

SPACE-TIME DISCONTINUOUS GALERKIN METHOD
FOR COMPRESSIBLE FLOW

Colophon

The research presented in this thesis was done within the Chair Numerical Analysis and Computational Mechanics, Department of Applied Mathematics, Faculty EWI, University of Twente.

This work was conducted as part of the Technologiestichting STW project TWI.5541, entitled *Advanced simulation techniques for vortex dominated flows in aerodynamics*. The financial support from STW and the National Aerospace Laboratory NLR is gratefully acknowledged.

This work was part of the research program of the J.M. Burgers School for Fluid Dynamics.

This thesis was typeset in \TeX by the author and printed by Wöhrmann Printing Service, Zutphen. The figures were made with either Tecplot, Matlab or Xfig.

All editing was done with GNU-Emacs; the AUCT \TeX and Ref \TeX front ends were used for \TeX editing and `matlab.el` was used for editing Matlab scripts.

All programming was done in Fortran 90. The source code was compiled with either the GNU or Intel Fortran compiler and simulations were executed on regular PC's or on SGI Altix supercomputers.

All computers used the GNU/Linux operating system. Without the availability of free, open source software this work would not have been possible.

© C.M. Klaij, Enschede, 2006.

No part of this work may be reproduced by print, photocopy or any other means without the permission in writing from the author.

ISBN 90-365-2403-2

SPACE-TIME DISCONTINUOUS GALERKIN METHOD
FOR COMPRESSIBLE FLOW

PROEFSCHRIFT

ter verkrijging van
de graad van doctor aan de Universiteit Twente,
op gezag van de rector magnificus,
prof.dr. W.H.M. Zijm,
volgens besluit van het College voor Promoties
in het openbaar te verdedigen
op vrijdag 29 september 2006 om 13.15 uur

door

Christiaan Marijn Klaij
geboren op 3 november 1977
te Hyderabad, India

Dit proefschrift is goedgekeurd door de promotoren
prof.dr.ir. J.J.W. van der Vegt
prof.dr.ir. B.J. Geurts

Preface

During the last four years, I have had the pleasure of doing scientific research; meeting many people along the way and collaborating with some. This thesis could not have been completed without their help and support, for which I am deeply grateful.

I would like to start by thanking Jaap van der Vegt for introducing me to discontinuous Galerkin methods when I was a student. Jaap first supervised me during my Master's, then offered me a PhD position and is now my promotor. His profound insight in the topic proved invaluable and his enthusiasm and optimism kept me motivated. I would also like to thank Harmen van der Ven from the NLR for sharing his knowledge and offering his help and advice, which was indispensable for the development and implementation of the algorithms, first in my own program and later in the NLR program. This work builds on the work by Jaap and Harmen; both have had a great impact on the contents of this thesis.

I would further like to thank Bernard Geurts for his interest and advice. As my copromotor, Bernard was closely involved in the project and his insight in physics, modeling and numerics helped me gain the broader perspective needed to distinguish the (in)essential. He also organized weekly discussions with, among others, Fedderik van der Bos, Arek Kuzaĵ and Sander Rhebergen, whom I would like to thank for their highly beneficial comments.

I would like to thank Marc van Raalte from the Centrum voor Wiskunde en Informatica (CWI) for introducing me to multigrid analysis. His knowledge of the intricacies of multigrid and of Fourier analysis was essential for chapter five. He also helped me being more precise and concise in my writing.

Furthermore, I would like to thank Alexander Bell for his help with the

PETSc package used in chapter four and Enno Oosterhuis for keeping our computers running and for answering my Linux related questions.

Special thanks go to my roommates over the past four years, Michael Tchesnokov, Vijay Ambati, Sena Sopaheluwakan and Dom Sármany for the pleasurable office hours. I would also like to thank the other people in the NACM group for their helpfulness with various issues and for contributing to the friendly atmosphere in our group.

Last but not least, I would like to thank my friends and family for their continuous support and encouragements.

Christiaan Klaij

Enschede, September 2006

Contents

Preface	v
1 Introduction	1
2 Mathematical model	11
2.1 Navier-Stokes equations	11
2.2 Non-dimensionalization	13
3 Space-time method	17
3.1 Space-time discretization	17
3.2 Algebraic system	27
3.3 Numerical simulations	32
4 Pseudo-time integration	43
4.1 Pseudo-time stepping methods	43
4.2 Stability analysis	46
4.3 Convergence results	58
5 Multigrid analysis	67
5.1 Model problem	67
5.2 Fourier analysis	74
5.3 Results	77
6 Multigrid application	87
6.1 Grid-dependent stabilization parameter	87
6.2 Multigrid algorithm	89
6.3 Numerical simulations	94

7	Conclusions and recommendations	103
A	Computational details	107
A.1	Homogeneity tensor	107
A.2	IMEX method	109
A.3	EXV method	112
	Bibliography	115
	Summary	121
	Samenvatting	123
	Curriculum Vitae	125

Chapter 1

Introduction

Flows are common phenomena in our daily live which explains the prominent part fluid dynamics has been playing in the scientific endeavor over the past centuries. The behavior of Newtonian fluids is presently well understood and expressed in a complete set of physical models based on the principle of conservation. This principle states that mass, momentum and energy are conserved, but its simplicity belies the diversity and complexity found in fluid dynamics.

The model of interest in this thesis describes compressible flow with viscous effects and is mathematically formulated by the Navier-Stokes equations. Compressibility is considered a property of the flow: air may be modeled as incompressible at moderate flow speed since compressibility effects such as shocks only occur as the speed approaches the speed of sound. Viscous effects are important near solid surfaces where the fluid adheres to the object, influencing lift and drag. The model is valid for many applications in aerodynamics, however, exact solutions of the equations can only be found in the simplest of cases.

During the past decades, emphasis has shifted from experiments and analytical techniques to the numerical solution of the model equations, made possible by the development of micro-processors. Numerical simulations effectively complement wind tunnel experiments, which are essential in designing aircraft but have their own limitations. The standing shock on the wing of an aircraft in transonic flight, for example, may reflect on the walls of the wind tunnel and disturb the measurements.

To be beneficial in engineering, numerical methods for solving the model equations must be both accurate and efficient. Generally, a numerical method starts with the discretization of the flow domain: the solution is computed on a finite number of grid points around the object instead of on the whole domain. This process induces an error. But, as the computational grid¹ is refined, the error decreases according to the method's order. The discretization error of a second order accurate method, for example, is inversely proportional to the squared number of grid points: tripling the number of grid points reduces the error not by a factor three but by a factor nine. Engineers can thus, in principle, obtain the accuracy needed for their applications.

In practice, the accuracy is limited because, as the number of grid points increases, so does the computational cost. Therefore, the efficiency of the method is important. It is influenced by several factors. In the first place, on the discretized flow domain the numerical method transforms the complicated model equations into a large system of simpler algebraic equations. The size of this system is proportional to the number of grid points. Second, the system of algebraic equations is solved numerically. This procedure is most efficient if the computational effort remains proportional to the number of algebraic equations. The computational efficiency can be further improved by reducing the number of grid points: high densities are only needed in regions of interest, such as boundary layers, shocks or vortices. In other regions, the density of grid points can be reduced, provided the numerical method can handle such local grid refinement.

An additional complication considered in this thesis is the movement and deformation of certain parts of the object, such as the rotor blades on a helicopter or the flaps and slats on a wing. The movement is considered to be known in advance as true fluid-structure interaction is beyond our scope. The computational grid is no longer fixed, instead, it must be constantly adjusted to the movement. Doing so generally deteriorates a numerical method's accuracy.

The challenge, therefore, is threefold: to develop a numerical method for compressible flow which *a)* allows grid movement, *b)* is accurate and *c)* is efficient. We meet this challenge with the space-time discontinuous Galerkin (DG) discretization and a multigrid pseudo-time integration method to solve the system of algebraic equations.

¹also called *mesh*

Space-time DG discretization. The accurate solution of the compressible Navier-Stokes equations on a domain with time-dependent boundaries frequently requires moving and deforming meshes and it is non-trivial to maintain a conservative and accurate scheme on this type of meshes [19, 31]. Finite volume arbitrary Lagrangian Eulerian (ALE) methods, for example, do not automatically satisfy the geometric conservation law on deformed and adapted meshes, which was proven to be essential [31] for the accuracy of the solution. These issues have been the main motivation in [44, 47] to develop a space-time DG finite element method for inviscid compressible flows. This algorithm combines a fully conservative ALE approach to deal with deforming meshes with the well known benefits of the compact stencil of a DG method, such as optimal flexibility for local mesh refinement (h -adaptation), adjustment of the polynomial order in each element (p -adaptation) and excellent performance on parallel computers. The method has been demonstrated for a variety of aerodynamic applications, including rotorcraft [9, 46] and deforming wings [48].

The space-time DG method discussed in [44, 47] has been limited so far to inviscid compressible flow. The first topic of this thesis is extending the space-time DG method to the compressible Navier-Stokes equations, which significantly enlarges its range of applications. The key feature of the space-time DG method discussed here is that no distinction is made between space and time variables and the discretization is directly performed in four-dimensional space. This provides optimal flexibility to deal with time dependent boundaries and deforming elements and naturally yields a conservative discretization, even on deforming locally refined meshes with hanging nodes. The space-time algorithm results in an implicit time-integration method which is unconditionally stable and preserves accuracy also on non-smooth meshes. A complete hp -error and stability analysis of the space-time DG discretization for the linear advection-diffusion equation is given in [41].

DG methods have recently received significant attention and are applied to a wide range of hyperbolic and (incompletely) parabolic problems. For a survey, see [1, 13, 14, 15, 17]. An important step towards a DG discretization for the compressible Navier-Stokes equations was made in the pioneering work of Bassi and Rebay [3] and in a different formulation by Baumann and Oden [8]. These algorithms provide discretization techniques

for the diffusion operator and extend the DG formulation for hyperbolic equations developed by Cockburn and Shu [17] to incompletely parabolic equations. Improvements to the original formulation [3], which showed a weak instability, have been provided in [6] and analyzed in [1, 10]. Applications to the solution of the compressible Reynolds averaged Navier-Stokes equations are discussed in [2, 5, 18, 23]. A slightly different approach to deal with the diffusion operator has been proposed by Cockburn and Shu [16] with the local DG method. Although the various DG formulations for the diffusion operator are quite different, there are no major differences in terms of accuracy, computational cost and complexity between the methods which proved to be consistent, adjoint consistent and of optimal order in the analysis given in [1]. In this thesis, we follow the approach of Brezzi [10] for the diffusion operator and include this technique in the space-time discretization for compressible flows presented in [44, 47], to which we refer for details on the inviscid part of the algorithm.

Pseudo-time integration. When applying DG methods to time dependent equations, the usual approach is to take discontinuous basis-functions in space and a Runge-Kutta method for the time integration, resulting in the so-called RKDG method, see, for example, the survey by Cockburn and Shu [17]. The space-time DG method, however, uses discontinuous basis-functions both in space *and* time and a numerical time-flux ensures causality in time. The method is fully implicit in *physical time* and results in a system of non-linear algebraic equations for each physical time step. Therefore, the overall efficiency of the space-time DG method highly depends on the efficiency with which this system of algebraic equations is solved.

The algebraic system may be solved with a fully implicit method, such as the implicit Euler method [4] with the Generalized Minimal Residual Method (GMRES) to solve the non-linear system or the Newton method [24] for the steady-state equations. The advantage of implicit methods is that the number of iterations needed to solve the algebraic system is significantly reduced in comparison to explicit methods. Their main disadvantage, however, is the increased cost per iteration, linked to the global linear system (based on the Jacobian of the flux) which has to be assembled and solved. The storage of this (sparse) matrix can become problematic in

3D ‘real-life’ cases which typically require a few million elements. Another issue is the small basin of attraction which demands an accurate initial guess in order to converge. In practice, the initial guess is based on the free stream conditions, and special attention is needed to overcome this problem, either by initially using small physical time-steps [4] before increasing the step-size or by grid sequencing [24].

For the space-time discretization of the Euler equations [44], the algebraic system was solved with an explicit *pseudo-time* stepping Runge-Kutta method (with the correction by Melson e.a. [33]). When applied as a smoother in a full approximation multigrid scheme, this approach proved very efficient. The main advantage of the pseudo-time stepping method is its locality, which matches the locality of the DG method and dispenses with the assembly and storage of a global matrix. Also, the pseudo-time stepping method is insensitive to the initial condition and remains stable for large physical time steps. Therefore, we aim at extending the pseudo-time stepping approach in [44] to the space-time DG discretization of the Navier-Stokes equations. This is the second topic addressed in this thesis.

It is not trivial to extend the pseudo-time stepping method to the space-time DG discretization of the compressible Navier-Stokes equations for two reasons. First, for the Euler equations, the pseudo-time Courant-Friedrichs-Levy (CFL) condition must be satisfied for stability of the Runge-Kutta method in [44]. Applying the same method to the space-time discretization of the compressible Navier-Stokes equations requires an additional stability constraint, the diffusive Von Neumann condition, which is more restrictive than the CFL condition in flow regions with small *cell* Reynolds numbers, i.e., boundary layers. Therefore, the Runge-Kutta method will converge much slower. Second, in [44], the Runge-Kutta method is used as a smoother in a full approximation multigrid algorithm. Since the efficiency of the multigrid algorithm greatly depends on the quality of the smoother, we first have to find an effective Runge-Kutta method for viscous flows, to be combined with multigrid later on.

To relieve the viscous stability constraint without losing locality, we turn to the family of explicit Runge-Kutta methods derived by Kleb e.a. [28]. These methods are especially designed for viscous flows and have stability domains which are much more stretched along the negative real axis than the Runge-Kutta methods used for hyperbolic equations. Even though the

diffusive Von Neumann condition still has to be satisfied for stability, it may no longer be prohibitively restrictive. Since in aerodynamic applications, we encounter both inviscid flow (far-field) and viscous flow (boundary layers) in the same simulation, we apply *local* pseudo-time stepping and combine the method developed for the Euler equations in [44], which is optimal in the inviscid regime, with the method presented in [28] for the viscous regime. This is feasible as both methods are local and permissible because accuracy is not an issue in pseudo-time.

We investigate the effectiveness of this combination in relieving the viscous stability constraint by comparison with an implicit-explicit (IMEX) Runge-Kutta method. In the IMEX method, the inviscid part is treated explicitly and the viscous part implicitly, such that only the CFL condition has to be satisfied for stability. In that sense, it represents the ideal situation where viscosity does not affect stability, against which we can test the influence of the diffusive Von Neumann condition on fully explicit methods. Since the implicit-explicit method shares the disadvantage of fully implicit methods (i.e., a global linear system) without sharing the advantage (i.e., a small the number of iterations) it is only recommended for this type of testing.

Multigrid method. To efficiently solve the system of algebraic equations, we aim at multigrid pseudo-time integration, which is the third and final topic addressed in this thesis. We begin by studying the convergence of multigrid iteration for solving a time-dependent advection-diffusion equation that is discretized by the space-time DG method. In this technique we discretize the time variable as a spatial variable and hence decoupling of space and time is avoided. A consequence of such a space-time approach is that the resulting methods can handle problems with moving and/or deforming meshes. Moreover, the space-time approach applied to the incompressible Navier-Stokes equations does not suffer from the second order barrier in time as is the case of the classical pressure-correction methods. In theory, solutions of arbitrary order of accuracy can be computed.

For solving the underlying system of algebraic equations we prefer to rely on multigrid iteration because of its expected efficiency [43, 51]. However, in contrast to the block iterative schemes considered in [21, 25, 27], we study the application of explicit Runge-Kutta smoothing. For computa-

tionally demanding boundary value problems, the use of explicit multigrid methods is preferable because the iterative solution process remains local and does not need large data storage, thus ensuring low computational costs.

To compute the multigrid convergence rates we introduce a similar two-level local mode Fourier analysis as described in [25, 49]. The difference in our analysis is that we avoid the cell-staggering problem of transferring cell data from coarse to fine cells, by associating the data in a cell with a nodal point. The resulting analysis can be used for an arbitrary polynomial basis and is directly extendable to higher-dimensional problems by the tensor product principle [26]. For various cell Péclet numbers we compute multigrid convergence rates and we find that explicit Runge-Kutta smoothing is efficient for solving time dependent advection-diffusion equations.

The multigrid methods analyzed in the context of DG methods [21, 25, 27] use a sequence of nested meshes (h -multigrid) and are based on the embedding of function spaces associated with these meshes. On non-uniform grids where the embedding of spaces does not formally hold, an alternative is the approach followed by Fidkowski e.a. [20], who keep the mesh fixed and use a sequence of different order polynomials (p -multigrid). In [44], h - and p -multigrid are combined to solve the (non-linear) system of algebraic equations arising from the space-time DG discretization of the Euler equations: the discretization is second order on the fine grid and first order on the coarse grids. The advantage of such an approach is that the construction of inter-grid transfer operators is possible without requiring that the function spaces are embedded. Therefore, we wish to extend this approach to the space-time DG discretization of the compressible Navier-Stokes equations.

