third-order finite-volume residual-based scheme (denoted by FV-RB *O2* and FV-RB *O3* respectively) described in the previous chapter are now applied to a series of test cases in order to assess its accuracy, efficiency and robustness properties. Our analysis will be especially focused on the comparison between the FV-RB *O2* and *O3* schemes as well as the comparison between the FV-RB schemes and a classical second- or third-order upwind scheme (typically the Roe scheme with linear or quadratic solution reconstruction). The comparison is performed in the following progressive way :

- Firstly we deal with *2D* and *3D* steady smooth inviscid flows in order to analyze the schemes without interference effects of the solution limiting strategy. In each case (*2D* or *3D*), a preliminary series of computations on a model problem with known exact solution are performed so as to assess the order of accuracy achieved in practice by the schemes implemented within our unstructured finite-volume solver. Next, these schemes (extended for Euler equations) are applied to the computation of a well-known *2D* inviscid subsonic flow over a NACA0012 airfoil and a subsonic inviscid flow over a *3D* body proposed in the ADIGMA project. No exact solutions are available for these cases but numbers of references in the literature are available for the *2D* test-problem. However we adopt the solution evaluation strategy proposed by the ADIGMA project, rather than performing comparisons with results in the literature, which are often not only dependent on the numerical scheme but also on the choice of grid and the treatment of boundary conditions. The strategy used here is the grid convergence study with an appropriate criteria on the solution accuracy. This study is performed for the developed FV-RB schemes, and it provides elements for a clear comparison between FV-RB *O2* and *O3* schemes.
- Next, the analysis of schemes applied to the Euler equations is performed for *2D* and *3D* flows involving shocks in order to assess the performance of the solution limiting strategies for cases with discontinuities described in the previous chapter.
- In the case of *2D* viscous flows, we return to the line of analysis followed in the inviscid case : a model advection-diffusion problem of known exact solution is first computed in order to assess the actual order of accuracy provided by the developed FV schemes. These schemes (extended to the Navier-Stokes equations) are then applied to the computation of a laminar flow over the NACA0012 airfoil, where the grid-convergence study is also conducted.
- At last this chapter closes with the assessment of the FV-RB schemes for unsteady flow computations, both in the smooth case, a vortex propagation problem and a much more challenging case, the double Mach reflection problem.

3.1 Steady inviscid smooth flows

3.1.1 2D circular advection problem

The model scalar advection problem solved in this section takes the following form :

$$\frac{\partial W}{\partial t} + y \frac{\partial W}{\partial x} + (1 - x) \frac{\partial W}{\partial y} = 0, \quad (3.1)$$

where the flow domain corresponds to the square $x \in [0, 1]$, $y \in [0, 1]$. The initial condition in this domain is:

$$W(x, y, 0) = 0,$$

The edges $y = 0$ and $x = 0$ of the computational domain are inlet flow conditions on which the solution is imposed as :

$$W(x, y, t) = e^{-50(r-0.5)^2},$$

with $r = \sqrt{(x-1)^2 + y^2}$ the distance to the lower right corner of the flow domain (point $(1, 0)$). The edges $x = 1$ and $y = 1$ are outlet flow boundaries along which the solution is extrapolated from the interior domain.

Equation (3.1) is discretized using its conservative form $W_t + F(W)_x + G(W)_y = 0$ with $F(W, x, y) = yW$ and $G(W, x, y) = (1-x)W$. The wavespeed associated with each physical flux is respectively $a(W, x, y) = y$ for $F(W)$ and $b(W, x, y) = (1-x)$ for $G(W)$.

The exact solution of (3.1) with associated initial and boundary conditions is readily obtained in the space of characteristics : the characteristic lines of the problem are circular trajectories centered on point $(1, 0)$, entering the domain through $x = 0$ and $y = 0$ and leaving it through $x = 1$ and $y = 1$; along each of this line the inlet value is conserved. A typical numerical solution illustrating these features is displayed in Figure 3.1.

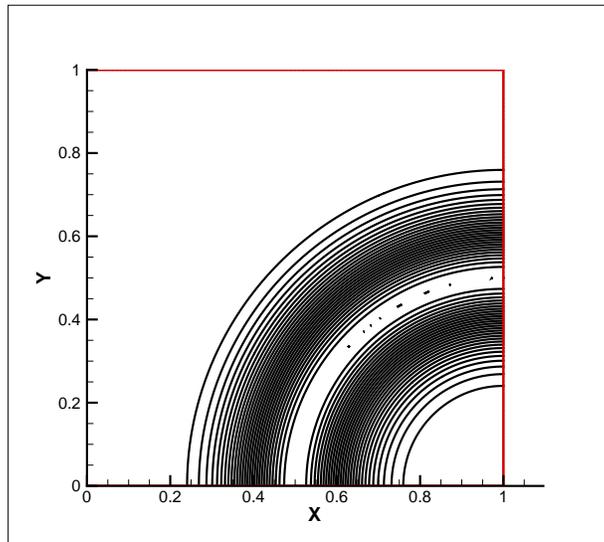


Figure 3.1: Contours of the computed solution W (30 levels from 0 to 1) obtained by the $O3$ RB scheme on mesh **Tri_irreg5** (irregular triangular grid with 6400 boundary faces).

In order to provide a more complete analysis of the scheme properties, the computations are carried out on two different series of meshes : one made of increasingly refined regular triangle meshes and the other made of increasingly refined irregular triangle meshes. The regular triangular grids are derived from regular quadrilateral grids in which each cell is cut along the top-left / bottom-right diagonal (see the left-side of Figure 3.2). The irregular triangular grids are obtained from an automatic Delaunay triangulation performed by a commercial grid generator when asked to mesh the

computational domain by using triangular cells with a prescribed number of regularly-spaced nodes on the domain boundaries (see the right-side of Figure 3.2). It is important to point out that at this stage the FV schemes developed in this work can be used indifferently on grids made of triangles or rectangles (or even hybrid grids). The analysis presented here is focused on triangular grids but some comments will be provided on the case of grids based on quadrilaterals.

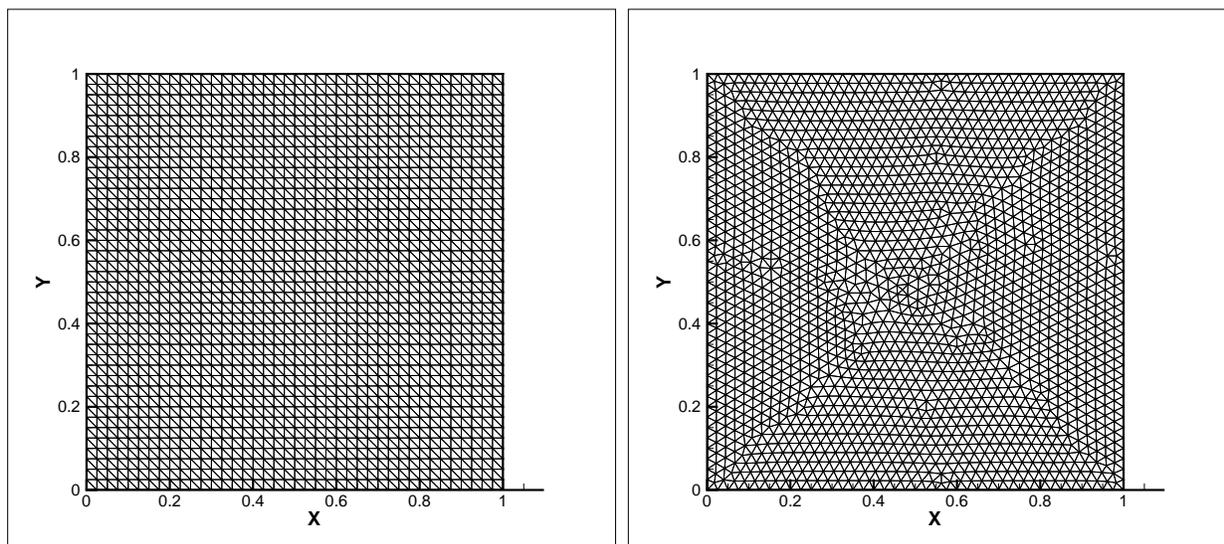


Figure 3.2: Mesh examples used for 2D circular advection problem: regular triangle mesh(left), irregular triangle mesh(right).

The main features of the computational meshes are summarized in Table 3.1 and 3.2 : the regular triangular grids are obtained from successive Cartesian grids with 10×10 , 20×20 , 40×40 , 80×80 and 160×160 nodes regularly spaced along each side of the unit square computational domain; the irregular triangular grids rely on the same boundary nodes but are obtained from a triangulation based on these boundary distributions. Note **dof** denotes the total number of degrees of freedom used for the computation; in the case of the finite-volume method used in this work, the number of degrees of freedom is equal to the number of mesh cells regardless of the accuracy order, while this is not the case when a Spectral Volume approach is developed for the RB schemes in the next chapter : each cell contains a certain number of degrees of freedom which increases with the desired accuracy order.

Mesh	Dof	Number of boundary faces
Tri_reg1	200	40
Tri_reg2	800	80
Tri_reg3	3200	160
Tri_reg4	12800	320
Tri_reg5	51200	640

Table 3.1: Basic information on the regular triangle-based meshes used for the 2D circular advection problem.

Mesh	Dof	Number of boundary faces
Tri_irreg1	226	40
Tri_irreg2	894	80
Tri_irreg3	3588	160
Tri_irreg4	14412	320
Tri_irreg5	57518	640

Table 3.2: Basic information on the irregular triangle-based meshes used for the 2D circular advection problem.

Accuracy analysis on regular triangular meshes The first series of computations is performed on the increasingly refined regular triangle-based meshes. For each steady solution achieved on a given grid, the logarithm of the L_2 -norm of the difference between the exact solution in the grid and the numerical solution provided by the scheme, also called L_2 error of the scheme, is computed and plotted as a function of the characteristic mesh size h , which is taken as the minimum of the cell mesh sizes computed by $h_i = \sqrt{\Omega_i}$ (in 2D), and $h_i = \Omega_i^{1/3}$ (in 3D). This L_2 error is computed both for the FV-RB scheme in its second- and third-order version and for the Roe scheme, which is taken as the representative of conventional upwind schemes, also in its second and third-order version. The numerical errors are plotted in Figure 3.3 and also summarized in Table 3.3 for the FV-RB scheme. The orders (of accuracy) appearing in this table correspond to the estimation computed for the slope of the curve *error vs mesh size* using the error and mesh size for the current grid and the previous one. The practical accuracy order should correspond to the asymptotic value reached by this slope when the finest grids are used for this estimation.

It can be observed on Figure 3.3 the expected theoretical accuracy orders are roughly achieved with the Roe numerical flux and linear then quadratic solution reconstruction, yielding respectively an asymptotic slope of 2.32 and 2.98 (the regularity of the grids is likely to introduce some error compensations which would explain the practical accuracy order exceeds 2 with the linear reconstruction). When the RB numerical flux is used, the error slopes obtained between two finest grids are respectively 1.98 for FV-RB O2 and 3.29 for FV-RB O3, which corresponds to the expected orders. More importantly maybe, it is interesting to note the error level achieved on the finest grid with the third-order RB numerical flux is much lower than the error level obtained with the third-order Roe scheme. In fact the error level associated with this third-order conventional upwind scheme on the finest grid (51200 **dof**) corresponds to the error level obtained with the third-order RB scheme on a grid with a number of **dof** between 3200 and 12800. This point will be detailed in the efficiency analysis in the last paragraph of this section.

Mesh	L_2 error of RBO2	order	L_2 error of RBO3	order
Tri_reg1	-1.32978	-	-1.37869	-
Tri_reg2	-2.05291	2.40	-2.30685	3.08
Tri_reg3	-2.71613	2.20	-3.63751	4.42
Tri_reg4	-3.31275	1.98	-4.92621	4.28
Tri_reg5	-3.90832	1.98	-5.91761	3.29

Table 3.3: L_2 norm of numerical errors and mesh convergence order obtained with FV-RB schemes on regular triangle meshes.

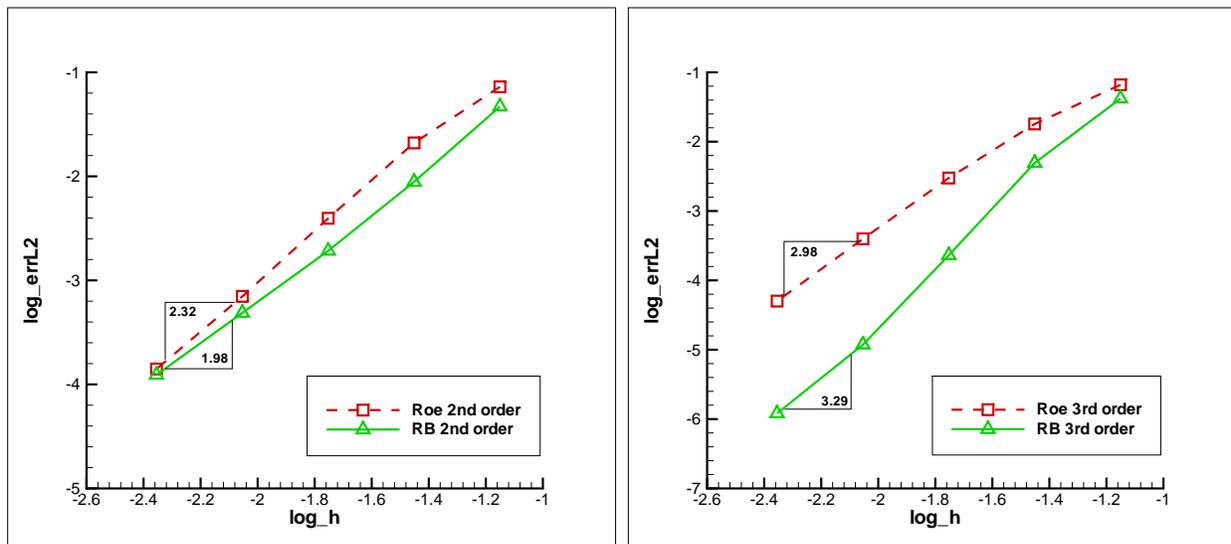


Figure 3.3: L_2 norm of numerical errors obtained by the FV-RB scheme and FV-Roe scheme on regular triangle meshes.

Accuracy analysis on irregular triangular meshes The (theoretically) second and third-order Roe and RB schemes are now applied on the series of irregular triangular meshes. The numerical errors associated with these schemes are summarized in Table 3.4 and plotted in Fig.3.4. With respect to the previous regular case, the changes for the Roe scheme remain modest : the practical error orders are again very close to their respective theoretical value, namely 2.11 instead of 2 for FV-Roe $O2$ and 3.04 instead of 3 for FV-Roe $O3$. These results indicate the linear and most of all the quadratic solution reconstruction are correctly implemented in our numerical solver. As for the FV-RB $O2$ scheme, it still provides a practical accuracy order (1.98) very close to the expected value of 2. However, the behaviour of the FV-RB $O3$ scheme is disappointing since the practical accuracy order of the scheme on irregular triangular grids drops down to 2.52. A possible explanation for this discrepancy between theoretical and practical accuracy order could be an insufficiently careful treatment of outflow boundary conditions. If the numerical flux is the cause of this behavior, it is necessarily the dissipative flux which is involved (since the non-dissipative centered flux is also used in the Roe scheme); the simplifications retained for computing this dissipative flux (with a single residual evaluation per face) could be incriminated but we did not manage to understand precisely why these simplifications would work perfectly well on the regular triangular grids and not on the finest irregular triangular grids. It must be emphasized though that the error level provided by the FV-RB $O3$ scheme remains very low and, in particular, lower than the error level associated with the third-order Roe scheme in a given grid.

Efficiency analysis We shall now compare the second-order RB scheme with its third-order extension as well as compare the RB schemes with the second and third-order Roe schemes not purely in terms of accuracy but rather in terms of the cost required (CPU, memory) to achieve a similar level of accuracy. It should be mentioned here that almost all of computations (unless pointed out specially) in this work are carried out on a PC with 2 processors of 1.86Ghz, 4MB memory,

Mesh	L_2 error of RBO2	order	L_2 error of RBO3	order
Tri_irreg1	-1.69010	-	-1.90880	-
Tri_irreg2	-2.40014	2.41	-3.02975	3.81
Tri_irreg3	-3.05994	2.46	-4.03660	3.76
Tri_irreg4	-3.67028	1.92	-4.87415	2.63
Tri_irreg5	-4.26599	1.98	-5.63012	2.52

Table 3.4: L_2 norm of numerical errors and mesh convergence order obtained with FV-RB schemes on irregular triangle meshes.

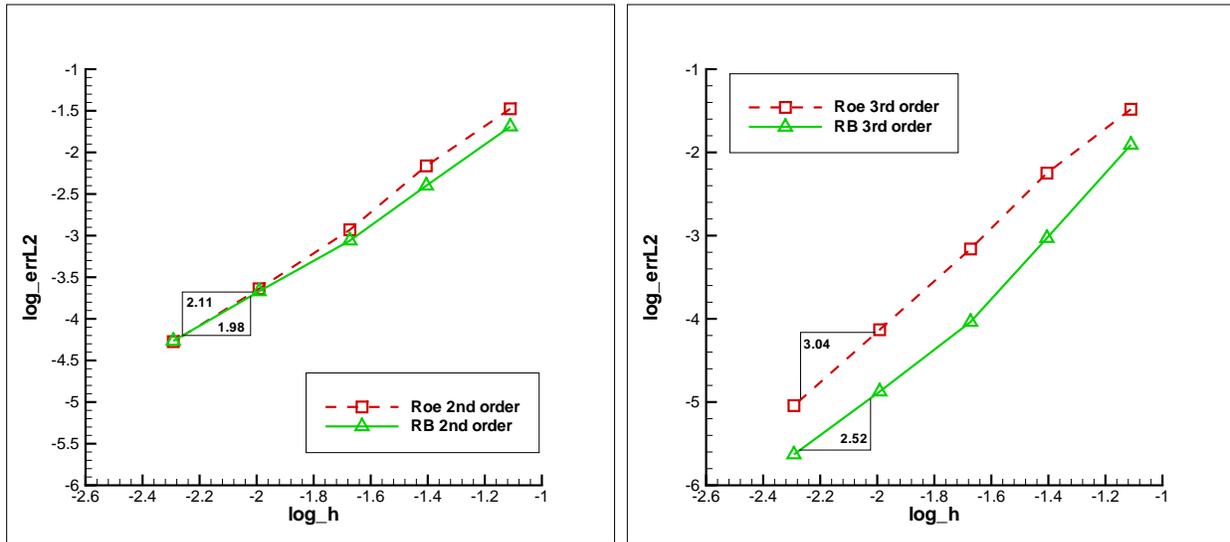


Figure 3.4: L_2 norm of numerical errors obtained by the FV-RB scheme and FV-Roe scheme on irregular triangle meshes.

windows xp system and **Digital/Compaq Visual Fortran** compiler. But each computation uses only one processor because the code is not parallelized. To make such a comparison possible, we use FV-RB and FV-Roe second and third-order computations on the finest irregular mesh **Tri_irreg5** and obtain from these computations the CPU cost and memory storage for each scheme. For each computation on this grid with 57518 **dof**, we report the total CPU time and number of iterations needed to reach machine-zero steady-state from which we derive a CIPID (CPU time Per Iteration Per **Dof**). The values reported in Table 3.5 are also normalized by the smallest values (obtained with the second-order Roe scheme) and displayed in Table 3.6. Although this comparison may be to some extent machine-dependent and no special effort was made regarding the cost and memory optimization in our solvers, these values provide a basis to draw a few conclusions regarding the estimation of the relative costs of the RB schemes developed in this work :

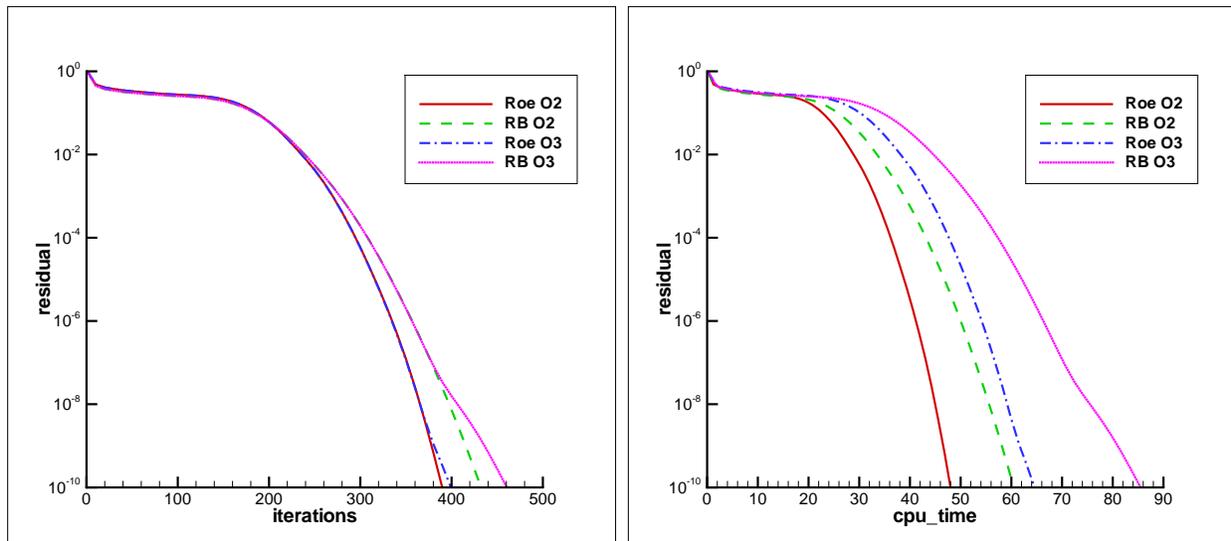
- the FV-RB schemes are systematically more expensive and time-consuming than the corresponding FV-Roe scheme. When comparing RB *O3* with Roe *O2* the extra CIPID and extra memory reaches roughly a factor of 1.5 and 2 correspondingly.
- The third-order RB scheme requires about the same amount of memory than the third-order Roe scheme with a CIPID only slightly larger (about 10%).

Keeping these numbers in mind, we first analyze the efficiency of the schemes when applied to compute the 2D advection problem. On the finest irregular grid, retained as the sample of the observed behavior, we note that the convergence to a zero-machine steady-state is achieved for all schemes, with a number of iterations ranging from about 400 for Roe *O2* and *O3*, up to 440 for RB *O2* and 470 for RB *O3* (see Figure 3.5). Taking into account the cost per iteration, this leads to a factor of about 1.7 on CPU time between the fastest method Roe *O2* and the slowest one RB *O3*. However, such a comparison is not fair since the error level achieved by RB *O3* at convergence is much lower than Roe *O2*. A more interesting and meaningful comparison is to analyze the convergence behavior of the schemes for a similar accuracy level. It can be inferred from the error curves displayed in Figure 3.4 that the minimum error level on the series of irregular triangle-based grids is achieved with the second-order schemes on the finest grid *Tri_irreg5*, while is achieved on the grid *Tri_irreg4* with Roe *O3* and on the grid *Tri_irreg3* with RB *O3*. The associated convergence curves both in iterations and in CPU time are display in Figure 3.6. The hierarchy is now widely different: for this fixed level of accuracy, the RB *O3* scheme provides the solution about 30 times faster than the Roe *O2* scheme. This gain is achieved because of the reduction in the number of **dof** (from 57518 for the grid level 5 down to 3588 for the grid level 3) and also because of a faster convergence (in iterations) on coarser grids. At the same time, the RB *O2* scheme remains more expensive than the Roe *O2* scheme because it achieves the same level of accuracy on a given grid for a cost per iteration 15% higher and a slightly slower convergence. This can be further interpreted as the fact that the dominant error term for the second-order schemes comes from the centered non-dissipative contribution to the flux balance; when going to third-order, the dominant error contribution comes from the dissipative contribution to the flux balance and the gain offered by the RB scheme demonstrates the good properties of the residual-based dissipation, although third-order is not reached between two finest grids.

Scheme	Memory (MB)	CPU time (s)	Iterations	CPIPD
RB $O2$	52	60.7	440	2.4×10^{-6}
RB $O3$	92	83.6	470	3.1×10^{-6}
Roe $O2$	45	48	390	2.1×10^{-6}
Roe $O3$	86	64.7	400	2.8×10^{-6}

Table 3.5: Computation cost of the FV schemes obtained on mesh **Tri_irreg5**, $CFL = 5$.

Scheme	Normalized Memory Requirement	Normalized CPIPD
RB $O2$	1.16	1.14
RB $O3$	2.04	1.48
Roe $O2$	1	1
Roe $O3$	1.91	1.33

Table 3.6: Normalized memory requirement and Cost Per Iteration Per **Dof** for the second and third-order RB and Roe schemes.Figure 3.5: Convergence history for the second and third-order Roe and RB schemes with $CFL = 5$ on the finest regular grid **Tri_reg5**. Left : L_2 norm of the residual vs iterations; right : L_2 norm of the residual vs CPU.

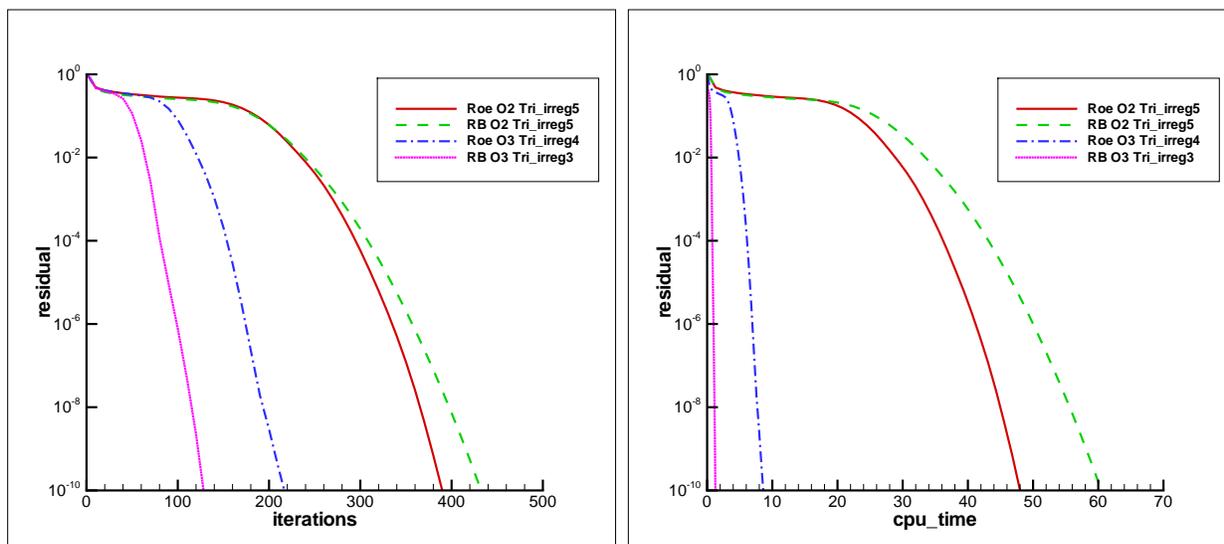


Figure 3.6: Convergence history for the second and third-order Roe and RB schemes with $CFL = 5$ on grids where a similar error level is achieved at steady state. Left : L_2 norm of the residual vs iterations; right : L_2 norm of the residual vs CPU.

3.1.2 Subsonic inviscid flow over a NACA0012 airfoil

We proceed now to the application of the second- and third-order RB schemes to the solution of the Euler equations. Our focus here will not be on a comparison between the RB scheme and a conventional upwind scheme such as the Roe scheme, but rather on a detailed analysis of the possible benefit of using a third-order scheme instead of a second-order scheme. This is precisely the main motivation of the ADIGMA project in which this thesis has taken place. The idea is to *quantify* for a given method the cost reduction offered by the third-order scheme with respect to the second-order one, taking into account most of the aerodynamic solvers in use for industrial applications are still limited to second-order accuracy. This quantification has been performed for a panel of test-cases representing the typical problems encountered in the aeronautical industry. For each test-case, industrial partners have specified grid convergence criteria for some aerodynamic quantities of interest (for instance, lift coefficient, drag coefficient and moment coefficient) and a series of increasingly refined grids has been generated. The second and third-order schemes have then been run on these various grids. The grid size required for both schemes to achieve the grid-independent results within the prescribed tolerance intervals have been analyzed and compared. A systematic faster grid convergence of the third-order scheme should be expected but the third-order scheme will be proved valuable if the gain in the number of **dof** required to achieve a prescribed accuracy level is not compromised by an excessive over-cost of the third-order scheme with respect to the second-order one. In this section our focus is on such a demonstration of interest for the RB schemes developed in this work.

Methodology The first case of demonstration retained in the ADIGMA project and analyzed now is the subsonic inviscid flow (upstream Mach number $M_\infty = 0.5$, angle of attack $\alpha = 2^\circ$) over the NACA0012 airfoil. Meshes used for computation are a series of 9 unstructured meshes composed of mainly quadrilateral elements and provided within the ADIGMA project; their main features are

summarized in Table 3.7.

Mesh	Dof	No. of faces on airfoil
mesh1	206	16
mesh2	365	32
mesh3	664	46
mesh4	1197	70
mesh5	2249	112
mesh6	4417	182
mesh7	9046	306
mesh8	19316	528
mesh9	41685	926

Table 3.7: Basic information on the unstructured meshes used for the inviscid flow over airfoil computations.

The performance provided by the RB *O2* and RB *O3* is analyzed as follows :

- computations with both schemes are run for the whole series of grids.
- for each grid the lift, drag and moment coefficients are computed (respectively **Cl**, **Cd** and **Cm** with a moment center located at the quarter-chord of the airfoil).
- the evolution of these coefficients with grid refinement is plotted on Fig.3.7
- the convergence zone corresponds to the approximation level considered as acceptable in industrial simulations. This convergence zone is defined by the value obtained on the finest grid and a tolerance interval which has been provided in the ADIGMA projet based on industrial experience. In the present case, the extent of the tolerance intervals for the lift, drag and moment coefficient were respectively $E_{Cl} = 1 \times 10^{-3}$, $E_{Cd} = 1 \times 10^{-4}$ and $E_{Cm} = 2 \times 10^{-4}$. The corresponding convergence zone for the lift coefficient is then given by $Cl_{mesh9} \pm E_{Cl}$, and similarly for the drag and moment coefficients.
- grid convergence is considered as achieved when the curve "result vs dof" enters for the first time the convergence zone (without leaving it for further refinement).
- once the grid level necessary to achieve grid convergence for each scheme has been defined, the final efficiency analysis must take into account the **dof** of the first grid ensuring convergence and the CPIPD of the scheme.

Overview of the flow Figure 3.8 provides a view of the finest quadrilateral grid (**mesh9**) and displays some results illustrative of the flow physics - obtained with the RB *O3* scheme on this finest grid. As explained above, the comparison between schemes will be performed following the very matter-of-fact approach adopted in the ADIGMA project, based on global indicators (aerodynamic coefficients) and not on contours or wall-distributions analysis. The Mach contours, wall-pressure distribution and wall-entropy deviation ($C_s = (S - S_\infty)/S_\infty$) displayed in Figure 3.8 illustrate the smoothness of this subsonic inviscid flow. The entropy variation obtained with the RB *O3* scheme on this fine grid does not exceed 1%. The computed aerodynamic coefficients on this finest grid are

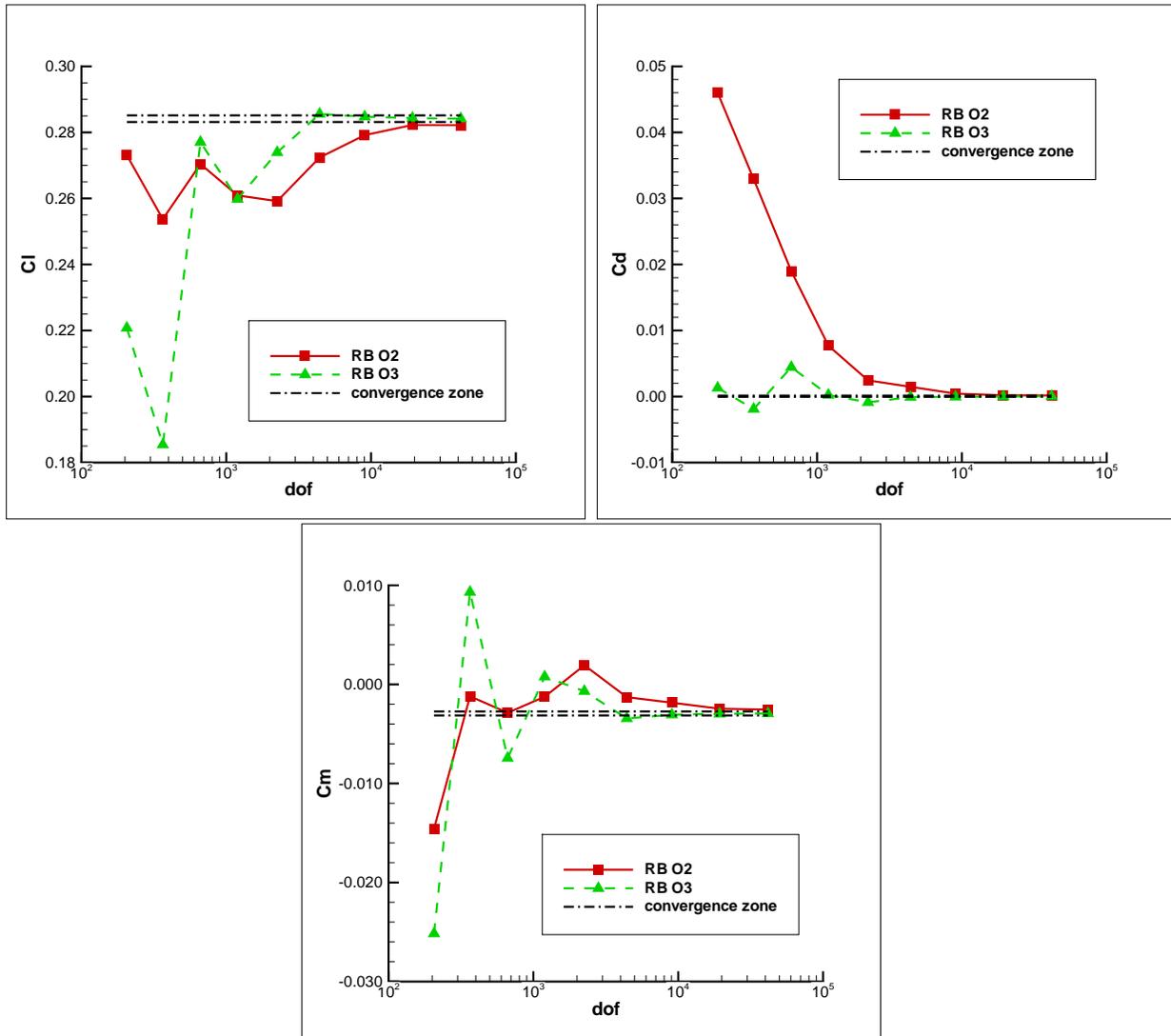


Figure 3.7: Aerodynamic coefficients convergence with **dof** based on meshes from **mesh1** to **mesh9** obtained by the second and third-order FV-RB scheme.

also displayed in Table 3.8 both for RB *O2* and RB *O3* scheme. The difference in the lift coefficient predicted by the two schemes amounts to 0.7%. For this subsonic inviscid case, the drag should be zero of course and any deviation from this value is a symptom of numerical dissipation; the computed drag with RB *O3* is 3.6 times lower than the one with RB *O2*.

Scheme	C_l	C_d	C_m
<i>O2</i>	0.282181	1.67×10^{-4}	-0.002545
<i>O3</i>	0.284150	4.62×10^{-5}	-0.002936

Table 3.8: Aerodynamic coefficients obtained with the FV-RB scheme on **mesh9**.

Results analysis from the point view of accuracy/efficiency It can be deduced from the curves displayed in Figure 3.7 that :

- the lift coefficient is converged on **mesh8** for RB *O2* and on **mesh6** for RB *O3*.
- the drag coefficient is converged on **mesh7** for RB *O2* and on **mesh6** for RB *O3*.
- the moment coefficient is converged on **mesh8** for RB *O2* and on **mesh7** for RB *O3*.

Summing things up it can be stated that the RB *O3* schemes provides a grid converged solution of all of aerodynamic coefficients by using **mesh7**, a grid with 19316 **dof** while RB *O2* needs to be applied with **mesh8**, a grid with 41685 **dof** to achieve this same grid convergence. The ratio of **dof** in favor of the third-order scheme is therefore about 2.2. To make this comparison complete, it is of course crucial to know the relative CPIPD of both schemes. Reliable values for these unit costs are obtained from the series of computations performed on grids **mesh6** to **mesh9** (see Table 3.9 and 3.10, where the CPU time corresponds to the time needed to achieve a fully converged state defined by a residual drop of 8 orders of magnitude). The average CPIPD for RB *O2* is $1.25 \times 10^{-5} s$ and $1.575 \times 10^{-5} s$ for RB *O3*; the extra-cost introduced by the third-order scheme is about 26% (in agreement with the 29% observed in the scalar advection case). On a given grid, the second- and third-order schemes need roughly the same number of iterations to achieve steady-state (typically RB *O3* needs between 3% and 10% more iterations). Of course, using a coarser grid ensures a faster convergence to steady-state : thus, RB *O3* needs 8140 iterations to reach steady-state and yield a grid-converged solution on **mesh7** while RB *O2* needs 11760 iterations to achieve steady-state and yield a grid-converged solution on **mesh8**. The reduction factor on the iterations in favor of RB *O3* amounts to 0.69 going up to 0.87 when the extra-cost (26%) per iteration for RB *O3* is taken into account. Eventually the net efficiency gain offered by the third-order RB scheme is 2.2 (ratio of the **dof**) divided by 0.87 (CPU gain on convergence to steady-state) which is 2.5. This factor corresponds also of course to the direct CPU time ratio, 10016 s for RB *O2* over 3821 s for RB *O3*, but we wanted to give some more details on how this gain is built (intrinsic convergence, cost per iteration, grid size). If the grid convergence for a specific aerodynamic coefficient is under consideration, the following is observed :

- the lift coefficient convergence is obtained with a CPU time gain of 87% for RB *O3* with respect to RB *O2* (and a corresponding memory gain of 56%).
- the drag coefficient convergence is obtained with a CPU time gain of 60% for RB *O3* with respect to RB *O2* (and a corresponding memory gain of 8%).

- the moment coefficient convergence is obtained with a CPU time gain of 59% for RB *O3* with respect to RB *O2* (and a corresponding memory gain of 16%).

Dof	Memory (MB)	CPU time (s)	Iterations	CIPD
4417	7	277	5100	1.23×10^{-5}
9046	12	888	7900	1.24×10^{-5}
19316	25	2862	11760	1.26×10^{-5}
41685	52	10016	18860	1.27×10^{-5}

Table 3.9: Computation cost of the FV-RB *O2* scheme obtained from **mesh6** to **mesh9**.