The space-time DG method for the Euler equations reduces to a first order finite volume method when constant basis functions are used. For the viscous terms in the Navier-Stokes equations, however, a first order DG discretization depends solely on the stabilization term, which (because of the non-uniform grid) becomes grid-dependent. We analyze the multigrid algorithm with linear basis functions on the fine grid and constant basis functions on the coarse grids with two-level Fourier analysis of the advection-diffusion equation. The quality of the predictions obtained with this analysis is then evaluated by comparing with the multigrid convergence results in actual simulations, for the compressible Navier-Stokes equations.

Outline. After describing the mathematical model for compressible flow in Chapter 2, we present the space-time DG discretization of the Navier-Stokes equations in Chapter 3. We show that accuracy is maintained on a moving and deforming grid and that local mesh refinement is a relatively inexpensive way of improving accuracy.

To efficiently solve the system of algebraic equations resulting from the discretization, we propose multigrid pseudo-time integration with explicit Runge-Kutta smoothing. In Chapter 4, we focus on the Runge-Kutta methods and show that a combination of two methods, one for the inviscid part of the flow and one for the viscous part, effectively relieves the viscous stability constraint. The combined Runge-Kutta methods can, in principle, be used as smoother in the h -multigrid context, which we show in Chapter 5 by two-level Fourier analysis of an advection-diffusion equation.

Application of h -multigrid to the compressible Navier-Stokes equations, however, is complicated by the fact that the grid levels are generally not nested. In Chapter 6, we circumvent this problem by limiting ourselves to a first order discretization on the coarse grids, which facilitates the construction of inter-grid transfer operators. The multigrid algorithm with linear basis functions on the fine grid and constant basis functions on the coarse grids is analyzed for the advection-diffusion equation with the technique presented in Chapter 5. We show that the analysis gives good predictions for the compressible Navier-Stokes equations.

Finally, conclusions are drawn in Chapter 7 and computational details needed to implement the methods are given in Appendix A. The contents of this thesis is largely based on the following work:

- C.M. Klaij, J.J.W. van der Vegt and H. van der Ven. Space-time discontinuous Galerkin method for the compressible Navier-Stokes equations. *J. Comput. Phys. (in press)*, 2006.
- C.M. Klaij, J.J.W. van der Vegt and H. van der Ven. Pseudo-time stepping methods for space-time discontinuous Galerkin discretizations of the compressible Navier-Stokes equations. *J. Comput. Phys. (in press)*, 2006.
- M.H. van Raalte and C.M. Klaij. Runge-Kutta multigrid analysis for space-time discontinuous Galerkin discretization of an advection-diffusion equation. *Submitted to SIAM J. Sci. Comput.*, 2006.

- J.J.W. van der Vegt, C.M. Klaij, F. van der Bos and H. van der Ven. Space-time discontinuous Galerkin method for the compressible Navier-Stokes equations on deforming meshes. ECCOMAS CFD 2006.
- H. van der Ven, O.J. Boelens, C.M. Klaij and J.J.W. van der Vegt. Extension of a discontinuous Galerkin finite element method to viscous rotor flow simulations. *Proceedings of the 31th European Rotorcraft Forum*, Florence, 2005.

Chapter 2

Mathematical model

In this chapter, we summarize the mathematical model used for applications in aerodynamics. The model consists of the compressible Navier-Stokes equations with initial and boundary conditions, the equations of state of calorically perfect gas and a set of dimensionless parameters. These parameters describe the flow problem and are obtained through non-dimensionalization of the equations.

2.1 Navier-Stokes equations

The equations of motion considered in this thesis are the Navier-Stokes equations that govern the dynamics of viscous compressible flows. These five partial differential equations form a non-linear system, expressing conservation of mass, momentum and energy. Using the summation convention on repeated indices and the comma notation to denote partial differentiation, the compressible Navier-Stokes equations can be written as:

$$U_{,t} + F_k^e(U)_{,k} - F_k^v(U, \nabla U)_{,k} = 0,$$

with the vector of conservative variables $U \in \mathbb{R}^5$, the inviscid flux $F^e \in \mathbb{R}^{5 \times 3}$ and the viscous flux $F^v \in \mathbb{R}^{5 \times 3}$ given by:

$$U = \begin{bmatrix} \rho \\ \rho u_j \\ \rho E \end{bmatrix}, \quad F_k^e = \begin{bmatrix} \rho u_k \\ \rho u_j u_k + p \delta_{jk} \\ u_k (\rho E + p) \end{bmatrix}, \quad F_k^v = \begin{bmatrix} 0 \\ \tau_{jk} \\ \tau_{jk} u_j - q_k \end{bmatrix},$$

where $j, k = 1, 2, 3$. The conservative variables are the density ρ , the momentum density vector $\rho\vec{u}$ and the total energy density ρE , with \vec{u} the velocity vector and E the total energy. The pressure is denoted by p and δ represents the Kronecker delta function. The total stress tensor τ is defined as:

$$\tau_{jk} = \lambda u_{i,i} \delta_{jk} + \mu(u_{j,k} + u_{k,j}),$$

with $i = 1, 2, 3$ and the dynamic viscosity coefficient μ given by Sutherland's law:

$$\frac{\mu}{\mu_\infty} = \frac{T_\infty + T_S}{T + T_S} \left(\frac{T}{T_\infty} \right)^{3/2},$$

where T is the temperature and T_S a constant. The subscript ∞ denotes free-stream values. The second viscosity coefficient λ relates to μ following the Stokes hypothesis: $3\lambda + 2\mu = 0$. The heat flux vector \vec{q} is defined as:

$$q_k = -\kappa T_{,k},$$

with κ the thermal conductivity coefficient.

In aerodynamical applications such as the flow around aircraft, the initial and far-field boundary conditions are based on uniform flow:

$$\rho = \rho_\infty, \quad \vec{u} = \vec{u}_\infty, \quad p = p_\infty,$$

and, at the body surface, the isothermal no-slip boundary condition is applied:

$$\vec{u} = 0, \quad T = T_\infty.$$

For a calorically perfect gas in thermodynamic equilibrium, the pressure p and internal energy e are given by the equations of state:

$$p = \rho RT, \quad e = c_v T,$$

where $R = c_p - c_v$ is the specific gas constant and c_p and c_v the specific heats at constant pressure and constant volume, respectively. Since the total energy is the sum of the internal and kinetic energy:

$$E = e + \frac{1}{2} u_i u_i,$$

the pressure and temperature can be expressed in terms of the conservative variables as:

$$p = (\gamma - 1) \left(\rho E - \frac{1}{2} \rho u_i u_i \right), \quad T = \frac{1}{c_v} \left(E - \frac{1}{2} u_i u_i \right),$$

where $\gamma = c_p/c_v$ is the ratio of specific heats. These two equations of state close the system. A summary of all physical quantities involved in the compressible Navier-Stokes equations is given in Table 2.1, together with their symbols and physical magnitudes.

We notice that the viscous flux F^v is homogeneous with respect to the gradient of the conservative variables ∇U :

$$F_{ik}^v(U, \nabla U) = A_{ikrs}(U)U_{r,s},$$

where the homogeneity tensor $A \in \mathbb{R}^{5 \times 3 \times 5 \times 3}$ is defined as:

$$A_{ikrs}(U) = \frac{\partial F_{ik}^v(U, \nabla U)}{\partial (U_{r,s})},$$

with $i, r = 1, \dots, 5$ and $k, s = 1, 2, 3$. This tensor is given in Appendix A.1. The homogeneity property will play a crucial role in the derivation of the weak formulation of the compressible Navier-Stokes equations in the next chapter.

2.2 Non-dimensionalization

The magnitudes of the physical quantities involved in the Navier-Stokes equations can be expressed in terms of the four fundamental magnitudes mass $[M]$, length $[L]$, time $[T]$ and temperature $[\theta]$. According to Buckingham's Pi theorem [32], the dimensionless form of the equations is obtained using a recurrent set of four reference values; we choose the set $\{\rho_\infty, a_\infty, T_\infty, L\}$, where a_∞ is the free-stream speed of sound defined as:

$$a_\infty = \sqrt{\frac{\gamma p_\infty}{\rho_\infty}}, \quad (2.1)$$

and L is the characteristic length scale of the problem. The formulae in Table 2.1 give the relation between the dimensional and dimensionless quantities (denoted with a tilde), for example $p = \rho_\infty (a_\infty)^2 \tilde{p}$. Using these formulae, the dimensionless equations can be obtained in the following way. Consider the dimensional mass equation and substitute the dimensional

Table 2.1: Physical quantities in the Navier-Stokes equations, their symbol, magnitude and formula in terms of the recurrent set $\{\rho_\infty, a_\infty, T_\infty, L\}$.

Quantity	Symbol	Magnitude	Formula
mass	m	$[M]$	$\rho_\infty L^3$
length	x	$[L]$	L
time	t	$[T]$	L/a_∞
absolute temperature ^a	T	$[\theta]$	T_∞
density	ρ	$[ML^{-3}]$	ρ_∞
velocity	u	$[LT^{-1}]$	a_∞
total internal energy	E	$[L^2T^{-2}]$	$(a_\infty)^2$
pressure	p	$[ML^{-1}T^{-2}]$	$\rho_\infty (a_\infty)^2$
stress	τ	$[ML^{-1}T^{-2}]$	$\rho_\infty (a_\infty)^2$
dynamic viscosity	μ	$[ML^{-1}T^{-1}]$	$\rho_\infty a_\infty L$
second viscosity	λ	$[ML^{-1}T^{-1}]$	$\rho_\infty a_\infty L$
heat flux	q	$[MT^{-3}]$	$\rho_\infty (a_\infty)^3$
thermal conductivity	κ	$[MLT^{-3}\theta^{-1}]$	$\rho_\infty (a_\infty)^3 L/T_\infty$
specific heats	c_v, c_p	$[L^2T^{-2}\theta^{-1}]$	$(a_\infty)^2/T_\infty$

^a Do not confuse the symbol T for temperature with the magnitude time $[T]$.

quantities by their expression in terms of the reference values and the dimensionless quantities:

$$\begin{aligned} \frac{\partial \rho}{\partial t} + \frac{\partial(\rho u_k)}{\partial x_k} = 0 &\Leftrightarrow \frac{\partial(\rho_\infty \tilde{\rho})}{\partial(L\tilde{t}/a_\infty)} + \frac{\partial(\rho_\infty a_\infty \tilde{\rho} \tilde{u}_k)}{\partial(L\tilde{x}_k)} = 0, \\ &\Leftrightarrow \frac{\rho_\infty a_\infty}{L} \left(\frac{\partial \tilde{\rho}}{\partial \tilde{t}} + \frac{\partial(\tilde{\rho} \tilde{u}_k)}{\partial \tilde{x}_k} \right) = 0. \end{aligned}$$

In the same way, the j -th dimensionless momentum equation and the dimensionless energy equation are obtained:

$$\frac{\rho_\infty (a_\infty)^2}{L} \left(\frac{\partial(\tilde{\rho} \tilde{u}_j)}{\partial \tilde{t}} + \frac{\partial(\tilde{\rho} \tilde{u}_j \tilde{u}_k + \tilde{p} \delta_{jk})}{\partial \tilde{x}_k} - \frac{\partial \tilde{\tau}_{jk}}{\partial \tilde{x}_k} \right) = 0,$$

and

$$\frac{\rho_\infty (a_\infty)^3}{L} \left(\frac{\partial(\tilde{\rho} \tilde{E})}{\partial \tilde{t}} + \frac{\partial((\tilde{\rho} \tilde{E} + \tilde{p}) \tilde{u}_k)}{\partial \tilde{x}_k} - \frac{\partial(\tilde{\tau}_{jk} \tilde{u}_j - \tilde{q}_k)}{\partial \tilde{x}_k} \right) = 0.$$

Note that the form of the dimensionless equations is identical to the dimensional equations, therefore the tilde notation is omitted from here on.

For aerodynamical applications, the following eleven parameters define the flow problem [29]:

- the characteristic length scale L ,
- the free-stream density ρ_∞ , velocity¹ \vec{u}_∞ and temperature T_∞ ,
- the gas constant R and the specific heat at constant pressure c_p ,
- the free-stream dynamic viscosity μ_∞ and thermal conductivity κ_∞ ,
- the constant T_S in Sutherland's law.

Since the equations contain four fundamental magnitudes, seven dimensionless Pi groups can be formed according to Buckingham's Pi theorem. These are:

- the angles of attack α and β ,
- the free-stream Mach number $M_\infty = u_\infty/a_\infty$,
- the ratio of specific heats $\gamma = c_p/c_v$,
- the free-stream Reynolds number $\text{Re}_\infty = \rho_\infty u_\infty L/\mu_\infty$,
- the Prandtl number $\text{Pr} = c_p \mu_\infty/\kappa_\infty$,
- the ratio $\theta_S = T_S/T_\infty$ in Sutherland's law.

For the simulations presented in this thesis, the angles of attack α and β as well as the free-stream Mach and Reynolds numbers M_∞ and Re_∞ are given as input² of the simulation. We use the constants $\gamma = 1.4$, $\theta_S = 0.4$ and $\text{Pr} = 0.72$ and, by definition of our non-dimensionalization, $\rho_\infty = 1$, $a_\infty = 1$, $T_\infty = 1$ and $L = 1$. The values of the other parameters needed in the simulation are calculated using their expression in terms of the Pi groups and the recurrent set $\{\rho_\infty, a_\infty, T_\infty, L\}$ which gives:

$$u_\infty = M_\infty a_\infty, \quad \vec{u}_\infty = [\cos \alpha \sin \phi \quad \sin \beta \quad \sin \alpha]^T u_\infty,$$

¹The three components of the vector \vec{u}_∞ are counted separately.

²In all simulations $\beta = 0$ was used.

with $\phi = \arcsin(\sin \beta / \cos \alpha)$,

$$R = \frac{p_\infty}{\rho_\infty T_\infty}, \quad c_p = \frac{\gamma}{\gamma - 1} R$$

with the pressure from the definition of the speed of sound (2.1) and

$$\mu_\infty = \frac{\rho_\infty u_\infty L}{\text{Re}_\infty}, \quad \kappa_\infty = \frac{c_p \mu_\infty}{\text{Pr}}, \quad T_S = \theta_S T_\infty.$$

The mathematical model is now well defined and, in the next chapter, we focus on the numerical solution method for the equations.

Chapter 3

Space-time method

This chapter covers the space-time discontinuous Galerkin discretization of the compressible Navier-Stokes equations. We first define the geometry of the space-time domain, then the necessary function spaces and operators. This setting is used to define the weak formulation and a crucial part is the discussion of the space-time numerical fluxes. The proper definition of these fluxes allows the transformation of the space-time formulation into an arbitrary Lagrangian Eulerian formulation which combines well with upwind schemes based on approximate Riemann solvers. Finally, we derive the non-linear system of algebraic equations for the expansion coefficients of the solution in each element and demonstrate the method with several test cases.

3.1 Space-time discretization

3.1.1 Geometry of the space-time domain

The space-time discontinuous Galerkin finite element method does not distinguish between space and time variables; instead the equations are considered in an open domain $\mathcal{E} \subset \mathbb{R}^4$, where a point with position $\bar{x} = (x_1, x_2, x_3)$ at time $t = x_0$ has Cartesian coordinates (x_0, \bar{x}) . At time t , the flow domain $\Omega(t)$ is defined as $\Omega(t) := \{\bar{x} \in \mathbb{R}^3 : (t, \bar{x}) \in \mathcal{E}\}$. Let t_0 and T be the initial and final time of the evolution of the flow domain, then the space-time domain boundary $\partial\mathcal{E}$ consists of the hypersurfaces

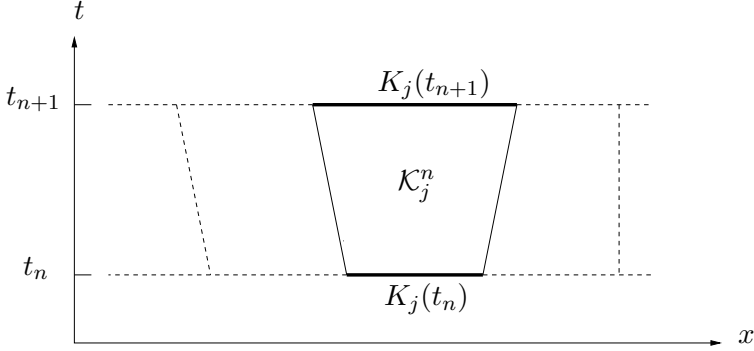


Figure 3.1: The spatial element $K_j(t_n)$ moves and deforms into $K_j(t_{n+1})$. The space-time element \mathcal{K}_j^n is constructed by linear interpolation in time.