Dof	Memory (MB)	CPU time (s)	Iterations	CIPD
4417	11	359	5220	1.56×10^{-5}
9046	21	1161	8140	1.57×10^{-5}
19316	43	3821	12300	1.61×10^{-5}
41685	91	13421	20640	1.56×10^{-5}

Table 3.10: Computation cost of the FV-RB *O3* scheme obtained from **mesh6** to **mesh9**.

3.1.3 A 3D helicoidal advection problem

Following a methodology similar to the one adopted in *2D*, another model scalar advection problem is used to check the actual order of accuracy brought by the FV-RB scheme in *3D*. This helicoidal advection of a Gaussian profile problem is defined as follows :

$$\frac{\partial W}{\partial t} + z \frac{\partial W}{\partial x} + 0.1 \frac{\partial W}{\partial y} - x \frac{\partial W}{\partial z} = 0, \quad (3.2)$$

on the flow domain $x \in [-1, 0]$, $y \in [0, 1]$, $z \in [0, 1]$. The initial condition is simply $W(x, y, z, 0) = 0$ in the whole domain with the inlet boundary condition:

$$W(x, y, 0, t) = e^{-50((x+\frac{1}{2})^2+(y-\frac{1}{2})^2)}.$$

on lower boundary $z = 0$.

For this 3D problem, two types of meshes will be used : so-called "structured" meshes because they appear as Cartesian grids since based on regular hexahedral elements, and unstructured meshes based on tetrahedral elements. The main features of these meshes are listed in Table 3.11 and 3.12.

A view of the steady solution obtained with the RB *O3* scheme on the fine mesh **Tetra_5** is plotted in Figure 3.9. It is provided to give an overview of the solution of the model flow problem as well as to illustrate the good accuracy of the numerical solution (on what is a fine grid) : the computed solution distribution on the outlet boundary is hardly distinguishable from the imposed distribution on the inlet boundary. The analysis of the accuracy will now be performed using a grid convergence study on the series of structured and unstructured grids, completed by considerations on the respective cost of each scheme (namely the second- and third-order Roe schemes on one hand and the second- and third-order RB schemes on the other hand).

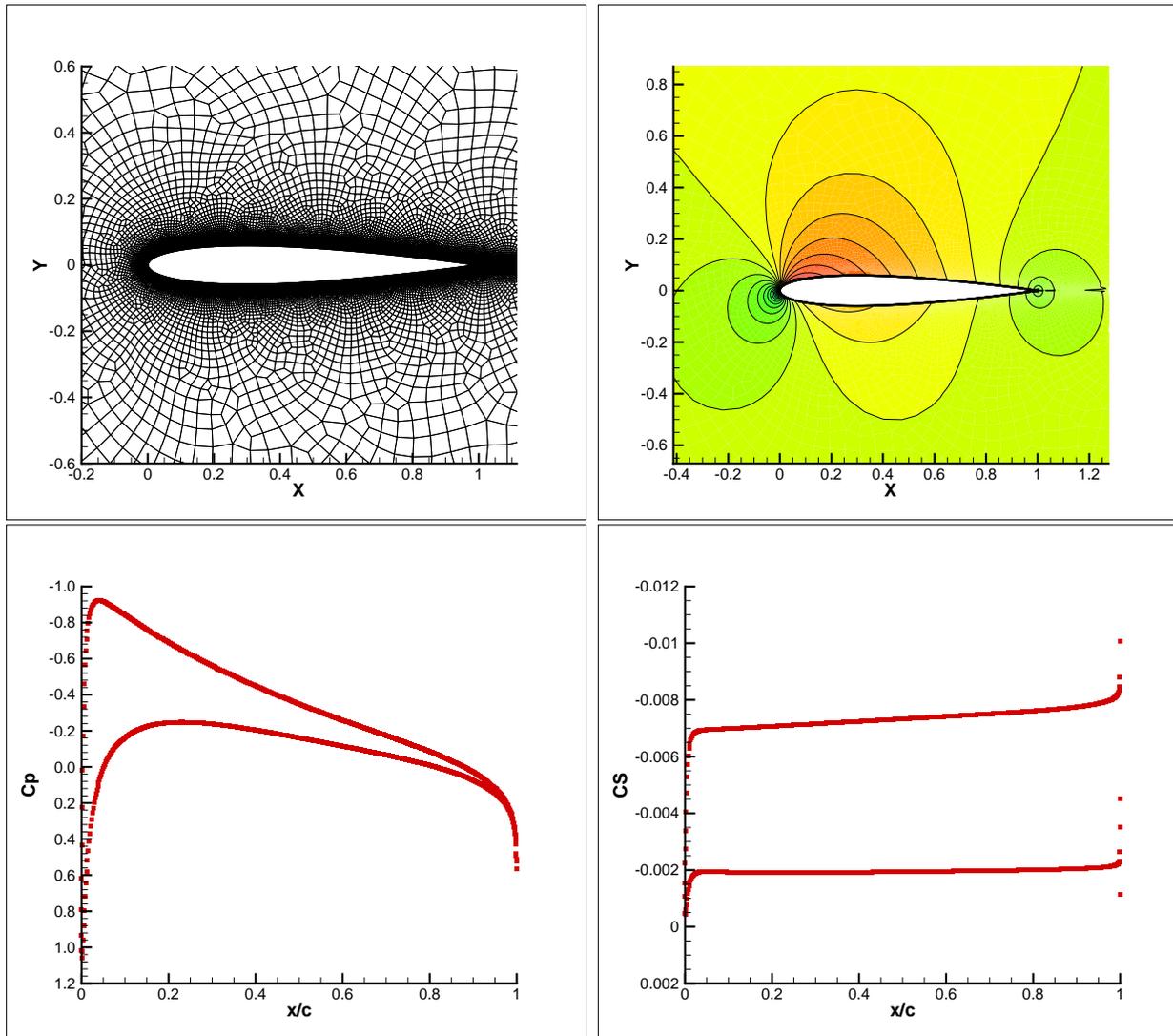


Figure 3.8: **Mesh9** (top left) and the corresponding results obtained with the *O3* RB scheme. Top right is the Iso-Mach contour with 30 levels from 0 to 0.7, bottom left is the pressure coefficient distribution on the airfoil, bottom right is the entropy coefficient distribution on the airfoil.

Mesh	cells	boundary faces
Hex_1	1000	600
Hex_2	8000	2400
Hex_3	27000	5400
Hex_4	64000	9600
Hex_5	125000	15000

Table 3.11: structured hexahedral meshes for the 3D helicoidal advection problem.

Mesh	cells	boundary faces
Tetra_1	7506	1364
Tetra_2	56076	5402
Tetra_3	191110	12140
Tetra_4	433439	21602
Tetra_5	870128	33676

Table 3.12: Unstructured tetrahedral meshes for the 3D helicoidal advection problem.

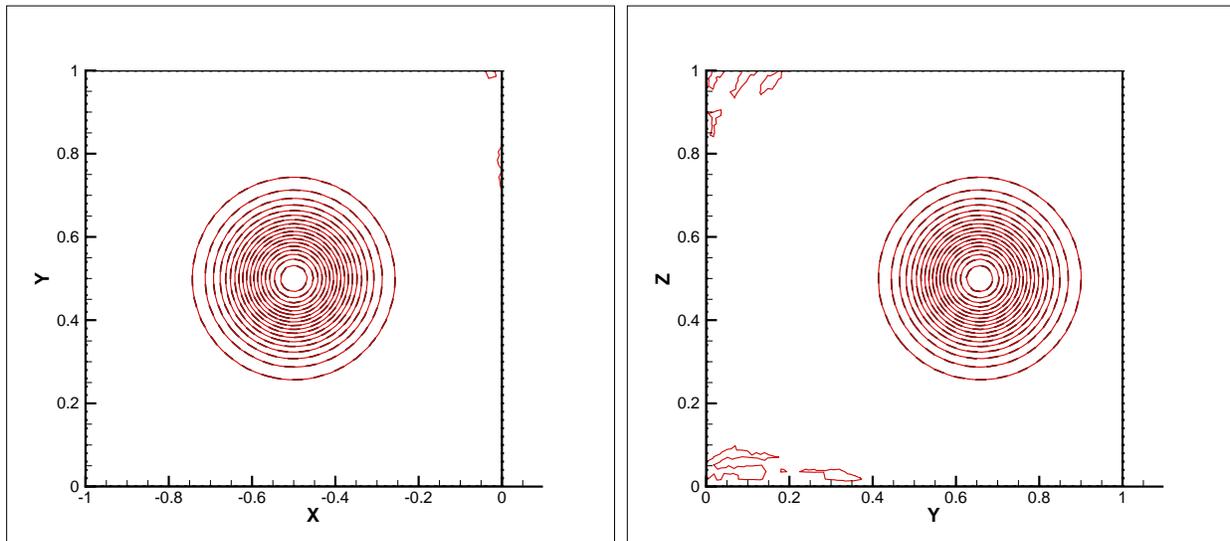


Figure 3.9: W iso-contours (20 levels from 0 to 1) obtained by the FV-RB $O3$ scheme with mesh **Tetra_5** on the face " $z = 0$ " (left) and " $x = 0$ " (right). The dashed black line is exact solution, solid red line is the numerical solution.

Accuracy analysis on regular hexahedral meshes Figure 3.10 displays the error convergence curve for the second- and third-order Roe and RB schemes when the series of regular hexahedral meshes is used. Although some error cancelations seem sometimes to take place, which is artificially increasing the practical order of accuracy, each scheme is found to yield the expected behavior; in particular, the second-order RB scheme yields a slope exactly equal to 2 for the $\log(\text{error})$ vs $\log(\text{mesh size})$ curve and the third-order RB scheme displays a curve with a slope equal to 3.3 (see also Table 3.13). Even more interestingly, the error level produced by the third-order RB scheme is particularly low : when the second-order Roe and RB schemes as well as the third-order Roe scheme produce about the same error level (about 10^{-3}) with the finest grid **Hex_5**, the third-order RB scheme already yields this same error level on grid **Hex_3**, with 27000 **dof** compared to the 125000 **dof** on the finest grid.

Mesh	L_2 error of RBO2	order	L_2 error of RBO3	order
Hex_1	-1.82430	-	-1.68127	-
Hex_2	-2.39261	1.89	-2.59874	3.05
Hex_3	-2.73202	1.93	-3.21879	3.52
Hex_4	-2.98182	2.00	-3.64636	3.42
Hex_5	-3.17614	2.00	-3.96648	3.30

Table 3.13: L_2 norm of numerical errors obtained by the FV-RB scheme on structured hexhedral meshes.

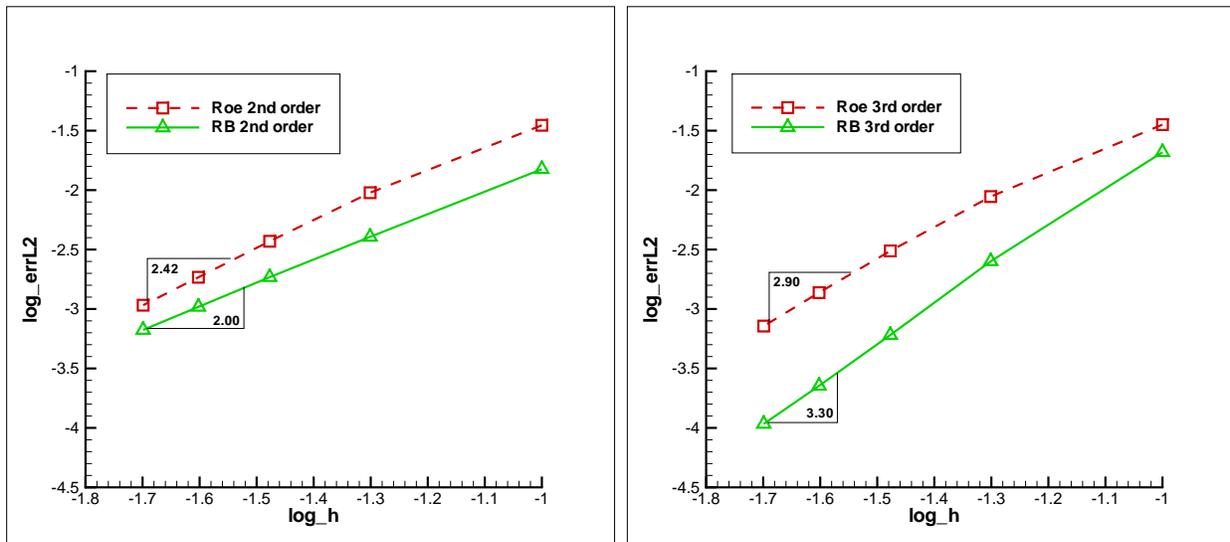


Figure 3.10: L_2 norm of numerical errors obtained by the FV-RB scheme and FV-Roe scheme on structured meshes from **Hex_1** to **Hex_5**.

Accuracy analysis on tetrahedral meshes When the previous mesh convergence analysis is carried out on the series of irregular tetrahedral meshes, the results of the third-order RB scheme are unfortunately disappointing. As can be observed from Table 3.14 and Figure 3.11, the RB O_3 scheme gives only second-order on these irregular tetrahedral meshes, while it does give third-order

convergence on the previous regular hexahedral meshes. Moreover, the error level provided by RB *O3* on the finest grid is about the same as the one given by RB *O2*, itself almost identical to the error level provided by Roe *O2*. In the meantime, the Roe *O3* scheme does yield a third-order practical accuracy, this means the quadratic solution reconstruction is correctly implemented in *3D*. This disappointing behavior of the RB *O3* scheme on irregular meshes most likely results from the evaluation of the RB dissipation, which is quite geometry dependent. In spite of our careful investigations, we did not manage to understand the reason for this loss of accuracy on irregular meshes until now.

Mesh	L_2 error of RBO2	order	L_2 error of RBO3	order
Tetra_1	-2.01724	-	-1.96477	-
Tetra_2	-2.67112	2.30	-2.62094	2.30
Tetra_3	-2.99440	1.84	-3.02481	2.30
Tetra_4	-3.22007	2.05	-3.26260	2.16
Tetra_5	-3.41852	2.02	-3.45300	1.94

Table 3.14: L_2 norm of numerical errors obtained by the FV-RB scheme on unstructured tetrahedral meshes.

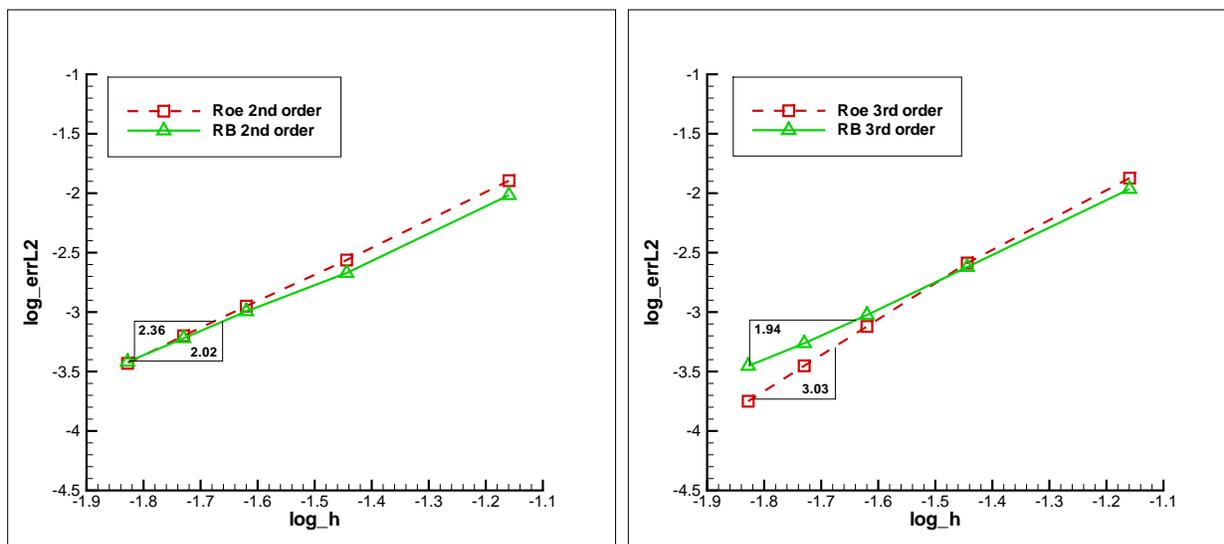


Figure 3.11: L_2 norm of numerical errors obtained by the FV-RB scheme and FV-Roe scheme on unstructured meshes from **Tetra_1** to **Tetra_5**.

Efficiency analysis Since the computations on the tetrahedral unstructured meshes seem to be flawed, we shall restrict our analysis of efficiency onto the much more encouraging (and consistent with the theory) results obtained on the hexahedral meshes. The analysis method adopted here is similar to the one used for 2D circular-advection case. First of all, residual convergence histories of second and third-order RB and Roe scheme obtained on the finest mesh **Hex_5** are showed in Figure 3.12. It is found that the third-order scheme takes few more iterations to reach the convergence than the second-order scheme for the Roe numerical flux, which is also the case for 2D problems; while

the second-order RB scheme takes many more iterations than its third-order scheme for this case, which is not the situation in 2D. This must be related to the implicit phase. The residual evolution with CPU time obtained with 4 schemes is quite reasonable, the third-order scheme is more complex than the second-order scheme and therefore takes more CPU time to reach the convergence. As the CIPD(Cost Per Iteration Per **Dof**) of the RB scheme is globally higher than the Roe scheme, the CPU time of computation with RB schemes is longer than the Roe schemes. Although RB *O2* scheme takes about 70 more iterations than the RB *O3* scheme to reach the convergence, the CPU time necessary for RB *O3* to reach the convergence is still longer than the one needed for the RB *O2* scheme because the CIPD of RB *O3* is higher than RB *O2*.

Table 3.15 shows detailed information on the computation cost of FV based 4 schemes. As RB schemes need more iterations than the Roe schemes to reach the residual convergence and the CIPD of RB schemes is higher than the Roe scheme, the total CPU time needed by the RB schemes for the residual convergence is therefore longer than the one for Roe schemes. And the RB scheme needs more memory for a computation than the Roe schemes. On the same mesh the third-order scheme is always more expensive on memory and CPU time than the second-order scheme. If values in Table 3.15 are normalized by the smallest values (results obtained with Roe *O2* scheme), Table 3.16 is obtained, which makes the computation cost of each scheme more clear. For Roe schemes, *O3* scheme consumes 3.68 times more memory and 1.63 times more CIPD than the *O2* scheme. And RB *O3* scheme is 2.68 and 2.40 times more expensive on memory and CIPD respectively than the RB *O2* scheme. Comparison between RB scheme and Roe scheme shows that RB *O2* scheme spends 1.6 times more memory and 1.53 times more CIPD than the Roe *O2* scheme, and the memory and CIPD of RB *O3* scheme are correspondingly 1.16 and 2.26 times more than those of Roe *O3* scheme. On the same mesh, the most expensive scheme on memory and CIPD is RB *O3* scheme.

Scheme	Memory (MB)	CPU time (s)	Iterations	CIPD
RB <i>O2</i>	203	145	200	5.8×10^{-6}
RB <i>O3</i>	543	248	140	1.4×10^{-5}
Roe <i>O2</i>	127	42.9	90	3.8×10^{-6}
Roe <i>O3</i>	467	78.0	100	6.2×10^{-6}

Table 3.15: Computation cost of the FV schemes obtained on mesh **Hex_5**, $CFL = 10^6$.

Scheme	Normalized Memory Requirement	Normalized CIPD
RB <i>O2</i>	1.6	1.53
RB <i>O3</i>	4.28	3.68
Roe <i>O2</i>	1	1
Roe <i>O3</i>	3.68	1.63

Table 3.16: Normalized memory requirement and Cost Per Iteration Per **dof** for the second and third-order RB and Roe schemes.

As mentioned before in 2D advection case, it is not fair to compare the efficiency of second and third-order schemes on the same mesh. This comparison should be made for the same accuracy level. From Figure 3.10 it can be seen that Roe *O2*, *O3* and RB *O2* scheme achieve about the same

error level on mesh **Hex_5**, while this error level is already obtained by the RB *O3* scheme on mesh **Hex_3**. So it is interesting to compare the efficiency of these schemes based on this error level. In Figure 3.13 it is found that to reach the same level of accuracy, Roe *O2* scheme takes the least iterations and RB *O2* scheme takes the most iterations, but it is RB *O3* scheme who takes the least CPU time to reach the convergence, which shows the real advantages of using the high-order scheme.

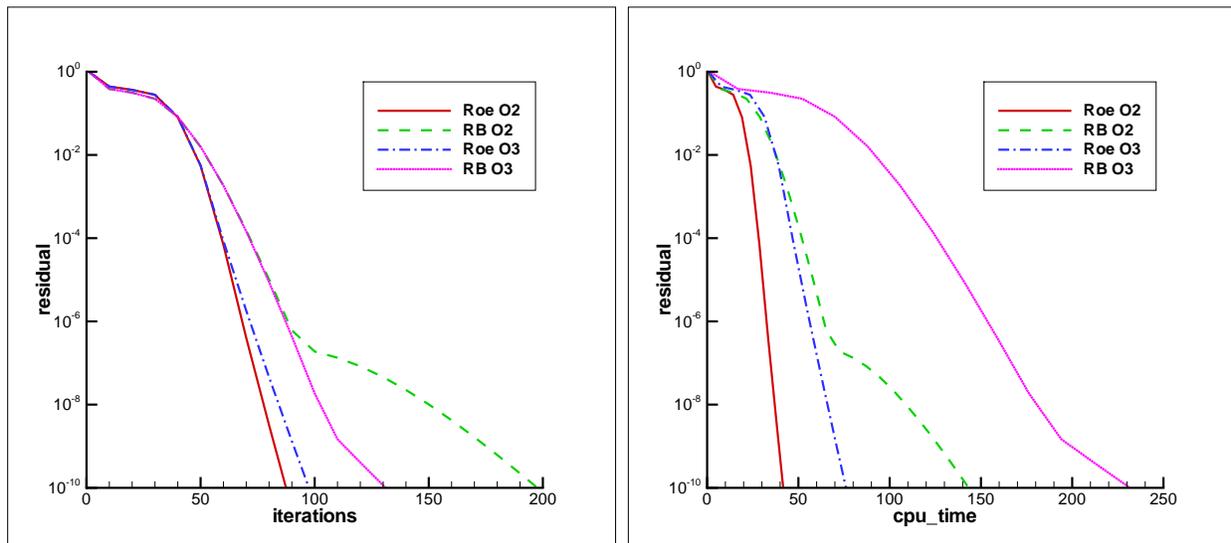


Figure 3.12: 3D helicoidal advection. Convergence history for the second and third-order Roe and RB schemes with $CFL = 10^6$ on grid **Hexa_5**. Left : L_2 norm of the residual vs iterations; right : L_2 norm of the residual vs CPU.

3.1.4 Subsonic inviscid flow over a 3D airfoil body

The 3D schemes are extended to the system of the Euler equations and applied to the computation of the subsonic flow over a 3D geometry. A conventional or typical geometry would have been a wing but it was decided in the ADIGMA project to compute the flow at $M_\infty = 0.5$, angle of attack $\alpha = 1^\circ$, over a streamlined body. The body surface is made from surface of revolution produced by a 10% thick airfoil which are joined by a central part corresponding to a simple extrusion of this thick airfoil (see Figure 3.14).

A grid convergence study similar to the one carried out in 2D is also performed but on a limited number of grid levels, in order to reduce the computational cost of the study. The meshes used for the computations are unstructured meshes based on pure tetrahedral elements, which have been provided within the ADIGMA project; the **dof** associated with each grid are summarized in Table 3.17.

Overview of the flow Figure 3.15 displays Mach contours in selected cut-planes, as computed by the FV-RB *O3* scheme on **Mesh3** (finest grid available). As the flow is in the x positive direction

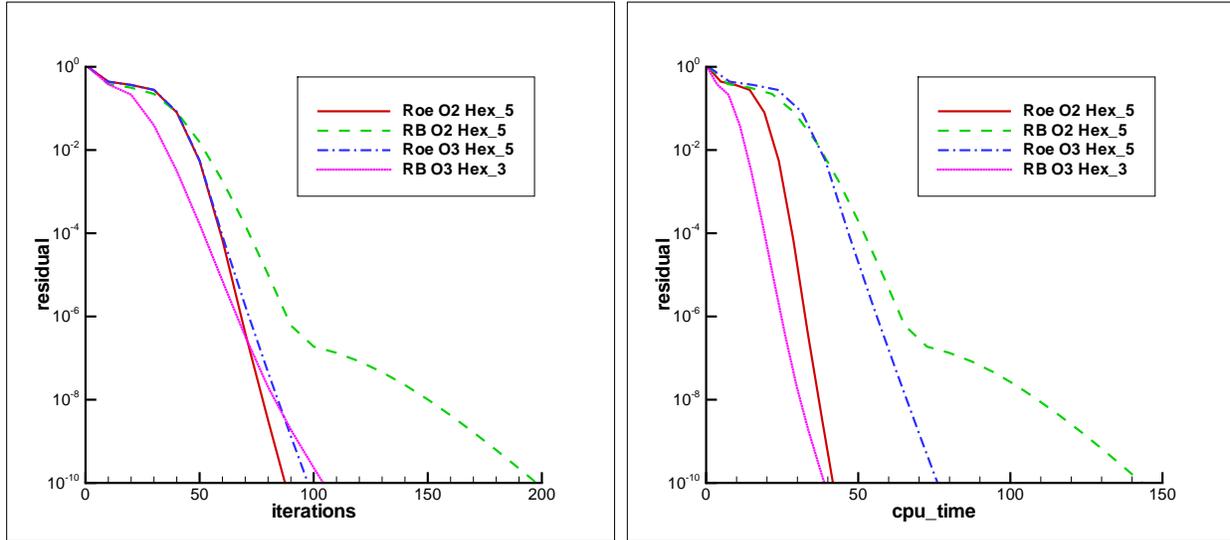


Figure 3.13: 3D helicoidal advection. Convergence history for the second and third-order Roe and RB schemes with $CFL = 10^6$ on grids in which about the same level of accuracy is achieved at steady-state. Left : L_2 norm of the residual vs iterations; right : L_2 norm of the residual vs CPU.

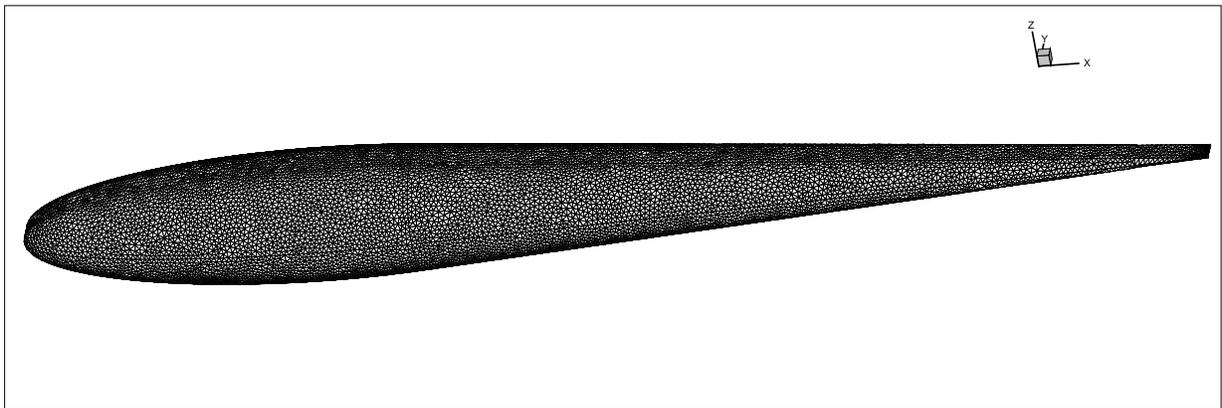


Figure 3.14: Surface mesh for the airfoil body corresponding to the ADIGMA Baseline Test Case n°0 (BTC0). The level of refinement of the mesh shown in the picture is **Mesh3** (finest grid).

Mesh	Dof
Mesh1	191753
Mesh2	254960
Mesh3	440494

Table 3.17: Unstructured tetrahedral meshes for the 3D airfoil body.

and its attack angle 1° in z positive direction, the asymmetric Mach contours on the plane " $y=0$ " (top left figures) are found. The zoom of leading edge (top right figure) shows that contours obtained by the RB $O3$ scheme on this fine mesh are smooth. Because the airfoil body is symmetric to the plane " $z=0$ " and the flow has no angle with the direction y , symmetric contours around the airfoil body are found on the plane " $z=0$ " (bottom figures), and it is interesting to see two symmetric vortex structures around the trailing edge (bottom right figure).

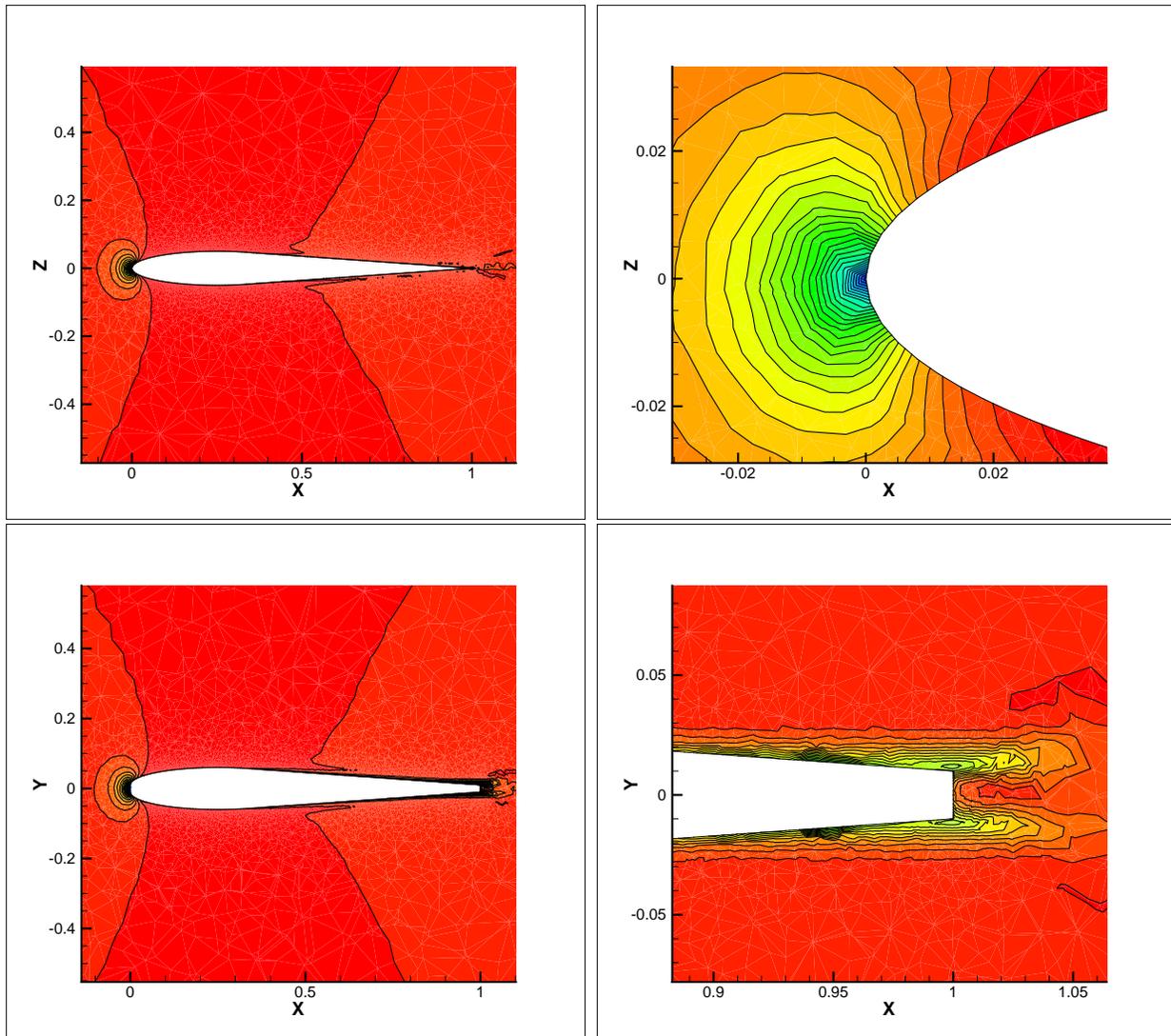


Figure 3.15: Mach contours (30 levels from 0.01 to 0.5) obtained with the RB $O3$ scheme on **Mesh3**. Top : global view of flow around the airfoil body on the cut plane " $y=0$ " (left) and a zoom of the leading edge (right). Bottom : global view of flow around the airfoil body on the cut plane " $z=0$ " (left) and a zoom of the trailing edge (right).

Accuracy and efficiency analysis Figure 3.16 shows aerodynamic coefficients evolution with the increase of **dof** used by the computation. **Cm** is the aerodynamic force moment coefficient with the moment center located at the point (0.25, 0, 0). The convergence zone is obtained with the permitted

error given by the ADIGMA project: $E_{Cl} = 1 \times 10^{-3}$, $E_{Cd} = 3 \times 10^{-4}$ and $E_{Cm} = 5 \times 10^{-4}$. It is found that globally the result difference between the FV-RB second and third-order scheme is very small, as showed in results of the 3D helicoidal advection case. For the lift coefficient, FV-RB O2 and O3 scheme stays in the convergence zone from the coarse mesh **Mesh1**, while the steady converged value seems achieved on **Mesh2** because the difference between the value obtained on **Mesh2** and on **Mesh3** is small. The drag coefficient convergence is achieved by the RB O2 and O3 scheme until **Mesh3**. Like the lift coefficient convergence, the moment coefficients obtained by both RB O2 and O3 scheme always stay in the convergence zone, and the converged value is almost achieved on the **Mesh2**.

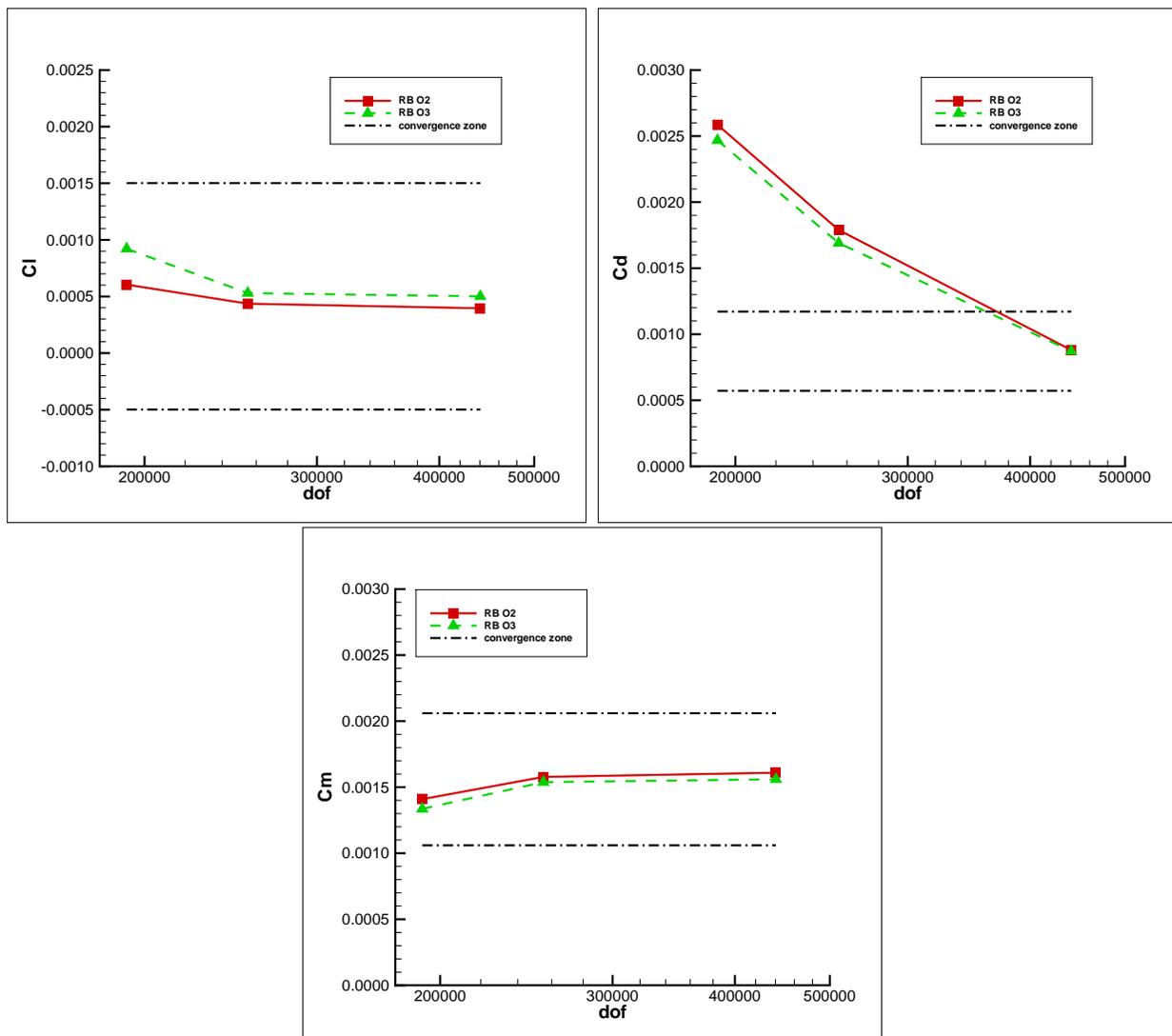


Figure 3.16: Aerodynamic coefficients convergence with **dof** based on meshes from **Mesh1** to **Mesh3** obtained by the second and third-order FV-RB scheme.

In Table 3.18, **Iterations** showed are those necessary to achieve the full converged steady state, which is defined as a density residual drop of 5 orders for this case. By comparing with Table 3.19, it

Dof	Memory (MB)	CPU time (s)	Iterations	CPIPD
191753	322	33790	5950	3.0×10^{-5}
254960	429	56200	6790	3.2×10^{-5}
440494	747	130800	8800	3.4×10^{-5}

Table 3.18: Computation cost of the FV-RB *O2* scheme obtained from **Mesh1** to **Mesh3**.

Dof	Memory (MB)	CPU time (s)	Iterations	CPIPD
191753	865	61050	7600	4.2×10^{-5}
254960	1100	99100	8830	4.4×10^{-5}
440494	2000	260000	11680	5.1×10^{-5}

Table 3.19: Computation cost of the FV-RB *O3* scheme obtained from **Mesh1** to **Mesh3**.