$\Omega(t_0) := \{x \in \mathcal{E} : x_0 = t_0\}$, $\Omega(T) := \{x \in \mathcal{E} : x_0 = T\}$ and $\mathcal{Q} := \{x \in \partial\mathcal{E} : t_0 < x_0 < T\}$.

First, consider the partitioning of the time interval $[t_0, T]$ by an ordered series of time levels $t_0 < t_1 < \dots < T$. The space-time domain \mathcal{E} is divided into N_t space-time slabs $\mathcal{E}^n = \mathcal{E} \cap I_n$, with $I_n = (t_n, t_{n+1})$ the n -th time interval. Each space-time slab \mathcal{E}^n is bounded by $\Omega(t_n)$, $\Omega(t_{n+1})$ and $\mathcal{Q}^n = \partial\mathcal{E}^n / (\Omega(t_n) \cup \Omega(t_{n+1}))$.

Second, consider an approximation $\Omega_h(t_n)$ of $\Omega(t_n)$ and divide $\Omega_h(t_n)$ into N_n non-overlapping hexahedral spatial elements $K_j(t_n)$, where $\Omega_h(t) \rightarrow \Omega(t)$ as $h \rightarrow 0$, with h the radius of the smallest sphere completely containing each element $K_j(t_n)$. Similarly, $\Omega_h(t_{n+1})$ approximates $\Omega(t_{n+1})$. Each element K^n is related to the master element $\hat{K} = (-1, 1)^3$ through the mapping F_K^n :

$$F_K^n : \hat{K} \rightarrow K^n : \bar{\xi} \mapsto \bar{x} = \sum_{i=1}^8 x_i(K^n) \chi_i(\bar{\xi}),$$

with x_i the spatial coordinates of the vertices of the hexahedron K^n and χ_i the usual tri-linear finite element shape functions for hexahedra. The space-time elements \mathcal{K}_j^n of \mathcal{E}^n are constructed by connecting $K_j(t_n)$ with $K_j(t_{n+1})$ using linear interpolation in time, which results in the mapping

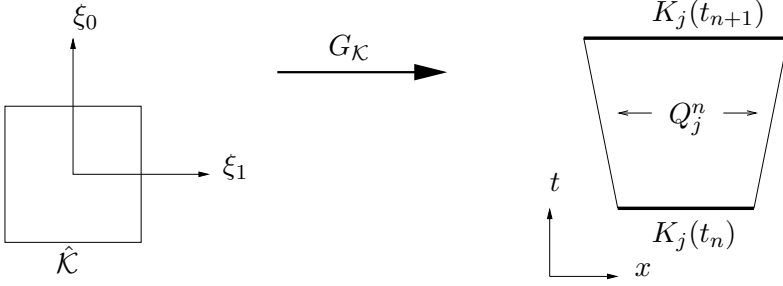


Figure 3.2: Mapping $G_{\mathcal{K}}$ between the master element $\hat{\mathcal{K}}$ and the space-time element \mathcal{K}_j^n with boundaries $\partial\mathcal{K}_j^n = K_j(t_n^+) \cup K_j(t_{n+1}^-) \cup Q_j^n$.

$G_{\mathcal{K}}$ from the master element $\hat{\mathcal{K}} = (-1, 1)^4$ to the space-time element \mathcal{K}^n :

$$G_{\mathcal{K}}^n : \hat{\mathcal{K}} \rightarrow \mathcal{K}^n : \xi \mapsto (t, \bar{x}) = \left(\frac{1}{2}(t_{n+1} + t_n) + \frac{1}{2}(t_{n+1} - t_n)\xi_0, \right. \\ \left. \frac{1}{2}(1 - \xi_0)F_K^n(\bar{\xi}) + \frac{1}{2}(1 + \xi_0)F_K^{n+1}(\bar{\xi}) \right).$$

The tessellation \mathcal{T}_h^n of the space-time slab \mathcal{E}_h^n consists of all space-time elements \mathcal{K}_j^n , thus the tessellation \mathcal{T}_h of the discrete flow domain \mathcal{E}_h is simply $\mathcal{T}_h = \cup_{n=0}^{N_t-1} \mathcal{T}_h^n$.

Finally, consider the element boundary $\partial\mathcal{K}$ which is the union of open faces of \mathcal{K}_j^n and consists of three parts: $K_j(t_n^+) = \lim_{\epsilon \downarrow 0} K_j(t_n + \epsilon)$, $K_j(t_{n+1}^-) = \lim_{\epsilon \downarrow 0} K_j(t_{n+1} - \epsilon)$ and $Q_j^n = \partial\mathcal{K}_j^n / (K_j(t_n^+) \cup K_j(t_{n+1}^-))$. The space-time normal vector at an element boundary point moving with velocity \vec{v} is given by:

$$n = \begin{cases} (1, 0, 0, 0) & \text{at } K(t_{n+1}^-), \\ (-1, 0, 0, 0) & \text{at } K(t_n^+), \\ (-v_k \bar{n}_k, \bar{n}) & \text{at } Q^n, \end{cases} \quad (3.1)$$

with \bar{n} the spatial outward normal vector. It is often convenient to consider the faces separately instead of the whole element boundary. Therefore, in addition to the previously defined faces $K_j(t_n^+)$ and $K_j(t_{n+1}^-)$, we also define interior and boundary faces as follows. A face \mathcal{S} is an interior face if it is shared by two neighboring elements \mathcal{K}_i^n and \mathcal{K}_j^n , such that $\mathcal{S} = Q_i^n \cap Q_j^n$, and a boundary face if $\mathcal{S} = \partial\mathcal{E}^n \cap Q_j^n$. The set of all interior faces in time slab I^n is denoted by \mathcal{S}_I^n , the set of all boundary faces by \mathcal{S}_B^n , and the total set of faces by $\mathcal{S}_{I,B}^n = \mathcal{S}_I^n \cup \mathcal{S}_B^n$.

3.1.2 Function spaces and operators

Each element \mathcal{K} of the tessellation \mathcal{T}_h is an image of the master element $\hat{\mathcal{K}}$: $\mathcal{K} = G_{\mathcal{K}}(\hat{\mathcal{K}})$, where $\hat{\mathcal{K}} = (-1, 1)^4$ is the open unit hypercube in \mathbb{R}^4 . The finite element space associated with the tessellation \mathcal{T}_h is given by:

$$W_h = \{W \in (L^2(\mathcal{E}_h))^5 : W|_{\mathcal{K}} \circ G_{\mathcal{K}} \in (P^k(\hat{\mathcal{K}}))^5, \quad \forall \mathcal{K} \in \mathcal{T}_h\},$$

where $L^2(\mathcal{E}_h)$ is the space of square integrable functions on \mathcal{E}_h and $P^k(\hat{\mathcal{K}})$ denotes the space of polynomials of degree at most k in element $\hat{\mathcal{K}}$. We will also use the following space:

$$V_h = \{V \in (L^2(\mathcal{E}_h))^{5 \times 3} : V|_{\mathcal{K}} \circ G_{\mathcal{K}} \in (P^k(\hat{\mathcal{K}}))^{5 \times 3}, \quad \forall \mathcal{K} \in \mathcal{T}_h\}.$$

Note that $\nabla_h W_h \subset V_h$, where the broken gradient ∇_h of W_h is defined as $(\nabla_h W_h)|_{\mathcal{K}} = \nabla(W_h|_{\mathcal{K}})$. This relation between the function spaces is essential for the discretization.

The trace of a function $f \in W_h$ at the element boundary $\partial\mathcal{K}^L$ is defined as:

$$f^L = \lim_{\epsilon \downarrow 0} f(x - \epsilon n^L),$$

with n^L the unit outward space-time normal at $\partial\mathcal{K}^L$. We will also use the notation \bar{n}^L when only the space components of the outward normal vector are considered. Because of the discontinuous function approximation, a function f in W_h and V_h can have a double-valued trace at the element boundaries $\partial\mathcal{K}$. The traces of the function f at an internal face $\mathcal{S} = \bar{\mathcal{K}}^L \cap \bar{\mathcal{K}}^R$ are denoted by f^L and f^R , respectively. The jump of f at an internal face $\mathcal{S} \in \mathcal{S}_I^n$ in the space direction k of a Cartesian coordinate system is defined as:

$$[[f]]_k = f^L \bar{n}_k^L + f^R \bar{n}_k^R. \quad (3.2)$$

Furthermore, we define the average of f at $\mathcal{S} \in \mathcal{S}_I^n$ as:

$$\{\!\!\{ f \}\!\!\} = \frac{1}{2}(f^L + f^R). \quad (3.3)$$

The jumps and averages are not needed at faces other than internal faces. Note that the jump operator satisfies the following product rule at $\mathcal{S} \in \mathcal{S}_I^n$ for $f \in V_h$ and $g \in W_h$:

$$[[g_i f_{ik}]]_k = \{\!\!\{ g_i \}\!\!\} [[f_{ik}]]_k + [[g_i]]_k \{\!\!\{ f_{ik} \}\!\!\}, \quad (3.4)$$

which can be verified by straightforward substitution of (3.2) and (3.3) into (3.4). We will also use the following relation for the element boundary integrals which occur in the weak formulation:

$$\sum_{\mathcal{K} \in \mathcal{T}_h^n} \int_{\mathcal{Q}} g_i^L f_{ik}^L \bar{n}_k^L d\mathcal{Q} = \sum_{S \in \mathcal{S}_I^n} \int_S \llbracket g_i f_{ik} \rrbracket_k d\mathcal{S} + \sum_{S \in \mathcal{S}_B^n} \int_S g_i^L f_{ik}^L \bar{n}_k^L d\mathcal{S}. \quad (3.5)$$

To verify this relation, note that in the sum over all element boundary integrals, the internal faces are counted twice. Therefore, when summing over the internal faces, the contributions from the left and the right must be counted, which is done by taking the jump.

3.1.3 Arbitrary Lagrangian Eulerian form

Now that the space-time context is well defined, we proceed by expressing the compressible Navier-Stokes equations in the domain $\mathcal{E} \subset \mathbb{R}^4$ as:

$$\begin{cases} U_{i,0} + F_{ik,k}^e - (A_{ikrs} U_{r,s})_{,k} = 0 & \text{on } \mathcal{E}, \\ U = U_0 & \text{on } \Omega(t_0), \\ U = \mathcal{B}(U, U^b) & \text{on } \mathcal{Q}, \end{cases}$$

for $i, r = 1, \dots, 5$ and $k, s = 1, \dots, 3$. The initial flow field is denoted by $U_0 : \Omega(t_0) \rightarrow \mathbb{R}^5$, with U_0 derived from the initial condition described in Chapter 2. The boundary operator is denoted by $\mathcal{B} : \mathbb{R}^{5 \times 5} \rightarrow \mathbb{R}^5$ and is a function of the internal data U and the boundary data U^b derived from the boundary conditions in Chapter 2. At the far-field boundary, suitable in- and out-flow conditions can be derived using local characteristics. The main idea is that characteristic variables of incoming characteristics are set equal to their free-stream values, while the other variables are extrapolated from within the flow domain, see for example [29]. At solid surfaces, the isothermal no-slip boundary condition is applied.

Following the framework described in [1], we write the compressible Navier-Stokes equations as a first-order system by introducing the auxiliary variable¹ $\Theta(U)$:

$$U_{i,0} + F_{ik,k}^e - \Theta_{ik,k} = 0, \quad (3.6a)$$

$$\Theta_{ik} - A_{ikrs} U_{r,s} = 0. \quad (3.6b)$$

¹Here, the auxiliary variable corresponds to the viscous flux, see Chapter 2. An alternative would be to choose $\Theta = \nabla U$, see [2].

The *flux formulation* of (3.6a) is obtained after multiplying by a test function $W \in W_h$, integrating by parts in space-time over an element $\mathcal{K} \in \mathcal{T}_h$ and summing over all elements of the tessellation:

$$\begin{aligned} - \sum_{\mathcal{K} \in \mathcal{T}_h} \int_{\mathcal{K}} (W_{i,0} U_i + W_{i,k} (F_{ik}^e - \Theta_{ik})) d\mathcal{K} \\ + \sum_{\mathcal{K} \in \mathcal{T}_h} \int_{\partial\mathcal{K}} W_i^L (\widehat{U}_i n_0^L + (\widehat{F}_{ik}^e - \widehat{\Theta}_{ik}) \bar{n}_k^L) d(\partial\mathcal{K}) = 0, \quad (3.7) \end{aligned}$$

where n^L is the outward normal vector at $\partial\mathcal{K}$. At the element boundaries, U can be double-valued due to the discontinuous function approximation in each element. Therefore, in order to uniquely define the element boundary integrals and provide a coupling between neighboring elements, we introduce numerical fluxes $(\widehat{\cdot})$ which depend on both the left and right trace of U at the element boundary. The numerical fluxes will be defined later on.

The auxiliary variable Θ is only needed as an intermediate step in the derivation of the discretization and will be eliminated as we go from the flux formulation to the *primal formulation*, which is expressed solely in terms of the primary unknowns U .

But first we turn to the Arbitrary Lagrangian Eulerian (ALE) context in order to accommodate moving and deforming meshes. The flux formulation in ALE context is obtained following the approach described in Van der Vegt and Van der Ven [44]. Using the definition (3.1) of the space-time normal vector, the boundary integral in (3.7) becomes:

$$\begin{aligned} \sum_{\mathcal{K} \in \mathcal{T}_h} \int_{\partial\mathcal{K}} W_i^L (\widehat{U}_i n_0^L + (\widehat{F}_{ik}^e - \widehat{\Theta}_{ik}) \bar{n}_k^L) d(\partial\mathcal{K}) \\ = \sum_{\mathcal{K} \in \mathcal{T}_h} \left(\int_{K(t_{n+1}^-)} W_i^L \widehat{U}_i dK - \int_{K(t_n^+)} W_i^L \widehat{U}_i dK \right) \\ + \sum_{\mathcal{K} \in \mathcal{T}_h} \int_{\mathcal{Q}} W_i^L (\widehat{F}_{ik}^e - \widehat{U}_i v_k - \widehat{\Theta}_{ik}) \bar{n}_k^L d\mathcal{Q}. \end{aligned}$$

The numerical flux \widehat{U} at the faces $K(t_n^+)$ and $K(t_{n+1}^-)$ is defined as an upwind flux to ensure causality in time:

$$\widehat{U} = \begin{cases} U^L & \text{at } K(t_{n+1}^-), \\ U^R & \text{at } K(t_n^+). \end{cases}$$

With this numerical flux, the flux formulation in each space-time slab only depends on the previous space-time slab, therefore the summation over the space-time slabs can be dropped and the ALE flux formulation of (3.6a) becomes:

$$\begin{aligned}
& - \sum_{\mathcal{K} \in \mathcal{T}_h^n} \int_{\mathcal{K}} (W_{i,0} U_i + W_{i,k} (F_{ik}^e - \Theta_{ik})) d\mathcal{K} \\
& \quad + \sum_{\mathcal{K} \in \mathcal{T}_h^n} \left(\int_{K(t_{n+1}^-)} W_i^L U_i^L dK - \int_{K(t_n^+)} W_i^L U_i^R dK \right) \\
& \quad \quad + \sum_{\mathcal{K} \in \mathcal{T}_h^n} \int_{\mathcal{Q}} W_i^L (\widehat{F}_{ik}^e - v_k \widehat{U}_i - \widehat{\Theta}_{ik}) \bar{n}_k^L d\mathcal{Q} = 0. \quad (3.8)
\end{aligned}$$

3.1.4 The auxiliary variable

The mixed formulation (3.6) has the disadvantage that both U and Θ have to be stored and solved during a computation. Fortunately, it is possible to eliminate the auxiliary variable using a weak expression for Θ in terms of the primary unknowns U , so only U has to be stored. To derive this expression, we multiply (3.6b) by a test function $V \in V_h$, integrate by parts in space (twice) over an element $\mathcal{K} \in \mathcal{T}_h$ and sum over all elements of the tessellation:

$$\begin{aligned}
\sum_{\mathcal{K} \in \mathcal{T}_h^n} \int_{\mathcal{K}} V_{ik} \Theta_{ik} d\mathcal{K} &= \sum_{\mathcal{K} \in \mathcal{T}_h^n} \int_{\mathcal{K}} V_{ik} A_{ikrs} U_{r,s} d\mathcal{K} \\
& \quad + \sum_{\mathcal{K} \in \mathcal{T}_h^n} \int_{\mathcal{Q}} V_{ik}^L A_{ikrs}^L (\widehat{U}_r - U_r^L) \bar{n}_s^L d\mathcal{Q}, \quad (3.9)
\end{aligned}$$

where we introduced the numerical flux \widehat{U} after the first integration by parts. In this case, the numerical flux does not have a time contribution because we only integrated in space. Instead of using integrals over the element boundary \mathcal{Q} , it is more convenient to use integrals over the element faces \mathcal{S} . We therefore apply relation (3.5) to the element boundary integral

of equation (3.9):

$$\begin{aligned} & \sum_{\mathcal{K} \in \mathcal{T}_h^n} \int_{\mathcal{Q}} V_{ik}^L A_{ikrs}^L (\widehat{U}_r - U_r^L) \bar{n}_s^L d\mathcal{Q} \\ &= \sum_{\mathcal{S} \in \mathcal{S}_I^n} \int_{\mathcal{S}} \llbracket V_{ik} A_{ikrs} (\widehat{U}_r - U_r) \rrbracket_s d\mathcal{S} + \sum_{\mathcal{S} \in \mathcal{S}_B^n} \int_{\mathcal{S}} V_{ik}^L A_{ikrs}^L (\widehat{U}_r - U_r^L) \bar{n}_s^L d\mathcal{S}. \end{aligned}$$

Now that we explicitly distinguish between internal and boundary faces, we can follow the approach by Bassi and Rebay [3, 5, 6] and define the numerical flux as:

$$\widehat{U} = \begin{cases} \{\{U\}\} & \text{at } \mathcal{S}_I^n, \\ U^b & \text{at } \mathcal{S}_B^n. \end{cases}$$

With this choice for the numerical flux at the internal faces and using relation (3.4) we obtain: $\llbracket V_{ik} A_{ikrs} (\widehat{U}_r - U_r) \rrbracket_s = -\{\{V_{ik} A_{ikrs}\}\} \llbracket U_r \rrbracket_s$, which leads to the following expression for the auxiliary variable:

$$\begin{aligned} & \sum_{\mathcal{K} \in \mathcal{T}_h^n} \int_{\mathcal{K}} V_{ik} \Theta_{ik} d\mathcal{K} = \sum_{\mathcal{K} \in \mathcal{T}_h^n} \int_{\mathcal{K}} V_{ik} A_{ikrs} U_{r,s} d\mathcal{K} \\ & - \sum_{\mathcal{S} \in \mathcal{S}_I^n} \int_{\mathcal{S}} \{\{V_{ik} A_{ikrs}\}\} \llbracket U_r \rrbracket_s d\mathcal{S} - \sum_{\mathcal{S} \in \mathcal{S}_B^n} \int_{\mathcal{S}} V_{ik}^L A_{ikrs}^L (U_r^L - U_r^b) \bar{n}_s^L d\mathcal{S}. \end{aligned}$$

In order to obtain an explicit expression for the auxiliary variable, we need to define a *global lifting operator*. The global lifting operator $\mathcal{R} \in \mathbb{R}^{5 \times 3}$ is defined in a weak sense as: *Find an $\mathcal{R} \in V_h$, such that for all $V \in V_h$:*

$$\begin{aligned} & \sum_{\mathcal{K} \in \mathcal{T}_h^n} \int_{\mathcal{K}} V_{ik} \mathcal{R}_{ik} d\mathcal{K} = \sum_{\mathcal{S} \in \mathcal{S}_I^n} \int_{\mathcal{S}} \{\{V_{ik} A_{ikrs}\}\} \llbracket U_r \rrbracket_s d\mathcal{S} \\ & + \sum_{\mathcal{S} \in \mathcal{S}_B^n} \int_{\mathcal{S}} V_{ik}^L A_{ikrs}^L (U_r^L - U_r^b) \bar{n}_s^L d\mathcal{S}. \quad (3.10) \end{aligned}$$

More details on the lifting operator are given in Section 3.2. According to this definition, the face integrals in the expression for Θ can now be written as element integrals, leading to the weak expression of the auxiliary variable:

$$\sum_{\mathcal{K} \in \mathcal{T}_h^n} \int_{\mathcal{K}} V_{ik} \Theta_{ik} d\mathcal{K} = \sum_{\mathcal{K} \in \mathcal{T}_h^n} \int_{\mathcal{K}} V_{ik} (A_{ikrs} U_{r,s} - \mathcal{R}_{ik}) d\mathcal{K}, \quad \forall V \in V_h. \quad (3.11)$$

In other words, $\Theta_{ik} = A_{ikrs}U_{r,s} - \mathcal{R}_{ik}$ almost everywhere in \mathcal{E}_h^n . The role of the lifting operator \mathcal{R} is to penalize the jumps at the faces [1]. For smooth solutions $\mathcal{R} = 0$.

3.1.5 Primal formulation

The primal formulation can be obtained using the expression (3.11) for the auxiliary variable Θ . Since $\nabla_h W_h \subset V_h$, the special case $V_{ik} = W_{i,k}$ can be considered in (3.11), and the auxiliary variable Θ can be replaced in the element integral of (3.8):

$$\sum_{\mathcal{K} \in \mathcal{T}_h^n} \int_{\mathcal{K}} W_{i,k} \Theta_{ik} d\mathcal{K} = \sum_{\mathcal{K} \in \mathcal{T}_h^n} \int_{\mathcal{K}} W_{i,k} (A_{ikrs}U_{r,s} - \mathcal{R}_{ik}) d\mathcal{K}.$$

Now, only the numerical fluxes \widehat{F}^e and $\widehat{\Theta}$ remain to be chosen. We therefore consider the element boundary integrals of (3.8) and use relation (3.5) to get the element face integrals:

$$\begin{aligned} \sum_{\mathcal{K} \in \mathcal{T}_h^n} \int_{\mathcal{Q}} W_i^L (\widehat{F}_{ik}^e - v_k \widehat{U}_i - \widehat{\Theta}_{ik}) \bar{n}_k^L d\mathcal{Q} \\ = \sum_{\mathcal{S} \in \mathcal{S}_I^n} \int_{\mathcal{S}} \llbracket W_i (\widehat{F}_{ik}^e - v_k \widehat{U}_i - \widehat{\Theta}_{ik}) \rrbracket_k d\mathcal{S} \\ + \sum_{\mathcal{S} \in \mathcal{S}_B^n} \int_{\mathcal{S}} W_i^L (\widehat{F}_{ik}^e - v_k \widehat{U}_i - \widehat{\Theta}_{ik}) \bar{n}_k^L d\mathcal{S}. \end{aligned}$$

The inviscid numerical flux \widehat{F}^e is based on the HLLC approximate Riemann solver [7, 42, 44], because of its computational efficiency, accuracy and straightforward implementation. The HLLC flux is consistent and conservative and is obtained by interpreting the discontinuity between U^L and U^R at a face \mathcal{S} as a local Riemann problem, which is solved approximately while taking into account the grid velocity \vec{v} . Following [44] and using the fact that $n^R = -n^L$, we have:

$$\llbracket W_i (\widehat{F}_{ik}^e - v_k \widehat{U}_i) \rrbracket_k = (W_i^L - W_i^R) H_i$$

with $H = H(U^L, U^R, v, \bar{n}^L)$ the HLLC flux. At a face moving with velocity v , the HLLC flux is given by:

$$H_i = \frac{1}{2}((F_{ik}^e)^L \bar{n}_k^L - (F_{ik}^e)^R \bar{n}_k^R) + \frac{1}{2}(|S^M - v| - |S^L - v|)U_{i*}^L - (v - |S^L - v|)U_i^L + \frac{1}{2}(|S^R - v| - |S^M - v|)U_{i*}^R - (v + |S^R - v|)U_i^R,$$

with $(F^e)^{L,R} = F^e(U^{L,R})$. The intermediate states U_*^L and U_*^R are given by:

$$U_*^{L,R} = \frac{S^{L,R} - q^{L,R}}{S^{L,R} - S^M} U^{L,R} + \frac{1}{S^{L,R} - S^M} \begin{bmatrix} 0 \\ (p_* - p^{L,R}) \bar{n}_k^L \\ p_* S^M - p^{L,R} q^{L,R} \end{bmatrix},$$

with $q = \bar{n}_k^L u_k$ the normal velocity and p_* the intermediate pressure:

$$p_* = \rho^L (S^L - q^L)(S^M - q^L) + p^L = \rho^R (S^R - q^R)(S^M - q^R) + p^R.$$

The middle wave speed is defined as:

$$S^M = \frac{\rho^R q^R (S^R - q^R) - p^R - \rho^L q^L (S^L - q^L) + p^L}{\rho^R (S^R - q^R) - \rho^L (S^L - q^L)},$$

and the left and right wave speeds as:

$$S^L = \min\{q^L - a^L, q^R - a^R\}, \quad S^R = \max\{q^L + a^L, q^R + a^R\},$$

with $a = \sqrt{\gamma p / \rho}$ the speed of sound. At the boundary faces, we use $H^b = H(U^L, U^b, v, \bar{n}^L)$.

The numerical flux $\hat{\Theta}$ is defined following Brezzi [10] as a central flux $\hat{\Theta} = \{\{\Theta\}\}$, using the weak expression (3.11) for the auxiliary variable. This is a suitable choice as viscosity does not have a preferred direction. The numerical flux can thus be written as:

$$\hat{\Theta}_{ik}(U^L, U^R) = \begin{cases} \{\{A_{ikrs} U_{r,s} - \eta \mathcal{R}_{ik}^S\}\} & \text{for } \mathcal{S} \in \mathcal{S}_I^n, \\ A_{ikrs}^b U_{r,s}^b - \eta \mathcal{R}_{ik}^S & \text{for } \mathcal{S} \in \mathcal{S}_B^n, \end{cases}$$

where η is a stabilization constant and $A^b = A(U^b)$ and $U_{r,s}^b$ denotes the derivatives of U at the boundary. The *local lifting operator* \mathcal{R}^S is an approximation of the global lifting operator \mathcal{R} and is preferable because it

reduces the width of the stencil to the minimum, see [10]. The local lifting operator $\mathcal{R}^S \in \mathbb{R}^{5 \times 3}$ is defined as follows: *Find an $\mathcal{R}^S \in V_h$, such that for all $V \in V_h$:*

$$\sum_{\mathcal{K} \in \mathcal{T}_h^n} \int_{\mathcal{K}} V_{ik} \mathcal{R}_{ik}^S d\mathcal{K} = \begin{cases} \int_{\mathcal{S}} \{V_{ik} A_{ikrs}\} \llbracket U_r \rrbracket_s d\mathcal{S} & \text{for } \mathcal{S} \in \mathcal{S}_I, \\ \int_{\mathcal{S}} V_{ik}^L A_{ikrs}^L (U_r^L - U_r^b) \bar{n}_s d\mathcal{S} & \text{for } \mathcal{S} \in \mathcal{S}_B. \end{cases} \quad (3.12)$$

With these numerical fluxes the space-time weak formulation of the compressible Navier-Stokes equations in terms of the primary unknown U can be written as follows: *Find a $U \in W_h$, such that for all $W \in W_h$:*

$$\begin{aligned} & - \sum_{\mathcal{K} \in \mathcal{T}_h^n} \int_{\mathcal{K}} (W_{i,0} U_i + W_{i,k} (F_{ik}^e - A_{ikrs} U_{r,s} + \mathcal{R}_{ik})) d\mathcal{K} \\ & + \sum_{K \in \mathcal{T}_h^n} \left(\int_{K(t_{n+1}^-)} W_i^L U_i^L dK - \int_{K(t_n^+)} W_i^L U_i^R dK \right) \\ & + \sum_{\mathcal{S} \in \mathcal{S}_I^n} \int_{\mathcal{S}} (W_i^L - W_i^R) H_i d\mathcal{S} + \sum_{\mathcal{S} \in \mathcal{S}_B^n} \int_{\mathcal{S}} W_i^L H_i^b d\mathcal{S} \\ & - \sum_{\mathcal{S} \in \mathcal{S}_I^n} \int_{\mathcal{S}} \llbracket W_i \rrbracket_k \{A_{ikrs} U_{r,s} - \eta \mathcal{R}_{ik}^S\} d\mathcal{S} \\ & - \sum_{\mathcal{S} \in \mathcal{S}_B^n} \int_{\mathcal{S}} W_i^L (A_{ikrs}^b U_{r,s}^b - \eta \mathcal{R}_{ik}^S) \bar{n}_k^L d\mathcal{S} = 0 \end{aligned} \quad (3.13)$$

where we used the relation $\llbracket W_i \widehat{\Theta}_{ik} \rrbracket_k = \llbracket W_i \rrbracket_k \widehat{\Theta}_{ik}$, which follows from the viscous numerical flux being conservative: $\widehat{\Theta}(U^L, U^R) = \widehat{\Theta}(U^R, U^L)$.

Discontinuous Galerkin methods are known to suffer from numerical oscillations around shocks and sharp gradients. This problem can be overcome using a slope limiter (see for example [15]), but we prefer the artificial dissipation proposed in [44] as it allows convergence to steady-state up to machine precision. We refer to [44] for a detailed description of the artificial dissipation operator.

3.2 Algebraic system

In this section, the space-time discretization of the compressible Navier-Stokes equations is completed by defining the basis functions, computing the local lifting operator and constructing the system of algebraic equations.

3.2.1 Basis functions

Here, we use linear polynomials to represent the trial function U and the test function W in each element $\mathcal{K} \in \mathcal{T}_h^n$:

$$U(t, \bar{x})|_{\mathcal{K}} = \hat{U}_m \psi_m(t, \bar{x}), \quad W(t, \bar{x})|_{\mathcal{K}} = \hat{W}_l \psi_l(t, \bar{x}), \quad m, l = 0, \dots, 4$$

with $(\hat{\cdot})$ the expansion coefficients and ψ the basis functions. The basis functions are defined such that the test and trial functions are split into an element mean at time t_{n+1} and a fluctuating part. This construction facilitates the definition of the artificial dissipation operator and of the multigrid convergence acceleration method [44]. The basis functions ψ are given by:

$$\psi_m = \begin{cases} 1, & \text{for } m = 0 \\ \phi_m(t, \bar{x}) - \frac{1}{|K_j(t_{n+1}^-)|} \int_{K_j(t_{n+1}^-)} \phi_m(t, \bar{x}) dK, & \text{for } m = 1, \dots, 4 \end{cases}$$

where the functions ϕ in an element \mathcal{K} are related to the basis functions $\hat{\phi}$ on the master element $\hat{\mathcal{K}}$ through the mapping G :

$$\phi_m = \hat{\phi}_m \circ G_{\mathcal{K}}^{-1}$$

with $\hat{\phi}_m(\xi) \in P^1(\hat{\mathcal{K}})$ and ξ the local coordinates in the master element $\hat{\mathcal{K}}$ defined in Sec. 3.1.

3.2.2 Lifting operators

The global and local lifting operators contained in the primal formulation (3.13) must be computed first in order to obtain the system of algebraic equations for the expansion coefficients \hat{U} of the trial function U . The volume integral containing the global lifting operator can simply be replaced by face integrals using its definition (3.10):

$$\begin{aligned} \sum_{\mathcal{K} \in \mathcal{T}_h^n} \int_{\mathcal{K}} W_{i,k} \mathcal{R}_{ik} d\mathcal{K} &= \sum_{\mathcal{S} \in \mathcal{S}_I^n} \int_{\mathcal{S}} \{W_{i,k} A_{ikrs}\} [[U_r]]_s d\mathcal{S} \\ &+ \sum_{\mathcal{S} \in \mathcal{S}_B^n} \int_{\mathcal{S}} W_{i,k}^L A_{ikrs}^L (U_r^L - U_r^b) \bar{n}_s^L d\mathcal{S}. \end{aligned}$$

These face integrals can be directly computed by replacing the test and trial functions by their polynomial expansions. The local lifting operator, however, cannot be computed directly. Like the test and trial functions, it is represented by a linear polynomial:

$$\mathcal{R}^{\mathcal{S}}(t, \bar{x})|_{\mathcal{K}} = \hat{R}_j \psi_j(t, \bar{x}),$$

and a small linear system must be solved for the expansion coefficients \hat{R}_j . The linear system follows from the definition of the local lifting operator (3.12). By this definition, the local lifting operator is only non-zero on the two elements \mathcal{K}^L and \mathcal{K}^R connected to the face $\mathcal{S} \in \mathcal{S}_I^n$, hence:

$$\int_{\mathcal{K}^R} V_{ik} \mathcal{R}_{ik}^{\mathcal{S}} d\mathcal{K} + \int_{\mathcal{K}^L} V_{ik} \mathcal{R}_{ik}^{\mathcal{S}} d\mathcal{K} = \int_{\mathcal{S}} \{ \{ V_{ik} A_{ikrs} \} \} [U_r]_s d\mathcal{S}.$$

Since V is an arbitrary test function, this is equivalent with the two following equations:

$$\int_{\mathcal{K}^{L,R}} V_{ik} \mathcal{R}_{ik}^{\mathcal{S}} d\mathcal{K} = \frac{1}{2} \int_{\mathcal{S}} V_{ik}^{L,R} A_{ikrs}^{L,R} [U_r]_s d\mathcal{S},$$

where the superscript L, R refers to the traces from either the left or right element. Replacing $\mathcal{R}^{\mathcal{S}}$ by its polynomial approximation leads to two systems of linear equations for the expansion coefficients \hat{R}_{ikj} of $\mathcal{R}_{ik}^{\mathcal{S}}$ on $\mathcal{S} \in \mathcal{S}_I$:

$$\hat{R}_{ikj}^{L,R} \int_{\mathcal{K}^{L,R}} \psi_l \psi_j d\mathcal{K} = \frac{1}{2} \int_{\mathcal{S}} \psi_l^{L,R} A_{ikrs}^{L,R} [U_r]_s d\mathcal{S}.$$

The element mass matrices on the l.h.s. are denoted by $M_{lj}^{L,R}$ and can easily be inverted leading to following expression for the expansion coefficients of the local lifting operator on $\mathcal{S} \in \mathcal{S}_I$:

$$\hat{R}_{ikj}^{L,R} = \frac{1}{2} (M^{-1})_{jl}^{L,R} \int_{\mathcal{S}} \psi_l^{L,R} A_{ikrs}^{L,R} [U_r]_s d\mathcal{S}. \quad (3.14)$$

Similarly, the expression for the expansion coefficients of the local lifting operator for the faces $\mathcal{S} \in \mathcal{S}_B$ is:

$$\hat{R}_{ikj}^L = (M^{-1})_{jl}^L \int_{\mathcal{S}} \psi_l^L A_{ikrs}^L (U_r^L - U_r^b) \bar{n}_s^L d\mathcal{S}. \quad (3.15)$$

Note that the mass matrices M only have to be inverted once per element in each space-time slab, after which the local lifting operator can be computed as a small matrix-vector multiplication.