Scheme	C_l	C_d	C_m
<i>O2</i>	3.9564×10^{-4}	8.8020×10^{-4}	1.6096×10^{-3}
<i>O3</i>	5.0117×10^{-4}	8.7147×10^{-4}	1.5598×10^{-3}

Table 3.20: Aerodynamic coefficients obtained with the FV-RB scheme on **Mesh3**.

is found that the FV-RB *O3* scheme has a averagely 41% higer CPIPD and 2.7 times more memory requirement than the FV-RB *O2* scheme for a computation. Unfortunately the precision gain of the third-order scheme is very small. In Table 3.20 exact aerodynamic coefficient values obtained on the finest mesh are showed. For this subsonic case, ideally the drag is zero, so it is easy to see the precision gain on the drag coefficient. The RB *O3* scheme gives only a 1% smaller drag than the *O2* scheme. For other coefficients, the quality of RB scheme results can be seen from the comparison with a reference result of a project partner NLR(National Aerospace Laboratoy) showed in ADIGMA project report [1]. It gives $C_l = 2.1163 \times 10^{-4}$, $C_d = 5.3076 \times 10^{-4}$, $C_m = 1.7085 \times 10^{-3}$, which are obtained on a structured grid with 1572864 **dof** by a second-order discontinuous Galerkin method. It is found that on the lift coefficient, RB *O2* scheme has 87% difference from the reference result, and RB *O3* scheme has 137% difference; for the moment coefficient, the difference between the reference value and the RB *O2* result is 6%, and RB *O3* result has 9% difference.

3.2 Steady inviscid flows with shocks

3.2.1 Transonic inviscid flow over a NACA0012 airfoil

After having analyzed the performance of the FV-RB scheme for smooth flows in 2D and 3D, flow problems with discontinuities are taken into consideration in this section. First of all, a well-documented test case is considered, namely the steady transonic inviscid flow around the NACA0012 airfoil with an inlet Mach number equal to 0.8 and an angle of attack α equal to 1.25° . The main characteristics of this flow are a strong shock on the suction side and a weak shock on the pressure side of the airfoil. A rather fine unstructured mesh with 26384 pure triangle elements is used in our computation (see Figure 3.17). In fact, in the course of the ADIGMA project, a grid convergence study similar to the previous subsonic case has been performed and similar conclusions on the faster grid convergence of the FV-RB *O3* scheme have been drawn. Rather than repeating this study here,

we wish to illustrate in this section the shock-capturing properties of the second- and third-order RB schemes on a sufficiently fine grid.

Scheme	K	S	β	C_L	C_D	C_M
RB O2	2	-	-	0.35825417	0.02300606	-0.04090117
RB O3	2	160	0.03	0.36009461	0.02293367	-0.04137368

Table 3.21: Limiter parameters used by the FV-RB scheme and the corresponding aerodynamic coefficients obtained on a triangular mesh with 26384 **dof**.

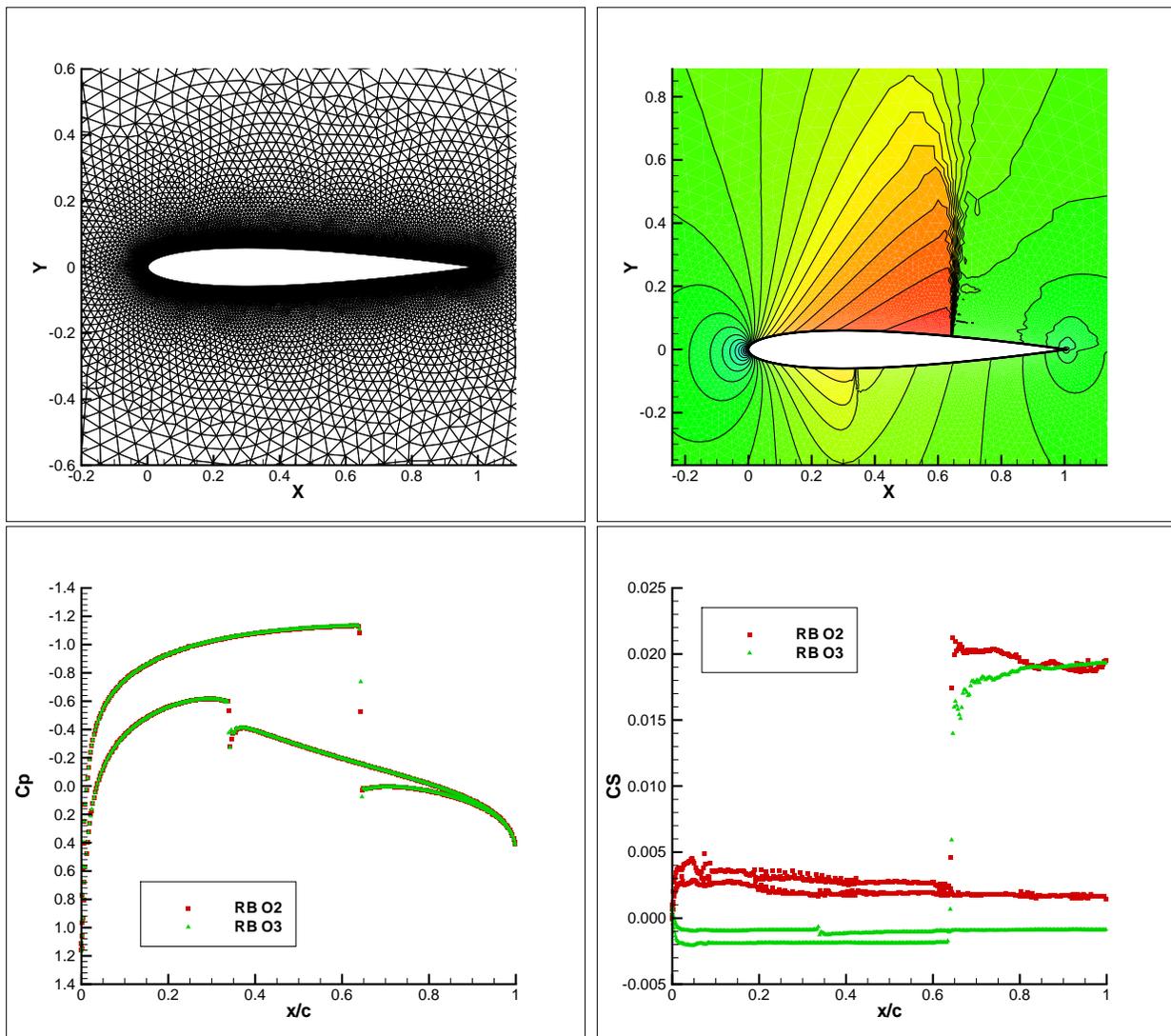


Figure 3.17: Computation mesh (top left) and the corresponding results obtained with FV-RB scheme. Top right is the Iso-Mach contour (30 levels from 0 to 1.4) obtained by the O3 RB scheme, bottom figures the pressure coefficient distribution (left) and the entropy coefficient distribution (right) on the airfoil obtained by both O2 and O3 RB scheme.

The limiting strategy described in 2.1.4 is used to remove oscillations. The sole parameter for the second-order scheme is the coefficient K used in the Venkatakrishnan limiter; for the third-order scheme, the parameter S and β coefficient appearing in the switch (2.53) from quadratic reconstruction to limited linear reconstruction must also be tuned. The values eventually retained are typically used for transonic flows and yield oscillation-free results. They are summarized in Table 3.21, where the computed aerodynamic coefficient are also displayed. A reference result is provided by a project partner DLR (German Aerospace Center) in the ADIGMA project report [2], which is obtained on a structured C-mesh with 115584 **dof** by a second-order scheme, and yields the following reference values for the lift, drag and moment coefficients : $C_L = 0.357895$, $C_D = 0.022736$ and $C_M = -0.038646$. It is found that the result of FV-RB O2 scheme has a 0.1% difference on C_L , 1.2% difference on C_D and 5.8% difference on C_M , while the FV-RB O3 gives a result with 0.6% difference on C_L , 0.9% difference on C_D and 7% difference on C_M . More importantly, it must be underlined the second- and third-order RB schemes yield very close results (which do not differ by more than 1%) which tends to validate the strategy followed for the third-order scheme in presence of flow discontinuities. Some representative results are displayed in Figure 3.17 : Mach contours computed using RB O3, pressure coefficient and entropy deviation distributions at the wall computed with RB O2 and RB O3. It is clear from these pictures that both the strong and weak shocks are well captured by the FV-RB O2 and O3 schemes. The difference between these two schemes on the \mathbf{C}_p is very small; meanwhile, on the \mathbf{CS} figure, the better accuracy of the third-order solution can be observed with, in particular, a lower entropy deviation level upstream of the shocks.

3.2.2 Transonic inviscid flow over the ONERA M6 wing

The performance of the FV-RB scheme is also assessed for a 3D transonic case, namely the flow over the ONERA M6 wing with an inlet Mach number $M_\infty = 0.84$ and an angle of attack $\alpha = 3.06^\circ$. At first, computations with both the second and third-order RB schemes have been performed on a very coarse "structured" mesh containing 48000 hexahedrons. The limiting parameters used for this case are listed in the Table 3.22. The mesh and the Mach number contours computed with the FV-RB O3 scheme are displayed in Figure 3.18.

Scheme	K	S	β
RB O2	6	-	-
RB O3	6	160	0.1

Table 3.22: Limiter parameters used by the FV-RB scheme for the transonic inviscid flow over the ONERA M6 wing.

Although the mesh is coarse, the λ -shock structure on the upper wing surface is already correctly captured by both FV-RB O2 and O3 scheme. A more detailed comparison between the RB O2 and O3 schemes can be found in the wall pressure coefficient distributions on the wing along some selected spanwise sections where experimental data (for an evidently turbulent flow) are available (see Figure 3.19). Globally the differences between the FV-RB O2 and O3 schemes on the computed C_p distributions are small. For the position around 20% of the span-wise length, counted from the wing root, the two branches of the λ -shaped shock are captured in a very similar way by both FV-RB O2 and O3 schemes. At 65% of the span-wise length away from the wing root, the result provided by the O3 scheme appears more dissipative, probably because the choice of the parameter β is not

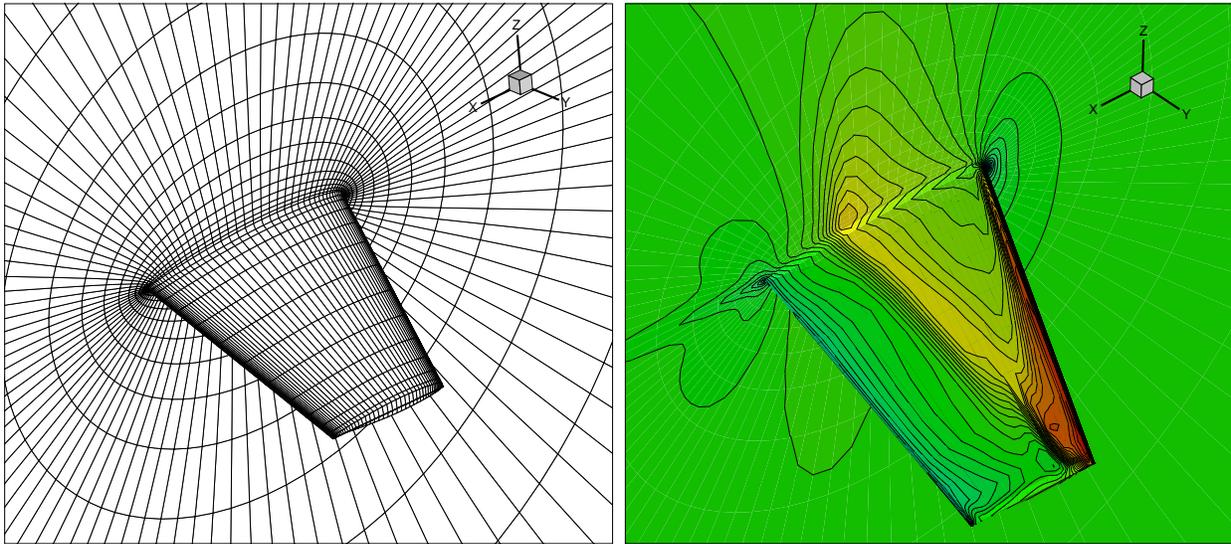


Figure 3.18: **Coarse structured mesh and the corresponding result.** Structured mesh(left) with 48000 cells, and the Mach iso-contour (right) with 40 levels from 0.2 to 1.5 obtained with $O3$ RB scheme.

optimal (the tuning of such a parameter becomes difficult to perform for costly 3D computations). When the position moves to 95% spanwise length away from the wing root, only one shock branch is left and it seems that the $O3$ scheme gives a slightly sharper shock. It must be underlined that both sets of results are quite close to the experimental data, which are obtained with the same Mach number and attack angle but for a turbulent flow with Reynolds number 1.172×10^7 [43].

Next, the FV-RB $O2$ and $O3$ schemes are applied to the computation of the same case but using an unstructured mesh with 818411 tetrahedrons provided by the ADIGMA project. This unstructured mesh has about 17 times more elements than the previous structured mesh and will of course yield more accurate results. This computation gives us in particular the opportunity to demonstrate the RB solver is fully operational on whatever type of 3D unstructured grids. Moreover, it is also a chance to assess the behavior of the second-order and third-order RB schemes for the typical values of the parameters K , β and S that we have decided to use for transonic flows in 3D. The Mach contours computed by the RB $O3$ scheme are displayed in Figure 3.20 and clearly illustrate the λ -shaped shock structure is much better captured on this refined grid. Figure 3.21 allows to appreciate in some detail the differences between the FV-RB $O2$ and $O3$ schemes. As could be expected from the already close agreement on the previous coarse grid, the second and third-order results remain very close to each other (and in good agreement with experimental values). Note also the choice of limiting coefficients previously adopted seem to work well again in this case, which is encouraging since we do not wish to fine-tune these coefficients for each new computation. The overshoots in the RB $O3$ C_p -prediction near the wing leading-edge are not oscillations (the flow is smooth in this region).

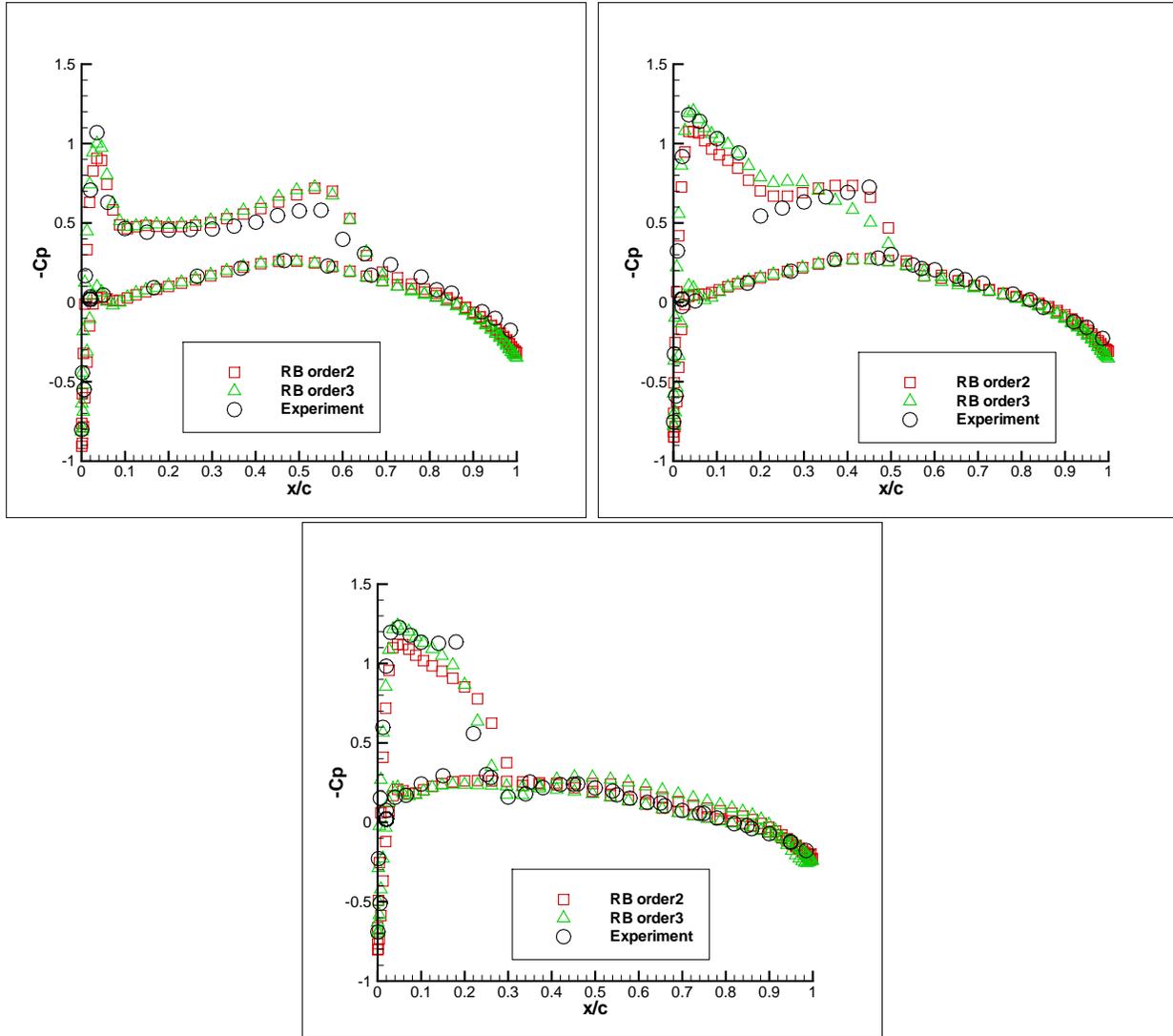


Figure 3.19: **Coarse structured mesh results.** C_p distribution on the wing body for spanwise wing sections located respectively 20% (top left), 65% (top right), and 95% (bottom) span-wise length away from the wing root, obtained with the second and third-order FV-RB schemes.

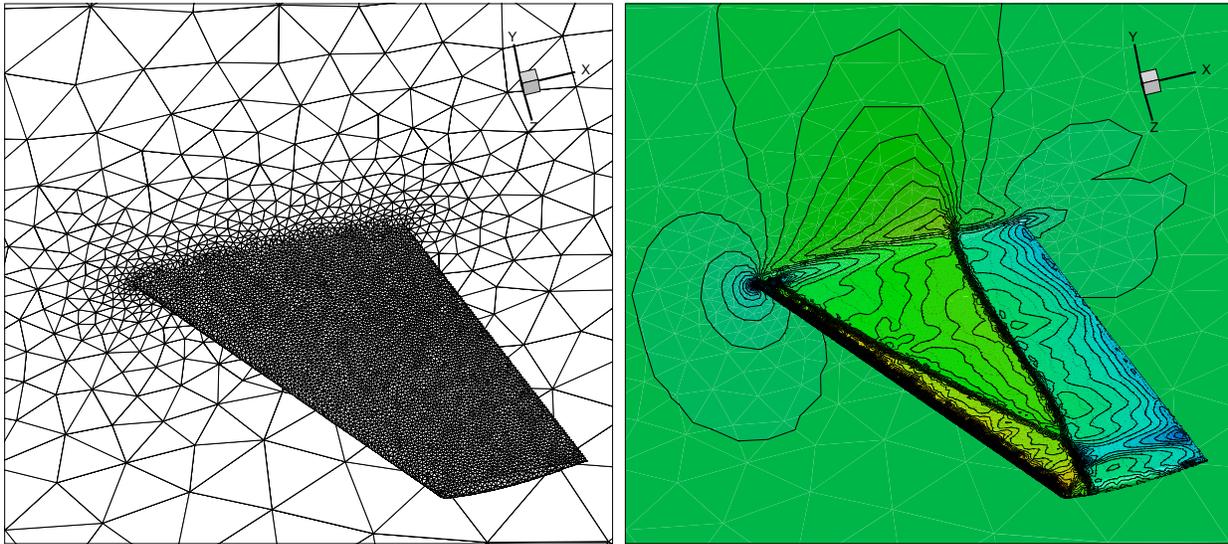


Figure 3.20: **Fine Unstructured mesh and the corresponding result.** Unstructured mesh (left) with 818411 cells. Mach contours (right) with 40 levels from 0.2 to 1.8 obtained with the *O3* RB scheme.

3.3 Steady viscous flows

3.3.1 2D advection-diffusion problem

Before solving the Navier-Stokes equations, the viscous formulation of the second and third-order FV-RB schemes are tested on a 2D advection-diffusion model problem. The governing equation of this problem reads:

$$\frac{\partial W}{\partial t} + a \frac{\partial W}{\partial x} = \nu \frac{\partial^2 W}{\partial y^2},$$

and the flow domain is $x \in [0, L]$, $y \in [0, L]$ ($L = 1$). The problem can be seen as a Poiseuille-type flow :

- the solution in the flow domain is initialized with $W = 1$;
- the inlet condition $W = \sin(\pi \frac{y}{L})$ is imposed on the boundary $x = 0$;
- "wall-like" conditions are imposed on the lower ($y = 0$) and upper ($y = L$) boundary : $W = 0$;
- the boundary $x = L$ is an outflow where the solution is extrapolated from the interior domain.

The exact solution of the problem at steady-state is given by :

$$W_{exact}(x, y) = \sin(\pi y) e^{-\pi^2 \frac{\nu}{a} x}.$$

Physically, the inlet signal is convected and diffused along the way (see also the plot of this exact solution in Figure 3.22). The computations performed in this section use the following values of the problem parameters : $a = 1$, $\nu = 0.005$ and $L = 1$, so that the Reynolds number is equal to $Re = aL/\nu = 200$. Two series of unstructured meshes are used to perform the grid convergence

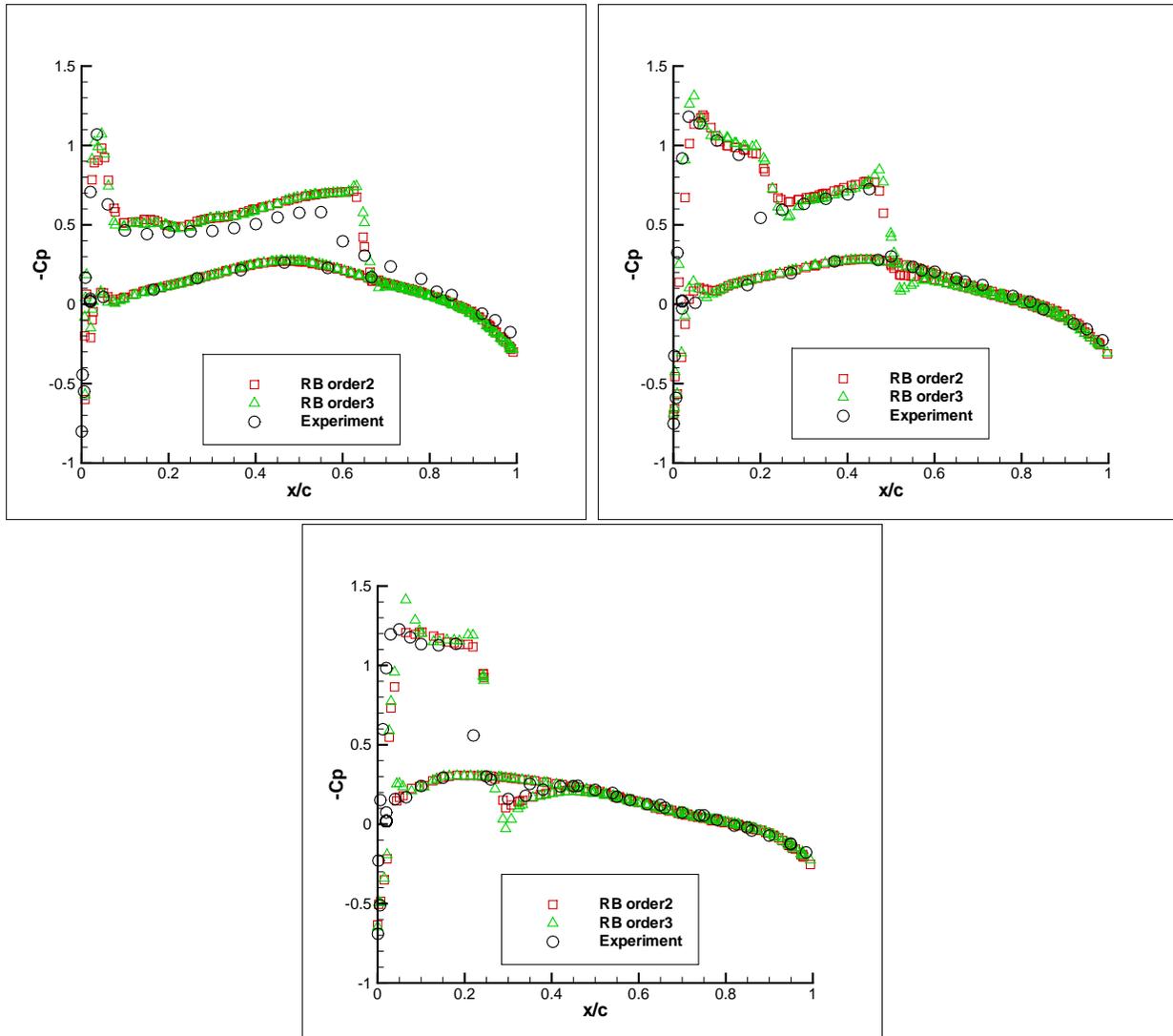


Figure 3.21: **Fine Unstructured mesh results.** C_p distribution on the wing body for span-wise wing sections located respectively 20% (top left), 65% (top right) and 95% (bottom) of the span-wise length away from the wing root, obtained by second and third-order FV-RB schemes.

study with the Roe $O2$, Roe $O3$, RB $O2$ and RB $O3$ schemes : a series of regular triangle meshes and a series of irregular triangle meshes which are precisely the ones used for the circular advection problem (see Table 3.1 and Table 3.2 for a summary of their main features).

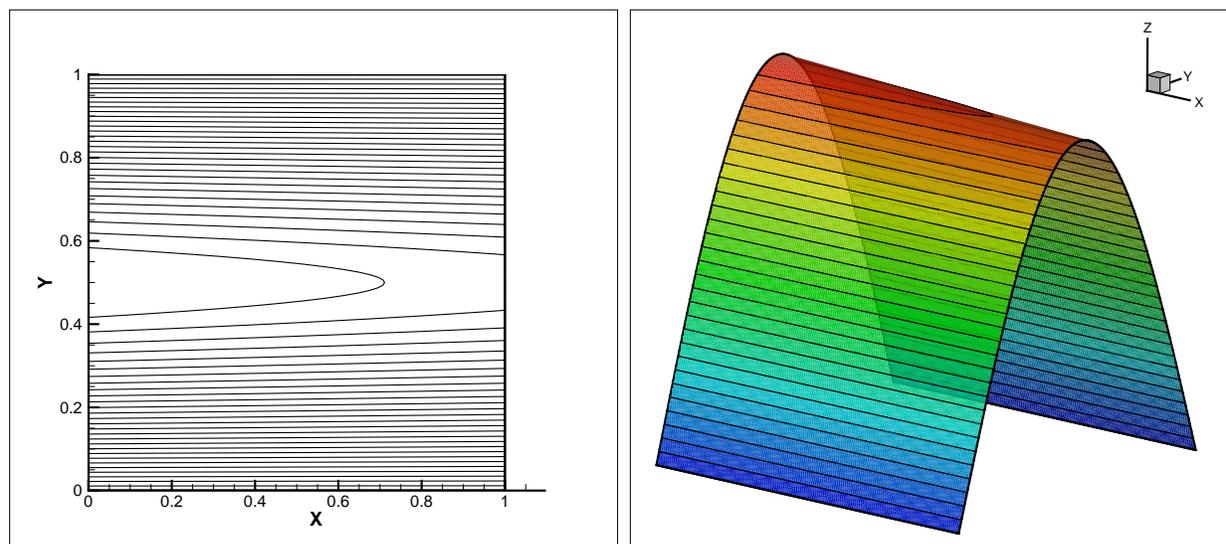


Figure 3.22: Left : contours of the solution W (with 30 levels from 0 to 1) obtained by the RB $O3$ scheme on mesh **Tri_reg5**. Right : 3D view of the exact solution.

Accuracy analysis on regular grids The numerical errors produced by the FV-RB schemes on regular triangle meshes are summarized in Table 3.23. Plots of the error versus the characteristic mesh size are also provided in Figure 3.23 along with the results given by the conventional Roe scheme. Let us recall that with the choice made in Chapter 2 for the design of the viscous extension of a conventional scheme and of the RB scheme, we expect in fact to recover in practice first-order accuracy only for the so-called Roe $O2$ and RB $O2$ schemes and second-order accuracy for the so-called Roe $O3$ and RB $O3$ schemes. Indeed, let us remind the reader the solution gradient is computed to first-order with the linear least-square reconstruction for the $O2$ scheme while the quadratic least-square reconstruction used by the $O3$ scheme gives a second-order approximation of the solution gradient.

In practice, the FV-RB $O2$ scheme gives indeed an order of precision close to 1 when the finer meshes are used while the RB $O3$ scheme is almost second-order accurate. Note the Roe and RB schemes of the same order yield close error levels which can be explained by the fact the dominant error for this low-Reynolds number flow comes from the centered discretization of the viscous flux, which is the same for Roe or RB scheme. It is important to note that the third-order schemes yield on the mesh **Tri_reg3** an error level comparable with the error level achieved by the second-order schemes on the finest mesh **Tri_reg5**.

Mesh	L_2 error of RBO2	order	L_2 error of RBO3	order
Tri_reg1	-2.59113	-	-3.52000	-
Tri_reg2	-3.19317	2.00	-4.32713	2.68
Tri_reg3	-3.78722	1.97	-4.98202	2.18
Tri_reg4	-4.32675	1.79	-5.58930	2.02
Tri_reg5	-4.70709	1.26	-6.16806	1.92

Table 3.23: L_2 norm of numerical errors obtained by the FV-RB scheme on regular triangle meshes.

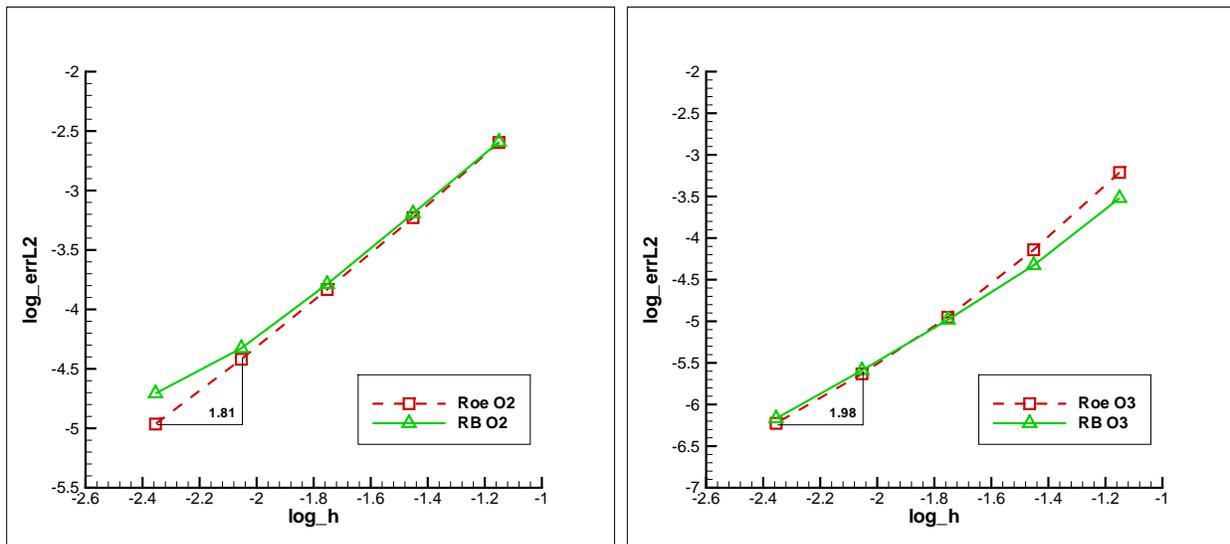


Figure 3.23: L_2 norm of numerical errors obtained by the FV-RB scheme and FV-Roe scheme on regular triangle meshes.

Accuracy analysis on irregular grids Table 3.24 and Figure 3.24 show the results obtained when performing the grid convergence analysis on the series of irregular triangle meshes. A bit strangely, it is found that the FV-RB $O2$ scheme gives a globally better convergence order on these irregular meshes than on the previous regular meshes, with an accuracy order equal to 1.62 when reaching the finest grids. A slightly better accuracy order is also found on irregular meshes for the RB $O3$ scheme, with an order equal to 2.17 on the finest grids. What is more interesting is the fact that, again, the $O3$ scheme needs a much coarser grid (**Tri_irreg3**) to produce the error level achieved with the $O2$ scheme on the finest grid **Tri_irreg5**.

Mesh	L_2 error of RBO2	order	L_2 error of RBO3	order
Tri_irreg1	-2.59034	-	-3.36309	-
Tri_irreg2	-3.22951	2.17	-4.12650	2.59
Tri_irreg3	-3.88792	2.46	-4.84348	2.68
Tri_irreg4	-4.48798	1.88	-5.56022	2.25
Tri_irreg5	-4.97336	1.62	-6.21312	2.17

Table 3.24: L_2 norm of numerical errors obtained by the FV-RB scheme on irregular triangle meshes.

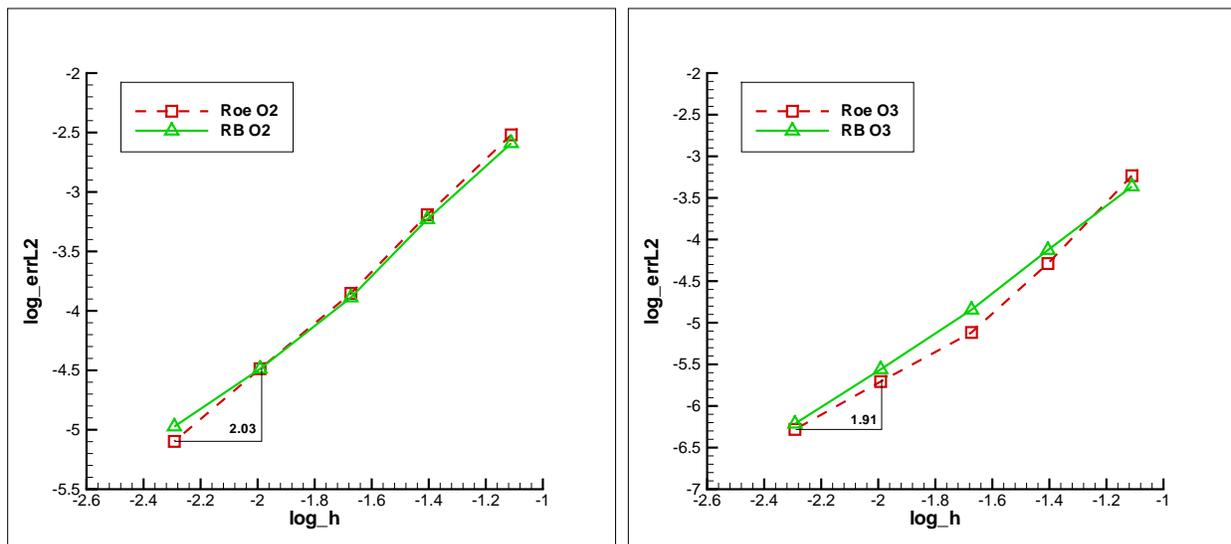


Figure 3.24: L_2 norm of numerical errors obtained by the FV-RB scheme and FV-Roe scheme on irregular triangle meshes.

Efficiency analysis The same analysis method used for 2D and 3D advection case is adopted for this case, which is to compare at first the efficiency of the second and third-order RB and Roe schemes on the same mesh, and then the cost of 4 schemes to achieve the same accuracy level. In Figure 3.25 it is interesting to see that on the same irregular triangle mesh, RB schemes take less iterations to reach the steady state than the Roe schemes, and iterations used by the second and third-order schemes to reach the convergence are comparable, with few iterations less necessary for the third-order schemes; and the most CPU-expensive scheme is Roe $O3$ scheme, RB $O2$ spends the

least CPU time to reach the steady state. The reason of these CPU time differences can be found in Table 3.25. It shows that although RB *O2* scheme takes 30 more iterations to reach the convergence than the RB *O3* scheme, its CPIPD is lower than the one of RB *O3*, which finally leads to a smaller CPU time of RB *O2* scheme. At the same time Roe *O3* scheme needs 10 iterations less than Roe *O2* scheme for the convergence, but a larger CPU time is obtained with the Roe *O3* scheme because of its higher CPIPD than the *O2* scheme. If values in this table are normalized by the Roe *O2* result, Table 3.26 can be obtained. It is found that Roe *O3* scheme is 1.29 and 1.98 times more expensive on CPIPD and memory respectively than the Roe *O2* scheme, RB *O3* scheme spends 1.27 times more CPIPD and 1.87 times more memory than the RB *O2* scheme. For the third-order schemes, RB *O3* scheme needs 1.1 times more CPIPD and 1.08 times more memory than the Roe *O3* scheme for the computation.

Scheme	Memory (MB)	CPU time (s)	Iterations	CPIPD
RB <i>O2</i>	47	80.1	520	2.7×10^{-6}
RB <i>O3</i>	88	96.4	490	3.4×10^{-6}
Roe <i>O2</i>	41	89.9	650	2.4×10^{-6}
Roe <i>O3</i>	81	115	640	3.1×10^{-6}

Table 3.25: Computation cost of the FV schemes obtained on mesh **Tri_irreg5**.

Scheme	Normalized Memory Requirement	Normalized CPIPD
RB <i>O2</i>	1.15	1.12
RB <i>O3</i>	2.15	1.42
Roe <i>O2</i>	1	1
Roe <i>O3</i>	1.98	1.29

Table 3.26: Normalized memory requirement and Cost Per Iteration Per **dof** for the RB and Roe second and third-order schemes.

As mentioned in **Accuracy analysis on irregular grids**, with mesh **Tri_irreg3** the third-order schemes achieved the same error level as the one obtained with second-order schemes on mesh **Tri_irreg5**, it is therefore useful to compare the efficiency of these schemes in this situation (Figure 3.26). It is seen that no matter in term of iterations or CPU time, the third-order schemes are far less expensive than the second-order schemes, and the difference between RB *O3* scheme and Roe *O3* scheme is very small. In summary, with less computation cost the third-order scheme can achieve the same accuracy level as the second-order scheme.

3.3.2 Subsonic laminar flow over a NACA0012 airfoil

In order to assess the performances of the FV-RB scheme for solving the Navier-Stokes equations, a steady laminar flow over the NACA0012 airfoil is computed, with the far-field conditions : $M_\infty = 0.5$, zero angle of attack and Reynolds number (based on the airfoil chord and the far-field incoming flow boundary conditions $Re_{\infty,c} = 500$). A series of 5 increasingly refined triangular meshes is used for the computations; the main features of these meshes are summarized in Table 3.27. An overview of the fine **mesh5** is also provided in Fig. 3.27. The grid convergence analysis is performed following the very same methodology used in the case of the subsonic inviscid case over the NACA0012 airfoil.