3.2.3 Equations for the expansion coefficients of the flow field

The system of algebraic equations for the expansion coefficients \hat{U} of the trial function U is obtained by replacing U and the test function W in (3.13) by their polynomial expansions. We distinguish between the inviscid and viscous part:

$$\mathcal{L}^e(\hat{U}^n, \hat{U}^{n-1}) + \mathcal{L}^v(\hat{U}^n) = 0. \quad (3.16)$$

The term \mathcal{L}^e corresponds to the inviscid part of the residuals and is defined as:

$$\mathcal{L}_{il}^e = - \sum_{\mathcal{K} \in \mathcal{T}_h^n} (A_{il} + B_{il}) + \sum_{K \in \mathcal{T}_h^n} C_{il} + \sum_{\mathcal{S} \in \mathcal{S}_{I,B}^n} E_{il},$$

with $i = 1, \dots, 5$ the equation number, $l = 0, \dots, 4$, the index of the expansion coefficients and the terms A , B , C and E defined as:

$$A_{il} = \int_{\mathcal{K}} \psi_{l,0} U_i d\mathcal{K}, \quad (3.17)$$

$$B_{il} = \int_{\mathcal{K}} \psi_{l,k} F_{ik}^e d\mathcal{K}, \quad (3.18)$$

$$C_{il} = \int_{K(t_{\bar{n}+1}^-)} \psi_l^L U_i^L dK - \int_{K(t_{\bar{n}}^+)} \psi_l^L U_i^R dK, \quad (3.19)$$

$$E_{il} = \begin{cases} \int_{\mathcal{S}} (\psi_l^L - \psi_l^R) H_i d\mathcal{S} & \text{for } \mathcal{S} \in \mathcal{S}_I, \\ \int_{\mathcal{S}} \psi_l^L H_i^b d\mathcal{S} & \text{for } \mathcal{S} \in \mathcal{S}_B, \end{cases} \quad (3.20)$$

with $F_{ik}^e = F_{ik}^e(U)$ the Euler flux and $H_i = H_i(U^L, U^R, v, \bar{n}^L)$ the HLLC flux. The term \mathcal{L}^v corresponds to the viscous part of the residual and is defined as:

$$\mathcal{L}_{il}^v = \sum_{\mathcal{K} \in \mathcal{T}_h^n} D_{il} + \sum_{\mathcal{S} \in \mathcal{S}_{I,B}^n} (-F_{il} - G_{il} + H_{il}),$$

with

$$D_{il} = \int_{\mathcal{K}} \psi_{l,k} A_{ikrs} U_{r,s} d\mathcal{K}, \quad (3.21)$$

$$F_{il} = \begin{cases} \int_{\mathcal{S}} \{ \psi_{l,k} A_{ikrs} \} [U_r]_s d\mathcal{S} & \text{for } \mathcal{S} \in \mathcal{S}_I, \\ \int_{\mathcal{S}} \psi_{l,k}^L A_{ikrs}^L (U_r^L - U_r^b) \bar{n}_s^L d\mathcal{S} & \text{for } \mathcal{S} \in \mathcal{S}_B, \end{cases} \quad (3.22)$$

$$G_{il} = \begin{cases} \int_{\mathcal{S}} \llbracket \psi_l \rrbracket_k \{ \{ A_{ikrs} U_{r,s} \} \} d\mathcal{S} & \text{for } \mathcal{S} \in \mathcal{S}_I, \\ \int_{\mathcal{S}} \psi_l^L (A_{ikrs}^b U_{r,s}^b) \bar{n}_k^L d\mathcal{S} & \text{for } \mathcal{S} \in \mathcal{S}_B, \end{cases} \quad (3.23)$$

$$H_{il} = \begin{cases} \eta \int_{\mathcal{S}} \llbracket \psi_l \rrbracket_k \{ \{ \mathcal{R}_{ik}^S \} \} d\mathcal{S} & \text{for } \mathcal{S} \in \mathcal{S}_I, \\ \eta \int_{\mathcal{S}} \psi_l^L \mathcal{R}_{ik}^S \bar{n}_k^L d\mathcal{S} & \text{for } \mathcal{S} \in \mathcal{S}_B, \end{cases} \quad (3.24)$$

with $\mathcal{R}_{ik}^S = \mathcal{R}_{ik}^S(U)$ the local lifting operator and $A_{ikrs} = A_{ikrs}(U)$ the homogeneity tensor.

Thus, the space-time discontinuous Galerkin discretization of the compressible Navier-Stokes equations results in a system of coupled non-linear equations for the expansion coefficients, which is solved by adding a pseudo-time derivative:

$$|K^n| \frac{\partial \hat{U}^n}{\partial \tau} = -\frac{1}{\Delta t} \left(\mathcal{L}^e(\hat{U}^n; \hat{U}^{n-1}) + \mathcal{L}^v(\hat{U}^n) \right), \quad (3.25)$$

and integrating to steady-state in pseudo-time. Different pseudo-time stepping methods suitable for this purpose are presented in Chapter 4, where we analyze their stability in pseudo-time and compare their efficiency. Computing the viscous part \mathcal{L}^v of the residual takes roughly twice the CPU time needed for the inviscid part \mathcal{L}^e , see Table 3.1. The stabilization term (3.24) is by far the most expensive as it requires the expansion coefficients of the local lifting operator, which must be computed first using (3.14) and (3.15).

Table 3.1: Relative computational effort in the NLR program HEXADAP.

	Equation	Notation	CPU time (%)
Time flux	(3.17)	A_{il}	1.7
	(3.19)	C_{il}	1.8
Euler flux	(3.18)	B_{il}	4
	(3.20)	E_{il}	25
Viscous flux	(3.21)	D_{il}	10
	(3.22)	F_{il}	10
	(3.23)	G_{il}	7
	(3.24)	H_{il}	40.5

3.3 Numerical simulations

The space-time discontinuous Galerkin method for the compressible Navier-Stokes equations is implemented in the NLR computer program HEXADAP and, in this section, numerical results are presented. We consider a model problem of two-dimensional laminar dynamic stall and the three-dimensional vortex flow around a delta wing.

3.3.1 Laminar dynamic stall of NACA0012 airfoil

We consider the laminar flow over a NACA0012 airfoil in rapid pitch-up maneuver, comparable to the situation described in [34] and [50]. The flow is characterized by a complex interaction of an unsteady leading-edge vortex, shear layer vortices and trailing edge vortex, resulting in the detachment of the leading edge vortex: the dynamic stall phenomenon. The complexity of the unsteady flow and the significant grid movement make this a challenging test case for the space-time discretization of the Navier-Stokes equations, where the deforming elements in the neighborhood of the moving airfoil are accommodated with the deformation algorithm proposed in [44].

In this case, the far-field Reynolds number is $Re_\infty = 10^4$ and the Mach number $M_\infty = 0.2$, based on the non-dimensionalization presented in Chapter 2 with L the chord of the airfoil. The pitch axis is situated at 25% from the leading edge and the airfoil rotates in such a way that the angle of attack α evolves as follows:

$$\alpha(t) = a + bt - a \exp(-ct). \quad (3.26)$$

The coefficients are $a = -1.2455604$, $b = 2.2918312$, $c = 1.84$ and the time t ranges from 0 to 25. With these coefficients, the movement of the airfoil is the same as the movement of the NACA0015 used in [50]. At time $t = 0$, both $\alpha = 0$ and $d\alpha/dt = 0$ and, after a short transition, the movement becomes mainly linear. The basis functions in the discretization are linear and the stabilization constant is $\eta = 5$.

Remark. Although, in this thesis, we limit ourselves to linear basis functions, the space-time method allows higher-order basis functions to be used both in space and time. We refer to [41] for the detailed analysis of high-order space-time approximations.

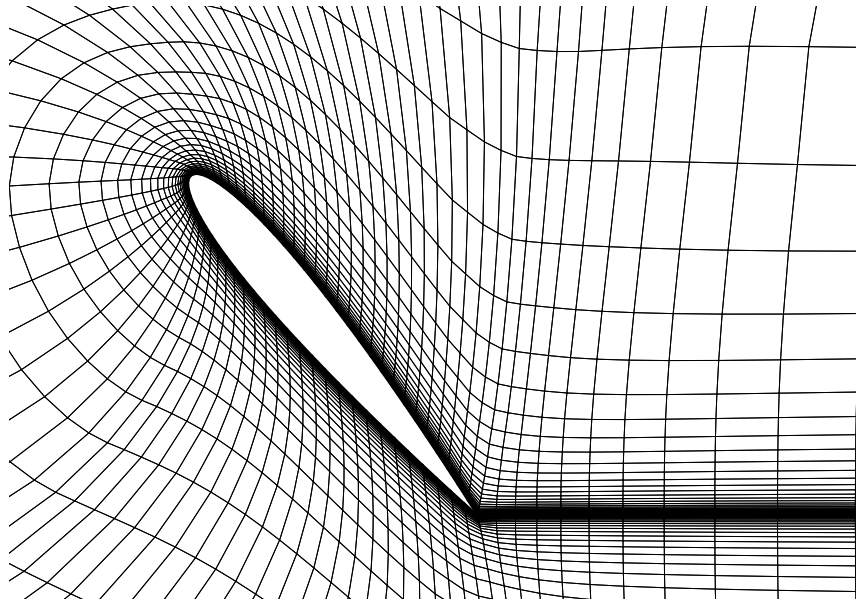
Since the flow is laminar, the boundary layer thickness is estimated as $\delta \approx 5/\sqrt{\text{Re}}$ and the computational mesh should be fine enough to accurately represent this layer. In this case, $\delta \approx 0.05$ and we use a C-type grid with 112×38 elements which results in 14 elements in this boundary layer. The dimensionless physical time step is $\Delta t = 0.005$. Each physical time step requires about 20 pseudo-time iterations to solve the algebraic system. At each step the mesh moves and deforms according to the motion of the airfoil prescribed by (3.26), see Figure 3.3 for the details of the mesh at a 50° angle of attack. At this point, the mesh lines are no longer perpendicular to the airfoil geometry and are sharply bend near the trailing edge. Yet, even on this mesh of reduced quality, the space-time discontinuous Galerkin method still performs well as can be seen in Figure 3.4 which shows the streamlines at angles of attack $\alpha = 30^\circ$, 40° and 50° , respectively. The sudden drop in lift and increase in drag associated with the detachment of the leading edge vortex (between $\alpha = 40^\circ$ and 50°) can clearly be seen in Figure 3.5, where we show the lift and drag coefficients as a function of the angle of attack.

We thus conclude that the space-time discontinuous Galerkin method combined with grid movement and deformation has significant potential to simulate the complex flow phenomenon which occur in dynamic stall situations.

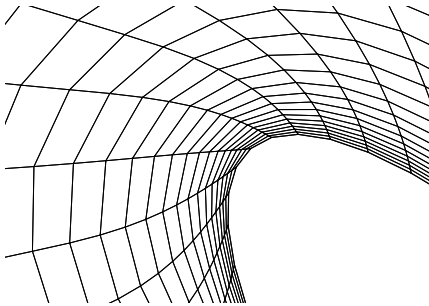
3.3.2 Delta wing with mesh adaptation

To test the performance of the space-time method with local mesh adaptation in a 3D situation, we consider the steady state flow around the 75° delta wing used in the experiments by Riley and Lawson [39], see Figure 3.6 for details on the geometry. The flow is symmetric with a large steady vortex and two secondary vortices along both sides of the wing, see for example the similar situation in [39].

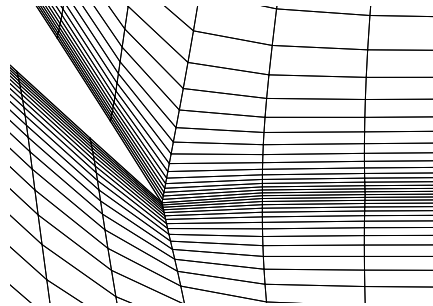
We consider the case with far-field Reynolds number $\text{Re}_\infty = 4 \cdot 10^4$, Mach number $M_\infty = 0.3$ and angle of attack $\alpha = 12.5^\circ$. We compute the solution on a coarse mesh with 208 896 elements and on a fine mesh with 1 671 168 elements. The basis functions are linear and the stabilization constant is $\eta = 7$. Since this is a steady-state case, we take one huge physical time step $\Delta t = 10^{21}$ and solve the algebraic system with the pseudo-time integration method. Figure 3.7 shows an overview of the streamlines and the vorticity in several cross sections of the flow field computed on the fine



(a) Overview



(b) leading edge



(c) trailing edge

Figure 3.3: Overview and details of the mesh deformation in the dynamic stall case at $\alpha = 50^\circ$.

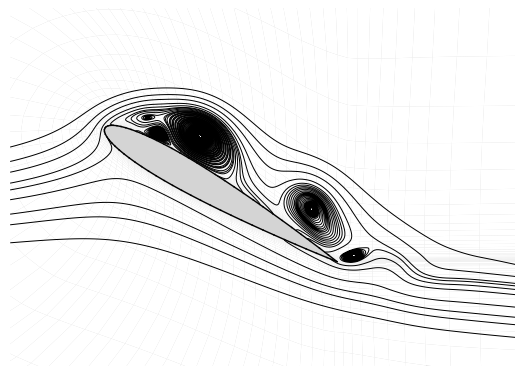
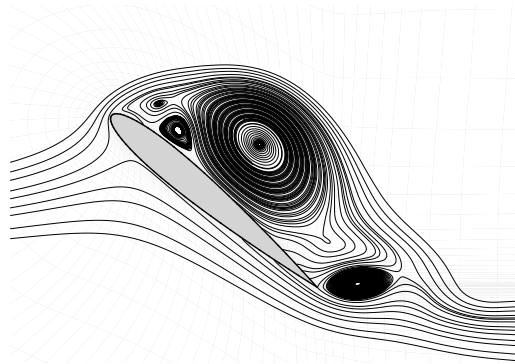
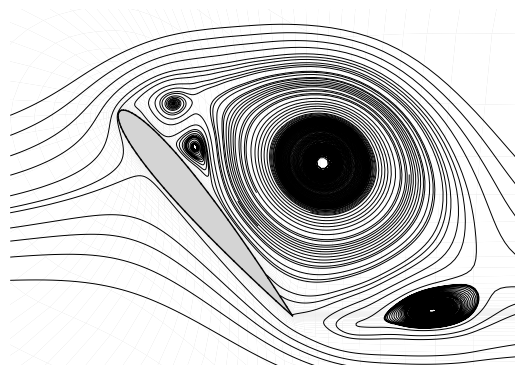
(a) $\alpha = 30^\circ$ (b) $\alpha = 40^\circ$ (c) $\alpha = 50^\circ$

Figure 3.4: Streamlines in the dynamic stall case.

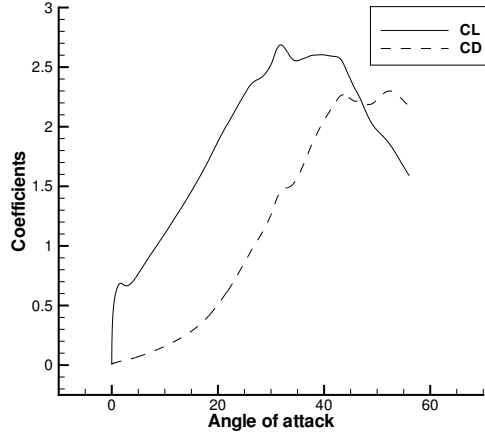


Figure 3.5: The lift and drag coefficients in the dynamic stall case.

mesh. The two main vortices are clearly visible as well as the secondary vortices near the edges of the wing, see also Figure 3.8 for the streamlines in cross-section $x/c = 0.6$.

In the local mesh adaptation procedure, we start with the solution on the coarse mesh, then refine the mesh in the regions with the highest vorticity, thereby increasing the number of elements by 10%. Then we compute the solution on the adapted mesh and repeat the same procedure until the mesh has been adapted three times. The final adapted mesh has 286 416 elements, see Figure 3.9 for an impression of the 3D adaptation. Note that the refinement mainly takes place in the stream-wise direction.

The effect of vorticity driven mesh adaptation is shown by comparing the pressure coefficient and the helicity ($u \cdot \omega$ with w the vorticity) obtained on the coarse and adapted mesh with those on the fine mesh. Figure 3.10 shows the pressure coefficient C_p on the delta wing at cross sections $x/c = 0.3$ and $x/c = 0.6$ respectively. In these figures, we also show the C_p computed with the NLR finite volume code ENSOLV [30] on the fine mesh and found some small differences. For instance, the suction peak with HEXADAP on the coarse and adapted mesh is higher than the one on the fine mesh and the one obtained with ENSOLV. Also, the sharp edge at the

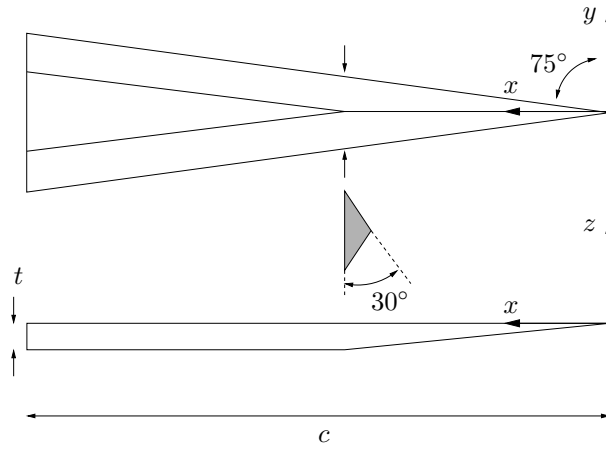


Figure 3.6: The geometry of the delta wing ($t/c = 0.024$).

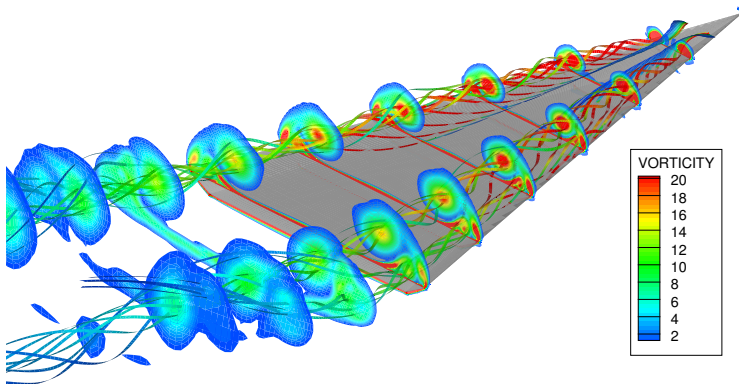


Figure 3.7: Streamlines and vorticity in several cross sections of the delta wing.

bottom of the wing induces a small oscillation in C_p with HEXADAP on the coarse and adapted mesh, while the fine mesh results of both HEXADAP and ENSOLV are smoother. We conclude that the pressure coefficient is not very sensitive to the mesh quality, even the coarse mesh gives reasonable results.

The helicity, however, is much more sensitive to the mesh quality as can be seen in Figures 3.11 and 3.12 where we show the mesh and helicity contours in cross sections $x/c = 0.9$ and $x/c = 1.1$, respectively. At $x/c = 0.9$, the results on the coarse mesh are rather poor, while the results on the adapted mesh are much closer to those on the fine mesh. Downstream of the delta wing ($x/c = 1.1$), the advantage of grid adaptation is even clearer: on the coarse grid the details in the helicity are almost lost, while on the adapted grid the helicity still strongly resembles the one on the fine mesh. Since the adapted mesh has five times fewer elements than the fine grid, the computational cost is much lower. This demonstrates that a solution adaptive space-time method can result in significant cost savings when applied to vortex dominated viscous flows.

Remark. Cost saving is important as DG methods are known to be computationally expensive in comparison to finite volume methods. In [47], the computational complexity of the space-time DG method for the Euler equations is considered in detail and in [45] it was shown that the CPU time per degree of freedom is comparable to a Jameson finite volume solver. The relative cost of the viscous part is given in Table 3.1.

The space-time DG discretization is now validated and, in the next chapter, we focus on the pseudo-time integration method used to solve the system of algebraic equations.

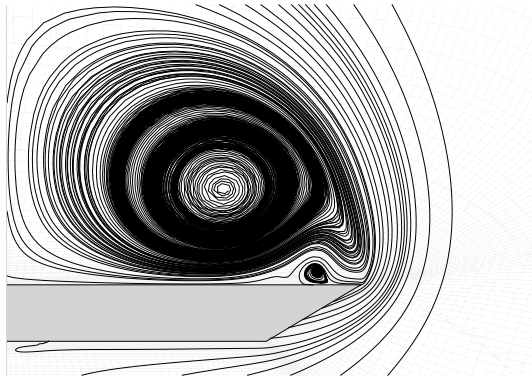


Figure 3.8: Streamlines around the delta wing (cross-section $x/c = 0.6$).

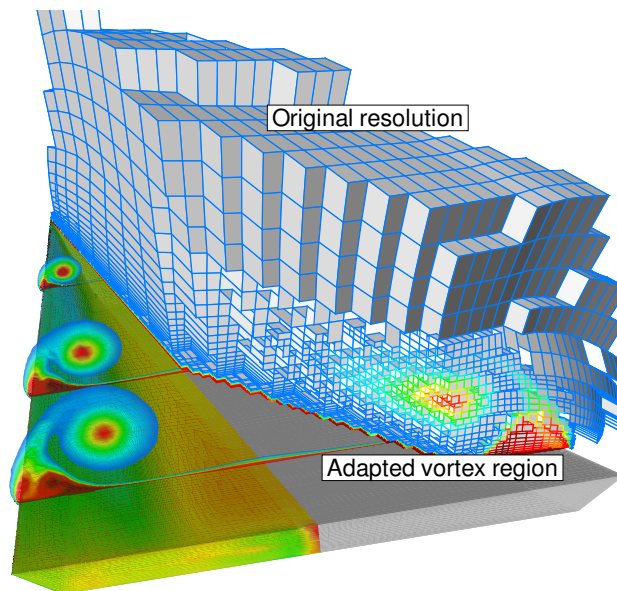
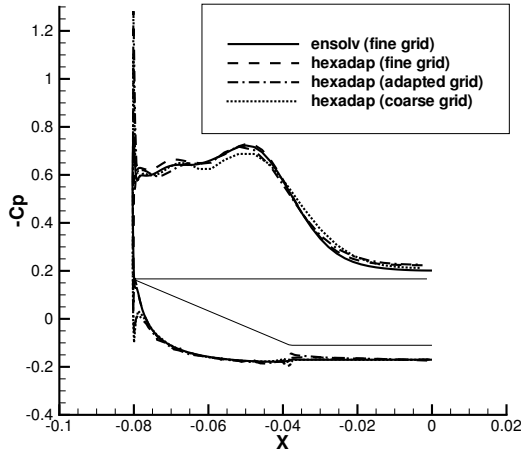
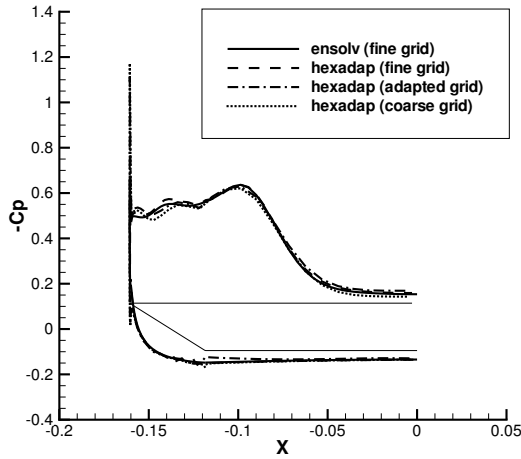


Figure 3.9: Impression of the vorticity based mesh adaptation.



(a) cross-section $x/c = 0.3$



(b) cross-section $x/c = 0.6$

Figure 3.10: Pressure coefficient C_p at of the delta wing.

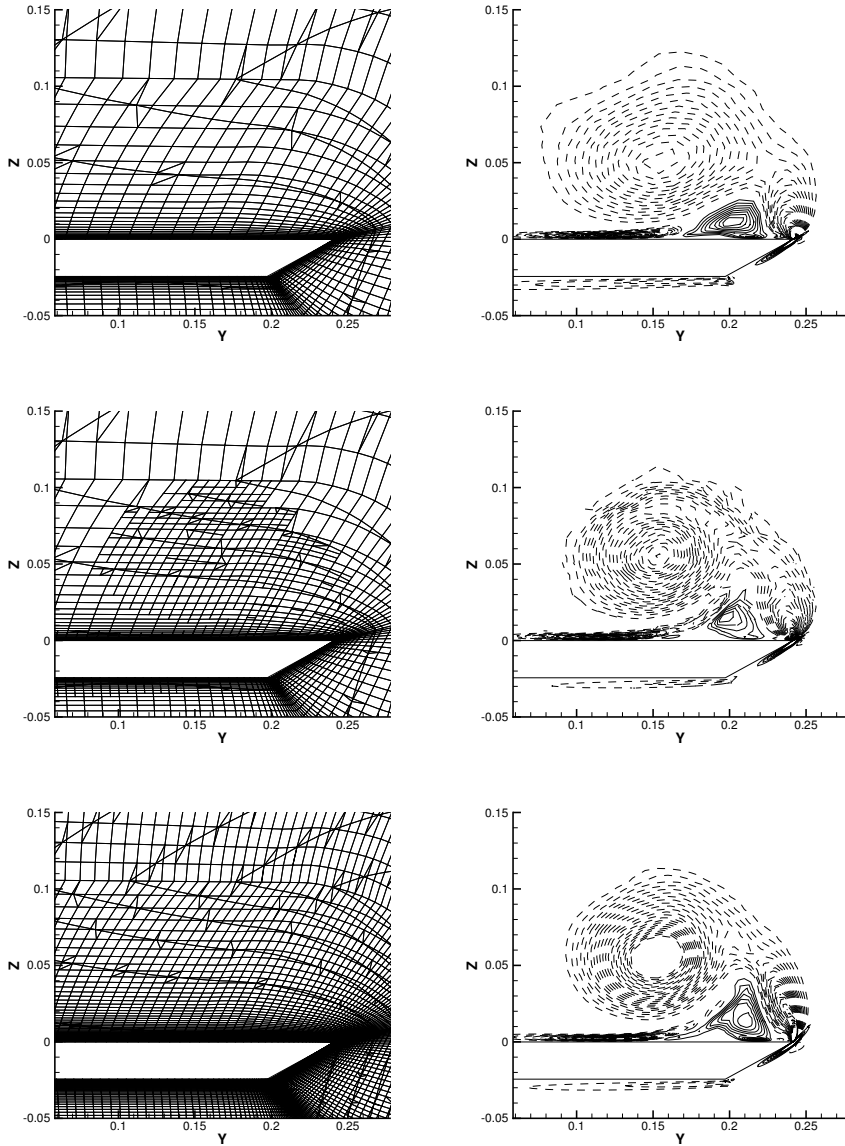


Figure 3.11: Grid and helicity isolines at cross section $x/c = 0.9$ for the coarse, adapted and fine mesh. The helicity ranges from -5 to 2 with step size 0.2 , the negative part being represented with solid lines, the positive part with dashed lines.

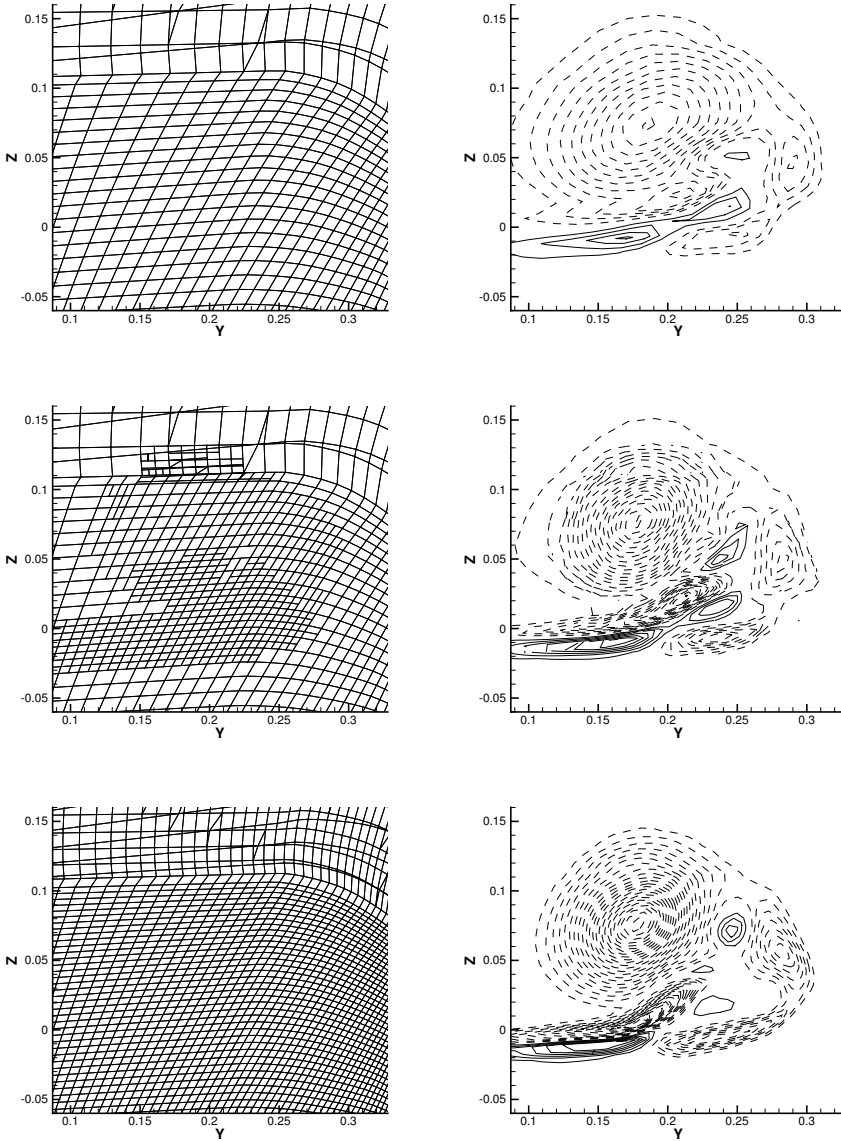


Figure 3.12: Grid and helicity isolines at cross section $x/c = 1.1$ for the coarse, adapted and fine mesh. The helicity ranges from -5 to 2 with step size 0.2 , the negative part being represented with solid lines, the positive part with dashed lines.

Chapter 4

Pseudo-time integration

In this chapter, we investigate a combination of two explicit Runge-Kutta schemes, one designed for inviscid flows and the other for viscous flows, to solve the non-linear system of algebraic equations of the space-time discretization of the compressible Navier-Stokes equations. To evaluate the effect of viscosity on the stability and performance of this method, we compare with an implicit-explicit Runge-Kutta method, where the viscous terms are treated implicitly and the inviscid terms explicitly.

4.1 Pseudo-time stepping methods

In this section, the different Runge-Kutta methods for the pseudo-time integration of system (3.25) are described. Here, the scaling in (3.25) with the diagonal matrix $|K^n|$ with entries $|K_j(t_{n+1})|$ is taken into the definition of the residual:

$$\mathcal{L}(\hat{U}^n; \hat{U}^{n-1}) \equiv \frac{1}{|K^n|} \left(\mathcal{L}^e(\hat{U}^n; \hat{U}^{n-1}) + \mathcal{L}^v(\hat{U}^n) \right),$$

in the remainder of this chapter.

First, we consider the explicit 5 stage Runge-Kutta method, which was successfully used to solve the system arising from the space-time discretization of the Euler equations in [44]. It is derived from a 5 stage Runge-Kutta method using the correction proposed by Melson e.a. [33] to enhance the stability of the pseudo-time integration. For details of the derivation and

the stability analysis for the Euler case we refer to [44]. This scheme is given by:

Algorithm 4.1 (EXI). Explicit Runge-Kutta method for inviscid flow with Melson correction.