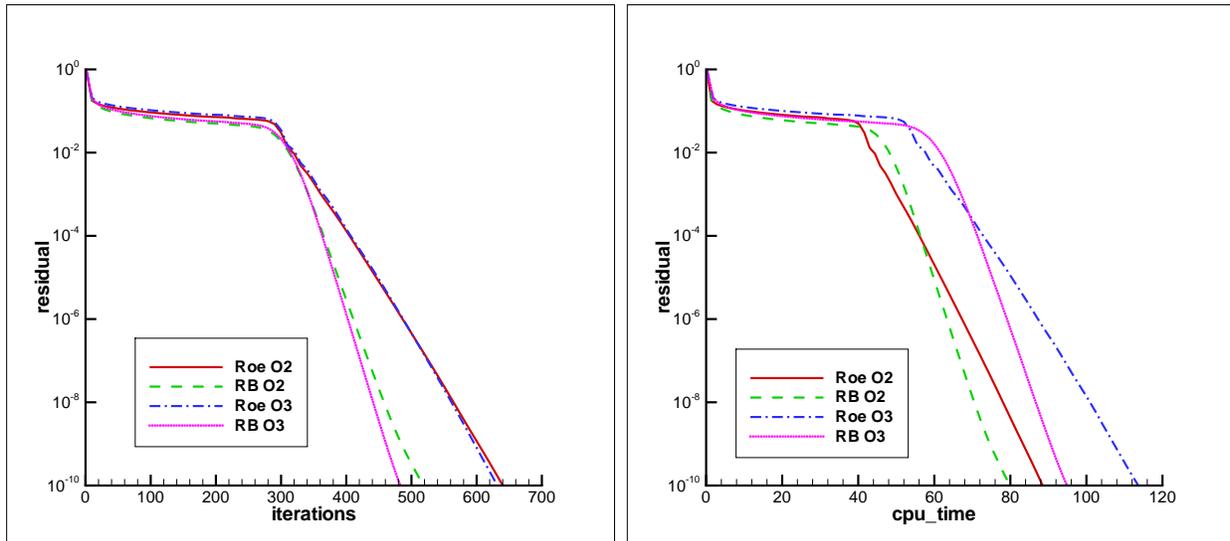


Figure 3.25: 2D advection-diffusion problem. Convergence history for the second and third-order Roe and RB schemes on the finest irregular grid `Tri_irreg5`. Left : L_2 norm of the residual vs iterations; right : L_2 norm of the residual vs CPU.

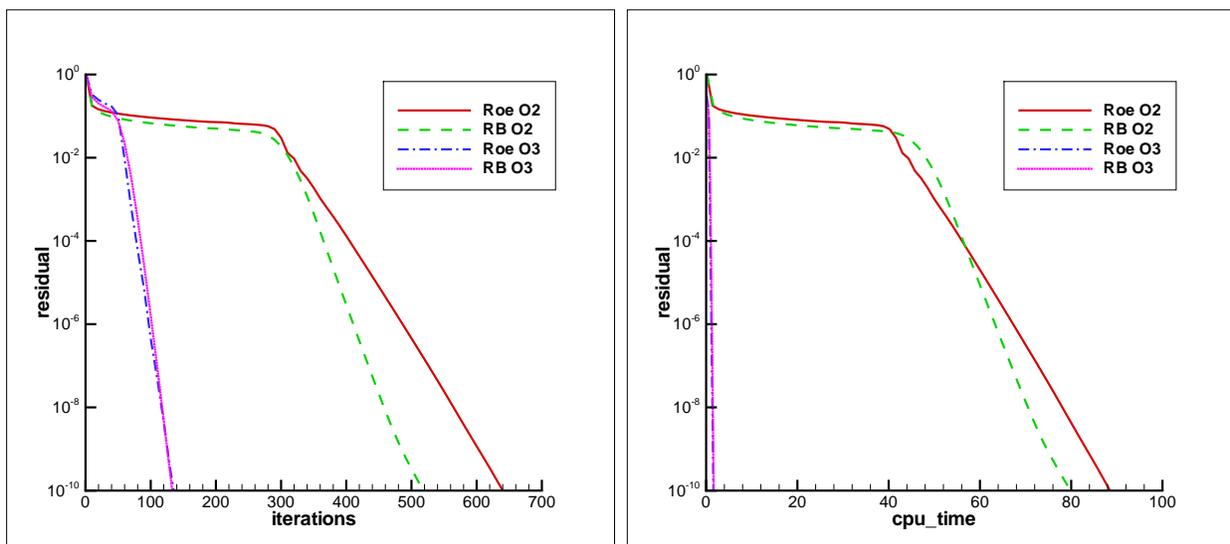


Figure 3.26: 2D advection-diffusion problem. Convergence history for the second- and third-order Roe and RB schemes on grids where a similar error level is achieved at steady state. Left : L_2 norm of the residual vs iterations; right : L_2 norm of the residual vs CPU.

The prescribed tolerances on the values of the aerodynamic coefficients computed in the finest grid are respectively $E_{C_l} = \pm 1 \times 10^{-3}$ for the lift coefficient, $E_{C_d} = \pm 5 \times 10^{-4}$ for the drag coefficient and $E_{C_m} = \pm 2 \times 10^{-4}$ for the moment coefficient.

Mesh	Dof	No. of faces on airfoil
mesh1	2262	16
mesh2	4518	50
mesh3	8526	150
mesh4	17210	400
mesh5	26384	700

Table 3.27: Unstructured triangle meshes used for subsonic laminar flow over the NACA0012 airfoil.

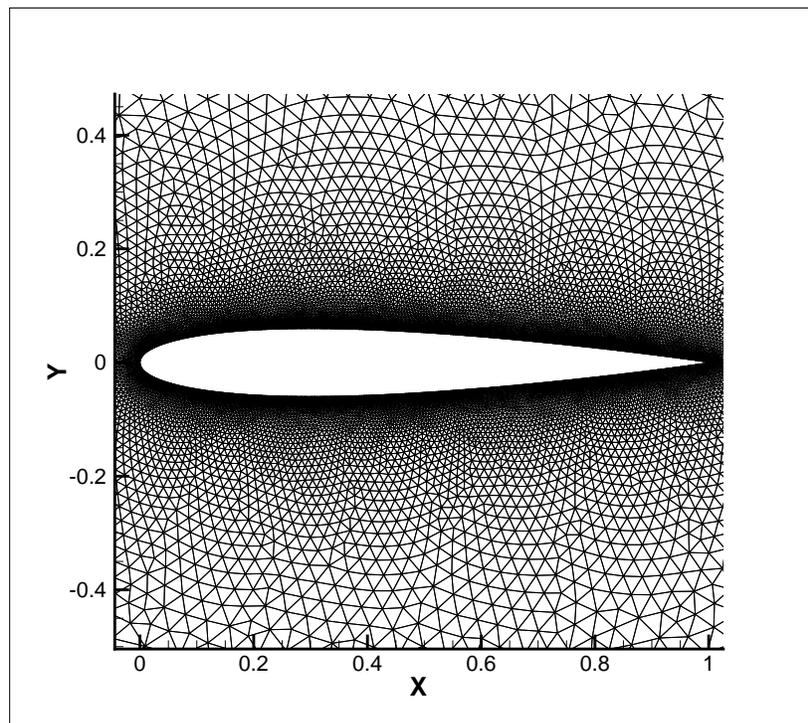


Figure 3.27: Global view of **mesh5**.

Overview of the flow The Mach contours obtained by using the RB *O3* scheme on the finest mesh **mesh5** are displayed on top of Fig. 3.29. The computed flow is perfectly (at least visually) smooth and symmetric with respect to the y axis. The flow seems to remain fully attached. These observations are confirmed by the plots of the wall-pressure coefficient distribution and skin-friction distribution along the airfoil. Table 3.28 displays the aerodynamic coefficients obtained on the finest mesh **mesh5** by the FV-RB *O2* and *O3* schemes. The ideal lift coefficient for this case should of course be zero because the flow is symmetric along the airfoil chord. The total drag coefficients C_d computed by both schemes are very close to each other, since their relative difference does not exceed

0.2%. Note however there is a compensation effect between the pressure drag C_{dp} and the viscous drag C_{dv} . Precisely the value of C_{dp} obtained by the RB *O3* scheme is 1.7% higher than the one obtained by the *O2* scheme, while the value of C_{dv} obtained with the *O3* scheme is 0.4% smaller than the one given by the *O2* scheme. Let us now proceed to analyze how the schemes evolve towards these values when the grid is refined.

Scheme	C_l	C_d	C_{dp}	C_{dv}	C_m
<i>O2</i>	-1.56×10^{-4}	0.181513	0.048751	0.132762	-2.31×10^{-5}
<i>O3</i>	-1.04×10^{-4}	0.181808	0.049580	0.132228	-2.59×10^{-5}

Table 3.28: Subsonic laminar flow over the NACA0012 airfoil. Aerodynamic coefficients obtained with the FV-RB scheme on **mesh5**.

Results analysis from the viewpoint of accuracy/efficiency In Figure 3.28 it is found that using FV-RB *O2* scheme, the lift coefficient remains within the convergence zone from **mesh3**. However, this coefficient is still decreasing slowly with further grid refinement. A fully converged value does not seem to be reached until the finest mesh **mesh5** is used. The drag coefficient enters the convergence zone from **mesh3** but is always increasing until **mesh5** where the convergence is achieved. The moment coefficient goes within the convergence zone from **mesh2** but its variation does not stop until **mesh5**, where its grid convergence can be assumed.

Using RB *O3* scheme, the lift coefficient enters the convergence zone from **mesh3** and remains almost constant until **mesh5**, which is to say the converged value is reached on **mesh3**, a very satisfying behavior with respect to RB *O2* scheme. The drag coefficient enters the convergence zone from **mesh3** and a fully steady state is obtained on **mesh4**. The computed moment coefficient enters the convergence zone from **mesh3** and reaches its steady state on **mesh4**. In summary, RB *O3* provides grid-converged values for the lift coefficient from **mesh3** and for the drag and moment coefficient from **mesh4**, while RB *O2* scheme needs **mesh5** for the grid convergence of all of coefficients. Note this analysis leads to think the convergence criteria provided within the ADIGMA project should have been more restrictive; in any case, the previous analysis assumes more stringent criteria of grid convergence than the ones initially provided.

Dof	Memory (MB)	CPU time (s)	Iterations	CIPPD
8526	11	2109	10170	2.43×10^{-5}
17210	21	6143	14620	2.44×10^{-5}
26384	32	10427	16380	2.41×10^{-5}

Table 3.29: Computation cost of the FV-RB *O2* scheme from **mesh3** to **mesh5**.

The performance of both RB *O2* and RB *O3* schemes is now compared in terms of CPU and memory cost. From Table 3.29 and Table 3.30, it can be first noticed that averagely FV-RB *O3* scheme has a 31% higher CIPPD (cost per iteration per **dof**) and 1.6 times more memory requirement than the one associated with the RB *O2* scheme. For a fully grid-converged lift coefficient, the RB *O2* scheme needs **mesh5**(26384 **dof**) with 10427s CPU time, while RB *O3* scheme only needs **mesh3** (8526 **dof**) with 2680s CPU time. The CPU time gain offered by RB *O3* with respect to RB *O2* is therefore

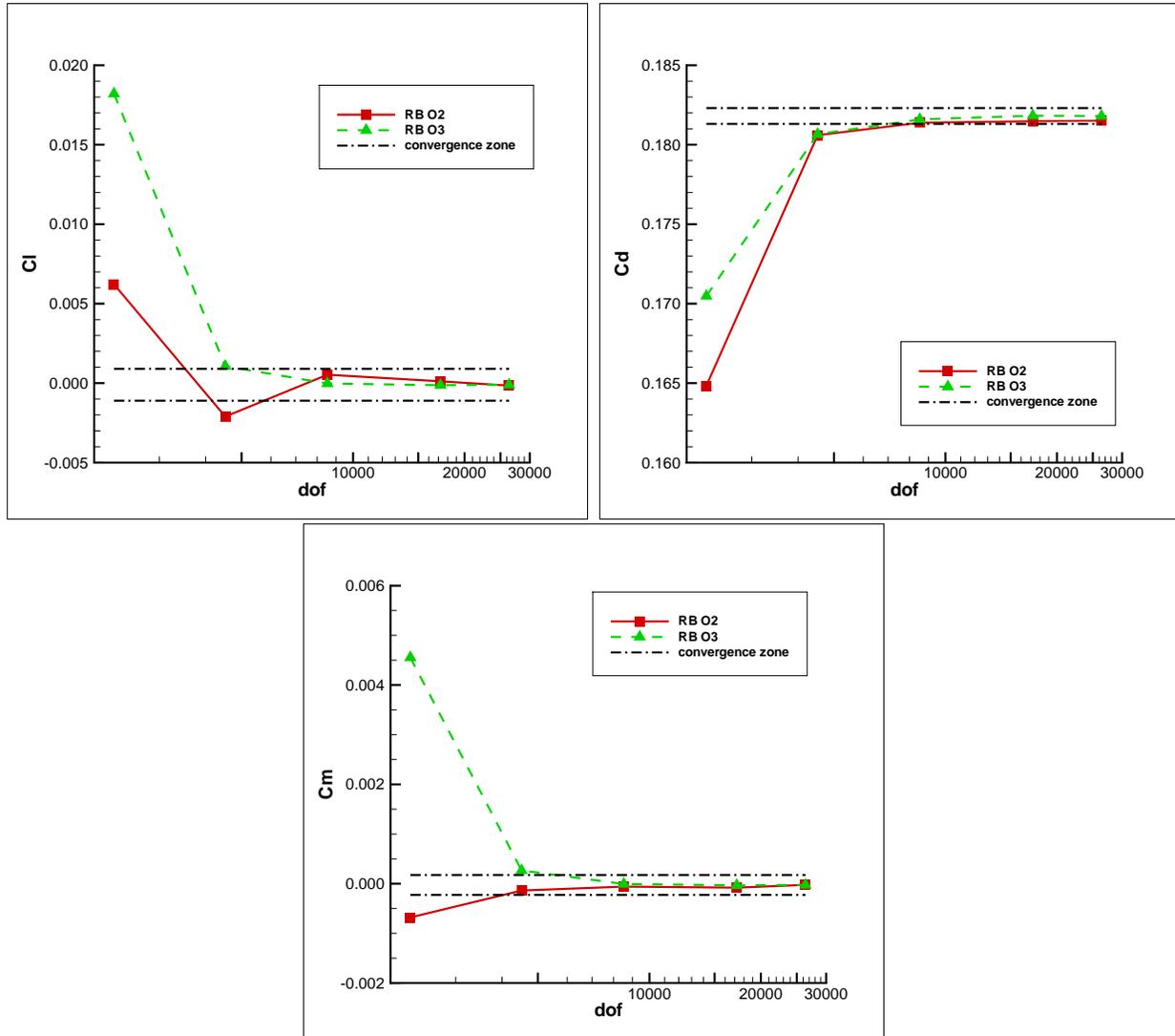


Figure 3.28: Aerodynamic coefficients convergence with dof based on meshes from **mesh1** to **mesh5** obtained by the FV-RB scheme.

Dof	Memory (MB)	CPU time (s)	Iterations	CIPD
8526	17	2680	10020	3.14×10^{-5}
17210	34	7952	14420	3.20×10^{-5}
26384	50	13729	16390	3.17×10^{-5}

Table 3.30: Computation cost of the FV-RB O3 scheme from **mesh3** to **mesh5**.

a factor of almost 4; the corresponding memory gain is 47%. The cost reduction achieved by RB *O3* scheme with respect to the RB *O2* scheme for the grid convergence of the drag and moment coefficients is less since **mesh4** has now to be used with RB *O3*. Precisely this reduction is about 24% in CPU time and the memory requirement is almost the same.

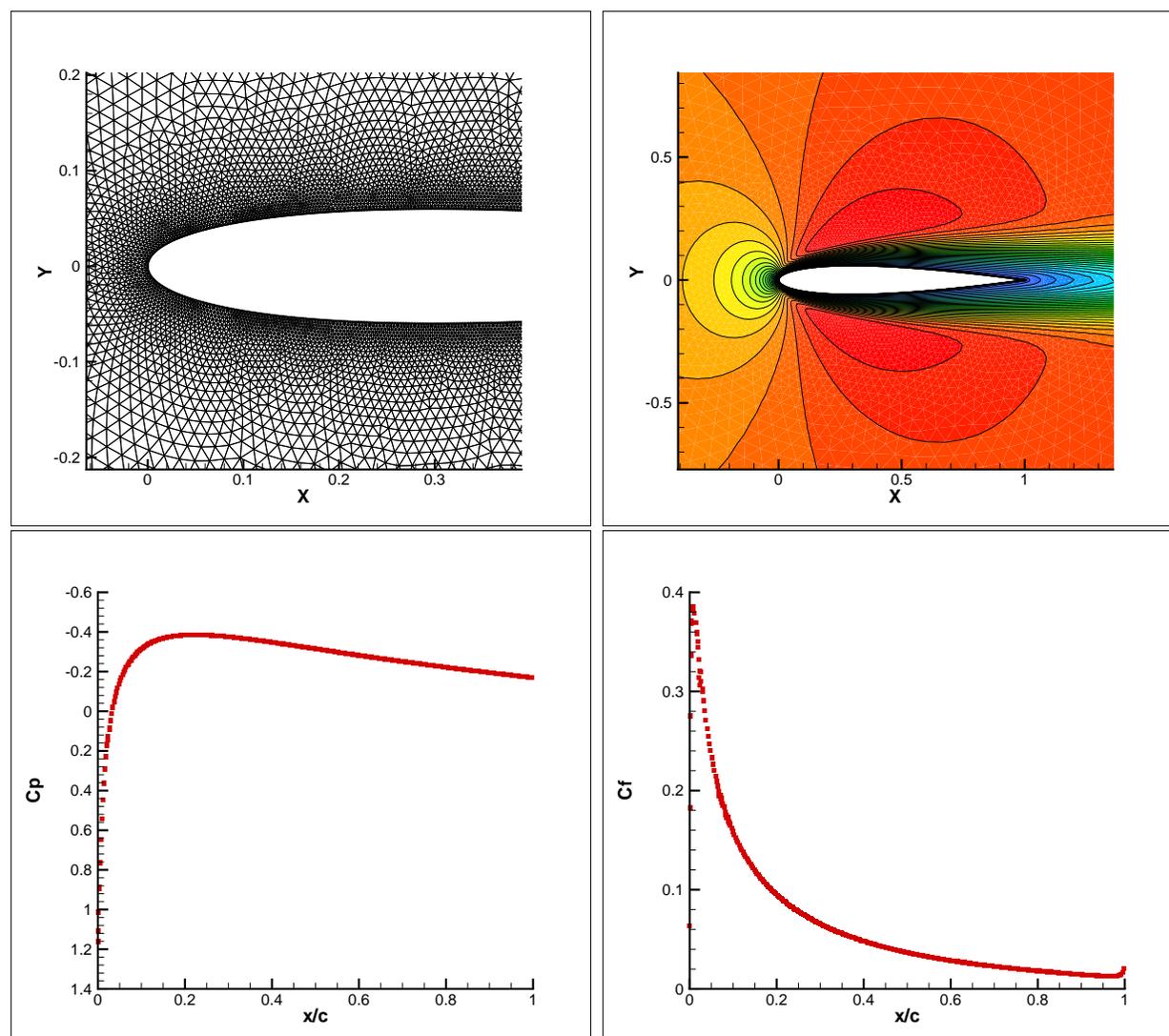


Figure 3.29: A local zoom of **mesh5**(top left) and the corresponding results obtained by the FV-RB *O3* scheme. Mach number contour (top right); Pressure coefficient distribution on the airfoil (bottom left); Friction coefficient distribution on the airfoil (bottom right).

3.4 Unsteady flows

We close this chapter with the application of the unsteady version of the RB schemes to the computation of unsteady flow problems :

- the propagation of an isotropic vortex along the diagonal of a flow domain will allow us to check whether the proposed unsteady extension of the RB scheme within a dual time framework is indeed second-order accurate.
- the computation of the (difficult) Double Mach Reflection problem will provide some insight on the robustness of the RB scheme when applied to the computation of flows involving strong moving shocks (in particular we will assess the interest of the variant proposed in 2.3.3).

3.4.1 Smooth flow: vortex propagation problem

Overview of the flow problem The unsteady Euler equations are solved on a computational domain $x \in [0, 10]$ and $y \in [0, 10]$. Initially, a mean flow defined by the following values for the primitive variables is set everywhere in the domain : $\{\rho, u, v, p\} = \{1, 1, 1, 1\}$. An isotropic vortex is added to this mean flow with the following perturbations on the velocity components, temperature and entropy :

$$\begin{aligned}\Delta u &= \frac{\varepsilon}{2\pi} e^{0.5(1-r^2)}(-\bar{y}), \\ \Delta v &= \frac{\varepsilon}{2\pi} e^{0.5(1-r^2)}(\bar{x}), \\ \Delta T &= \frac{(1-\gamma)\varepsilon^2}{8\gamma\pi^2} e^{(1-r^2)}, \\ \Delta S &= 0.\end{aligned}\tag{3.3}$$

In these expressions, ε denotes the vortex strength, $\varepsilon = 5$; the initial position of the vortex center is $x_c = 5$, $y_c = 5$ and $r^2 = \bar{x}^2 + \bar{y}^2$ with $\bar{x} = x - x_c$, $\bar{y} = y - y_c$ (see Figure 3.30). The vortex is convected through the domain with the velocity $(u, v) = (1, 1)$. Characteristic-based inflow and outflow boundary condition are used on all the boundaries. The solution is computed until $t = 2$ which ensures the vortex remains in the computational domain. It must be pointed out that our intent with this test-case is simply to check the correctness of the design principles of the unsteady RB schemes: we do not seek to assess for instance the behavior of the schemes when the vortex is propagated several times through the whole domain, where periodic boundary conditions should be used.

A series of regular triangular grid is used to perform the computations, from **Tri_reg1** with 200 **dof** to **Tri_reg4** with 12800 **dof** (see 3.31). Another important parameter to set up is the physical time-step : it must be large enough to limit the computational cost of the flow simulation but also small enough to make the time-discretization error small with respect to the space-discretization error. Besides, it must also be kept in mind that the physical time-step has a direct influence on the convergence speed to the pseudo-steady state at each physical time iteration. The physical time-step finally retained for the computations on the whole series of meshes has been $\Delta t = 0.002$, coupled with a criterion on the pseudo-time convergence consisting to let the computation run at each physical iteration until the residual on the dual-time derivative has been reduced by 4 orders of magnitude. An example of the evolution of this residual is showed in Figure 3.31, in each physical time step interval 0.002, the residual drops at least 4 orders.

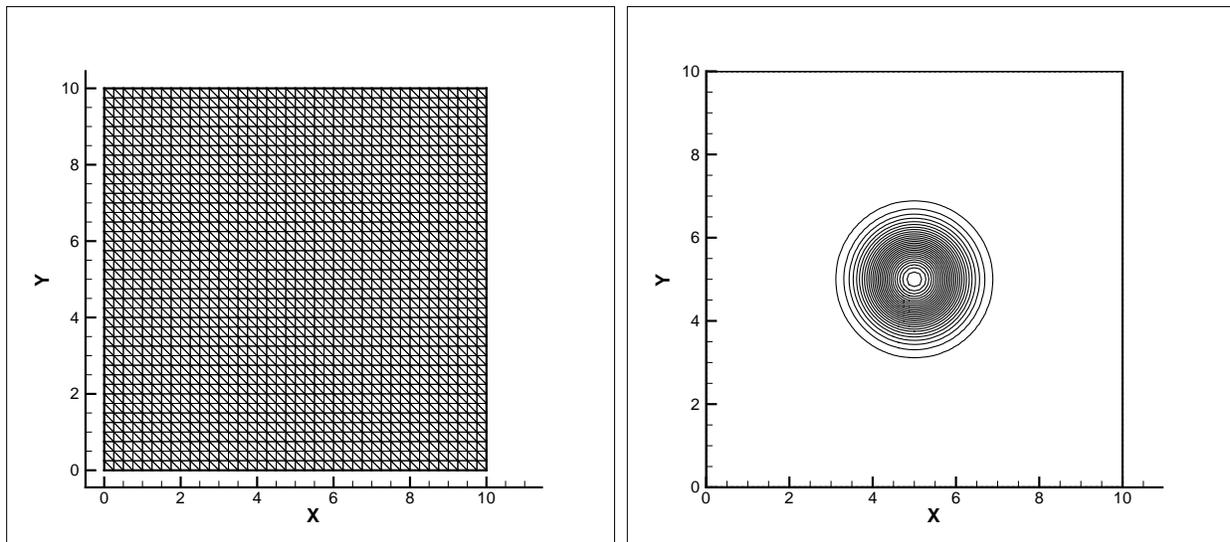


Figure 3.30: Left : mesh **Tri_reg3**. Right : initial vortex density contours (30 levels from 0.49 to 1.0).

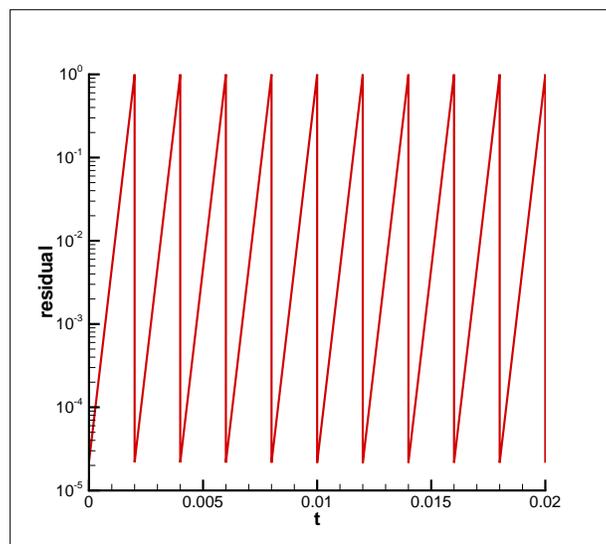


Figure 3.31: A zoom of residual convergence history vs physical time, obtained with RB $O3$ scheme on mesh **Tri_reg4**, time step $\Delta t = 0.002$.

Error analysis In Figure 3.32, the density of the exact solution and the numerical solution computed with RB $O3$ on the mesh **Tri_reg4** are displayed. On this fine grid, these two solutions are almost undistinguishable. Tables 3.31 and 3.32 summarize the computed error at $t = 2$ with RB $O2$ and RB $O3$ on this series of regular triangle grids. A comparison between these two sets of results allow to conclude that, with the same second-order time-discretization, the FV-RB $O3$ scheme yields more accurate results than the FV-RB $O2$ scheme. For instance, on the mesh **Tri_reg4** for example, the error reduction is about 5.5%.

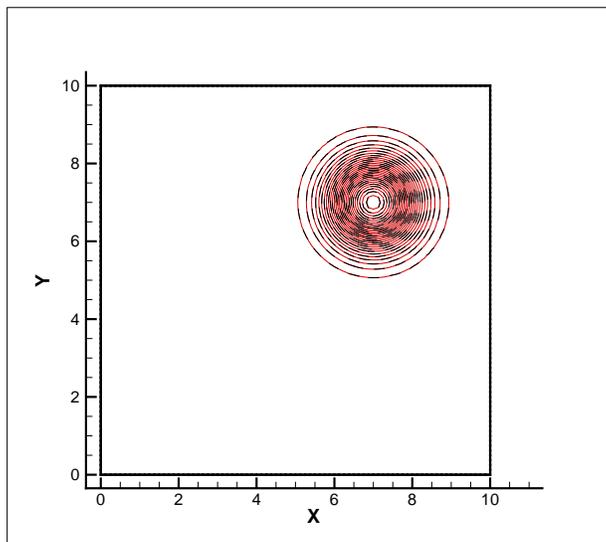


Figure 3.32: **Solution at $t = 2$.** Vortex density contours (30 levels from 0.49 to 1.003) obtained by the FV-RB $O3$ scheme with $\Delta t = 0.002$ on mesh **Tri_reg4** (red solid line), and exact solution (black dashed line).

Mesh	Dof	L_2 error	order
Tri_reg1	200	-2.66056	-
Tri_reg2	800	-3.40782	2.48
Tri_reg3	3200	-4.00367	1.98
Tri_reg4	12800	-4.51921	1.71

Table 3.31: L_2 norm of numerical errors of density obtained by the FV-RB $O2$ scheme.

Mesh	Dof	L_2 error	order
Tri_reg1	200	-2.56071	-
Tri_reg2	800	-3.29511	2.44
Tri_reg3	3200	-4.12090	2.74
Tri_reg4	12800	-4.76814	2.15

Table 3.32: L_2 norm of numerical errors of density obtained by the FV-RB $O3$ scheme.

3.4.2 Flow with shocks: double Mach reflection problem

Set up of the test-case The Double Mach Reflection (DMR) problem involves moving and reflecting strong shocks. The configuration computed here has been initially proposed by Woodward and Collela[51]. The flow domain is set as $x \in [0, 4]$, $y \in [0, 1]$. At initialization, a shock wave is sent diagonally into a reflecting wall (equivalently, the problem can also be seen as a shock moving horizontally and encountering a wedge). The initial flow is defined as the two constant states on each side of an inclined shock with shock Mach number $M_s = 10$, its foot attached at $(x = 1/6, y = 0)$ and forming an angle of 30° with respect to the y -axis (see Figure 3.33). The fluid located in the flow region in which the shock is advancing is initially at rest and such that :

$$\rho_1 = 1.4, p_1 = 1.$$

The upper boundary condition is set to describe the exact shock movement from the left to the right of the domain. With this condition and the Mach number of the shock, the flow variables can be computed in the following way. At the beginning, let us consider a moving vertical shock at velocity u_s with the same after-shock condition (Figure 3.34), it can be also considered as a static shock with fluid on both sides of it moves with different velocities (Figure 3.35). The Mach number of shock M_s is actually determined by the fluid on upstream of the shock: $M_s = M_1 = u'_1/a_1$, with the sound velocity defined by: $a_1 = \sqrt{\gamma p_1/\rho_1}$. As the fluid considered here is the perfect gas: $\gamma = 1.4$, the computed upstream fluid velocity is therefore $u_1 t = 10$, the shock velocity $u_s = 10$. And then the downstream fluid status can be obtained by the relationship across a vertical shock. The pressure is given by:

$$p_2 = \left(\frac{2\gamma}{\gamma + 1} M_1^2 - \frac{\gamma - 1}{\gamma + 1} \right) p_1,$$

the computed value is $p_2 = 116.5$.

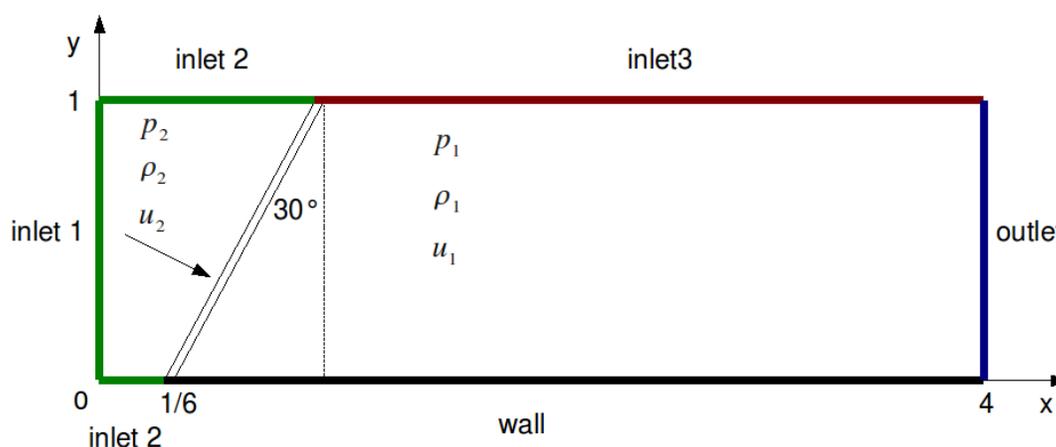


Figure 3.33: Initial shock position, and the corresponding boundary conditions.

The relationship for the fluid density and velocity is given by:

$$\frac{\rho_2}{\rho_1} = \frac{u'_1}{u'_2} = \frac{(\gamma + 1)M_1^2}{2 + (\gamma - 1)M_1^2},$$

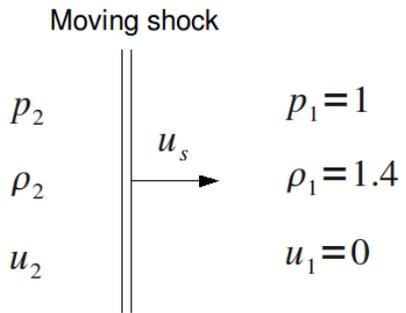


Figure 3.34: A moving vertical shock.

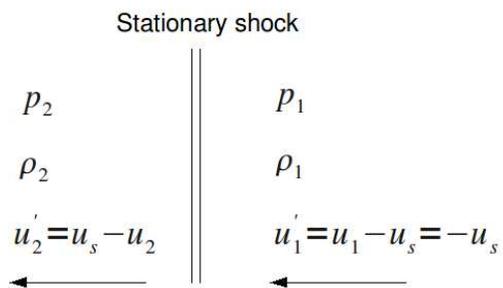


Figure 3.35: A static shock.

the computed downstream fluid density is: $\rho_2 = 8$, and velocity is $u_2' = 1.75$.

Now if we go back to the case with the moving vertical shock (Figure 3.34), status upstream the shock is:

$$p_2 = 116.5, \rho_2 = 8, u_2 = u_s - u_2' = 8.25.$$

For the moving shock with the presence of a inclined angle, the velocity component in x direction: $(u_2)_x = u_2 \cos(30^\circ)$, in y direction $(u_2)_y = -u_2 \sin(30^\circ) = -4.125$. Until now both of status upstream and downstream the inclined moving shock are obtained.

Figure 3.33 and Figure 3.36 show the boundary conditions chngement with time t . The shock foot is always located at $x = 1/6$, and the shock head position is changing with time:

$l_1 = \frac{1}{6} + \tan(30^\circ) + \frac{u_s t}{\cos(30^\circ)}$. On the left boundary, boundary condition **inlet1** is imposed, which represents that the physical flux computed from upstream shock conditions is used here. For the part of the top and bottom boundary upstream shock, the boundary condition **inlet2** represents that a numerical flux using upstream shock condition and extrapolated solution from internal domain is computed on this boundary. It is to note that the numerical flux of a classical approximate Riemann solver is used for the computation with FV-RB scheme, because the numerical flux of the FV-RB scheme cannot be used on the boundary. On the top boundary located at right hand side of the shock, condition **inlet3** means that the physical flux computed from the status downstream the shock is imposed. For the part of the bottom boundary downstream shock, the boundary condition **wall** represents that a physical flux is imposed, where the slip-boundary condition is used and the pressure is extrapolated from the internal domain. On the right boundary, the **outlet** condition impose zero solution gradient. The solution to be computed is the solution at $t = 0.2$.

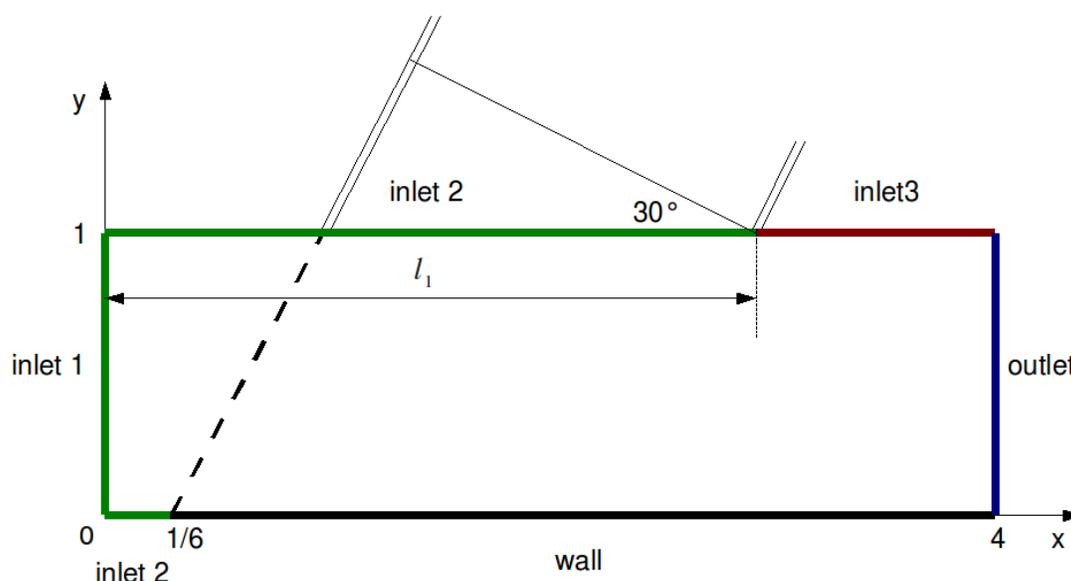


Figure 3.36: Shock movement on the upper boundary, and the corresponding boundary conditions.

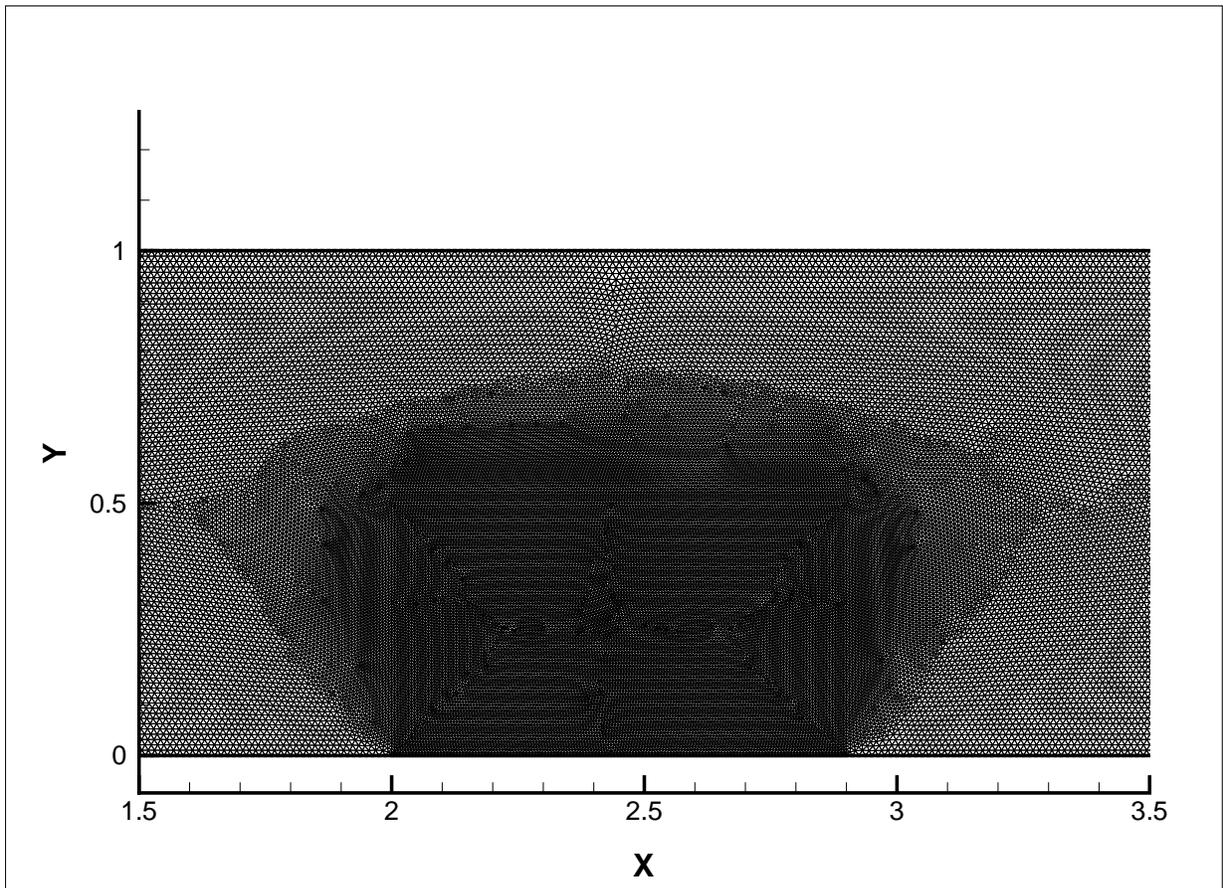


Figure 3.37: A local refined triangle mesh used for DMR case, zoom of the part $x \in [1.5, 3.5]$.

Analysis of the computed solutions The DMR test-case is computed using the FV-RB *O2* and *O3* schemes on a triangle-based mesh, which is locally refined in the region where the main flow structures will be located at the final time $t = 2$ of the computation. This mesh contains 98172 **dof** and a detailed view of the refined region is displayed in Figure 3.37. It is important to point out the baseline unsteady version of the RB schemes applied to this problem (that is the same version used for the previous vortex convection problem) fails to converge from the very start of the computation. The only way to obtain a developed solution at $t = 2$ with RB *O2* or RB *O3* is to make use of the RB time residual limiting described in 2.3.3). Unfortunately, this means that the coefficient β_{RB} which appears in equation 2.69 has to be tuned, on top of the coefficients K and β associated with the space discretization. When β_{RB} is taken large, it means the applied limiting is loose, so that many detailed structures such as the discontinuity vortex roll-up can be captured but many oscillation may appear and sometimes the computation can even explode before actually reaching $t = 2$. With a small value for β_{RB} (a strict limiting), the computation runs without difficulties and the result has few oscillations, but some fine structures are smeared out. The parameters used by both RB schemes are summarized in Table 3.33 and the corresponding results on this mesh are displayed in Figure (3.38). The global shock structure is correctly captured by both the RB *O2* and *O3* schemes with the applied limiting strategy but numerous oscillations remain present in the solution. A comparison with the reference result (Figure 3.39) taken from the computation results of a project partner UNST (University of Stuttgart) in the ADIGMA project report [3] clearly show some small flow structures (in particular the roll-up vortex structure) are not really well captured by RB schemes. Although the lack of grid refinement may be incriminated for this very demanding test-problem, it is clear however the RB scheme, in its present stage of development, is not well adapted to the accurate and efficient computations of such flows involving strong moving discontinuities.

Scheme	K	β_{RB}	β
RB <i>O2</i>	2.0	0.05	-
RB <i>O3</i>	2.0	0.05	0.05

Table 3.33: Limiter parameters used by the FV-RB scheme for DMR case.

3.5 Conclusion

In this chapter the grid convergence order of FV-RB second and third-order scheme is verified on simple model problems like a circular advection problem and advection-diffusion problem. Results show that the second-order RB scheme is robust to the mesh quality, while the result of the third-order RB scheme is quite mesh dependant. On a regular mesh it can give a very good result: small error and super convergence order, but a poor convergence order and big error could be obtained on a irregular mesh. This problem seems quite severe in 3D. And then some more complex problems are computed. For a smooth inviscid flow in 2D, the gain of third-order FV-RB scheme compared to the second-order scheme for computing a grid-converged aerodynamic coefficient is very clear. The precision advantage of the third-order scheme is also evident on a transonic case and viscous case in 2D. But for several 3D cases, the gain of third-order FV-RB scheme is not found. The possible reason is that a very large stencil is necessary for the third-order FV scheme, which compromised the performance of the third-order scheme to some extent.

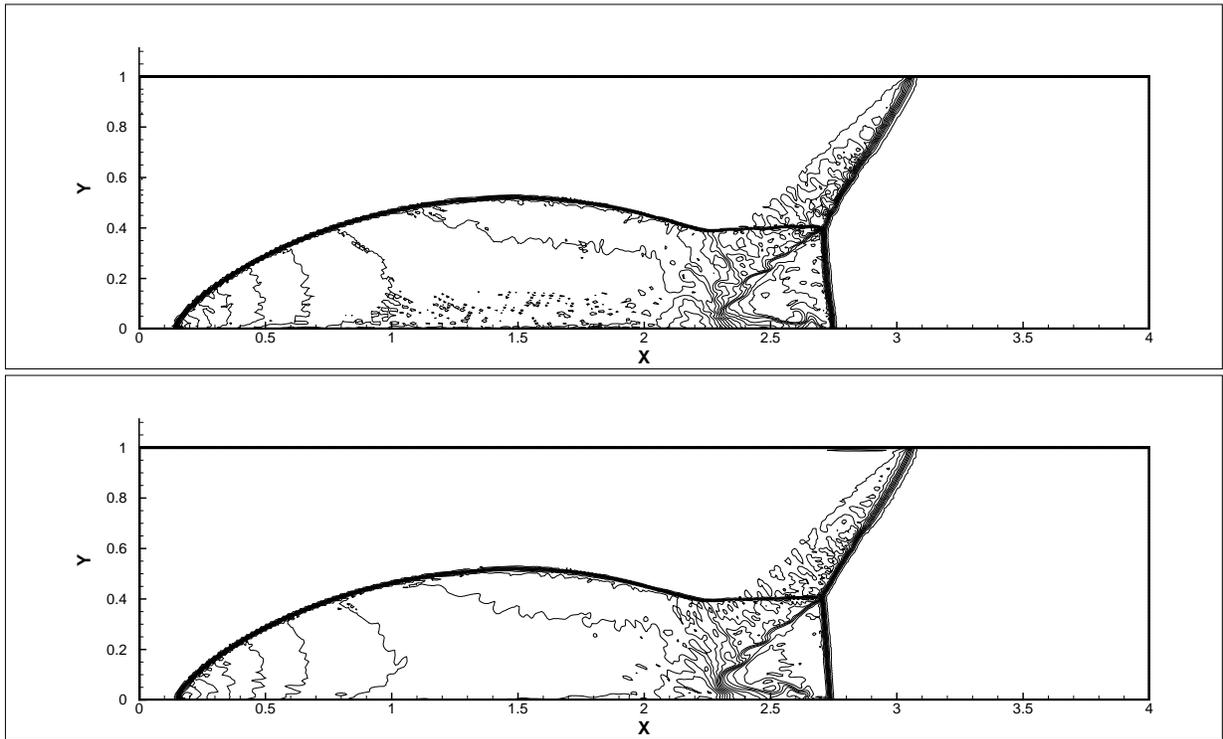


Figure 3.38: Density contours (30 levels from 1.5 to 21.5) obtained by the FV-RB O_2 scheme (top) and the O_3 scheme (bottom) with $\Delta t = 2 \times 10^{-4}$.

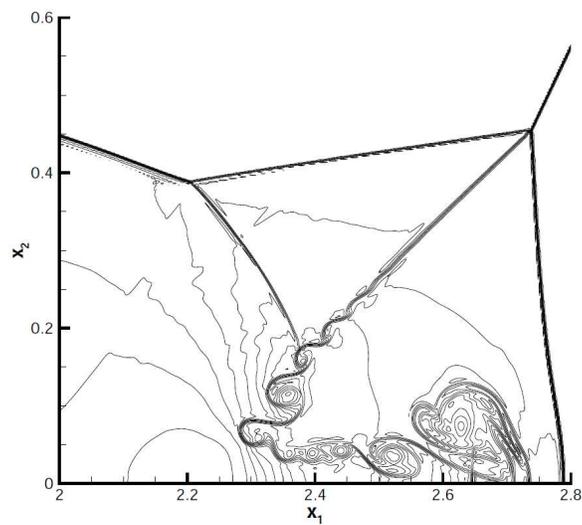


Figure 3.39: Reference result. Density contours obtained by a WENO-FV O_5 scheme on a structured mesh with 921600 **dof**.

4

Design principles for the SV-RB scheme

French La première partie de ce travail a été consacrée au développement et la validation d'un schéma d'ordre deux et d'un schéma d'ordre trois basés sur le résidu sous la formulation des volumes finis en maillages non-structurés. Ces schémas FV-RB $O2$ et $O3$ ont été appliqués à une série des problèmes d'écoulement et une comparaison systématique a démontré dans quelques cas le gain en efficacité due à la montée en ordre du schéma. Toutefois, lorsque l'on augmente l'ordre du schéma, les difficultés liées à la méthode des volumes finis sont apparus clairement : de l'ordre deux à l'ordre trois pour les problèmes $3D$ en particulier une augmentation très significative de la mémoire de stockage est introduite. En effet, il est difficile d'imaginer de concevoir, par exemple, une reconstruction cubique de moindres carrés en $3D$ en raison d'un stencil trop large qui lui serait associé. Une autre stratégie consiste à augmenter le nombre de degrés de liberté dans chaque cellule du maillage plutôt que d'étendre le support autour de chaque cellule. À cette fin, des essais ont été menés sur la méthode des différences spectrales [34][49] dans la phase initiale de cette thèse. Étant donné que certains problèmes de stabilité apparaissent lorsque le schéma RB est couplé avec la méthode des différences spectrales, on a finalement décidé d'utiliser la méthode des volumes spectraux (SV)[47] comme une base alternative de développement du schéma RB d'ordre très élevé en maillages non-structurés généraux. Dans cette thèse, on s'est concentré sur le développement d'un schéma d'ordre deux et surtout d'ordre trois, qui couple l'idée de SV et du flux numérique de RB (les schémas correspondants seront désignés par SV-RB $O2$ et SV-RB $O3$ à partir de maintenant).

Ce chapitre décrit le principe de l'approche SV et la façon dont on couple la SV avec le flux numérique du RB. Afin de présenter clairement les descriptions techniques, le cas d'application traité ici est un problème d'advection scalaire simple. Ensuite on va montrer comment la phase implicite sans matrice utilisée par le schéma FV-RB est implémentée avec succès pour les schémas SV. Finalement, les comparaisons seront faites non seulement entre les schéma SV-RB et les schémas décentrés-amont classiques couplés avec la SV, mais aussi entre les schémas SV-RB et les schémas FV-RB développés auparavant.

English The first part of this work has been devoted to the development and assessment of a second and third-order finite-volume formulation for the residual-based scheme on unstructured grids. These FV-RB $O2$ and $O3$ schemes have been applied to a large panel of flow problems; a systematic comparison has demonstrated in some cases the benefit for efficiency of increasing the accuracy order. However the difficulties associated with the finite-volume strategy when increasing this accuracy order were also made clear : going from second-order to third-order for $3D$ problems in particular

induces a very significant increase in memory storage. In fact, it is hardly conceivable to design for instance a cubic least-square reconstruction in 3D because of a huge stencil associated with it. An alternative strategy consists in increasing the number of degrees of freedom within each mesh cell rather than extending the support stencil for each cell. To this end, experiments were carried out in the initial stage of this thesis with the spectral difference method [34][49]. Since some stability problems emerged when the RB scheme is coupled with the spectral difference method, it was eventually decided to assess the Spectral Volume (SV) strategy proposed in [47] as an alternative framework for deriving potentially very high-order RB schemes on general unstructured grids. In the present work, we have focused on the design of a second and more importantly third-order accurate scheme combining the SV ideas with the RB numerical flux (the corresponding schemes will be denoted by SV-RB *O2* and SV-RB *O3* from now on).

This chapter describes the basic principles of the SV method and the way to combine SV with the RB numerical flux. In order to clarify as much as possible this technical description, the case of application treated here is a simple scalar advection problem. And then we will show how the matrix-free implicit method used by the FV-RB scheme is successfully implemented for SV schemes. Finally comparisons will be made not only between the SV-RB schemes and classical upwind schemes coupled with the SV method but also between these SV-RB schemes and the previously developed FV-RB schemes.

4.1 Introduction to the Spectral Volume method

Consider a computational domain Ω which can be divided into non-overlapping triangular cells, each cell is named a spectral volume (SV) and the i^{th} SV is denoted by S_i . In the present work we will restrict our study to triangular spectral volumes but the approach is general and applies to any type of cell (triangular or quadrilateral). Each SV is itself divided into a set of non-overlapping sub-cells in a structured way so that a polynomial of a certain degree can be reconstructed by using subcell-average values. The sub-cell is called control volume (CV); the j^{th} CV in the i^{th} SV is denoted by $C_{i,j}$. The triangular SV can be transformed into a simplex: a right triangle or an equilateral triangle; the first type is chosen in the present work. There are many choices for subdividing a SV into CVs in order to achieve a given accuracy order; the subdivisions or partitions initially used by Wang [48] have been found to be not the best ones [18]. A series of optimal partitions giving a smaller Lebesgue constant proposed by Chen [6] are used in the present work and displayed in Figure 4.1. For example, to construct a linear polynomial, at least 2 pieces of information are necessary in each direction, which results in 3 CVs (3 CV centroids i.e. 3 pieces information) totally in a SV. The details on the solution reconstruction will be described later.

Let us consider the following system of conservation laws :

$$\frac{\partial U}{\partial t} + \nabla \cdot \mathcal{F} = 0. \quad (4.1)$$

with the time t , the conservative variable U and the physical flux \mathcal{F} . The integral formulation of this

Note the choice was made to store the geometry-based data to be used in least-square linear or quadratic reconstruction. Not storing these data would make comparable memory requirements of the second and third-order schemes but, in the meantime, would severely increase the cost per iteration of the latter over the former.

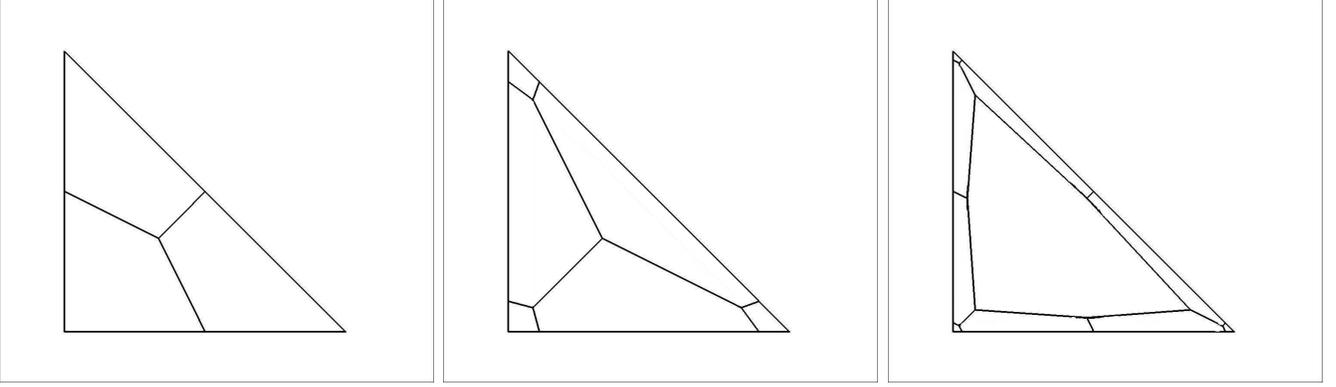


Figure 4.1: **Partitions of a right triangle simplex SV** Partitions for linear, quadratic, and cubic reconstruction (left to right).

equation leads to the following evolution equation in each CV:

$$\frac{\partial \bar{U}_{i,j}}{\partial t} + \frac{1}{|\Omega_{i,j}|} \int_{\Omega_{i,j}} \nabla \cdot \mathcal{F} d\Omega = 0. \quad (4.2)$$

where $\bar{U}_{i,j}$ is the average solution in the CV $C_{i,j}$ with its surface given by $|\Omega_{i,j}|$ (for the reason of simplicity, the bar on top of solution will be dropped from now on). By using Green-Gauss theorem, the integral term in the above equation becomes:

$$\int_{\Omega_{i,j}} \nabla \cdot \mathcal{F} d\Omega = \sum_{k=1}^{N_f} \int_{\Gamma_k} \mathcal{F} \cdot \mathbf{n} d\Gamma, \quad (4.3)$$

with Γ_k the k^{th} face of $C_{i,j}$, \mathbf{n} its normal outward-pointing unit vector, and N_f the total number of faces of $C_{i,j}$. The flux integral on each face can be approximated to the desired order by a Gauss quadrature rule :

$$\sum_{k=1}^{N_f} \int_{\Gamma_k} \mathcal{F} \cdot \mathbf{n} d\Gamma \simeq \sum_{k=1}^{N_f} \sum_{m=1}^{N_{gp}} \omega_m \mathcal{H}_{k,m} |\Gamma_k|. \quad (4.4)$$

where $\mathcal{H}_{k,m}$ is the numerical flux normal to the face Γ_k at the quadrature point m on this face, the corresponding quadrature weight is ω_m ; $|\Gamma_k|$ is the face area which is the length of face Γ_k in 2D; total number of quadrature points on the face k is N_{gp} .

A solution polynomial in the SV i is derived from the CV-averaged values $U_{i,j}$ and takes the form :

$$U_i(x, y) = \sum_{j=1}^{N_{dof}} L_{i,j}(x, y) U_{i,j}. \quad (4.5)$$

where N_{dof} is the number of degrees of freedom, which is the number of CVs in each SV. To construct a polynomial of degree n in 2D on a SV, a number $N_{dof} = (n+1)(n+2)/2$ of independent pieces of information is necessary (by piece of information we mean here a CV-averaged value). For example, $N_{dof} = 6$ for the third-order scheme, hence there are 6 CVs for the quadratic reconstruction (Figure

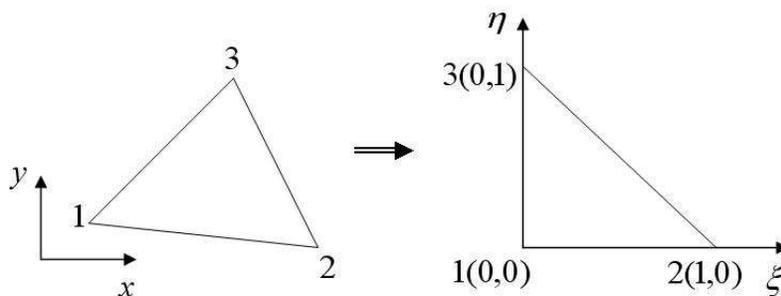


Figure 4.2: **SV mapping from physical domain to computational domain**

4.1), which will be used later when deriving third-order schemes. Moreover, the solution reconstruction coefficient $L_{i,j}(x, y)$ depends on the location of the centroid of $C_{i,j}$ in the SV i . It seems that this coefficient varies in each SV for every point $k(x_k, y_k)$. In fact it becomes constant for every point k if each SV is transformed into a standard cell (a simplex) which is then divided into straight edged CVs in a structured way, as those partitions showed in Figure 4.1. The mapping process from a normal mesh cell to the simplex is showed in Figure 4.2, where the original domain (with coordinates x and y) is called physical domain, the new domain (with coordinates ξ and η) being called computational domain. This mapping process can be also expressed by the following equation:

$$\mathbf{r} = \mathbf{r}_{i,1} + \xi(\mathbf{r}_{i,2} - \mathbf{r}_{i,1}) + \eta(\mathbf{r}_{i,3} - \mathbf{r}_{i,1}). \quad (4.6)$$

where $\mathbf{r} = [x \ y]^T$ is the vector of coordinates, $\mathbf{r}_{i,j}$ with $j = 1, 3$ are the coordinates of the three nodes of the triangular SV i , in which the polynomial reconstruction is going to be built.

The reconstruction coefficients are computed by expressing the fact that the average value of the reconstructed solution U_i over the CV $C_{i,j}$ is equal to the solution $U_{i,j}$:

$$\frac{1}{|\Omega_{i,j}|} \int_{\Omega_{i,j}} L_m(\xi, \eta) d\Omega = \delta_{j,m}, \quad (m = 1, N_{dof}), \quad (4.7)$$

with $\delta_{j,m}$ the Kronecker delta function. For example, in the case of a linear reconstruction the reconstruction coefficient can be developed as :

$$L(\xi, \eta) = a_0 + a_1\xi + a_2\eta. \quad (4.8)$$

where a_0 , a_1 and a_2 are polynomial coefficients to be determined. Since there are three CVs $C_{i,j}$ in the SV S_i for the linear reconstruction case ($N_{dof} = 3$), equation (4.7) yields three equations for each $L_{i,j}$ ($j = 1, N_{dof}$). The three polynomial coefficients for a given reconstruction coefficient L can be computed from these equations. In this way, all three reconstruction coefficients in function of the computational coordinates $L_1(\xi, \eta)$, $L_2(\xi, \eta)$ and $L_3(\xi, \eta)$ can be obtained.

After the mapping process equation (4.5) can be written in this form :

$$U_i(\xi, \eta) = \sum_{j=1}^{N_{dof}} L_j(\xi, \eta) U_{i,j}. \quad (4.9)$$

Given the computational coordinates of any point in the SV, the solution at this point could be estimated from (4.9). In practice only the solution at some points is necessary, typically at Gauss quadrature points. In a simplex, the coordinates (ξ_g, η_g) of these Gauss points are known once for all, therefore any reconstruction coefficient $L_j(\xi_g, \eta_g)$ is a constant for the Gauss point g . Coefficients for each Gauss point are computed and saved at the beginning of the computation.

Since an independent solution polynomial is reconstructed in each SV, the solution is continuous through the CV faces located inside the SV. Consequently, the flux at a Gauss point located on a so-called internal CV face is simply computed as the physical flux using the solution estimate at this Gauss point. The picture is of course different on the boundaries of a SV, or so-called external CV faces, coincident with CV faces from a neighboring SV. In that case, the solution reconstruction is *a priori* discontinuous since expressed by two distinct polynomials; a common numerical flux is then necessary for approximating the physical flux in equation (4.4) at Gauss points located on external CV faces which belong to the SV boundary. The numerical flux can be obtained by using an exact or approximate Riemann solver (Rusanov scheme, HLLC scheme or Roe scheme typically) or by using a RB numerical flux. The formulation of the RB scheme based on the SV method will be detailed in the next section. In the present section, we complete our description of the SV approach by using a classical approximate Riemann solver as the numerical flux, namely the Rusanov numerical flux given by :

$$\mathcal{H}_{Rusanov} = \frac{1}{2}(\mathcal{F}_L^\perp + \mathcal{F}_R^\perp) - \frac{1}{2}\lambda^{max}(U_R - U_L). \quad (4.10)$$

where \mathcal{F}_L^\perp and \mathcal{F}_R^\perp are the projection of the physical flux onto the outward face normal direction, respectively computed in the left and right CV of the corresponding interface, λ^{max} is the maximum wave-speed associated with the hyperbolic system to be solved, which is the maximum absolute value of the normal Jacobian matrix eigenvalues.

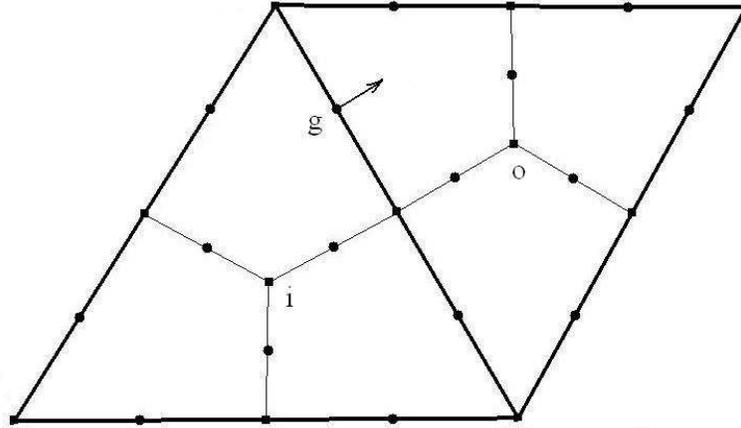
The partition of the SV into CVs and the Gauss points along the faces of each CV in the case of a second-order SV method are provided in Figure 4.3. At the so-called interior Gauss-points, located along faces of CVs which are not shared with another SV, the numerical flux appearing in (4.4) is directly computed as :

$$\mathcal{H}_g = \mathcal{F}^\perp(U_g) \quad (4.11)$$

where U_g is computed with (4.9). For boundary Gauss-points, located along faces of CVs which are shared with another SV, the physical flux is approximated by a numerical flux, written as follows in the case where the Rusanov flux is retained :

$$\mathcal{H}_g = \frac{1}{2}(\mathcal{F}^\perp(U_{g,L}) + \mathcal{F}^\perp(U_{g,R})) - \frac{1}{2}\lambda_g^{max}(U_{g,R} - U_{g,L}). \quad (4.12)$$

where $U_{g,L} = U_{g,i}$ is the solution obtained by the reconstruction polynomial in SV i and $U_{g,R} = U_{g,o}$ is the solution obtained by the reconstruction polynomial in SV o . The maximum signal speed, $\lambda^{max}(U_{g,L}, U_{g,R})$ is computed at the Gauss point g using typically a Roe-averaged state computed from $U_{g,L}$ and $U_{g,R}$.


 Figure 4.3: CVs and Gauss points for O_2 SV

4.2 Strategy for time-integration

4.2.1 Explicit time-advancement

Following several authors who have initially contributed to the development of the SV approach, a third-order Runge-Kutta scheme [20] can be used for computing both steady and unsteady problems [47], [18]. Time-accuracy will be ensured for unsteady flows while robustness in the choice of a CFL number will be obtained for steady flow computations. The method reads :

$$\begin{aligned}
 \bar{U}_{i,j}^1 &= \bar{U}_{i,j}^n + R(\bar{U}_{i,j}^n), \\
 \bar{U}_{i,j}^2 &= \frac{3}{4}\bar{U}_{i,j}^n + \frac{1}{4}[\bar{U}_{i,j}^1 + R(\bar{U}_{i,j}^1)], \\
 \bar{U}_{i,j}^3 &= \frac{1}{3}\bar{U}_{i,j}^n + \frac{2}{3}[\bar{U}_{i,j}^2 + R(\bar{U}_{i,j}^2)], \\
 \bar{U}_{i,j}^{n+1} &= \bar{U}_{i,j}^3.
 \end{aligned} \tag{4.13}$$

where the residual of equation (4.2) is given by :

$$R(U_{i,j}^n) = -\frac{\Delta t_{i,j}}{|\Omega_{i,j}|} \sum_{k=1}^{N_f} \sum_{m=1}^{N_{gp}} \omega_m \mathcal{H}_{k,m} |\Gamma_k|. \tag{4.14}$$

with $\Delta t_{i,j}$ the time step in the CV $C_{i,j}$. Based on the definition of the CFL number, the time step is computed by:

$$\Delta t_{i,j} = \frac{CFL \cdot h_{i,j}}{\lambda_{i,j}^{max}}. \tag{4.15}$$

where $h_{i,j}$ is the characteristic size of the CV $C_{i,j}$ and is computed as for the FV-RB scheme (refer to section 2.1.3); $\lambda_{i,j}^{max}$ is the maximum eigenvalue of the physical flux Jacobian matrix in equation (4.2). This approach has been implemented within our SV numerical solver; however since it was important for us to be in position to perform a fair comparison between the FV-based and SV-based

strategies for deriving third-order versions of the RB scheme, it was also decided to devote some effort to the building of an implicit version of the SV scheme for computing steady problems.

4.2.2 Matrix-free implicit method for SV

As previously mentioned, in the course of this work, implicit formulations for SV schemes have been derived by some authors (motivated in particular by the need to speed up the convergence to steady-state in the case of viscous flow problems). For instance, an implicit LU-SGS algorithm coupled with a p -multigrid strategy has been successfully used for the SV method in [40]. In this work, we have naturally decided to extend to the SV formulation the simple matrix-free implicit scheme presented in the first part of the work (see section 2.1.3, devoted to the FV formulation of the RB scheme).

For SV method, the implicit scheme is used for each CV instead of SV because the discretization equation is based on a CV. The basic idea of this matrix-free scheme is to use a Rusanov scheme for the numerical flux in the implicit phase, while the numerical flux used in explicit phase is independent on the implicit scheme. There are two choices: either using the implicit numerical flux only on the boundary CV faces (faces located on the SV boundary), or using the implicit scheme on all of CV faces. By considering that later choice could bring more dissipation to the whole scheme, this choice is therefore adopted in our computation. In order to implement the implicit scheme in this way, a technical issue in the programming comes, which is that all of CVs and the nodes and faces of each CV needs to be numbered globally in the whole physical domain, while it is not necessary when the explicit scheme is used, where CVs are numbered locally in a SV, i.e. $C_{i,j}$ is the j^{th} CV of SV i . As long as the global numbering is done, the connection between a local CV in a SV and a global CV is necessary, some detailed issues related to programming are discussed in Appendix.

Let us recall the discretized system of conservation law for a CV $C_{i,j}$:

$$\Delta U_{i,j}^n = R(U_{i,j}^n), \quad (4.16)$$

where $\Delta U^n = U^{n+1} - U^n$, and the residual $R(U_{i,j}^n)$ is given by (4.14). Now if we drop the notation for the local CV and use a new notation for any global CV p , the above discretization becomes:

$$\Delta U_p^n = -\frac{\Delta t_p}{|\Omega_p|} \sum_{k=1}^{N_f} \sum_{m=1}^{N_{gp}} \omega_m \mathcal{H}_{k,m} |\Gamma_k|. \quad (4.17)$$

Then the implicit scheme is used in the same way as described in section 2.1.3. The total implicit system is also solved with the Point-Jacobi relaxation technique, which gives:

$$\Leftrightarrow \begin{cases} \Delta U_p^{(0)} = 0 \\ \begin{cases} l = 1, L \\ \Delta U_p^{(l)} = \frac{1}{C_p} (\mathcal{R}_p^n - a \sum_k |\Gamma_k| (\Delta \mathcal{H}_p^{(l-1)} + \Delta \mathcal{H}_{o(p,k)}^{(l-1)} - (\rho_\perp)_{p,o(p,k)} \Delta U_{o(p,k)}^{(l-1)})) \end{cases} \\ \Delta U_p^{(n)} = \Delta U_p^{(L)} \end{cases} \quad (4.18)$$

where L is the number of sub-iteration; the coefficient $C_p = 1 + a\sigma \sum_k |\Gamma_k| (\rho_\perp)_{p,o(p,k)}$, with $a = \frac{\Delta t_p}{2|\Omega_p|}$, and a relaxation coefficient σ . Normally $\sigma = 1$ for a high efficiency of the implicit phase. But this high efficiency could result in a stability problem in a certain case. Therefore only a small CFL is permitted. By increasing the σ a big CFL can still be used.

4.3 Coupling the SV method and the RB numerical scheme

In the following a 2D second and third-order RB scheme is developed within the framework of the SV method. The proposed formulation can be "easily" generalized to higher orders of precision and to 3D problems. By "easily", we mean there is no such obstacle as the exceedingly large stencil associated with a high-order 3D FV extension. Let us start from the SV-RB O_2 scheme, the partition of the SV and choice of Gauss quadrature points on each CV face are showed in Figure 4.4. A shift cell formed by two CV centroids i, j and o, k , and two nodes $n1$ and $n2$ of the face on which the numerical flux is computed. Let us now proceed to a detailed description of the RB flux computation at a Gauss-point g on frace k .

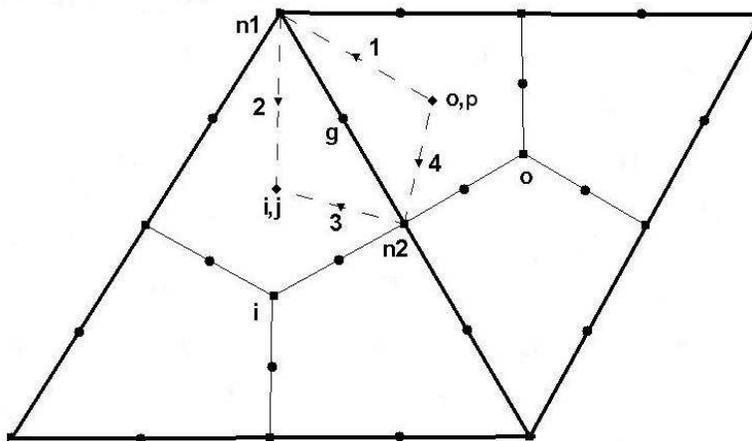


Figure 4.4: Shift cell used by the SV-RB O_2 scheme. For the RB numerical flux computed at Gauss point g on a boundary CV face $n1 - n2$, the shift cell formed by i, j (centroid of j^{th} CV in SV i), o, p (centroid of p^{th} CV in SV o), and two nodes ($n1$ and $n2$) of this face is used. Points 1 – 4 are quadrature points used for residual computation in this shift cell.

As in the FV method, the RB numerical flux still includes two parts, namely a purely centered flux and a numerical dissipation :

$$\mathcal{H}_g^{RB} = \frac{1}{2}(\mathcal{F}_{g,L}^\perp + \mathcal{F}_{g,R}^\perp) - d_g, \quad (4.19)$$

where $\mathcal{F}_{g,L}^\perp$ and $\mathcal{F}_{g,R}^\perp$ are the normal fluxes at the Gauss point g calculated correspondingly in the SV i and SV o . The definition of the numerical dissipation d_g is similar to the one defined for FV-RB scheme:

$$d_g = \frac{1}{2} L^\perp \Phi_k \mathcal{R}_k, \quad (4.20)$$

where $L^\perp = \Delta \mathbf{r}_{(i,j),(o,p)} \cdot \mathbf{n}_k$ is the projection of distance between the centroids of two CVs sharing the interface k on the face normal direction; the $\mathcal{O}(1)$ dissipation coefficient matrix Φ_k is calculated in the same way as for FV-RB (refer to equation (2.18) and (2.19)); and \mathcal{R}_k is an approximation of the residual integral:

$$\mathcal{R}_k = \frac{1}{|\Omega_k|} \int_{\Omega_k} r dV, \quad (4.21)$$

where r the residual of the considered system, here Euler equations are considered: $r = \nabla \cdot \mathcal{F}(U)$; and $|\Omega_k|$ is the surface of the shift cell (see Figure 4.4). Taking into account the residual r in equation (4.21) and using Gauss's theorem yields :

$$R_k = \frac{1}{|\Omega_k|} \int_{\partial\Omega_k} \mathcal{F} \cdot \mathbf{n} \, dS = \frac{1}{|\Omega_k|} \sum_{l \in \mathcal{I}(\Omega_k)} \int_{\Gamma_l} \mathcal{F} \cdot \mathbf{n} \, d\Gamma, \quad (4.22)$$

where $\mathcal{I}(\Omega_k)$ includes all the faces of the shift cell. For a second-order scheme, this residual should be approximated at least to first-order in order to obtain a second-order dissipation term. A second-order - at least - residual approximation is necessary for a third-order dissipation. Obviously the integral in equation (4.22) can be computed by a Gauss quadrature formula :

$$\int_{\Gamma_l} \mathcal{F} \cdot \mathbf{n} \, d\Gamma = \sum_{q=1}^{N_q} \omega_q \mathcal{F}_{l,q}^\perp |\Gamma_l| + \mathcal{O}(h^{2N_q+1}), \quad (4.23)$$

where N_q is the number of quadrature points, ω_q is the quadrature weight corresponding to point q , the length of the face l is $|\Gamma_l|$ and h is the typical mesh size. Clearly the exact flux $\mathcal{F}_{l,q}^\perp$ is not available because the exact solution at this point remains of course unknown. Therefore a numerical flux $\mathcal{H}_{l,q}$ is used as replacement, which is obtained by substituting the reconstructed solution into the physical flux, *i.e.* $\mathcal{H}_{l,q} = \mathcal{F}^\perp(U_{l,q}^R)$, so that :

$$\mathcal{H}_{l,q} = \mathcal{F}_{l,q}^\perp + \mathcal{O}(h^k). \quad (4.24)$$

Taking into account (4.24), equation (4.23) is rearranged in the form :

$$\int_{\Gamma_l} \mathcal{F} \cdot \mathbf{n} \, d\Gamma = \sum_{q=1}^{N_q} \omega_q \mathcal{H}_{l,q} |\Gamma_l| + \mathcal{O}(h^{k+1}) + \mathcal{O}(h^{2N_q+1}), \quad (4.25)$$

An extra order of accuracy will be gained if the face integral is summed up for all faces of the shift cell:

$$R_k = \frac{1}{|\Omega_k|} \left(\sum_{l \in \mathcal{I}(\Omega_k)} \sum_{q=1}^{N_q} \omega_q \mathcal{H}_{l,q} |\Gamma_l| + \mathcal{O}(h^{k+2}) + \mathcal{O}(h^{2N_q+2}) \right), \quad (4.26)$$

In 2D $|\Omega_k|$ is considered to be $\mathcal{O}(h^2)$, so that this equation becomes:

$$R_k = \mathcal{R}_k + \mathcal{O}(h^k) + \mathcal{O}(h^{2N_q}), \quad (4.27)$$

with the residual approximation:

$$\mathcal{R}_k = \frac{1}{|\Omega_k|} \sum_{l \in \mathcal{I}(\Omega_k)} \sum_{q=1}^{N_q} \omega_q \mathcal{H}_{l,q} |\Gamma_l|. \quad (4.28)$$

Equation (4.27) shows that for a second-order scheme ($k = 2$) as well as for a third-order scheme ($k = 3$), only one Gauss quadrature point along each face of the shift cell is needed for the first-order residual approximation or second-order approximation. Equation (4.28) finally becomes:

$$\mathcal{R}_k = \frac{1}{|\Omega_k|} \sum_{l \in \mathcal{I}(\Omega_k)} \mathcal{H}_l |\Gamma_l|. \quad (4.29)$$

When computing the non-dissipative flux for the third-order SV-RB scheme, two Gauss quadrature points are needed on each CV face, which means that two distinct RB numerical fluxes should be computed on each boundary face. On a given CV face, the purely centered flux is computed at each quadrature point (g_1 or g_2 in Figure 4.5) but the same numerical dissipation is adopted to reduce the computational cost :

$$\begin{aligned}\mathcal{H}_{g_1}^{RB} &= \frac{1}{2}(\mathcal{F}_{g_1,L}^\perp + \mathcal{F}_{g_1,R}^\perp) - d_g, \\ \mathcal{H}_{g_2}^{RB} &= \frac{1}{2}(\mathcal{F}_{g_2,L}^\perp + \mathcal{F}_{g_2,R}^\perp) - d_g.\end{aligned}\tag{4.30}$$

where it is emphasized d_g is computed once on each CV face, using the flux balance on a shift-cell previously described. This shift cell is constructed in the same way as for the second-order scheme. An example is displayed in Figure 4.5. It is important to make clear that no RB flux is computed on internal CV faces, and only one dissipation flux d_g is computed for each external face of a CV. For instance, in the case of the second-order SV-RB scheme (see Figure 4.4) there are 6 external CV faces, hence 6 distinct evaluations for the dissipation flux on a given SV and 6 distinct evaluations of the non-dissipative centered flux because there are 1 Gauss quadrature point on each CV face. In the case of the third-order SV-RB scheme, because there are 9 external CV faces, 9 distinct evaluations of the dissipation flux and 18 evaluations of the non-dissipative centered flux (since 2 Gauss points on each CV face) are made.