1. Initialize $\hat{V}^0 = \hat{U}$.
2. For all stages $s = 1$ to 5 compute \hat{V}^s as:

$$(I + \alpha_s \lambda I) \hat{V}^s = \hat{V}^0 + \alpha_s \lambda (\hat{V}^{s-1} - \mathcal{L}(\hat{V}^{s-1}; \hat{U}^{n-1})).$$

3. Return $\hat{U} = \hat{V}^5$.

The Runge-Kutta coefficients at stage s are denoted by α_s and defined as: $\alpha_1 = 0.0791451$, $\alpha_2 = 0.163551$, $\alpha_3 = 0.283663$, $\alpha_4 = 0.5$ and $\alpha_5 = 1.0$. The matrix I represents the identity matrix. The coefficients were optimized to ensure rapid convergence to steady state. The factor λ is the ratio between the pseudo-time step $\Delta\tau$ and the physical time step: $\lambda = \Delta\tau/\Delta t$. The Melson correction consists in treating \hat{V} semi-implicitly, without this the scheme would become unstable for values of λ around one.

Second, we consider the implicit-explicit version of the EXI method. The residual \mathcal{L} defined in (3.16) consists of two parts: $\mathcal{L} = \mathcal{L}^e + \mathcal{L}^v$, where \mathcal{L}^e stems from the inviscid part of the compressible Navier-Stokes equations and \mathcal{L}^v from the viscous part. The implicit-explicit method can be derived by introducing a Newton matrix \mathcal{D} , which approximates the Jacobian of the viscous part of the residual:

$$\mathcal{D}\hat{V}^s \cong \mathcal{L}^v.$$

Here, the approximation consists of freezing the (non-linear) homogeneity tensor A at the previous Runge-Kutta stage $s - 1$. Details on the construction of \mathcal{D} are given in Appendix A.2. This approximation is relatively inexpensive compared with the Jacobian of the inviscid flux which would be required by a Newton solver, since A is readily available in the discretization. The implicit-explicit Runge-Kutta method can thus be written as:

Algorithm 4.2 (IMEX). Implicit-explicit Runge-Kutta method.

1. Initialize $\hat{V}^0 = \hat{U}$.
2. For all stages $s = 1$ to 5 compute \hat{V}^s by solving:

$$(I + \alpha_s \lambda (I + \mathcal{D})) \hat{V}^s = \hat{V}^0 + \alpha_s \lambda ((I + \mathcal{D}) \hat{V}^{s-1} - \mathcal{L}(\hat{V}^{s-1}; \hat{U}^{n-1})).$$

3. Return $\hat{U} = \hat{V}^5$.

The coefficients α_s are the same as in Algorithm 4.1. Note that the diffusive terms \mathcal{L}^v in the residual \mathcal{L} are *not* replaced by the approximation, both methods solve the same non-linear system $\mathcal{L} = 0$ at steady-state. Clearly, the l.h.s. of the equation for \hat{V}^s is no longer a diagonal matrix, but a global sparse block matrix, therefore \hat{V}^s must be computed by solving the sparse linear system. We do so using the sparse iterative GMRES solver with Jacobi preconditioning, available in the PETSc package [36].

Finally, we consider the family of methods proposed by Kleb e.a. [28]. Based on an explicit 4 stage Runge-Kutta method, Kleb e.a. proposed a procedure to optimize the Runge-Kutta coefficients for various situations depending on the cell Reynolds number. The method is given by:

Algorithm 4.3 (EXV). Explicit Runge-Kutta method for viscous flows.

1. Initialize $\hat{V}^0 = \hat{U}$.
2. For all stages $s = 1$ to 4 compute \hat{V}^s as:

$$\hat{V}^s = \hat{V}^0 - \alpha_s \lambda \mathcal{L}(\hat{V}^{s-1}; \hat{U}^{n-1}).$$

3. Return $\hat{U} = \hat{V}^4$.

Here, the Runge-Kutta coefficients at stage s are defined as: $\alpha_1 = 0.0178571$, $\alpha_2 = 0.0568106$, $\alpha_3 = 0.174513$ and $\alpha_4 = 1$; a summary of the derivation of these values is given in Appendix A.3. With these coefficients, the stability domain of the Runge-Kutta method is very different from the one associated with the classic 4 stage Runge-Kutta method for inviscid flows. Notice that we do not apply the Melson correction to this scheme because

we will not use it for values of λ around one, for reasons which will become clear in the next section.

The EXI method is designed for inviscid flows, while the EXV method is designed for viscous flows. In aerodynamical applications, however, one encounters both flow regimes in the same simulation: the flow is inviscid in the far-field *and* viscous in boundary layers. Therefore, we will seek to combine both methods, based on their stability properties. The advantage of such a combination is that it remains local, contrary to implicit-explicit methods or fully implicit methods.

4.2 Stability analysis

The methods discussed in the previous section can all be applied to solve the system of non-linear equations (3.25) given by the space-time discretization of the compressible Navier-Stokes equations, provided a suitable pseudo-time stability constraint is satisfied. In this section, we derive these constraints.

4.2.1 The model problem

Rigorous stability analysis of numerical methods for the Navier-Stokes equations is extremely difficult and will not be attempted. Instead, in order to derive practical stability constraints, the method is required to be stable for the scalar advection-diffusion equation [28, 52]:

$$u_t + a u_x = d u_{xx}, \quad (x, t) \in \mathbb{R} \times \mathbb{R}^+,$$

with $a > 0$ the advection constant and $d > 0$ the diffusion constant. The domain is divided into uniform rectangular elements Δt by Δx . The space-time discontinuous Galerkin method using the linear basis functions described in Chapter 3 gives the following discrete system for the vector of expansion coefficients \hat{u}^n at time level n :

$$\mathcal{L}(\hat{u}^n; \hat{u}^{n-1}) \equiv (\mathcal{L}^a + \mathcal{L}^d)\hat{u}^n + \mathcal{L}^t\hat{u}^{n-1} = 0, \quad (4.1)$$

The (block tridiagonal) inviscid part of the stencil depends on the Courant number:

$$\sigma = \frac{a\Delta t}{\Delta x}, \quad (4.2)$$

and is given by:

$$\mathcal{L}^a = \left[\begin{array}{ccc|ccc} -\sigma & -\sigma & \sigma & 1 + \sigma & \sigma & -\sigma & 0 & 0 & 0 \\ \sigma & \sigma & -\sigma & -\sigma & \frac{1}{3} + \sigma & \sigma & 0 & 0 & 0 \\ \sigma & \sigma & -\frac{4}{3}\sigma & -2 - \sigma & -\sigma & 2 + \frac{4}{3}\sigma & 0 & 0 & 0 \end{array} \right].$$

The right block is zero because the advective numerical flux is upwind ($a > 0$). The (block tridiagonal) viscous part of the stencil depends on the diffusive Von Neumann number:

$$\delta = \frac{d\Delta t}{(\Delta x)^2}, \quad (4.3)$$

as well as on the stabilization constant η and is given by:

$$\mathcal{L}^d = \delta \left[\begin{array}{ccc|ccc} -2\eta & 1 - 2\eta & 2\eta & 4\eta & 0 & -4\eta & -2\eta & -1 + 2\eta & 2\eta \\ -1 + 2\eta & -2 + 2\eta & 1 - 2\eta & 0 & 4\eta & 0 & 1 - 2\eta & -2 + 2\eta & -1 + 2\eta \\ 2\eta & -1 + 2\eta & -\frac{13}{6}\eta & -4\eta & 0 & \frac{13}{3}\eta & 2\eta & 1 - 2\eta & -\frac{13}{6}\eta \end{array} \right].$$

The (block diagonal) part of the stencil related to the previous space-time slab is given by:

$$\mathcal{L}^t = \left[\begin{array}{ccc} -1 & 0 & 0 \\ 0 & -\frac{1}{3} & 0 \\ 2 & 0 & 0 \end{array} \right].$$

Note that the matrix has a periodic block Toeplitz structure with 3×3 blocks, written symbolically as:

$$\mathcal{L} = [L | D | U],$$

with L the block-lower, D the block-diagonal and U the block-upper part of the matrix. The system of algebraic equations (4.1) resulting from the space-time discontinuous Galerkin discretization of the model problem is solved using the pseudo-time stepping methods described in the previous section. Since the stability in pseudo-time of the Runge-Kutta methods is only affected by the transients, we only consider the homogeneous part of the linear system (4.1). Thus, the pseudo-time equation for the model problem becomes:

$$\frac{\partial \hat{u}^n}{\partial \tau} = -\frac{1}{\Delta t} (\mathcal{L}^a + \mathcal{L}^d) \hat{u}^n. \quad (4.4)$$

4.2.2 Stability of the EXI and EXV method

The stability analysis of the EXI and EXV method is similar and therefore treated simultaneously in this section. The vector of expansion coefficients in element j is assumed to be a Fourier mode:

$$\hat{u}_j^n = \hat{u}^F \exp(i\theta j)$$

with \hat{u}^F the amplitude of the mode, $i = \sqrt{-1}$ and $\theta \in (-\pi, \pi]$. With this assumption, the Fourier transform of the discrete system becomes:

$$\mathbf{FT}(\mathcal{L})(\theta) = L \exp(-i\theta) + D + U \exp(i\theta),$$

with L the block-lower, D the block-diagonal and U the block-upper part of the matrix. We begin by noticing that the matrix $\mathbf{FT}(\mathcal{L})(\theta)$ is non-singular and can be diagonalized as QMQ^{-1} , with Q the matrix of right eigenvectors and M the diagonal matrix with the (complex) eigenvalues $\mu_i(\theta)$ with $i = 1, 2, 3$. Using this property and introducing the new vector $w = Q^{-1}\hat{u}^n$, reduces equation (4.4) to the simple scalar test model:

$$\frac{\partial w_i}{\partial \tau} = -\frac{\mu_i(\theta)}{\Delta t} w_i, \quad \text{for } i = 1, 2, 3. \quad (4.5)$$

Note that the summation convention does not apply here and to avoid confusion, we consider the generic scalar model problem of the form:

$$\frac{\partial w}{\partial \tau} = -\frac{\mu}{\Delta t} w,$$

with the understanding that the scalars w and μ can be any of the three components of the corresponding vectors. When applying the EXI method to this model equation, the Runge-Kutta stages w^s are computed as:

$$(1 + \alpha_s \lambda) w^s = w^0 + \alpha_s \lambda (1 - \mu) w^{s-1},$$

with $\lambda = \Delta \tau / \Delta t$ and for the EXV as:

$$w^s = w^0 - \alpha_s \lambda \mu w^{s-1}.$$

Using these equations the relation between two consecutive pseudo-time steps can easily be derived and is written in generic form as:

$$w^n = G(-\lambda \mu) w^{n-1},$$

with G the algorithm dependent amplification factor. Starting with an initial condition w^{init} , we obtain after n steps:

$$w^n = G(-\lambda\mu)^n w^{\text{init}}.$$

In stability analysis, we are interested in the behavior of a perturbation of the initial condition (see for example [52]), and, due to linearity, the amplification of the perturbation is the same as the amplification of w . Clearly, the perturbation is bounded if $\|G^n\|$ is bounded, where $\|\cdot\|$ denotes the Euclidian (or discrete l_2) norm [22, 52]. Therefore, in view of (4.5), a sufficient condition for stability is that the values $-\lambda\mu_i(\theta)$ for $i = 1, 2, 3$ and $\theta \in (-\pi, \pi]$ all lie inside the stability domain S given by:

$$S = \{z \in \mathbb{C} : |G(z)| \leq 1\}.$$

Remember that the discretization of the advection-diffusion equation only depends on the Courant number (4.2), the diffusive Von Neumann number (4.3) and the constant η . For given values of these numbers, the factor λ of the Runge-Kutta algorithm should be chosen such that $-\lambda\mu_i(\theta)$ lies inside the stability domain S . Once a suitable λ is found, it is convenient to express the stability in terms of the *pseudo-time* Courant and diffusive Von Neumann numbers: $\sigma_{\Delta\tau} = \lambda\sigma$ and $\delta_{\Delta\tau} = \lambda\delta$. Hence, for stability, the pseudo-time step $\Delta\tau$ must satisfy the pseudo-time Courant-Friedrichs-Levy (CFL) condition and the pseudo-time diffusive Von Neumann condition:

$$\Delta\tau \leq \Delta\tau^a \equiv \frac{\sigma_{\Delta\tau}\Delta x}{a} \quad \text{and} \quad \Delta\tau \leq \Delta\tau^d \equiv \frac{\delta_{\Delta\tau}(\Delta x)^2}{d}.$$

We distinguish between flow regimes by introducing the *cell* Reynolds number, defined as:

$$\text{Re}_{\Delta x} \equiv \frac{a\Delta x}{d}. \quad (4.6)$$

In aerodynamical computations, the flow is inviscid in most of the domain, yet significant viscous effects occur in the boundary layer near the airfoil. Therefore we will consider the following regimes:

1. Steady-state, inviscid: $\sigma = 100$ and $\text{Re}_{\Delta x} = 100$,
2. Steady-state, viscous: $\sigma = 100$ and $\text{Re}_{\Delta x} = 0.01$,

3. Time-dependent, inviscid: $\sigma = 1$ and $\text{Re}_{\Delta x} = 100$,
4. Time-dependent, viscous: $\sigma = 1$ and $\text{Re}_{\Delta x} = 0.01$.

The pseudo-time diffusive Von Neumann condition can be expressed in terms of the pseudo-time Courant number and the cell Reynolds number as:

$$\Delta\tau^d = \frac{\delta_{\Delta\tau}\text{Re}_{\Delta x}}{\sigma_{\Delta\tau}} \Delta\tau^a$$

Thus, the CFL condition is the most restrictive for the inviscid flow regime, the diffusive Von Neumann condition for the viscous flow regime and the threshold between both is given by $\delta_{\Delta\tau}\text{Re}_{\Delta x} = \sigma_{\Delta\tau}$.

The stability domains of the EXI and EXV method and the values $-\lambda\mu_i(\theta)$ are plotted in Figures 4.1, 4.2, 4.3 and 4.4 for $i = 1, 2, 3$ and a discrete series of $\theta = -0.96\pi, 0.92\pi, \dots, 1.00\pi$. For inviscid flow regimes with pseudo-time Courant number around $\sigma_{\Delta\tau} = 1.7$, the EXI method is stable and the EXV is unstable, but for viscous flow regimes with pseudo-time diffusive Von Neumann number $\delta_{\Delta\tau} = 0.8$, the converse holds. Stability constraints for which both methods are stable are given in Table 4.1, confirming that the EXI method is preferable in the inviscid regime and the EXV in the viscous regime. Therefore, we combine the EXI and EXV by looking at the cell Reynolds number, and, for that particular cell, deploy whichever scheme has the mildest stability restriction.

Remark. The stabilization parameter η has a significant effect on the stability of the pseudo-time integration: as η increases, the pseudo-time diffusive Von Neumann number decreases proportionally. Therefore η should be taken as small as allowed in the discontinuous Galerkin discretization, in general equal to the number of faces of an element [10, 41].

Remark. The Melson correction is applied to the EXI scheme to ensure stability for values of λ around one, which is the case for the time-dependent inviscid flow regime (Figure 4.3). For all other flow regimes, λ is small and the Melson correction vanishes. Since we only apply the EXV scheme in the viscous flow regime, the Melson correction is unnecessary for this scheme.

4.2.3 Stability of the IMEX method

The IMEX method solves the inviscid part of the equations with the EXI method and treats the viscous part implicitly. The main idea is that the

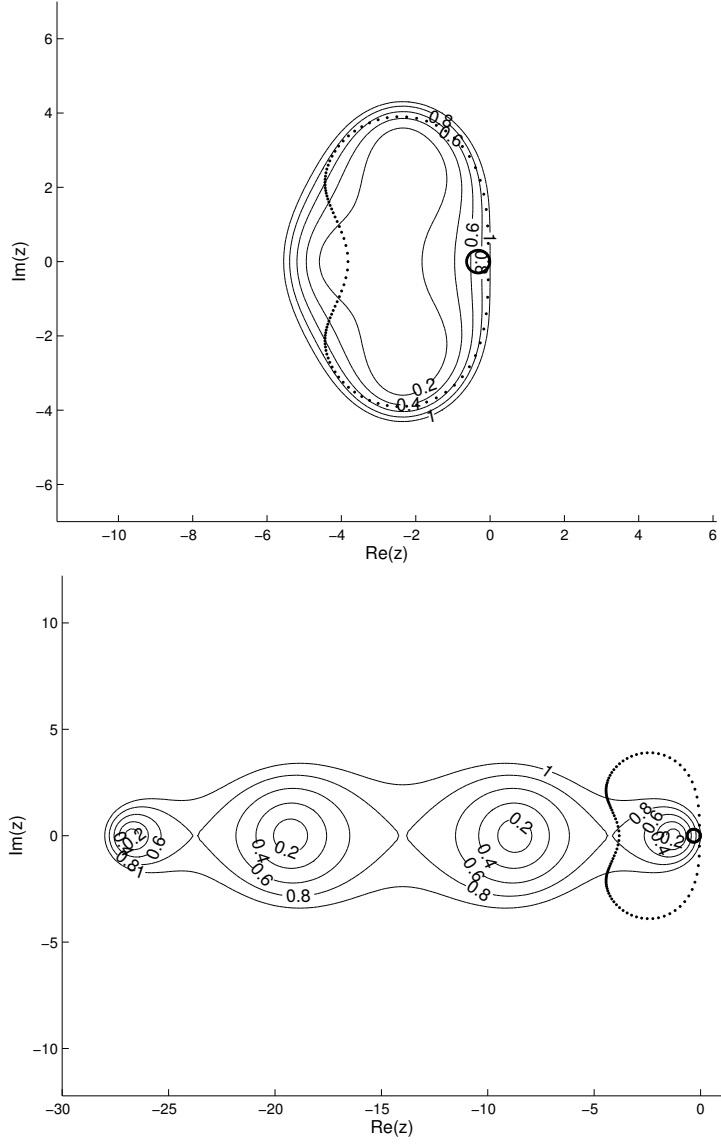


Figure 4.1: The stability domain S and values $-\lambda\mu_i$ (dots) for the EXI method (top) and EXV method (bottom) in the steady-state inviscid flow regime with $\lambda = 1.8 \cdot 10^{-2}$. The pseudo-time CFL number is 1.8 and for this constraint only the EXI method is stable.