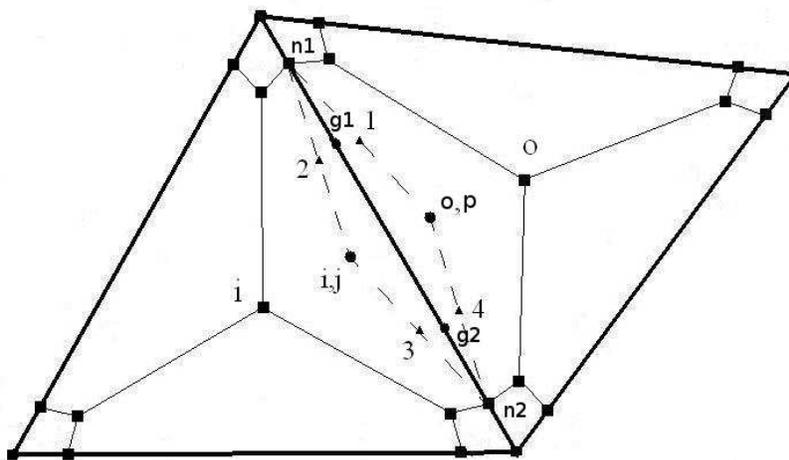


Figure 4.5: Shift cell used by the SV-RB O_3 scheme. For the RB numerical flux computed at Gauss point g_1 and g_2 on a boundary CV face $n_1 - n_2$, only one numerical dissipation is computed by using the shift cell formed by i, j (centroid of j^{th} CV in SV i), o, p (centroid of p^{th} CV in SV o), and two nodes (n_1 and n_2) of this face. Points 1 – 4 are quadrature points used for residual computation in this shift cell.

The reconstruction coefficients corresponding to the face centers of the shift cell, which are specific to the use of the RB numerical flux, are computed from their known coordinates in the simplex. They are computed only once at the very beginning of the calculation and stored for later usage, so as to save on computational time.

4.4 Numerical results

The 2D circular advection problem presented in section 3.2.1 is now considered and computed using the second and third-order SV schemes. For each order of precision, the RB scheme and a representative of conventional upwind schemes, Roe scheme, are used and compared. The series of unstructured triangular meshes already used for the FV computations in Chapter 3 (see Table 3.2 for details) are used for the computation with each scheme. Let us recall that these meshes are made of triangles, with the coarsest mesh, **Tri_irreg1**, containing 226 cells and the finest mesh, **Tri_irreg5**, containing 57518 cells. For each mesh the error between the exact solution and the numerical solution obtained using the SV approach is computed in each CV and the resulting norm is plotted against the characteristic grid size in order to estimate the actual order of accuracy that can be attained with the numerical schemes under study. It is important to point out at this stage the difference between the SV results and the previously obtained FV results. On the finest mesh **Tri_irreg5** for instance, the number of **dof** associated with the FV $O2$ and $O3$ methods are the same, namely the number of grid cells, 57518. If the SV $O2$ schemes are applied on this same grid, since each cell, *i.e.* SV, contains 3 CVs, the number of **dof** amounts to $57518 \times 3 = 172554$. Similarly for $O3$ SV scheme, with 6 CVs in each SV, $14412 \times 6 = 86472$ **dof** is used on this mesh.

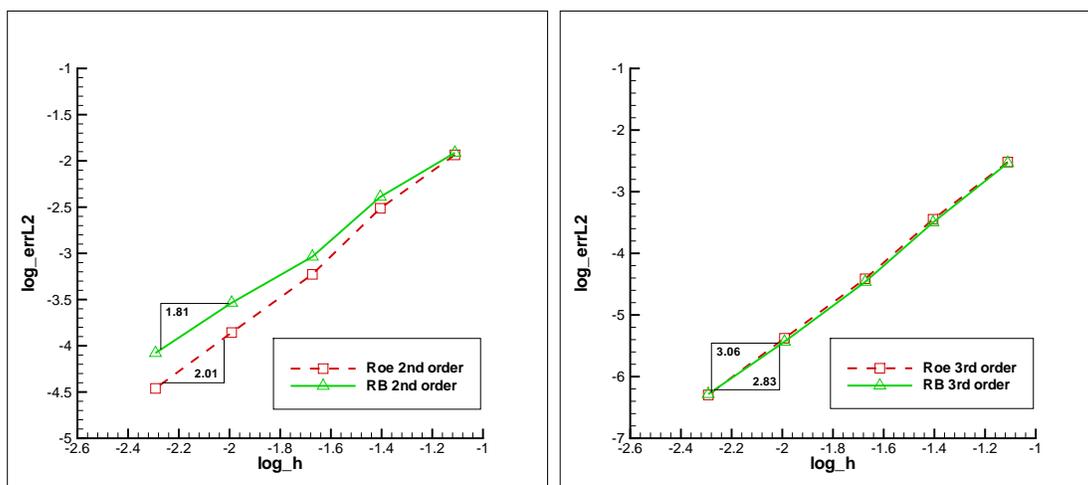


Figure 4.6: The convergence of L_2 norm of numerical error with the mesh typical size h obtained with SV $O2$ schemes (left) and SV $O3$ schemes (right).

Analysis of the SV schemes accuracy The evolution of the error between exact and numerical solutions for the second-order SV schemes (SV-Roe $O2$ and SV-RB $O2$ denoting respectively the SV approach with the Roe and RB numerical flux as numerical flux) and the third-order schemes: SV-Roe $O3$ and SV-RB $O3$ are plotted in Figure 4.6. The analysis of these plots shows that SV-RB $O2$ scheme produces a slightly larger error than the SV-Roe $O2$ scheme for this case, with an order of convergence (1.81) a little lower than the theoretical order 2, while the SV-Roe $O2$ scheme yields the exact theoretical mesh convergence order. For the third-order SV schemes, the difference between SV-RB $O3$ scheme and SV-Roe $O3$ scheme is very small on the numerical error level as well as on the mesh convergence order (close to 3, as expected, in both cases). It is interesting to observe that

the difference between the results obtained with two distinct numerical fluxes is decreasing with the scheme order increasing. In other words, the numerical flux seems to play a role less and less important in the solution accuracy. The reason for this behavior is probably that with the increasing order of accuracy, the number of CV boundary faces - *i.e.* CV faces located on a SV boundary, hence on which the numerical flux is computed - tend to decrease with respect to the total number of CV faces in a SV, thus making less significant the influence of the numerical flux formula on the numerical solution. For instance, each CV has two boundary faces for the second-order SV scheme; there are CVs that have only 1 boundary face for third-order SV schemes and for fourth-order SV schemes, a CV without boundary faces exists (see Figure 4.1).

Analysis of the SV schemes efficiency In the following the computation cost of the SV-RB scheme is compared to the SV-Roe scheme. At first, the results obtained on the finest mesh **Tri_irreg5** are considered. In Table 4.1 CPIPD is the CPU time Per Iteration Per **Dof**. Iterations and CPU time in this table are those needed by the computation to reach the steady state, which is defined as the residual drop of order 10 here. This table shows that the computation with SV-RB *O2* scheme is about 12% more expensive on CPIPD than the SV-Roe *O2* scheme for this case, which is understandable because the RB numerical flux is more complex to compute than the Roe numerical flux. A higher CPIPD is the source which leads to a global higher CPU time used by the RB *O2* scheme to reach the convergence with about the same iterations used by the Roe *O2* scheme. The memory use of both schemes is almost the same. Table 4.2 shows that between different numerical fluxes for the third-order scheme, the RB scheme has a 10% higher CPIPD than the Roe scheme. As more iterations are necessary to reach the computation convergence with the RB *O3* scheme, the total CPU time needed by this scheme is higher than the one of Roe *O3* scheme. Again both schemes use almost the same quantity of memory. By comparing the third-order results with those of second-order scheme, there is no surprise to see that the CPIPD of third-order SV scheme is globally higher than the one of second-order scheme, because there are more numerical fluxes to compute for each CV in the third-order scheme. Precisely, the CPIPD of Roe *O3* scheme is 19% higher than the one of Roe *O2* scheme, and compared to RB *O2* scheme, RB *O3* scheme has a 17% higher CPIPD. As there are more CVs and quadrature points to be stored in the computation, the memory usage of the third-order scheme is 2.2 times higher than the second-order scheme on the same mesh. But it should be noted that the numerical error level achieved by the second-order schemes on the finest mesh **Tri_irreg5** can already be obtained by third-order schemes on a coarse mesh **Tri_irreg3**, and the corresponding computation cost is very low (see Table 4.3). This is to say that to achieve the same precision level, with third-order schemes the computation cost is far less than the one of second-order schemes.

Scheme	Dof	Memory(M)	CPU time(s)	Iterations	CPIPD
Roe <i>O2</i>	172554	167	281	550	2.96×10^{-6}
RB <i>O2</i>	172554	168	332	580	3.32×10^{-6}

Table 4.1: Computation cost comparison for SV *O2* schemes on mesh **Tri_irreg5**, $CFL = 10^6$.

Comparison between FV and SV schemes Since our objective in this last part of the work is to investigate the interest of turning to the SV approach as a general framework for deriving

Scheme	Dof	Memory(M)	CPU time(s)	Iterations	CPIPD
Roe <i>O3</i>	345108	373	1031	850	3.52×10^{-6}
RB <i>O3</i>	345108	375	1246	930	3.88×10^{-6}

Table 4.2: Computation cost comparison for SV *O3* schemes on mesh **Tri_irreg5**, $CFL = 10^6$.

Scheme	Dof	Memory(M)	CPU time(s)
Roe <i>O3</i>	21528	24	21.7
RB <i>O3</i>	21528	24	32.4

Table 4.3: Computation cost comparison for SV *O3* schemes on mesh **Tri_irreg3**, $CFL = 10^6$.

high-order formulations of the RB scheme, we wish to compare SV-RB *O3* with FV-RB *O3* both in terms of accuracy, efficiency and memory requirements. The comparison will be made for a regular triangular mesh, but observations remain valid for the computations performed on other meshes. The mesh used by the FV-RB scheme has 19208 elements and the one used by the SV-RB scheme has 6 times less elements, which is the mesh **Tri_reg3** in Table 3.1 (3200 triangles, see Figure 4.7), so that both schemes will have almost the same **dof** (19208 for FV-RB *O3*, number of cells, and $6 \times 3200 = 19200$ for SV-RB *O3*, number of CVs). The evolution of the numerical error (L_2 norm) against the CPU time obtained by both schemes with the same (large) CFL is showed in Figure 4.8. On this picture, we also plot for reference the evolution of the numerical error obtained on the coarse grid made of 3200 triangles with FV-RB *O3* (in that case the number of **dof** is the number of cells or SVs). Naturally for such a reduced number of **dof** the convergence is fast, the memory requirement is low but the numerical error is large. With this case, we want to emphasize the fact that, in practice, the SV method will be applied on this same coarse grid; the automatic process of SV subdivision will then take place to yield the aforementioned 19200 **dof**. In the case of FV computations, a grid refinement would have to take place in order to increase the number of cells in the grid : this process can also be made automatic but the simplicity of generating CVs within each SV is better.

It is clear that the computation using FV-RB *O3* and SV-RB *O3* with the same number of **dof** yield a converged numerical error in about the same computational time as showed in Table 4.4. Note both schemes are coupled with the same previously described first-order implicit stage (applied at the cell level for the FV approach and at the CV level for the SV approach). In Figure 4.8, the calculation is run for each scheme until the same level of residual is attained (typically 10^{-8}); it can be observed the asymptotic convergence rate for the SV-RB *O3* scheme is poorer than the one achieved with the FV-RB *O3* scheme, thus leading to a larger number of iterations (though we emphasize the steady-state on the numerical error has been prior achieved with both schemes). More interestingly, the level of numerical error provided by the FV-RB *O3* scheme for the same number of **dof** is smaller than the one achieved with SV-RB *O3*. It can be deduced from Table 4.4 the intrinsic cost (CPIPD) of SV-RB *O3* is about 31% higher than the intrinsic cost of FV-RB *O3* but with memory requirements 25% lower. This last positive point is worthy of interest since the memory requirements clearly become a crucial concern when extending the FV approach to higher-order.

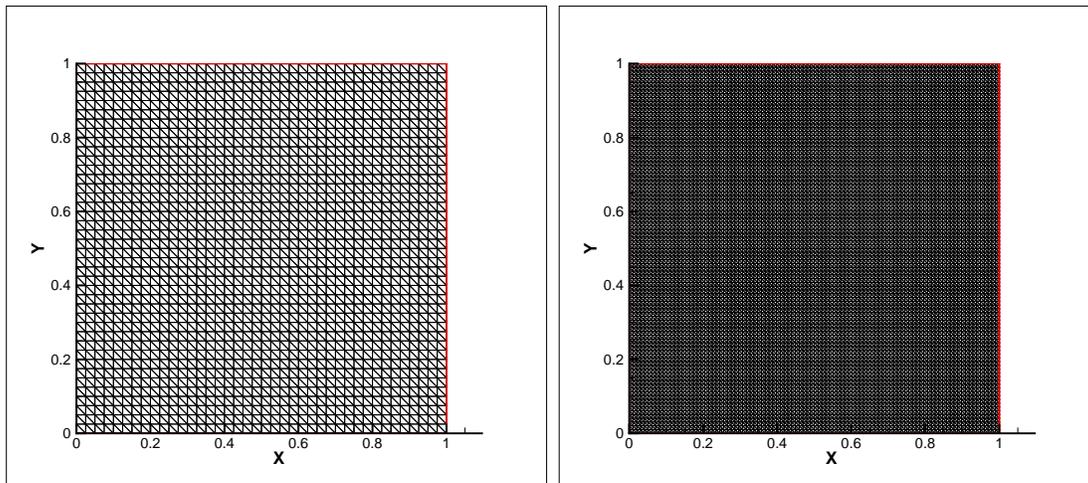


Figure 4.7: Mesh with regular triangles used by the SV-RB scheme (left) and the FV-RB scheme (right).

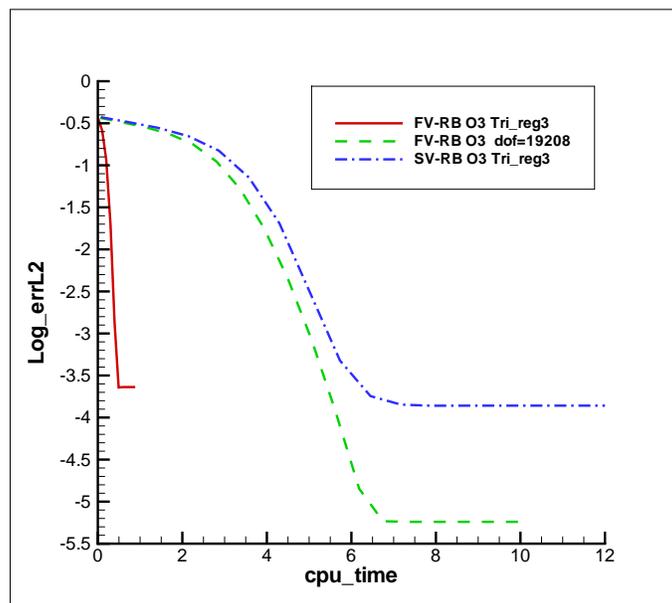


Figure 4.8: Evolution of numerical error with CPU time obtained by the FV-RB $O(3)$ scheme on mesh **Tri_reg3**(3200 dof) and on a regular triangle mesh (19208 dof), by the SV-RB $O(3)$ scheme with the mesh **Tri_reg3**(19200 dof), $CFL = 10^6$.

Scheme	Dof	Memory(M)	CPU time(s)	Iterations	CIPD
SV-RB $O(3)$	19200	21	11	150	3.82×10^{-6}
FV-RB $O(3)$	19208	28	8.4	150	2.92×10^{-6}

Table 4.4: Computation efficiency comparison for third-order RB scheme, $CFL = 10^6$

4.5 Conclusion

The second and third-order SV method based RB scheme are obtained for solving a pure advection problem on unstructured triangular grids. The grid convergence study shows that the RB scheme is no more accurate than the SV method coupled with the numerical flux of a classical approximate Riemann solver. And the computation cost of the SV-RB scheme is higher than the one of the classical numerical flux. When the SV-RB scheme is compared to the FV-RB scheme at third-order with the same **dof** and same implicit scheme, it is seen that the FV-RB converges a little faster and gives a much smaller numerical error.

5

Extension of the SV-schemes to compressible flows

French Le chapitre précédent a été consacré à la description du principe de l'approche SV. Les contributions du présent travail ont aussi été détaillées: développement d'un schéma SV-RB à l'ordre 2 et 3; adjonction d'une phase implicite associée à la phase explicite du système basé sur la discrétisation du SV d'ordre élevé. Le schéma SV-RB a d'abord été développé pour calculer des problèmes d'advection scalaire et une première analyse a été menée sur la précision et l'efficacité des schémas SV pour un problème d'advection circulaire. Dans ce dernier chapitre, on va étendre ces versions du schéma au calcul des écoulements compressibles autour d'un obstacle (par résolution des équations d'Euler). Comme on s'intéresse seulement ici aux problèmes stationnaires, l'extension des schémas scalaires au système non-linéaire des équations d'Euler n'introduit pas de grande difficulté. Cependant on a besoin d'un traitement spécial à la paroi afin que la solution d'ordre élevé ne soit pas dégradée par une représentation imprécise d'une paroi courbe dans un maillage grossier, qui est souvent utilisé avec des schémas SV d'ordre élevé. A la fin de ce dernier chapitre, on fera des comparaisons non seulement entre les schémas SV-RB et les schémas décentré-amont classiques couplés avec la SV, mais aussi entre les schémas SV-RB et les schémas FV-RB développés auparavant. Cette dernière comparaison faite dans ce chapitre et le précédent montrera l'intérêt potentiel de la SV comme base du développement du schéma RB d'ordre élevé.

English The previous chapter has been devoted to the description of the design principles of the SV approach. The specific contributions introduced in this work have also been detailed : derivation of a SV-RB scheme at second and third-order accuracy; development of an implicit stage associated with the high-order SV-based explicit stage. SV-RB scheme has been developed for scalar advection problems. A first analysis of the accuracy and efficiency properties of SV schemes has been performed for the computation of a circular advection problem. In this final chapter, we intend to extend these schemes to the computation of compressible flows over obstacles. Since we will limit ourselves to steady inviscid flows in this chapter, the extension from the scalar linear advection to the non-linear Euler system does not introduce any specific difficulties. The presence of a wall boundary however requires a careful treatment if one wishes to avoid that the whole high-order solution be spoiled by an insufficiently accurate representation of a curved boundary in the a-priori coarse grids which are often used with the high-order SV schemes. At the end of this last chapter we will perform detailed comparisons not only between the SV-RB schemes and other classical upwind schemes coupled with the SV method but also between the SV-RB schemes and the previously developed FV-RB schemes. This last comparison made in last chapter and this chapter will reveal the potential interest of the SV framework for the higher-order RB scheme development.

5.1 Extension of some classical upwind schemes

When reconstructing the solution of the Euler equations, formula (4.9) remains valid but can be applied either to the vector of conservative variables U or to the vector $Q = [\rho \ p \ u \ v]^T$ of so-called primitive variables. The later choice is sometimes favored when compressible flows with shocks is computed by using the FV approach because it allows a better control of the pressure oscillations (the limiting strategy associated with the FV approach is then directly applied upon the pressure). In the case of the SV approach, the first choice is adopted mainly because it is slightly more efficient since it avoids a conversion from conservative to primitive variables and vice-versa.

5.1.1 Rusanov scheme

We have already pointed out in the previous chapter the very simple Rusanov scheme [41] could be retained as the numerical flux for the SV approach, precisely because of its great simplicity hence low intrinsic cost. The expression of the Rusanov numerical flux is given by (4.10). For the 2D Euler equations, the Jacobian matrix in the face normal direction has 4 eigenvalues:

$\lambda^n = [V_n \ V_n \ V_n - a \ V_n + a]^T$, with $V_n = u \cdot n_x + v \cdot n_y$ the velocity in the face normal direction and a the speed of sound. Hence the maximum absolute value of the wave speeds is $\lambda_{max} = |V_n| + a$; its value on the face is computed using typically a Roe-averaged value of the velocity components, density and total enthalpy between the left and right states given at the considered Gauss point by the respective reconstruction polynomials in the SVs on each side of the interface. For smooth flows, a simple arithmetic average can also be used to further reduce the cost of the scheme.

5.1.2 HLLC scheme

In order to check the SV framework can accommodate a large spectrum of numerical fluxes and also to perform a fair comparison between the SV-RB scheme and the coupling of SV with a conventional upwind scheme, we have decided to perform computations with the HLLC scheme, reputed less dissipative than the previous Rusanov scheme while offering a low intrinsic cost. The HLLC scheme used here is precisely the one developed by Toro et al.[45]; its numerical flux is given by:

$$\mathcal{H}_{HLLC} = \begin{cases} \mathcal{F}_L^\perp & \text{if } S_L > 0 \\ \mathcal{F}^\perp(U_L^*) & \text{if } S_L \leq 0 < S_M \\ \mathcal{F}^\perp(U_R^*) & \text{if } S_M \leq 0 \leq S_R \\ \mathcal{F}_R^\perp & \text{if } S_R < 0 \end{cases}, \quad (5.1)$$

where

$$U_{L/R}^* = \begin{pmatrix} \rho_{L/R}^* \\ (\rho u)_{L/R}^* \\ (\rho v)_{L/R}^* \\ (\rho E)_{L/R}^* \end{pmatrix} = C_{L/R} \begin{pmatrix} \rho_{L/R} \Delta V_{L/R} \\ \Delta V_{L/R} (\rho u)_{L/R} + \Delta p_{L/R} n_x \\ \Delta V_{L/R} (\rho v)_{L/R} + \Delta p_{L/R} n_y \\ \Delta V_{L/R} (\rho E)_{L/R} - p_{L/R} (V_n)_L + p^* S_M \end{pmatrix}, \quad (5.2)$$

with $\Delta V_{L/R} = S_{L/R} - V_{L/R}^n$, $\Delta p_{L/R} = p^* - p_{L/R}$, $(V_n)_{L/R}$ the velocity in the interface normal direction. Moreover :

$$C_{L/R} = \frac{1}{S_{L/R} - S_M}, \quad (5.3)$$

$$p^* = \rho_L (V_L^n - S_L)(V_L^n - S_M) + p_L = \rho_R (V_R^n - S_R)(V_R^n - S_M) + p_R, \quad (5.4)$$

$$\mathcal{F}_{L/R}^{\perp*} = \mathcal{F}^{\perp}(U_{L/R}^*) = \begin{pmatrix} \rho_{L/R}^* S_M \\ (\rho u)_{L/R}^* S_M + p^* n_x \\ (\rho v)_{L/R}^* S_M + p^* n_y \\ ((\rho E)_{L/R}^* + p^*) S_M, \end{pmatrix}, \quad (5.5)$$

The signal velocities or wave speeds S_M , S_L and S_R are given by :

$$S_M = \frac{\rho_R V_R^n (S_R - V_R^n) V_L^n - \rho_L V_L^n (S_L - V_L^n) V_R^n + p_L - p_R}{\rho_R (S_R - V_R^n) - \rho_L (S_L - V_L^n)},$$

$$S_L = \min(\lambda_{\min}(U_L), \lambda_{\min}(U_{Roe})), \quad (5.6)$$

$$S_R = \max(\lambda_{\max}(U_{Roe}), \lambda_{\max}(U_R)).$$

with $\lambda_{\min}(U_{Roe})$ and $\lambda_{\max}(U_{Roe})$ the smallest and largest eigenvalue of the normal Jacobian matrix computed with the Roe averaged state between the Left and Right solution vectors. This numerical flux formula is then directly inserted into the SV formulation detailed in the previous chapter, which is computed at each Gauss point of external faces of the CVs.

5.2 Extension of the RB scheme

The numerical flux of the RB scheme for the Euler equations remains computed by equation (4.19) with a formally unchanged expression for the dissipation term, except that the dissipation flux is now the product of a matrix dissipation coefficient by the vector residual. The dissipation matrix is computed as described in Chapter 2 and the residual vector is obtained with equation (4.29). It is therefore especially important to adopt the simplification consisting of computing the dissipation flux only once on each CV face in order to limit the cost of the RB numerical flux with respect to that of the Rusanov or HLLC numerical flux previously described.

5.3 Boundary conditions

5.3.1 General strategy

We will consider only two types of boundaries in the forthcoming external flow applications : far-field boundary and solid wall boundary (slip-wall condition for the inviscid flow under study).

The treatment of the far-field boundary does not differ from what has been briefly described in the FV case : a boundary state is computed on the CV faces located on the boundary, by using characteristic-based formula which depends on the neighboring internal state (computed at the Gauss point on this boundary CV with the solution polynomial in the SV which includes this CV) and the far-field state. The physical normal flux computed with this boundary state is then taken as the numerical flux.

On a solid wall, the physical normal flux formula is applied with an extrapolated state, which is computed using the internal solution polynomial. More precisely, consider a cell i close to wall boundary; the extrapolated solution at the Gauss point on this wall boundary is computed with

$U_g = \sum_j L_{j,g} U_{i,j}$. Taking into account the slip-wall condition, the flux in the normal direction to the boundary face at this Gauss point is given by :

$$\mathcal{F}_w^\perp = [0 \ p_g n_x \ p_g n_y \ 0]^T. \quad (5.7)$$

where $\mathbf{n} = [n_x \ n_y]^T$ is the unit normal vector to the wall mesh face. The pressure p_g is deduced from the reconstructed value of the vector of conserved variables U_g . The face normal is usually computed by supposing the face of the CV is a straight line. As to be demonstrated this choice may be insufficient to ensure an accurate representation of the solution and some extra-care is needed in the treatment of a solid wall-boundary for general curved geometries.

5.3.2 Improved treatment for a curved wall-boundary

It is well known that a precise description of the boundary is very important for high-order methods. Given a curved boundary, unstructured mesh with straight edge cells cannot represent the boundary exactly. This approximation can be improved with the refinement of the mesh. For some high-order methods, like Discontinuous Galerkin (DG) method and spectral volume method, the number of degrees of freedom in each mesh cell increases with the scheme order but the boundary representation remains unchanged since still represented by a series of straight edges. A high-order scheme with a low order boundary representation cannot achieve a truly high-order accurate solution. Unlike the finite volume method which is moderately sensitive to the boundary representation, the DG method [5] and SV method [50] are on the contrary quite sensitive to this representation.

To represent the boundary in a more precise way, one could use iso-parametric elements on the boundary for SV method. But this would lead to different reconstruction coefficients and residual computations for boundary cells with respect to inner cells, which would make the corresponding CFD code rather complicated. Therefore a simpler method has been proposed by Harris [22] for Spectral Volume schemes and used in our computation. Note this idea, referred as high order implementation of wall boundary conditions, has been originally proposed in [29] for Discontinuous Galerkin method. Its general principles are described in the following; it is next coupled with the SV version of the HLLC and RB schemes previously described and applied to the computation of the flow over a cylinder and an airfoil in this work.

The general principles of the improved wall boundary treatment to be implemented and assessed in our computation can be summarized as follows :

- First, a set of interpolation points to be used to represent the boundary face at a prescribed (high) order is determined for each boundary SV.
- Next, the normal vector associated with each curved face constructed from the above interpolation points is computed.
- Finally, a ghost solution vector at each Gauss point on the wall mesh face is computed, by using this normal vector to the curved face and the solution vector in the internal domain. The internal and ghost solution on each side of the Gauss point(s) located on the boundary face are then used to compute a numerical flux through this face. Alternatively, the physical flux (5.7) computed with the physically correct/accurate normal vector could be used; this choice was not retained for the sake of robustness.

We now proceed to give a detailed description of the above steps in the following paragraphs.

5.3.2.1 Interpolation points determination

Let us suppose a curved wall boundary exists in the 2D flow domain. A first-order mesh discretizes it only with some straight lines (see Figure 5.1) leading to some possibly large numerical errors. As the analytic form of the wall boundary is often unknown, we have to find a way to recover the curved boundary from the mesh data. If a quadratic representation of the curved boundary is desired, three interpolation points on the boundary are necessary. For example, in order to represent the real curved face between points 1 and 2 quadratically in Figure 5.1, the coordinates of the middle point 4 has to be known. The idea is to suppose the curved face to be part of a circle. By using nodes of this face (1 and 2) and two neighbor mesh nodes (5 and 6), two circles can be obtained from two combinations of three neighbor points (point 5, 1 and 2 can determine a circle for instance). A new circle is then computed by averaging parameters of these two circles. With the parameters of this new circle and the face nodes coordinates, point 4 can be determined. If the face is located on a boundary corner (point 6 for example), only one neighbor node (point 1) can be found, which gives only one circle that will be directly used to determine the location of point 4.

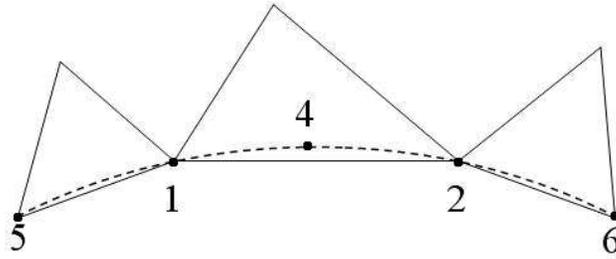


Figure 5.1: **Real curved boundary and straight mesh boundary**

Let us consider a face 1 – 2 in Figure 5.1, with two neighboring mesh nodes: point 5 and 6. Taking the equation of the circle with its center located at (a, b) , and a radius r :

$$(x - a)^2 + (y - b)^2 = r^2. \quad (5.8)$$

Substituting the coordinates of the three points 5, 1 and 2 into this equation, the parameters of the first circle defined by these three points can be computed. The circle centroid coordinates are denoted by (a_1, b_1) , its radius is r_1 . In the same way, the parameters of the second circle defined by points 1, 2 and 6 are obtained: centroid at (a_2, b_2) , with its radius r_2 . We suppose that the point 4 is located on the averaged circle obtained by:

$$a_m = \frac{(a_1 + a_2)}{2}, b_m = \frac{(b_1 + b_2)}{2}, r_m = \frac{(r_1 + r_2)}{2}. \quad (5.9)$$

In order to compute the coordinates of point 4, which is located at the middle of the arc connecting point 1 and 2, the equation of the averaged circle is written in the parametric form :

$$\begin{cases} x = a_m + r_m \cos\theta \\ y = b_m + r_m \sin\theta \end{cases} \quad (5.10)$$

Substitution of the coordinates of point 1 (x_1, y_1) in this equation gives the corresponding angle θ_1 , in the same way θ_2 is obtained with the coordinates of point 2. Obviously the angle of point 4 is $\theta_4 = \frac{(\theta_1 + \theta_2)}{2}$. By using again the equation (5.10) the coordinates of point 4 can be finally obtained.

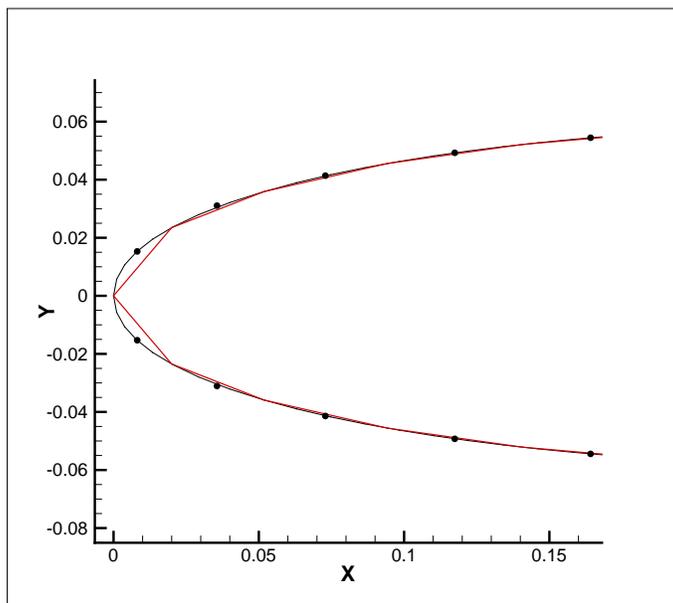


Figure 5.2: Extra interpolation points for a quadratic representation of the NACA0012 airfoil leading edge. Black line is the airfoil obtained on a fine mesh, red line is a baseline coarse mesh on which the interpolation points are obtained.

In Figure 5.2 an example of the quadratic representation of the leading edge of the NACA0012 airfoil is displayed. This figure illustrates the good agreement of the real curved geometry with these interpolation points computed by above described method based on a coarse mesh.

5.3.2.2 Physical normal vector computation

In this next step the objective is to compute the normal vector of the curved face at Gauss points on the wall mesh face. For a general mapping process from a curved element to an isoparametric simplex, the coordinates transformation is expressed by:

$$\mathbf{r} = \sum_{i=1}^n M_i(\xi, \eta) \mathbf{r}_i. \quad (5.11)$$

where n is the number of nodes necessary for a certain isoparametric simplex, the shape function corresponding to each node is $M_i(\xi, \eta)$.

In our computations the cell element has often only one curved edge as showed in Figure 5.3. The

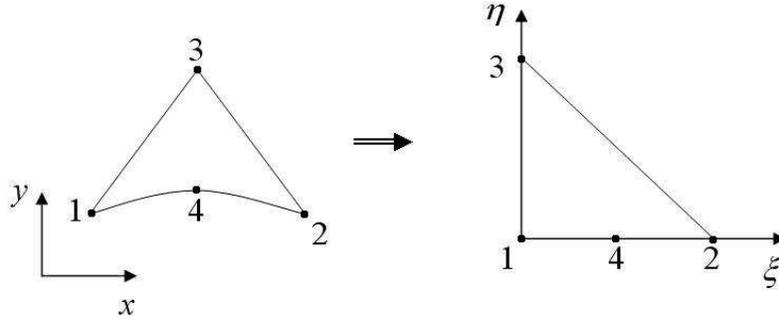


Figure 5.3: Mapping from a curved element to a quadratic simplex

shape function for the simplified quadratic isoparametric simplex is given by:

$$\begin{aligned} M_1 &= 1 - 3\xi + 2\xi(\xi + \eta), \\ M_2 &= -\xi + 2\xi(\xi + \eta), \\ M_3 &= \eta, \\ M_4 &= 4\xi(1 - \xi - \eta). \end{aligned} \quad (5.12)$$

Only Gauss points on face 1 – 2 in the simplex are considered; this face is such that :

$$\eta(\xi) = 0. \quad (5.13)$$

With the expression for shape functions and the mapping equation (5.11), the equation of face 1 – 2 is finally given by:

$$\mathbf{r} = (1 - 3\xi + 2\xi^2)\mathbf{r}_1 + (-\xi + 2\xi^2)\mathbf{r}_2 + (4\xi - 4\xi^2)\mathbf{r}_4. \quad (5.14)$$

It is known that the unit outward normal vector N of a physical face is computed by:

$$N_x = \frac{dy}{dA}, N_y = -\frac{dx}{dA}. \quad (5.15)$$

where:

$$\begin{aligned} dx &= \frac{\partial x}{\partial \xi} d\xi + \frac{\partial x}{\partial \eta} d\eta, \\ dy &= \frac{\partial y}{\partial \xi} d\xi + \frac{\partial y}{\partial \eta} d\eta, \\ dA &= \sqrt{dx^2 + dy^2}. \end{aligned} \quad (5.16)$$

From equation (5.13) we have: $d\eta(\xi) = 0$. Finally the normal vector is computed by:

$$N_x = \frac{\frac{dy}{d\xi}}{dA}, N_y = \frac{-\frac{dx}{d\xi}}{dA}. \quad (5.17)$$

with $dA = \sqrt{\left(\frac{dx}{d\xi}\right)^2 + \left(\frac{dy}{d\xi}\right)^2}$. The derivatives of the coordinates are easily obtained from equation (5.14):

$$\begin{aligned} \frac{dx}{d\xi} &= (-3 + 4\xi)x_1 + (4\xi - 1)x_2 + (4 - 8\xi)x_4, \\ \frac{dy}{d\xi} &= (-3 + 4\xi)y_1 + (4\xi - 1)y_2 + (4 - 8\xi)y_4. \end{aligned} \quad (5.18)$$

where the coordinates of point 1, 2 and 4 are now known. The coordinates of each Gauss point (ξ_g, η_g) on the face 1 – 2 are known in the quadratic simplex, so that the normal vector of the curved face at this point (N_x^g, N_y^g) can be obtained with equation (5.17). Compared to the normal vector of the mesh face, the computed normal vector is referred to as the physical normal vector, because it is an approximated normal vector to the true curved physical face.

5.3.2.3 Numerical flux computation on the curved wall

Since the physical normal vector is computed at each Gauss point, a ghost state can be obtained by using the algorithm proposed by Krivodonova et al [29]. At a Gauss point on the boundary mesh face, a solution vector can be obtained by the internal solution polynomial, and the corresponding primitive variable vector is denoted by : $Q_i = [\rho_i \ p_i \ u_i \ v_i]^T$. The corresponding ghost state is constructed by :

$$\begin{aligned} \rho_g &= \rho_i, \\ u_g &= u_i[(N_y^g)^2 - (N_x^g)^2] - 2N_x^g N_y^g v_i, \\ v_g &= v_i[(N_x^g)^2 - (N_y^g)^2] - 2N_x^g N_y^g u_i, \\ p_g &= p_i. \end{aligned} \tag{5.19}$$

and is such that the average value between the internal and ghost states satisfies the slip-wall boundary condition written with the physical normal vector on this face ($\mathbf{V} \cdot \mathbf{N} = 0$ with the velocity vector $\mathbf{V} = [u \ v]^T$). A classical approximate Riemann solver is finally used to approximate the numerical flux on the boundary: $\mathcal{H}_g = (U_i, U_g, \mathbf{n}_g)$ with \mathbf{n}_g the unit normal vector of the mesh face at the Gauss point g .

It should be pointed out that the RB numerical flux was not used as numerical flux approximating the physical flux on the boundary face because of the extra-complexity will be introduced with the computation of the RB dissipation on a boundary face. Consequently, for a computation with the RB scheme and this high order boundary representation method, the RB numerical flux is used everywhere in the flow domain except on the wall, where the numerical flux of a classical Riemann solver is used instead.

5.4 Numerical assessment

5.4.1 Influence of the numerical flux for SV method

In order to study the convergence order of the SV-RB scheme for solving the Euler equations, a classical test case is considered, namely the subsonic inviscid flow over a cylinder with Mach number 0.38 and zero attack angle [5]. The cylinder radius is 1 and the far-field boundary is located 20 chords away from the cylinder. A fine 128×32 quadrilateral structured mesh is built at first and then each cell is divided diagonally into 2 triangles. Three other triangular meshes are obtained by successively coarsening the fine structured mesh with a factor of 2 and performing the division of the quadrilateral cells. These meshes are displayed in Figure 5.4, and their main features are summarized in Table 5.1.

Assessment of accuracy In the first approach, aimed at demonstrating the need for higher-order boundary representation, a second-order SV method taking the HLLC scheme as numerical flux is

Mesh	No. of cells	No. of points on the cylinder
Mesh1	128	16
Mesh2	512	32
Mesh3	2048	64
Mesh4	8192	128

Table 5.1: Main features of the regular triangle meshes used for computation

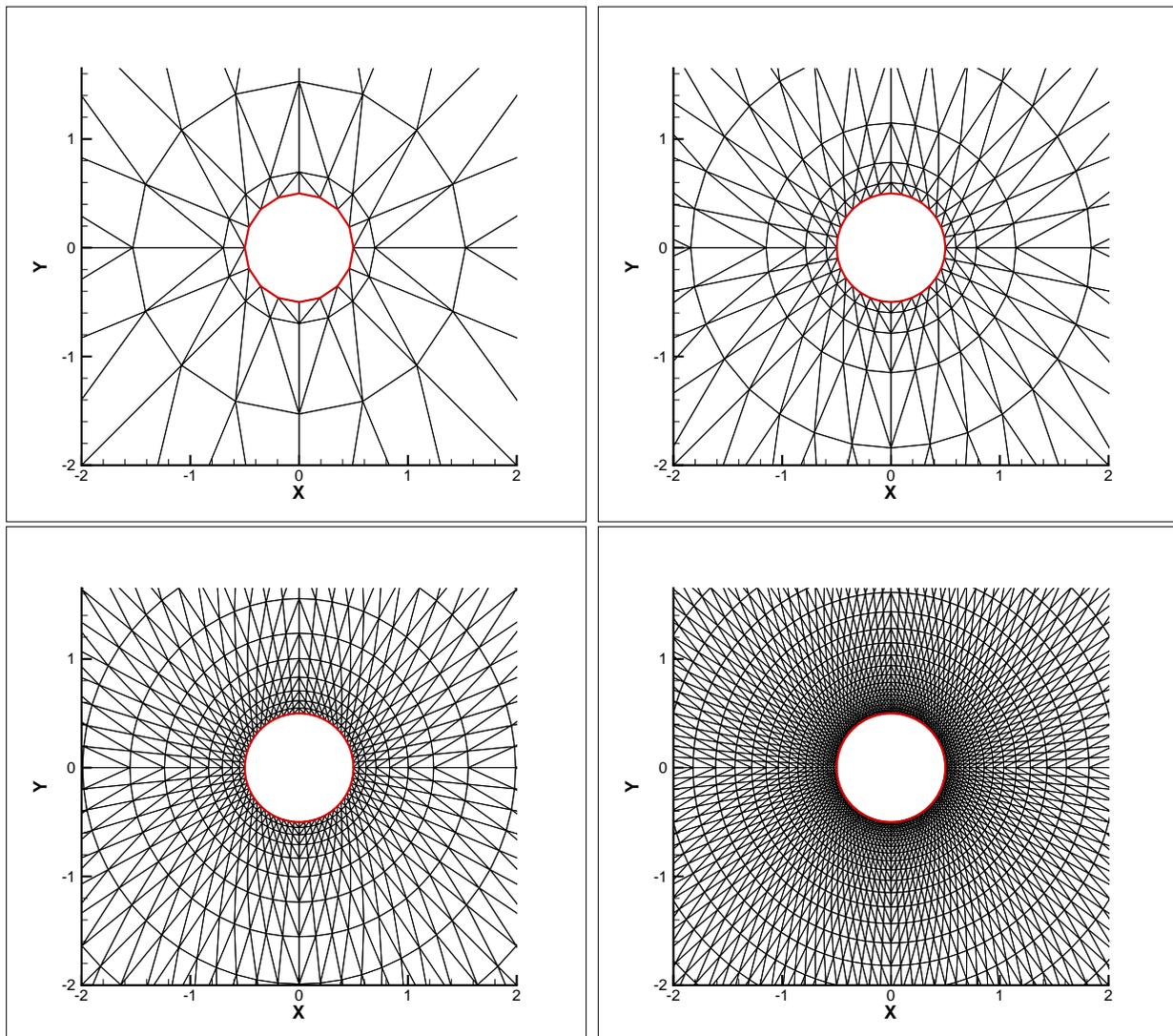


Figure 5.4: Meshes for cylinder case. Mesh1(top left) to Mesh4(bottom right).

used for the computation, with a first-order boundary representation. This numerical strategy will be denoted by SV-HLLC *O2B1* to emphasize the scheme order is 2 but the boundary representation order is 1. Although a relatively fine mesh **Mesh3** is used, it can be observed on the computed Mach contours in Figure 5.5 that the solution is spurious : instead of the expected symmetrical flow (upper/lower and upflow/downflow), two vortex structures appear behind the cylinder, a clear sign of a spurious entropy generation. The reason for this behavior is that the SV scheme is very sensitive to the boundary representation : its solution in that case is strongly influenced by the sharp corners introduced by the first-order grid in the wall boundary representation - turning the smooth cylinder into a polygon. In order to compute a correct solution, the same second-order SV scheme is used with the afore-explained second-order boundary condition - an approach denoted by SV-HLLC *O2B2*. The improvement brought by this improved boundary treatment is dramatic, as can be clearly seen on Figure 5.6 : the Mach contours are now visually almost perfectly symmetric.

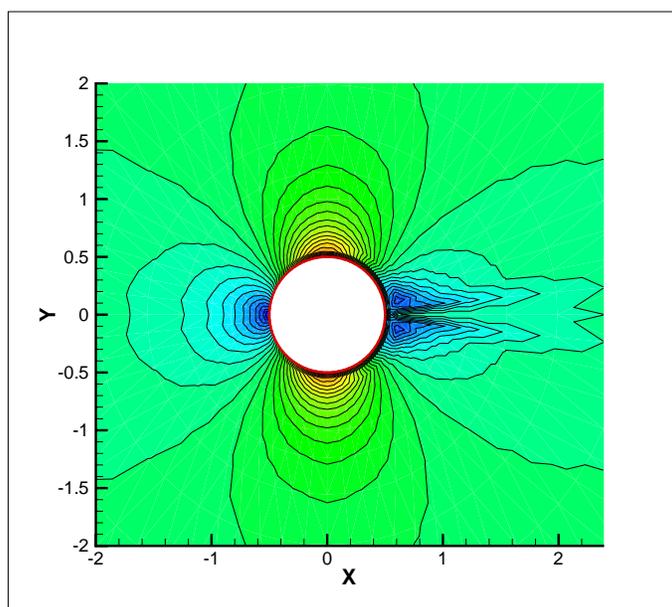


Figure 5.5: **SV-HLLC *O2B1* result.** Computed Mach contours from 0.05 to 0.9 with 30 levels on **Mesh3**.

Since the entropy is theoretically constant for this subsonic inviscid flow, an entropy-based error is defined as:

$$\epsilon_s = \frac{S - S_\infty}{S_\infty}. \quad (5.20)$$

where S_∞ denotes the entropy level in the incoming flow; the entropy is computed as $S = p/\rho^\gamma$. This numerical error on entropy, or entropy deviation, is used to perform a grid convergence analysis on the series of triangular grids previously introduced. Table 5.2 shows that both the second-order SV-HLLC scheme and the second-order SV-RB scheme achieve a super-convergence with a second-order boundary condition since their respective asymptotic order of accuracy exceeds 2. Note also that the levels of entropy errors computed by the SV-HLLC scheme are globally smaller than those obtained with the SV-RB scheme on all the meshes.

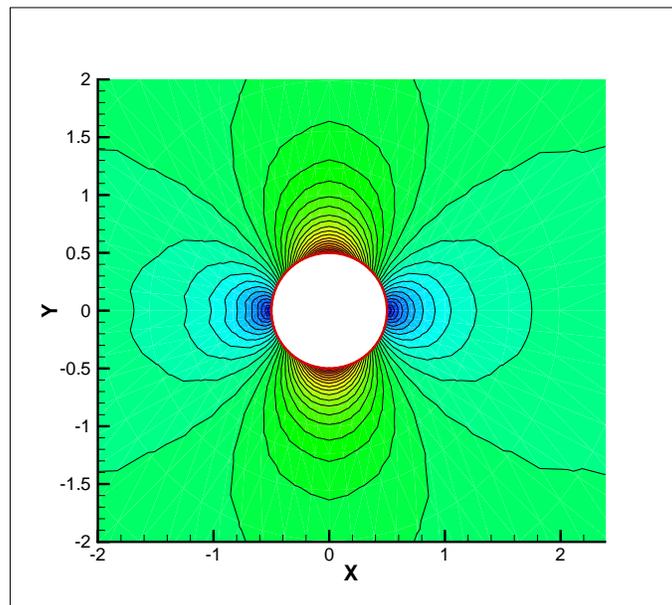


Figure 5.6: **SV-HLLC O2B2 result.** Computed Mach contours from 0.05 to 0.9 with 30 levels on **Mesh3**.

Next, the third-order SV schemes are used to compute this case with the same second-order boundary condition. Results are summarized in Table 5.3. It is found that both third-order schemes yield a convergence order a bit less than 3 (about 2.8), only very slightly better than the one obtained with the second-order SV schemes. It must be emphasized however the entropy error levels achieved by the third-order SV schemes are systematically lower than the ones obtained with the second-order SV schemes. In particular, the error level obtained with SV-RB O3 or SV-HLLC O3 on grid level 3 is a bit lower than the minimum error level achieved by SV-RB O2 and SV-HLLC O2 on the finest grid level 4.

The numerical error computed by SV-RB scheme is systematically a bit larger than the one obtained by the SV-HLLC scheme. This differs from the result obtained on the circular advection problem where the third-order SV-RB scheme provided the same accuracy as the classical upwind scheme. The reason for this worse performance of the third-order RB scheme on the cylinder case could be related to the details of the wall boundary treatment, known to be particularly crucial in the generation of entropy. Let us recall indeed we have not yet developed a specific high-order SV boundary treatment for the RB scheme and have retained instead the HLLC numerical flux for approximating the physical flux on a wall-boundary face. An improper combination of this flux and the RB fluxes on the other external CV faces in the wall SV might explain this error increase with respect to a purely HLLC calculation.

Assessment of efficiency Figure 5.7 and 5.8 display the convergence behavior of the second- and third-order SV schemes (with second-order wall-boundary representation). The convergence is analyzed both on the residual history and on the evolution of the numerical error on entropy. It can be observed that both second-order HLLC and RB schemes achieve a steady state on the numerical

Scheme	Mesh	Dof	L_2 norm ϵ_s (log10)	Order
HLLC	Mesh1	384	-1.39627	-
HLLC	Mesh2	1536	-2.04525	2.27
HLLC	Mesh3	6144	-2.76722	2.46
HLLC	Mesh4	24576	-3.52960	2.56
RB	Mesh1	384	-1.38527	-
RB	Mesh2	1536	-2.00906	2.18
RB	Mesh3	6144	-2.72502	2.22
RB	Mesh4	24576	-3.47125	2.72

Table 5.2: **SV O2B2 results.** L_2 norm of numerical entropy errors and the corresponding grid convergence order.

Scheme	Mesh	Dof	L_2 norm ϵ_s (log10)	Order
HLLC	Mesh1	768	-2.40296	-
HLLC	Mesh2	3072	-3.09008	2.40
HLLC	Mesh3	12288	-3.79667	2.42
HLLC	Mesh4	49152	-4.62274	2.78
RB	Mesh1	768	-2.40948	-
RB	Mesh2	3072	-2.95673	1.91
RB	Mesh3	12288	-3.60784	2.22
RB	Mesh4	49152	-4.42985	2.76

Table 5.3: **SV O3B2 results.** L_2 norm of numerical entropy errors and the corresponding grid convergence order.

error in about 1500 iterations, which corresponds to a residual drop by about 4 orders of magnitude.

Analyzing the third-order results, it is found that the SV-RB *O3* scheme achieves a steady state on the numerical error in less than 4000 iterations, corresponding to 7 orders of magnitude in the residual drop, while the SV-HLLC *O3* scheme achieves a steady state in about 5500 iterations, corresponding to a residual drop by almost 8 orders of magnitude. Note this observation indicates the computational cost of these methods should be compared when the steady state of the numerical error is achieved but not when the same level of residual decrease is achieved.

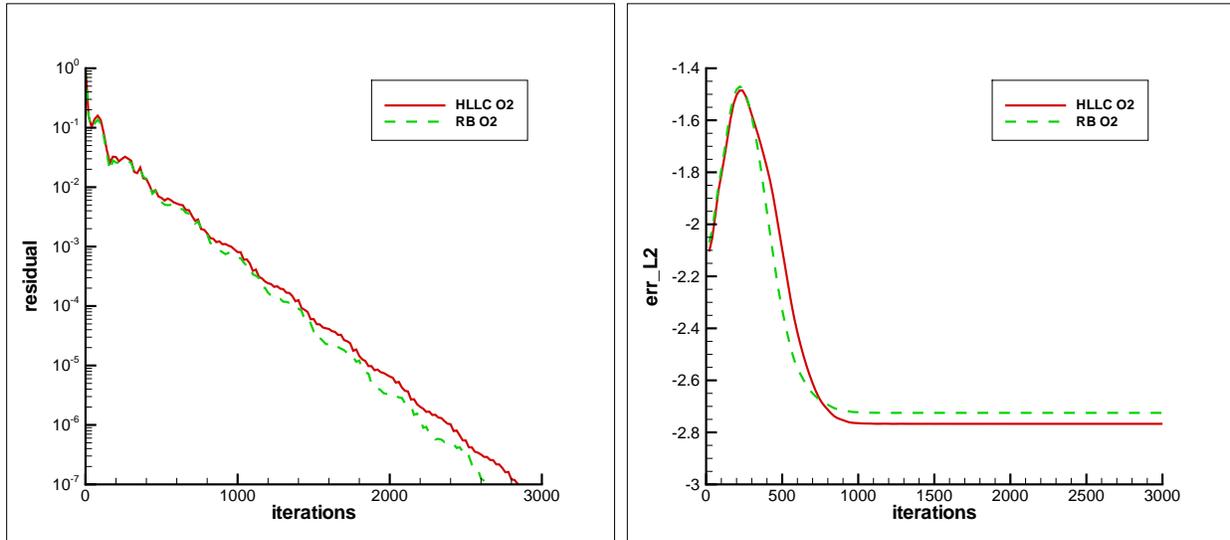


Figure 5.7: **SV *O2B2* results.** Evolution of density residual (left) and L_2 norm of the numerical error (right) with iterations on **Mesh3**.

Next the computational costs of these SV schemes are compared. We emphasize the computational cost of the second-order schemes is analyzed on **Mesh4** while the cost of the third-order schemes is analyzed on **Mesh3** because, doing so, the numerical errors obtained by second- and third-order methods are about the same and it is then fair to compare the cost of second and third-order schemes. It must also be noted that **Iterations** and **CPU time** in Table 5.4 and Table 5.5 are those necessary to reach a steady state on the numerical entropy error. Table 5.4 shows that the second-order SV-RB scheme needs 80 less iterations to reach the computation convergence than the second-order SV-HLLC scheme but it spends finally more CPU time because the CPIPD of SV-RB scheme is about 22% higher. In Table 5.5 it is interesting to see that the third-order SV-RB converged 1960 iterations earlier than the third-order SV-HLLC scheme, which makes the SV-RB *O3* scheme spend eventually less CPU time regardless of its roughly 10% larger CPIPD than the SV-HLLC *O3* scheme. Comparing the CPIPD of second and third-order schemes, HLLC *O3* scheme has a 22% higher CPIPD than the HLLC *O2* scheme, and the CPIPD of RB *O3* scheme is 10% higher than the RB *O2* scheme. It should be noted however that the third-order schemes yield almost the same precision (in fact an even smaller numerical error) than the second-order schemes by using **Mesh3** instead of **Mesh4**, that is with twice less **dof** than the second-order schemes, leading to a globally lower computational cost (both in terms of memory requirements and CPU time). Note the ratio of cell number between **Mesh4** and **Mesh3** is 4, which is that the finest grid contains 4 times more SVs than the previous grid level. Now since the number of CVs in each SV is doubled when going from second to third-order,

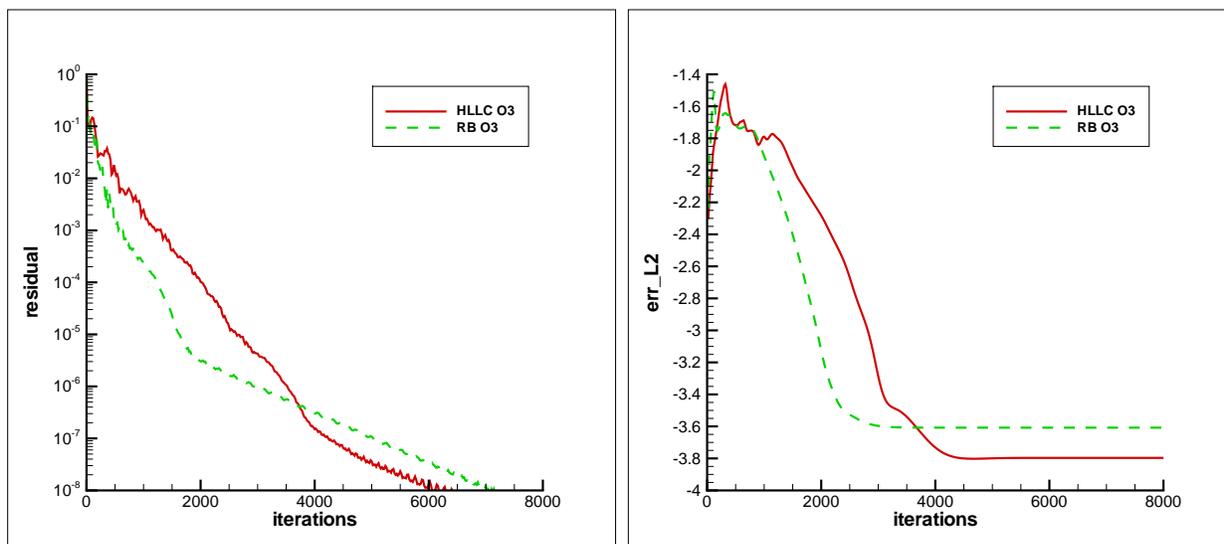


Figure 5.8: **SV O3B2 results**. Evolution of density residual (left) and L_2 norm of the numerical error (right) with iterations on **Mesh3**.

the ratio of **dof** between SV O2 on **Mesh4** and SV O3 on **Mesh3** is eventually 2.

Scheme	Dof	Memory(M)	CPU time(s)	Iterations	CPIP
HLLC	24576	54	1095	4400	1.01×10^{-5}
RB	24576	56	1285	4320	1.21×10^{-5}

Table 5.4: Computational cost comparison for SV O2 schemes on **Mesh4**, $CFL = 10^6$.

Scheme	Dof	Memory(M)	CPU time(s)	Iterations	CPIP
HLLC	12288	31	1007	6640	1.23×10^{-5}
RB	12288	31	764	4680	1.33×10^{-5}

Table 5.5: Computational cost comparison for SV O3 schemes on **Mesh3**, $CFL = 10^6$.

5.4.2 Comparison between SV-RB & FV-RB schemes

Until now, in this chapter, comparisons have been performed between SV based schemes only. It is also interesting to compare RB schemes based on two different methods, namely FV-RB and SV-RB schemes, in order to assess the expectations one may have for one strategy or the other. In order to perform a fair comparison, the same CFL number, same number of **dof** and same mesh structure should be used for the computation. With this idea in mind, a cylinder mesh based on the CVs used by the second-order SV method with the baseline **Mesh3** grid (baseline in the sense it produces the SVs for the SV method, which are subdivided into CVs) can be created (see Figure 5.9). This mesh is then used for the second-order FV-RB scheme computation, with no specific treatment for the wall boundary representation. The combination of SV-RB O2 with second-order boundary representation

is still denoted by SV-RB *O2B2* while the combination of FV-RB *O2* with a first-order or linear boundary representation is referred to as FV-RB *O2B1*. Table 5.6 summarizes the computation results obtained both approaches. It can be seen that both of schemes use the same quantity of memory and has roughly the same intrinsic cost (CPIP). And it must be noted that the standard FV-RB *O2B1* approach yields a faster convergence speed than the more sophisticated SV-RB *O2B2* scheme, while the numerical error obtained with the later is smaller, which must result from the fact that a quadratic boundary condition is adopted by the SV-RB scheme.

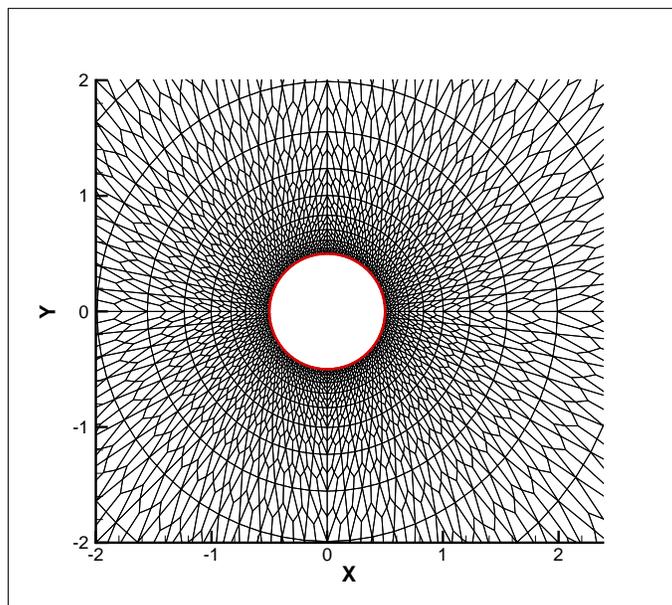


Figure 5.9: **Cylinder case.** The mesh created with CVs of the SV *O2* scheme used on **Mesh3**.

Scheme	L_2 norm of ϵ_s (log10)	Memory(M)	CPU time(s)	Iterations	CPIP
SV-RB <i>O2B2</i>	-2.72502	14	83	1100	1.23×10^{-5}
FV-RB <i>O2B1</i>	-2.09753	14	69	900	1.25×10^{-5}

Table 5.6: Computational cost comparison of FV-RB *O2B1* and SV-RB *O2B2*. Grid of 6144 dof, first-order implicit scheme with $CFL = 10^6$.

In order to perform a comparison between third-order SV and FV approaches which is not over-dominated by the effect of wall-boundary representation, we choose to retain a test-case with less pronounced curvature effects than in the cylinder case. Both FV-RB *O3* with linear boundary representation (thus FV-RB *O3B1*) and SV-RB *O3* with quadratic boundary representation (thus SV-RB *O3B2*) are applied to the computation of the subsonic inviscid flow over the NACA0012 airfoil with inlet Mach number 0.5 and angle of attack 2° . The mesh used for the third-order FV-RB scheme has 6 times more cells than the one used for the SV-RB scheme since the SV-RB scheme makes use of 6 **dof** inside each cell or SV while the FV-RB scheme uses a single **dof** per cell. For each mesh, a structured "O" type mesh is created and then every quadrilateral cell is divided into 2 triangles in the diagonal direction (see Figure 5.10). The mesh for the SV-RB *O3B2* has 90 points on the airfoil and 1800 cells; there are 180 points on the airfoil and 10800 cells in the mesh for the FV-RB *O3B1*.

The far-field boundary is located 50 chords away from the airfoil in both cases.

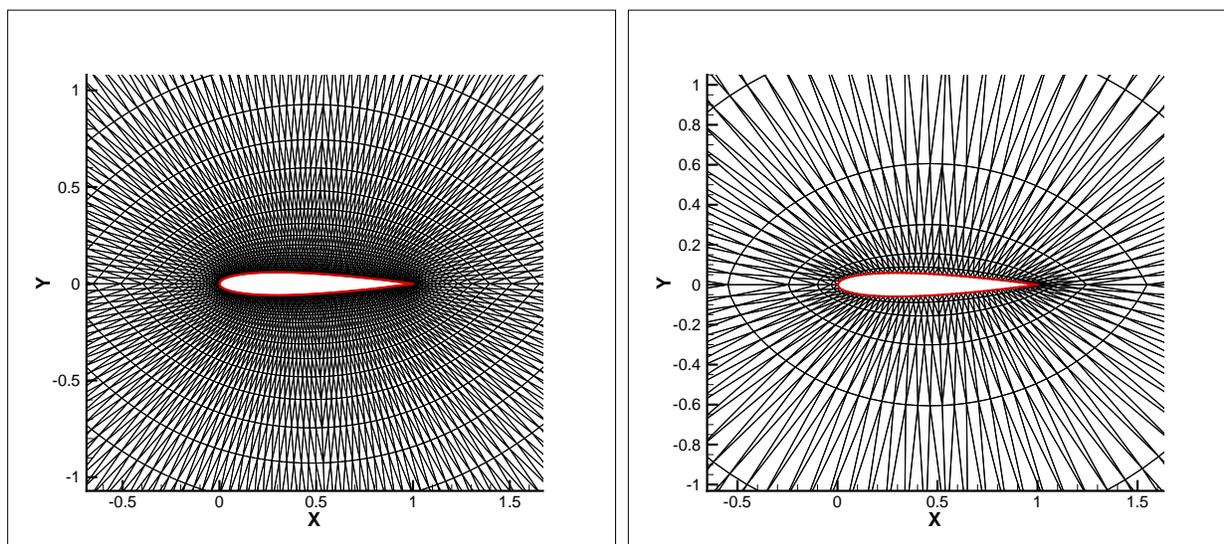
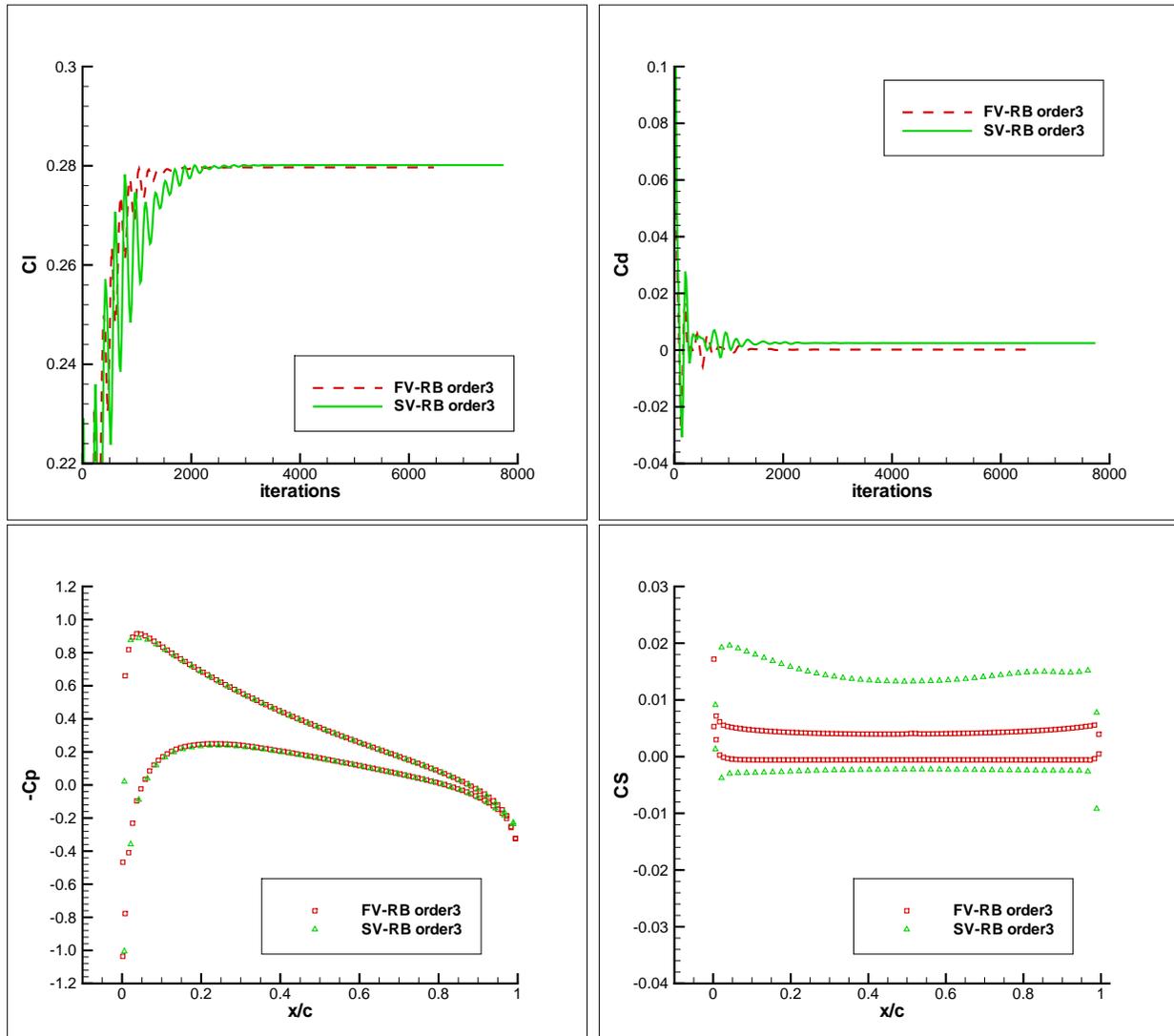


Figure 5.10: Mesh used for FV-RB *O3B1*(left) and SV-RB *O3B2*(right) scheme.

Accuracy assessment The evolution of the lift and drag coefficients computed by the two methods are plotted in Figure 5.11, along with the wall pressure coefficient and wall entropy deviation. Both schemes converge to almost the same value for the lift coefficient - a value which is not yet grid-converged, when compared with the results obtained in Chapter 3 for this same test problem. It can also be observed the FV-RB *O3B1* scheme displays a smoother and a bit faster convergence to a steady value (a more detailed efficiency comparison will be performed below). Regarding the drag coefficient, it must be pointed out that despite a simple (standard) first-order airfoil surface representation the FV-RB *O3B1* manages to yield a lower, hence more accurate, value for the drag coefficient than the SV-RB *O3B2* scheme. Moreover, a faster convergence to a steady value of C_D is also observed for FV-RB *O3B1* with respect to SV-RB *O3B2*. The distributions of the wall-pressure coefficient computed by both schemes are very close. The plot of wall entropy deviation, in line with the result observed on the drag coefficient, indicates a level of entropy error for SV-RB *O3B2* which is roughly twice the error level produced by FV-RB *O3B1*. Finally, the Mach contours obtained by both schemes are displayed in Figure 5.12. It can be observed on this plot the FV-RB *O3B1* scheme yields apparently smoother contours than the SV-RB *O3B2* scheme. In fact, this is only an artificial effect of visualization : in the case of the SV-RB *O3B2* method, the visualization makes use of values at SV nodes while the computed third-order solutions are available at CV centroids. Clearly, our procedure for computing these node values is not yet optimal, as testified by what may appear as a degraded representation of the real high-order solution. Note this is an open problem not only for the SV method but also for other high order methods, like the Discontinuous Galerkin method.

Efficiency assessment We conclude this comparison with a focus on the computational cost of the two schemes. The elements gathered in Table 5.7 indicate that FV-RB *O3B1* is 10% more efficient

Figure 5.11: Results obtained by FV-RB $O3B1$ and SV-RB $O3B2$ scheme.

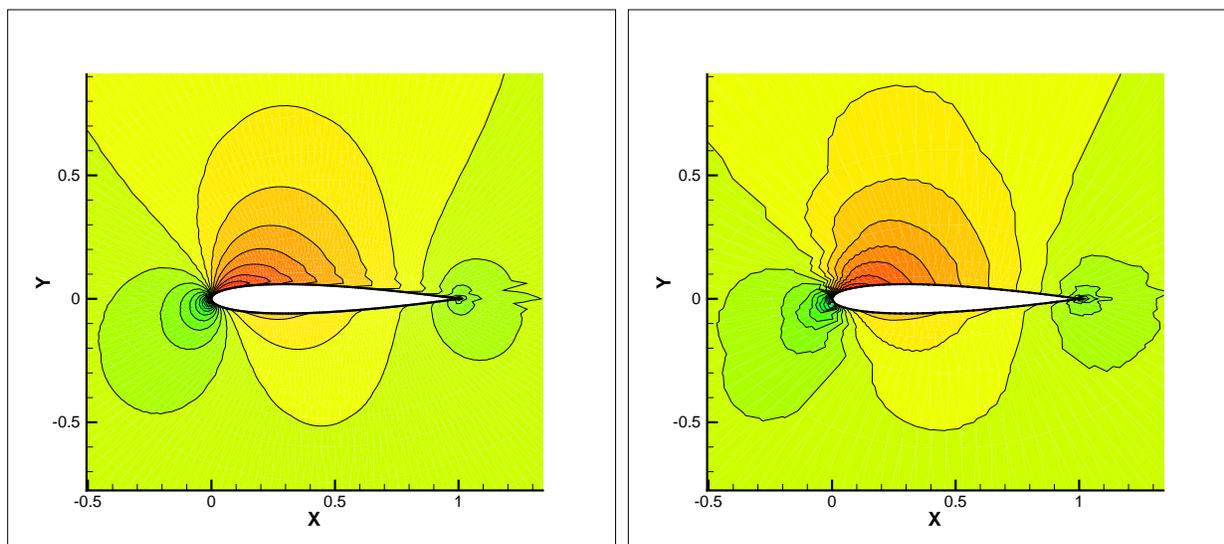


Figure 5.12: Mach contours obtained by FV-RB *O3B1* (left) and SV-RB *O3B2* (right) scheme.

on CIPD and requires 28% less memory than the SV-RB *O3B2* scheme for this case. We insist on the fact that both schemes are used with the simple first-order implicit stage and a large CFL number ($CFL = 10^6$).

scheme	Memory(M)	CPU time(s)	Iterations	CIPD
SV-RB <i>O3B2</i>	28	1099	7740	1.31×10^{-5}
FV-RB <i>O3B1</i>	20	826	6460	1.18×10^{-5}

Table 5.7: Computation cost comparison for RB *O3* schemes, $dof = 10800$, $CFL = 10^6$.

5.5 Conclusion

The second and third-order SV-RB schemes have been successfully extended to solve the Euler equations on unstructured triangular grids. Moreover, in order to deal properly with curved wall boundaries in the flow domain, a simple method for achieving improved (quadratic) representation of these boundaries has also been implemented. The matrix-free implicit method used by the FV based schemes has been extended without difficulty to the SV based schemes, where the sole difficulty is technical, which is managing the connectivity between neighboring CVs involved in the implicit stage. The numerical results obtained for a subsonic inviscid flow over a cylinder and a subsonic inviscid flow over a NACA0012 airfoil have been globally disappointing when the performances of the SV-RB schemes (SV-RB *O2* or SV-RB *O3*) are concerned. Indeed, it was found that the coupling of the SV method with the RB numerical flux does not improve the results obtained by coupling the SV method with a classical approximate Riemann solver such as the HLLC scheme : the SV-RB approach yields a slightly larger numerical error for a slightly higher computational cost, which is in agreement with what was previously observed for the preliminary model circular advection problem. When comparing the SV-RB schemes with FV-RB schemes at the same accuracy order, similar negative conclusions could be drawn : both a smaller numerical error and faster convergence are observed

for the FV-RB approach.

In this paragraph of conclusion, we wish to emphasize how crucial it is to be able to perform such a comparison by using our own developed numerical tools. Indeed, during the ADIGMA project, high-order methods have sometimes (often) been analyzed in a way that we did not judge truly fair. For instance, the last results obtained for the NACA0012 airfoil could be presented in a much more favorable light (for the SV strategy) if we insist on the fact the SV scheme makes use of a mesh that is 6 times coarser than the FV mesh, with twice less points on the airfoil surface. Naturally, such a picture would be biased since what makes sense is a comparison for the same number of **dof** - the type of comparison we have retained. It is true, though, the SV method allows to rely on the fast generation of coarse grids, which are then automatically enriched through the partition of the SVs / cells into CVs. For complex configurations, such a strategy may prove an interesting alternative to the generation of refined grids for FV computations of comparable quality.

We have not yet investigated the extension of the SV approach to orders higher than 3; it may be found that this strategy would demonstrate its advantages for fourth or fifth-order computations which are hardly tractable with the FV approach. The general framework provided by the SV approach makes this extension relatively easy to perform; we provide in the appendix of this work some details on the structure of the FV and SV solvers developed in this work, which might prove useful for future developers of such very high-order versions. Note that before proceeding to these developments on the spatial discretization, we would advise to investigate the implicit time integration associated with the explicit stage. Indeed, the choice of a simple first-order matrix-free implicit stage has proved quite advantageous for the second and even third-order schemes but it is probably not an optimal choice for higher-order schemes since its efficiency tends to decrease when the order difference between the explicit and implicit stage tends to increase. A more strongly implicit strategy could be advantageous in that case, of course more complex for the development.

Conclusion

The objective of this thesis was to derive, develop and analyze a high-order unstructured grid version of the RB scheme, initially developed in the context of Cartesian and curvilinear grid computations. The specific emphasis was on the assessment of the potential interest of a third-order version of the RB scheme for computing compressible flows on general unstructured grids over a more standard second-order version. This type of assessment was actually the general topic of the European project ADIGMA which funded the work. The choice is made to write a thesis report which is totally distinct from the deliverables produced within the ADIGMA project. Although test-cases and grids provided by the project were often used for flow computations, the systematic normalized grid-convergence studies required by the project are avoided in this report. However, this work is focused on a self-designed comparison. Namely, the third-order RB scheme is systematically compared with its second-order version as well as second and third-order schemes based on conventional upwind numerical fluxes (Roe, Rusanov, HLLC), which have been implemented inside the same code.

A natural way to develop a RB scheme on unstructured grids is to rely on the finite volume method. Therefore this path was first explored : second and third-order FV-based RB schemes (FV-RB $O2$ and $O3$) have been developed for solving the steady and unsteady Euler and Navier-Stokes equations on general unstructured grids. The choice has been made to derive these schemes as purely centered schemes with an added residual-based dissipation and to approximate the purely centered part with usual linear or quadratic least-square reconstruction, where the specificity of the RB scheme is condensed in the evaluation of the dissipation flux. The RB schemes as well as the conventional upwind schemes have been systematically coupled with a simple matrix-free implicit stage, allowing to perform computations with large CFL numbers and ensuring in this way fast convergence to a steady-state (be it in physical or dual-time, depending on the steady or unsteady nature of the flow problem). The accuracy, stability and efficiency properties of the RB schemes have then be assessed on a series of test-cases ranging from model test-problems of scalar steady advection in 2D and 3D to inviscid and viscous flows over $2D$ and $3D$ obstacles.

The model test problems with known analytical solutions allow to check the practical order of accuracy offered by the schemes. The main conclusions that can be drawn from these problems have two sides. On the negative side, a more pronounced level of grid-dependency has been observed for FV-RB $O3$ with respect to the other second and third-order schemes (FV-RB $O2$ as well as FV-Roe $O2$ and $O3$). The source of this behavior can be traced back to the calculation of the dissipation flux but could not be more clearly determined. On the bright side, it must be pointed out the gain of using the third-order scheme compared to the second-order scheme in 2D is very encouraging (not only for model problems but also for steady inviscid flows). For instance, to achieve the grid convergence of the drag coefficient for a subsonic flow over an airfoil, the third-order scheme saves 60% cpu time and 23% memory compared to the second-order scheme.

Unfortunately the advantage of using the third-order FV-RB rather than the second-order FV-RB scheme does not hold in 3D, at least for 3D flows in the implementation developed in this work. The interest of the third-order FV-RB scheme over the second-order FV-RB scheme is still established for the model advection problem but for the inviscid flow calculations performed over $3D$ bodies

the quality of the third-order scheme results remains very close to that of the second-order scheme results. Since the cost of FV-RB *O3* is naturally higher than the cost of FV-RB *O2*, this makes FV-RB *O3* unattractive for 3D cases. A possible reason for this behavior could be the very large stencil needed by the third-order FV scheme which may compromise the scheme accuracy on general grids. A bit similarly, the viscous extension of FV-RB *O2* and FV-RB *O3* did yield close results thus making the third-order scheme not really competitive with respect to the second-order version. Note however this behavior could also be explained by the fact that the viscous test-cases correspond to low-Reynolds laminar flows with dominating viscous effects.

When this stage of the work was reached, several options for the next step to take were evaluated : initially a further investigation of the weak points which have been pointed out for FV-RB *O3* was considered. However, it was judged more useful to put in perspective the general FV results obtained : it was clear that going very high-order (more than third-order) with a traditional FV approach, using for instance a cubic reconstruction instead of a quadratic one, was a strategy doomed to fail because of the explosion of the memory requirements which would be encountered, especially in *3D* computations. Therefore it was decided to explore another way to derive high-order schemes that we would assess for third-order computations. Note that it should be possible to extend the approach to higher-order.

After some preliminary attempts with the spectral difference method, We opted for the newly developed spectral volume method [47]. Basically, this approach relies on an increase of the number of available degrees of freedom inside a grid cell rather than an increase of the grid cell support for computing higher-order solution reconstruction within each cell. This more compact design principle keeps the higher-order 3D formulation of the SV schemes manageable. The coupling of the RB numerical flux with the spectral volume approach has been performed, giving birth to the SV-RB *O2* and SV-RB *O3* schemes. Moreover, a SV version of the existing FV matrix-free implicit stage was also developed in order to speed up the convergence of these schemes to a steady-state. The performances of these new schemes have then been assessed with respect to the previously developed FV-RB *O2* and *O3* schemes on one hand and with respect to conventional SV schemes (relying on conventional upwind numerical fluxes such as Rusanov or HLLC) at second and third-order on the other hand.

The possibility to insert the RB numerical flux into the general SV framework was successfully demonstrated for *2D* model problems first. However, a rather disappointing observation was that the numerical flux used for approximating the physical flux at the external faces of a SV seemed less and less influential when increasing the order of the reconstruction polynomial in each SV. In fact, this is not surprising because increasing the order of the SV scheme leads to a decreasing proportion of external faces on which the numerical flux is used, it is recalled that the physical flux through the increasing number of inner faces is directly computed with the reconstructed solution on these faces). When solving the *2D* Euler equations for smooth flows, no benefit was observed from using the SV-RB schemes rather than the SV-HLLC schemes for instance, which agrees with the result of model problem. In summary, the SV-RB schemes not only provide a less accurate solution but also needs a larger computation cost. Moreover, the SV-RB schemes did not prove its competitiveness compared to FV-RB schemes. In spite of these disappointing conclusions, it is believe that the development of the SV version of the RB schemes is truly useful when put in a broader perspective : they may be regarded as a first step for higher-order formulation of the RB scheme, where they might be

competitive since the FV-RB schemes could be hardly developed to very high order because of large memory constraints.

It should be pointed out that apart from the theoretical work on the RB schemes, another important part of this work has been devoted to the development of a CFD code, named NS^3 , in which all the numerical schemes developed in this work have been successfully implemented. The starting point was only a simple unstructured solver for $2D$ inviscid steady flows. This highly technical part was both essential and time-consuming since the unstructured history in the laboratory has started with the present work. The developed NS^3 code can now deal with steady inviscid and laminar flow in $2D$ and $3D$, unsteady inviscid flow in $2D$, subsonic, transonic and hypersonic cases on any type of unstructured grids. This code is also optimized to some extent in order to carry out computations with a reasonable memory and CPU time. For the purpose of evaluating the SV-RB schemes, the code NS^3 was adapted to the SV method and turned into a new version, NS^3_SV (the finite volume version is renamed similarly as NS^3_FV). The NS^3_SV code includes all the features presented in this work, in particular an efficient matrix free treatment for computing steady inviscid flows as well as the method for imposing high order boundary conditions.

Perspectives

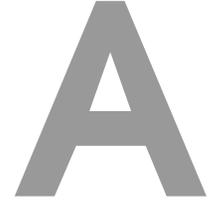
This first contribution to the development of high-order RB schemes on unstructured grids has raised a number of issues which remain to be solved. The advantage of using a third-order FV RB scheme for steady inviscid flows has been clearly demonstrated in $2D$, but this conclusion does not seem holding for $3D$. However, the $3D$ FV computations have been only performed for a subsonic case on a rather unconventional geometry and for a transonic case. To draw more definitive conclusions, it would be interesting to perform complementary tests for $3D$ subsonic flow, for more well-defined problems such as the flow over a sphere for instance. Turbulent test-cases needs also to be systematically performed to complete the $2D$ analysis. A RANS version of the solver, based on the Spalart-Allmaras model, has been developed by another research group working with NS^3 but the result is not ready to perform a proper assessment of the RB schemes for high-Reynolds flows.

It should also be recalled that the FV formulation of the RB schemes is based on an option in which the reconstructed solution is used to compute the non-dissipative part of the RB numerical flux. It would be interesting to investigate more deeply the possibility to use a least-square approach to estimate not only the flux gradient used in the dissipative flux but also the flux on the face, without resorting to a variable reconstruction. Such an approach agrees better with the strategy used by original RBC schemes since it preserves better the compactness of this scheme, while it is lost in the FV-RB schemes presented in this work.

Regarding the SV approach, it would probably be interesting to check whether the impact of the numerical flux is more meaningful when computing flows with shocks, because no real interest was found in using the RB flux over the HLLC flux for instance when computing smooth inviscid flows. More importantly however, performing a $3D$ extension of the SV approach, coupled with the proposed implicit treatment and RB numerical fluxes as well as conventional upwind fluxes should be a priority. Indeed, the developed codes including high-order approximations should provide a good support to perform some studies on large-eddy-simulations on unstructured grids as long as a $3D$ version is

made available. Note that recent results obtained in [17] when coupling the third-order SV method with a classical upwind scheme in $3D$ have revealed that the approach is weakly unstable; further $3D$ investigations are therefore crucial to determinate whether the SV approach is indeed a good candidate for performing flexible LES computations on general geometries.

Appendix



Structure of code NS^3_{FV}

The development of the CFD code NS^3 is a part of work of this thesis. Hence a brief introduction of the code structure is given here. Necessary directories for the code are listed in Figure A.1.

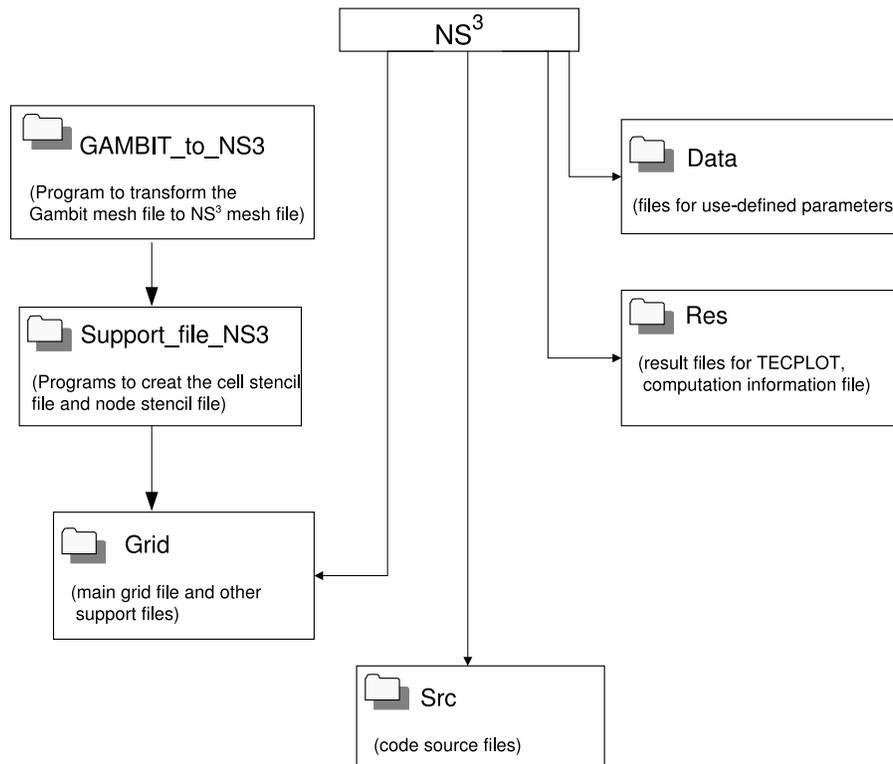


Figure A.1: Directories necessary for code NS^3

Files in each directory and their function are showed in Table A.1, Table A.2 and Table A.3. Files in **Data** are provided and to be modified by the user, files in **Res** will be created after the startup of computation, and files in **Grid** have to be created by using programs in directory **GAMBIT_to_NS3** and **Support_file_NS3**. The structure of directory **GAMBIT_to_NS3** is showed in Figure A.2. For the moment the code NS^3 uses GAMBIT as mesh generator, but the mesh format is different. Firstly the mesh is made inside the GAMBIT, before defining boundary conditions, 'Generic' in solver option is chosen. Then boundaries are defined with type 'Elementside'. Recognized boundary

names are: **WALL** for wall, **RIEMANN** for far-field boundary, **SUPIN** for inlet and **SUPOUT** for outlet. Finally a mesh file ***.neu** can be exported from GAMBIT. This file should be put into directory **GAMBIT**. In file **file_name.inp** the source mesh file name **'*.neu'** and the target file name **'*.inp'** are to be given. According to the mesh dimension (2D or 3D) an executive file can be created from source files in the corresponding **Src_*** directory. At last the target mesh file (***.inp**) will be created in directory **Grid** by using the created executive file.

After having the main mesh file, programs in directory **Support_file_NS3** will be used for support files creation. The structure of this directory is showed in Figure A.3. Copy the main mesh file ***.inp** into directory **Support_file_NS3/Grid** and give the name of main mesh file and its support files in file **Support_file_NS3/Data/file.inp**. According to mesh dimension use source files in directory **Src_*** to creat executive program, which can be used to obtain 3 support files for 2D mesh and 2 for 3D mesh. All of created files can be found in directory **Support_file_NS3/Grid**.

Files	Function
file.inp	provide names of main grid file and support files: main grid file: *.inp; support grid files: *_support_order2.inp, *_support_order3.inp, *_support_node.inp.
userchoice.inp	user-defined parameters: problem type, CFL, scheme order...
freestream.inp	free stream boundary condition: M, α, Re
ede.inp	give fluid type: 'pfg' for the perfect gas
dataDENSI.inp	fluid status constant: γ for the perfect gas

Table A.1: Files in directory **Data**

Files	Function
*.inp	main grid file
*_support_order2.inp	file providing cell stencils for second-order scheme
*_support_order3.inp	file providing cell stencils for third-order scheme
*_support_node.inp	file providing node stencils

Table A.2: Files in directory **Grid**

The code source files in the directory **Src** are listed in Table A.4. It is interesting to see the definition of some important variables in this code.

Files	Function
usr.info	computation information: grid file names, physical model, numerical parameters, CPU time...
residue.dat	evolution of density residual with iterations and CPU time
aero_stress.dat	evolution of aerodynamic coefficients with iterations and CPU time
tecplot_aero_fieldNNN.inp	solution field: ρ , p, T, M...obtained at iteration NNN
wall_coef_NNN.dat	solution distribution on the immersed body obtained at iteration NNN
restart_NNN.dat	conservative variables obtained at iteration NNN, for another computation restarting from the current solution.

Table A.3: Files in directory **Res**

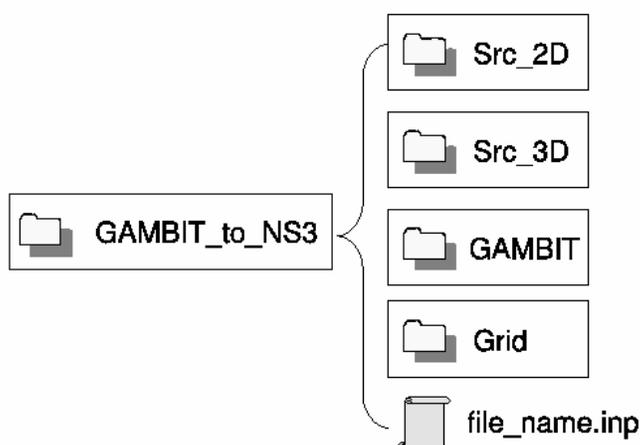


Figure A.2: Structure of directory **GAMBIT_to_NS3**

Files	Function
delaration.f90	declaration of all of variables in the code
allocate.f90	assemble of subroutines allocating variables
functions.f90	all of functions used in the code
ns3_2d.f90	main program
read_input.f90	grid file and support files reading, pre-processing
cell_update.f90	flux integration, solution reconstruction time step and residual computation
gradient_calculation.f90	computation of node values, cell gradients, limiter and error indicator
boundary.f90	boundary condition on the wall, far-field, inlet and outlet
num_flux.f90	numerical flux computation: RB scheme and AUSM+ scheme available
write_output.f90	write computation information file and result files, result files output management
file_operation.f90	open and close some files opened at the computation start and closed in the end of computation
impli.f90	implicit scheme: only matrix-free method available

Table A.4: Files in directory **Src**

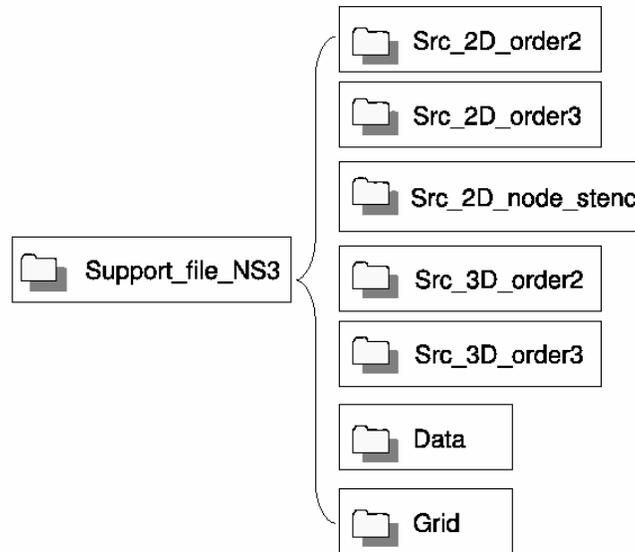


Figure A.3: Structure of directory **Support_file_NS3**

Variables	Meaning
np,nc,nf	number of mesh nodes, cells and faces
uold(i)%rho,%rho _u ,%rho _v ,%rho _E	conservative solution vector at centroid of cell i
qold(i)%p,%u,%v,%T	primitive solution vector at centroid of cell i
q_node(i)%p,%rho,%T,%u,%v	solution vector of node i
xc(i),yc(i)	cell centroid coordinates
surface(i)	cell volume
connection(4,i)	cell connectivity information: nodes belonging to the cell i
x(i),y(i)	mesh node coordinates
node_bc(i)	node type, same definition as face type
incell(i),outcell(i)	cell on the left and right side of the face
sx(i),sy(i)	mesh face area normal vector (pointing from left cell to the right cell)
bc(i)	face type: 0 for internal face, 1 for wall face 2 for inlet face, 3 for outlet face, 4 for far-field face
face(2,i)	face connectivity information: 2 nodes belonging to the face

Table A.5: Definition of some important variables in code NS^3_FV

B

Some important issues of code NS^3_SV

The code NS^3_SV is developed based on the code structure of TNS^3_FV and some adaptation are made in order to implement SV method in this version of code. The directory structure of code NS^3_SV is the same as code NS^3_FV . And this code needs only the main mesh file ***.inp** and the node stencil support file ***_support_node.inp**, which can be created in the same way as for the code NS^3_FV . Necessary source files for the code are the same as those showed in Table A.4. In this code the way of defining some important variables are quite different with the code NS^3_FV (see Table B.1).

Variables	Meaning
np,nc,nf	number of mesh nodes, cells and faces
ncv,nqp	number of CVs and quadrature points in a SV
$L(i)\%dofF(j)$	i th shape function of quadrature point j
$cell(i)\%dof_u(j)\%uold\%rho, \%rho...$	conservative solution vector at centroid of CV j in SV i
$cell(i)\%dof_f(j)\%U\%rho, \%rho...$	conservative solution vector at quadrature point j in SV i
$cell(i)\%sv_xc, \%sv_yc$	centroid coordinates of SV i
$cell(i)\%sv_volume$	SV volume
$cell(i)\%connection(4)$	SV connectivity information: nodes belonging to the SV
$node(i)\%x_n, \%y_n$	coordinates of mesh node i
$node(i)\%bc_n$	node type, same definition as face type
$face(i)\%incell, \%outcell$	cell on the left and right side of the face i
$face(i)\%sx_f, \%sy_f$	mesh face area normal vector (pointing from left cell to the right cell)
$face(i)\%bc$	face type: 0 for internal face, 1 for wall face 2 for inlet face, 3 for outlet face, 4 for far-field face
$face(i)\%nd(2)$	face connectivity information: 2 nodes belonging to the face
$face(i)\%spl(2), \%spr(2)$	CV centroid on the left and right side of the face
$face(i)\%fpl(2), \%fpr(2)$	quadrature points on the left and right side of the face
$face(i)\%svpl(3), \%svpr(3)$	SV partition points on the left and right side of the face
$face(i)\%rbpl(4), \%rbpr(4)$	quadrature points necessary for the RB scheme on the left and right side of the face

Table B.1: Definition of some important variables in code NS^3_SV

In this SV version code, an important issue is the numbering of CV centroid and quadrature points. As the solution polynomial is reconstructed in a SV, it is convenient to number CVs and quadrature

points locally in a SV. The numbering in a standard SV is showed in Figure B.1, where square points are partition points dividing one SV into 3 CVs (for second-order scheme), round points are CV centroids, and triangle points are all of quadrature points in a SV. For the RB scheme, its numerical dissipation term needs solutions at particular quadrature points on faces of shift cells formed between CVs, these points are therefore numbered and saved for later use (see triangle points showed in Figure B.2). Although all of these points in each SV are numbered in a same way in the physical domain, the orientation of each SV is random. Given a mesh face (SV face), the numeration of these points on both sides is therefore unknown, while this information is necessary for numerical flux computation. The numeration finding and saving is done in the pre-processing. An example of numeration for a face i is showed in Figure B.3, the variable saving numeration of CV centroids on the left side: **face(i)%spl(1) = 3**, **face(i)%spl(2) = 2**, on the right side: **face(i)%spr(1) = 1**, **face(i)%spr(2) = 2**; the variable saving numeration of quadrature points on the left side: **face(i)%fpl(1) = 4**, **face(i)%fpl(2) = 3**, on the right side: **face(i)%fpr(1) = 1**, **face(i)%fpr(2) = 2**. Numeration of SV partition points and RB quadrature points are saved in the same way.

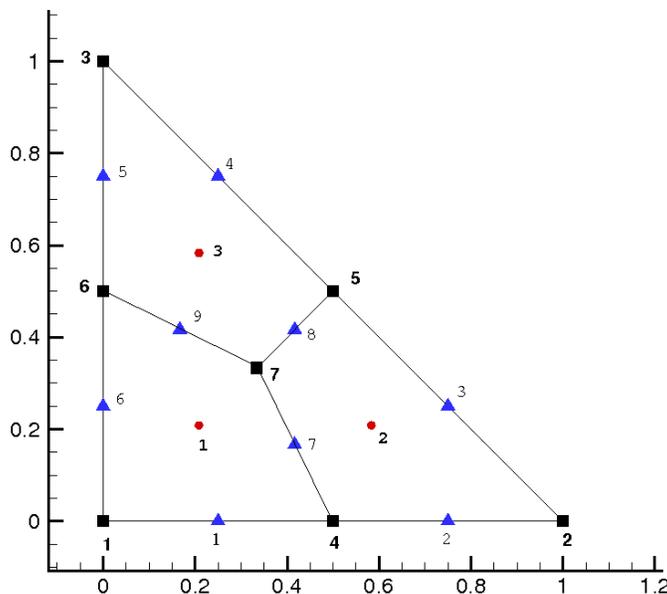


Figure B.1: CV centroids and quadrature points numbering in the standard SV for the second-order scheme.

As mentioned in Chapter 5 the matrix-free implicit method is used for the SV method, where the connectivity of each CV is necessary, which needs a global numbering of centroid, nodes and faces of each CV. This gives a new mesh with CVs as mesh cells. Variables related to the new mesh are listed in Table B.2. With the help of these variables, the solution can be passed easily from a local CV to the corresponding global cell. For example, solutions at the centroid of global cells can be obtained by the loop:

```
do i=1,nc
  do j=1,3
```

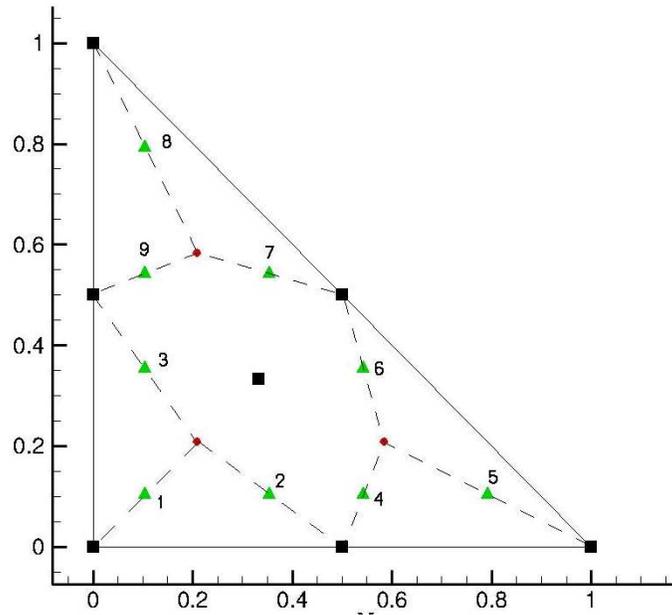


Figure B.2: RB quadrature points numbering in the standard SV for the second-order scheme.

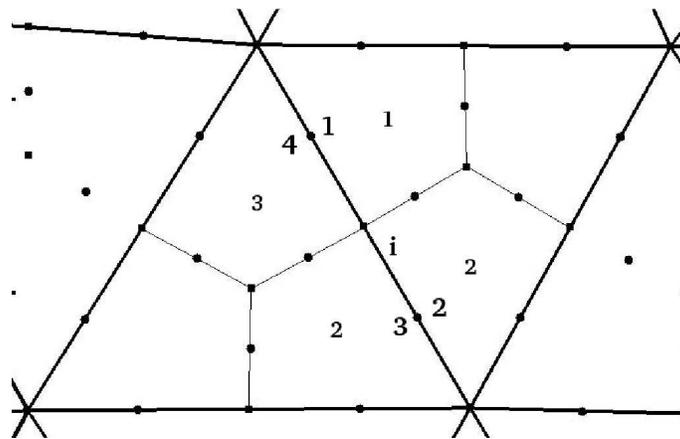


Figure B.3: Numeration saved for a face for the second-order scheme.

```
ncl=cell(i)%dof_u(j)%nwn
nwuold(ncl)=cell(i)%dof_u(j)%uold
end do
end do
```

where **nwuold(ncl)** is the solution at the centroid of global cell *ncl*. With information for the new global cells (CVs of the SV scheme), the implicit solution increment can be computed as the one for the FV scheme.

Variables	Meaning
nnp,nnc,nnf	number of new mesh nodes, cells and faces
nw_cell(i)%xc,%yc	centroid coordinates of new cell <i>i</i>
nw_cell(i)%connection(4)	connectivity information of new cell
nw_node(i)%x_n,%y_n	coordinates of new mesh node <i>i</i>
nw_face(i)%incell, %outcell	cell on the left and right side of new face <i>i</i>
nw_face(i)%sx, %sy	mesh face area normal vector (pointing from left cell to the right cell)
nw_face(i)%nd(2)	new face connectivity information
cell(i)%dof_u(j)%nwn	the global number of CV <i>j</i> in SV <i>i</i>

Table B.2: Definition of variables related to CV global numbering in code NS³_SV

Bibliography

- [1] ADIGMA test case report for BTC0 inviscid case. Technical report, National Aerospace Laboratory, Netherlands, 2 July 2008. [78](#)
- [2] ADIGMA test case report for MTC2 case. Technical report, German Aerospace Center, 25 April 2007. [80](#)
- [3] C. Altmann, G. Gassner, A. Taube, J. Uetzmann, and C. Munz. ADIGMA report: Shock-capturing on unstructured meshes. Technical report, University of Stuttgart, 3 December 2008. [101](#)
- [4] T. Barth. Aspects of unstructured grids and finite-volume solvers for the Euler and Navier-Stokes equations. In *VKI lecture series 1994-05*. Von Karman Institute for Fluid Dynamics, 1994. [22](#), [26](#)
- [5] F. Bassi and S. Rebay. High-order accurate discontinuous finite element solution of the 2D Euler equations. *Journal of Computational Physics*, 138:251–285, 1997. [122](#), [126](#)
- [6] Q.Y. Chen. Partitions for spectral (finite) volume reconstruction in the tetrahedron. *Journal of Scientific Computing*, 29:299–319, 2005. [104](#)
- [7] B. Cockburn, S. Hou, and C.W. Shu. The development of discontinuous Galerkin methods. *Discontinuous Galerkin methods*, 11:3–50, 1999. [11](#)
- [8] P.M. Congedo, C. Corre, and P. Cinnella. Airfoil shape optimisation for transonic flows of Beithe-Zel’Dolovich-Thompson fluids. *AIAA Journal*, 45(6):1303–1316, 2007. [27](#), [28](#)
- [9] C. Corre, A. Beccantini, and T. Kloczko. A residual based compact scheme for all-speed flows on unstructured grids. In Springer Berlin Heidelberg, editor, *Computational Fluid Dynamics 2006*, pages 107–112, 2009. [27](#), [28](#)
- [10] C. Corre and X. Du. A residual-based scheme for computing compressible flows on unstructured grids. *Computers and Fluids*, 38:1338–1347, 2009. [27](#), [28](#)
- [11] C. Corre, F. Falissard, and A. Lerat. High-order residual based compact schemes for compressible inviscid flows. *Computers and Fluids*, 36:1567–1582, 2007. [8](#), [12](#), [21](#), [27](#), [28](#), [52](#)
- [12] C. Corre, G. Hanss, and A. Lerat. A residual based compact scheme for the unsteady compressible Navier-Stokes equations. *Computers and Fluids*, 34:561–580, 2005. [8](#), [12](#), [27](#), [28](#), [52](#)
- [13] C. Corre and A. Lerat. *Stability and efficiency of implicit residual-based compact schemes*, chapter 6 of *Frontiers of Computational Fluid Dynamics 2006*, pages 107–128. World Scientific Publishing Company, 2005. [20](#)
- [14] C. Corre and A. Lerat. High-order residual based compact schemes for advection-diffusion problems. *Computers and Fluids*, 37:505–519, 2008. [8](#), [12](#), [21](#), [27](#), [28](#)
- [15] H. Deconinck, K. Sermeus, and R. Abgrall. Status of multidimensional upwind residual distribution schemes and applications in aeronautics. *AIAA Paper*, pages 2000–2328, 2000. [11](#)

- [16] M. Delanaye. *Polynomial reconstruction finite volume schemes for the compressible Euler and Navier-Stokes equations on unstructured adaptive grids*. PhD thesis, Université de Liège, Belgium, September 1996. [23](#), [26](#), [35](#), [44](#)
- [17] K. Van den Abeele, G. Ghorbaniasl, M. Parsani, and C. Lacor. A stability analysis for the spectral volume method on tetrahedral grids. *Journal of Computational Physics*, 228:257–265, 2009. [142](#)
- [18] K. Van den Abeele and C. Lacor. An accuracy and stability study of the 2D spectral volume method. *Journal of Computational Physics*, 226:1007–1026, 2007. [104](#), [108](#)
- [19] N.T. Frink and S.Z. Pirzadeh. Tetrahedral finite-volume solutions to the Navier-Stokes equations on complex configurations. volume 31, pages 175–187, 1998. [31](#)
- [20] S. Gottlieb and C.W. Shu. Total variation diminishing runge-kutta schemes. *Mathematics of computation*, 67,221:73–85, 1998. [108](#)
- [21] G. Hanss. *Schémas numériques compacts basés sur le résidu en maillage irrégulier pour les équations de Navier-Stokes en compressible*. Thèse de doctorat, ENSAM, France, October 2002. [8](#), [12](#), [27](#), [28](#)
- [22] R. Harris, Z.J. Wang, and Y. Liu. Efficient quadrature-free high-order spectral volume method on unstructured grids: Theory and 2D implementation. *Journal of Computational Physics*, 227:1620–1642, 2008. [122](#)
- [23] D.G. Holmes and S.D. Connell. Solution of the 2D Navier-Stokes equations on unstructured adaptive grids. *AIAA Paper*, 89-1932, 1989. [31](#), [35](#)
- [24] Y. Huang and A. Lerat. Second-order upwinding through a characteristic time-step matrix for compressible flow calculations. *Journal of Computational Physics*, 142:445–472, 1998. [19](#)
- [25] A. Jameson. Time dependent calculations using multigrid with applications to unsteady flows past airfoils and wings. *AIAA Paper*, 91-1596, 1991. [48](#)
- [26] A. Jameson and E. Turkel. Implicit schemes and LU decompositions. *Mathematics of computation*, 37:385–397, 1981. [41](#)
- [27] A. Jameson and S. Yoon. Lower-Upper implicit schemes with multiple grids for the Euler equations. *AIAA Journal*, 25:929–935, 1987. [41](#)
- [28] G.E.K. Karniadakis and S.J. Sherwin. *Spectral/hp element methods for computational fluid dynamics, second edition*. Oxford university press, New York, 2005. [11](#)
- [29] L. Krivodonova and M. Berger. High-order accurate implementation of solid wall boundary conditions in curved geometries. *Journal of Computational Physics*, 211:492–512, 2006. [122](#), [126](#)
- [30] A. Lerat and C. Corre. A residual based compact scheme for compressible Navier-Stokes equations. *Journal of Computational Physics*, 170:642–675, 2001. [8](#), [11](#), [12](#), [21](#), [27](#), [28](#)
- [31] A. Lerat and C. Corre. Residual based compact schemes for multidimensional hyperbolic systems of conservation laws. *Computers and Fluids*, 31:639–661, 2002. [8](#), [12](#), [20](#), [27](#), [28](#)

-
- [32] A. Lerat and C. Corre. High order residual-based compact schemes on structured grids. In *VKI Lecture Series 2006-01*. Von Karman Institute for Fluid Dynamics, 2006. [8](#), [12](#), [21](#)
- [33] A. Lerat, C. Corre, and G. Hanss. *Efficient high-order schemes on non-uniform meshes for multi-dimensional compressible flows*, chapter 6 of *Frontiers of Computational Fluid Dynamics 2002*, pages 89–112. World Scientific Publishing Company, 2002. [8](#), [12](#)
- [34] Y. Liu, M. Vinokur, and Z.J. Wang. Spectral difference method for unstructured grids I: Basic formulation. *Journal of Computational Physics*, 216:780–801, 2006. [11](#), [103](#), [104](#)
- [35] H. Luo, J.D. Baum, and R. Lohner. A fast, matrix-free implicit method for compressible flows on unstructured grids. *Journal of Computational Physics*, 146:664–690, 1998. [41](#)
- [36] H. Luo, J.D. Baum, and R. Lohner. An accurate, fast, matrix-free implicit method for computing unsteady flows on unstructured grids. *Computers and Fluids*, 30:137–159, 2001. [41](#)
- [37] R. Löhner. An adaptive finite element scheme for transient problems in CFD. *Journal of Computational Physics*, 61:323–338, 1987. [44](#)
- [38] B. Michel. *Contribution à la simulation numérique efficace des écoulements dans les prises d’air supersoniques*. Thèse de doctorat, ENSAM, France, June 2004. [8](#), [12](#), [27](#), [28](#)
- [39] A. Neja and C. Ollivier-Gooch. A high-order accurate unstructured finite volume Newton-Krylov algorithm for inviscid compressible flows. *Journal of Computational Physics*, 227:2582–2609, 2008. [44](#)
- [40] M. Parsani, K. Van den Abeele, and C. Lacor. Implicit LU-SGS time integration algorithm for high-order spectral volume method with p-multigrid strategy. Moscow, Russia, November 2007. West-east high speed flow field conference. [109](#)
- [41] V.V. Rusanov. Calculation of interaction of non-steady shock waves with obstacles. *J.Comput.Math.Phys.*, USSR1:267–279, 1961. [120](#)
- [42] B. Saget and C. Stella. Extension d’un schéma basé sur le résidu à des maillages non-structurés. Rapport de Fin d’Etudes 04073, ENSAM, Paris, 2004. [8](#), [12](#), [27](#), [28](#)
- [43] V. Schmitt and F. Charpin. Pressure distribution on the ONERA-M6-Wing at the transonic Mach numbers. *Report of the Fluid Dynamics Working Group 04*, AGARD AR 138, 1979. [81](#)
- [44] D. Sharov and K. Nakahashi. Reordering of 3-D hybrid unstructured grids for vectorized LU-SGS Navier-Stokes computations. *AIAA Paper*, 97-0617, 1997. [41](#)
- [45] E.F. Toro, M. Spruce, and W. Speares. Restoration of the contact surface in the HLL riemann solver. In *Shock waves*, volume 4, pages 25–34. Springer Berlin Heidelberg, 1994. [120](#)
- [46] V. Venkatakrishnan. Convergence to steady-state solutions of the Euler equations on unstructured grids with limiters. *Journal of Computational Physics*, 118:120–130, 1995. [43](#)
- [47] Z.J. Wang. Spectral (finite) volume method for conservation laws on unstructured grids: Basic formulation. *Journal of Computational Physics*, 178:210–251, 2002. [11](#), [103](#), [104](#), [108](#), [140](#)
-

- [48] Z.J. Wang and Y. Liu. Spectral (finite) volume method for conservation laws on unstructured grids II: Extension to two-dimensional scalar equation. *Journal of Computational Physics*, 179:665–697, 2002. [104](#)
- [49] Z.J. Wang and Y. Liu. The spectral difference method for the 2D Euler equations on unstructured grids. *AIAA Paper*, 2005-5112. [103](#), [104](#)
- [50] Z.J. Wang and Y. Liu. Extension of the spectral volume method to high-order boundary representation. *Journal of Computational Physics*, 211:154–178, 2006. [122](#)
- [51] P. Woodward and P. Colella. The numerical simulation of two-dimensional fluid flows with strong shocks. *Journal of Computational Physics*, 54:115–173, 1984. [97](#)

SCHEMA D'ORDRE ELEVE BASE SUR LE RESIDU POUR LA SIMULATION NUMERIQUE D'ÉCOULEMENTS COMPRESSIBLES EN MAILLAGES NON-STRUCTURES

RESUME : Un schéma compact de haute précision basé sur le résidu (RBC) a été développé au laboratoire SINUMEF pour la simulation numérique d'écoulements compressibles en maillages structurés. Certaines propriétés intéressantes font de ce schéma un bon choix pour les calculs d'écoulements compressibles. L'objectif de cette thèse est donc de développer un schéma basé sur le résidu (RB) en maillages non-structurés avec une précision d'ordre élevé. A cette fin, deux approches ont été explorées. La première est basé sur la méthode des volumes finies en non-structuré et conduit à un schéma basé sur le résidu appelé FV-RB. Le seconde approche s'appuie sur une nouvelle formulation spatiale dite volumes spectraux (SV) et mène au schéma SV-RB. Le schéma FV-RB a été développé à l'ordre 2 et 3. Avec cette version du schéma, de nombreux cas tests sont calculés: écoulement d'un fluide parfait et visqueux, subsonique, transonique et hypersonique, stationnaire et instationnaire, en 2D et en 3D. Une analyse de la précision et du coût de calcul est effectuée pour le schéma FV-RB. Dans la seconde approche, un schéma SV-RB est développé à l'ordre 2 et 3 pour résoudre le problème d'advection pure et les équations d'Euler. A travers quelques cas tests, une comparaison de la précision et l'efficacité est effectuée entre le schéma RB et un solveur de Riemman classique, et entre deux formulations du schéma RB développés ici.

Mots clés : schéma basé sur le résidu, ordre élevé, maillages non-structurés, écoulements compressibles

HIGH ORDER RESIDUAL BASED SCHEME FOR NUMERICAL SIMULATION OF COMPRESSIBLE FLOWS ON UNSTRUCTURED GRIDS

ABSTRACT : A very accurate compact residual based scheme (RBC) has been developed in SINUMEF laboratory for numerical simulation of compressible flows on structured grids. Some interesting properties of this scheme makes it a good choice for compressible flow computations. Objective of this thesis is therefore to develop a high-order residual based scheme (RB) on unstructured grids. For this purpose, two approaches have been explored. First one is based on the finite volume method for unstructured grids which gives a residual based scheme named FV-RB. Second approach uses a new spatial discretization method called Spectral Volume method (SV) giving a scheme named SV-RB. The FV-RB scheme is developed to 2nd and 3rd order. With this version of scheme, many test cases are computed: steady and unsteady, subsonic, transonic and hypersonic, inviscid and laminar flow in 2D and 3D. Analysis on the precision and cost of computation is made for this FV-RB scheme. For the second approach, the 2nd and 3rd order SV-RB schemes are obtained for solving pure advection problem and Euler equations. Through several test cases, the comparison of accuracy and computation efficiency is made between the RB scheme and a classical Riemann solver, and between two different versions of RB scheme developed in this work.

Keywords : residual based scheme, high order, unstructured grids, compressible flows