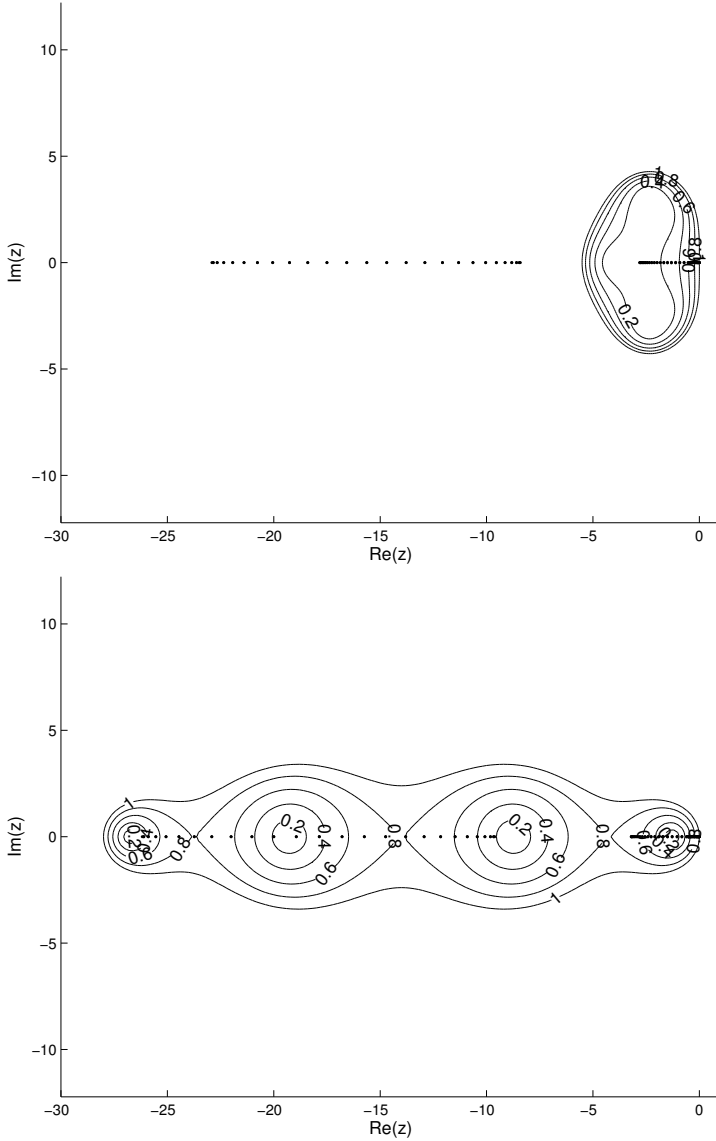


Figure 4.2: The stability domain S and values $-\lambda\mu_i$ (dots) for the EXI method (top) and EXV method (bottom) in the steady-state viscous flow regime with $\lambda = 8 \cdot 10^{-5}$. The pseudo-time diffusive Von Neumann number is 0.8 and for this constraint only the EXV method is stable.

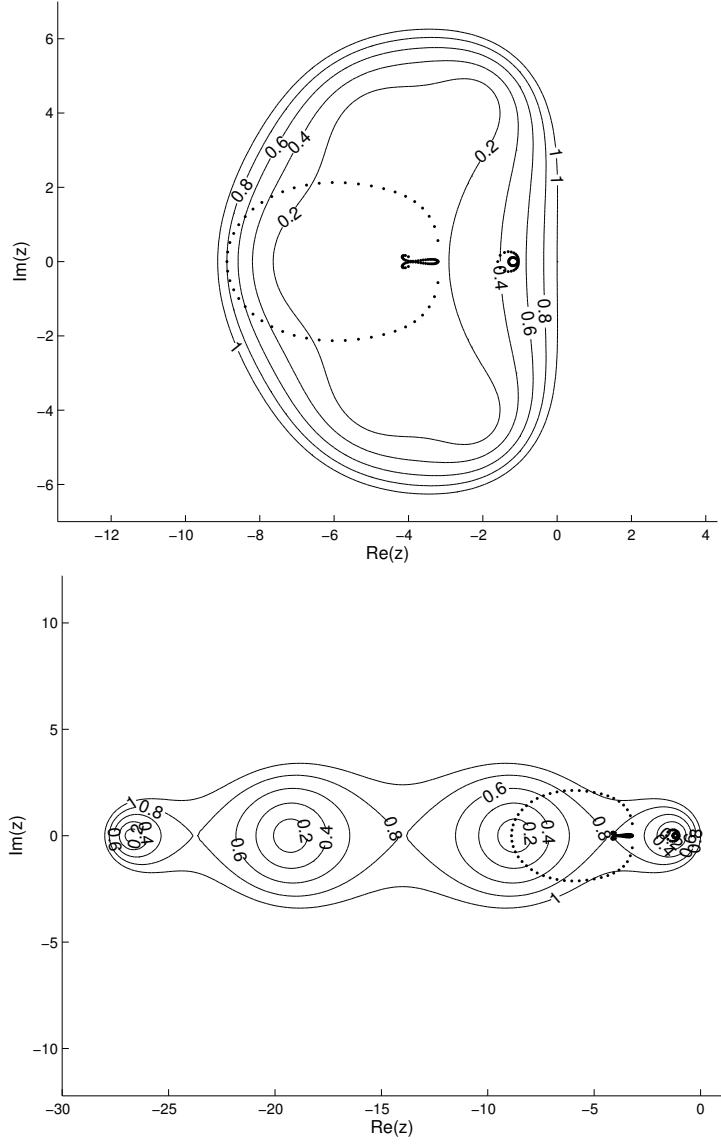


Figure 4.3: The stability domain S and values $-\lambda\mu_i$ (dots) for the EXI method (top) and EXV method (bottom) in the time-dependent inviscid flow regime with $\lambda = 1.6$. The pseudo-time CFL number is 1.6 and for this constraint only the EXI method is stable.

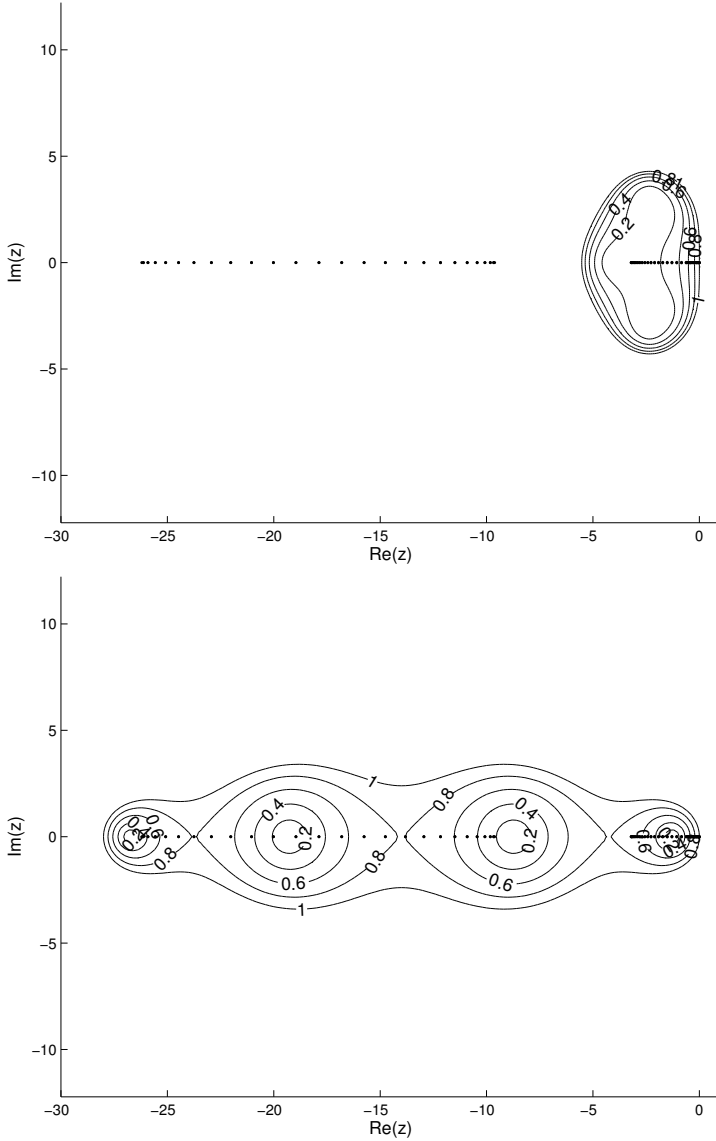


Figure 4.4: The stability domain S and values $-\lambda\mu_i$ (dots) for the EXI method (top) and EXV method (bottom) in the time-dependent viscous flow regime with $\lambda = 8 \cdot 10^{-3}$. The pseudo-time diffusive Von Neumann number is 0.8 and for this constraint only the EXV method is stable.

Table 4.1: Stability constraints of the EXI and EXV methods in various flow regimes.

	flow regime		stability constraints	
	σ	$\text{Re}_{\Delta x}$	EXI	EXV
Steady-state, inviscid	100	100	$\sigma_{\Delta\tau} \leq 1.8$	$\sigma_{\Delta\tau} \leq 0.3$
Steady-state, viscous	100	0.01	$\delta_{\Delta\tau} \leq 0.1$	$\delta_{\Delta\tau} \leq 0.8$
Time-dependent, inviscid	1	100	$\sigma_{\Delta\tau} \leq 1.6$	$\sigma_{\Delta\tau} \leq 1.0$
Time-dependent, viscous	1	0.01	$\delta_{\Delta\tau} \leq 0.1$	$\delta_{\Delta\tau} \leq 0.8$

stability should now only depend on the inviscid part, so only the CFL condition has to be satisfied. This represents the ideal situation where viscosity does not affect stability, needed to assess the influence of the diffusive Von Neumann condition on the fully explicit methods. Unfortunately, the matrices \mathcal{L}^a and \mathcal{L}^d in (4.4) do not commute, making it impossible to obtain a scalar model problem through diagonalization, as was done for the explicit method. Stability analysis of IMEX methods for general non-commuting matrices is still largely an open problem, although recently, for the related W-methods, results have been presented by Ostermann [35]. In this section, we will prove stability of the IMEX method by directly estimating the norm of the amplification factor G .

For the IMEX method the Runge-Kutta stages \hat{v}^s are computed by solving the sparse linear system:

$$(I + \alpha_s \lambda (I + \mathcal{L}^d)) \hat{v}^s = \hat{v}^0 + \alpha_s \lambda (I - \mathcal{L}^a) \hat{v}^{s-1}. \quad (4.7)$$

The starting point of our analysis is the fact that \mathcal{L}^d is a Hermitian matrix, therefore $\mathcal{L}^d = QMQ^T$ where Q is a unitary matrix and M the diagonal matrix with the eigenvalues μ_i of \mathcal{L}^d . The eigenvalues of \mathcal{L}^d are real and positive, and can be computed as the eigenvalues $\mu_i(\theta)$ with $i = 1, 2, 3$ and $\theta \in (-\pi, \pi]$ of the corresponding Fourier transform:

$$\mathbf{FT}(\mathcal{L}^d)(\theta) = L^d \exp(-i\theta) + D^d + U^d \exp(i\theta),$$

with L^d the block-lower, D^d the block-diagonal and U^d the block-upper part of the matrix \mathcal{L}^d . For a unitary matrix $Q^{-1} = Q^T$ and the l.h.s. of

(4.7) can be written as:

$$\begin{aligned} I + \alpha_s \lambda (I + \mathcal{L}^d) &= Q(I + \alpha_s \lambda (I + M))Q^T \\ &= QM_s Q^T, \end{aligned} \quad (4.8)$$

with M_s the diagonal matrix with values $1 + \alpha_s \lambda (1 + \mu_i)$. Introducing the decomposition (4.8) into (4.7) gives:

$$\begin{aligned} M_s w^s &= w^0 + \alpha_s \lambda Q^T (I - \mathcal{L}^a) Q w^{s-1}, \\ &= w^0 + \alpha_s \lambda P_a w^{s-1}, \end{aligned} \quad (4.9)$$

with $w^s = Q^T \hat{v}^s$ and $P_a = Q^T (I - \mathcal{L}^a) Q$. Therefore, the relation between two consecutive pseudo-time steps is: $w^n = G w^{n-1}$ with the amplification matrix G defined as:

$$G = M_5^{-1} (I + \alpha_5 \lambda P_a M_4^{-1} (I + \alpha_4 \lambda P_a \cdots M_1^{-1} (I + \alpha_1 \lambda P_a))).$$

If $\|G\| \leq 1$, then $\|G^n\| \leq 1$ and the method is stable. Our stability analysis aims at a direct estimation of this norm, therefore we consider the following upper bound:

$$\begin{aligned} \|G\| &\leq \|M_5^{-1}\| (1 + \alpha_5 \lambda \|P_a\| \|M_4^{-1}\| \\ &\quad (1 + \alpha_4 \lambda \|P_a\| \cdots \|M_1^{-1}\| (1 + \alpha_1 \lambda \|P_a\|))). \end{aligned}$$

The matrices M_s^{-1} are equal to:

$$M_s^{-1} = \text{diag} \left(\frac{1}{1 + \alpha_s \lambda (1 + \mu_1)}, \dots, \frac{1}{1 + \alpha_s \lambda (1 + \mu_n)} \right),$$

with μ_i the eigenvalues of \mathcal{L}^d . The Euclidian norm of M_s^{-1} can be estimated as:

$$\|M_s^{-1}\| = \max_{i \in \{1, \dots, n\}} \frac{1}{1 + \alpha_s \lambda (1 + \mu_i)} < \frac{1}{1 + \alpha_s \lambda},$$

since $\mu_i, \alpha_s, \lambda > 0$. Using this estimation, the upper bound for the Euclidian norm of G is then provided by the following estimate:

$$\begin{aligned} \|G\| &\leq \frac{1}{1 + \alpha_5 \lambda} \left(1 + \alpha_5 \lambda \|P_a\| \frac{1}{1 + \alpha_4 \lambda} \right. \\ &\quad \left. \left(1 + \alpha_4 \lambda \|P_a\| \cdots \frac{1}{1 + \alpha_1 \lambda} (1 + \alpha_1 \lambda \|P_a\|) \right) \right). \end{aligned}$$

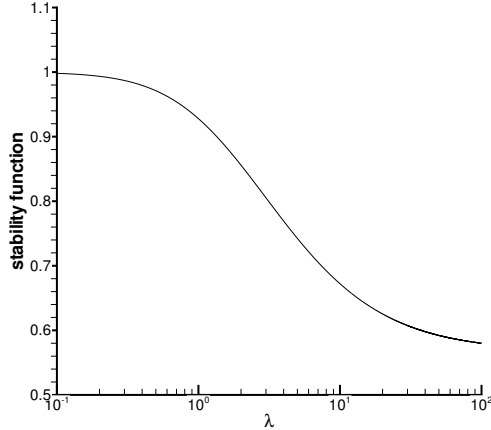


Figure 4.5: The stability function f for $\|P_a\| = 1$.

The r.h.s. of this equation is called the stability function, denoted by $f(\lambda, \|P_a\|)$ and plotted for $\|P_a\| = 1$ in Figure 4.5. If $\|P_a\| < 1$ we find ourselves below the curve in Figure 4.5, therefore: $\|P_a\| \leq 1 \Rightarrow f(\lambda, \|P_a\|) \leq 1 \Rightarrow \|G\| \leq 1$ meaning $\|P_a\| \leq 1$ is a sufficient condition for stability of the implicit-explicit method. Since the matrix P_a is defined as $P_a = Q^T(I - \mathcal{L}^a)Q$, with Q a unitary matrix (hence $\|Q\| = 1$), this implies that the stability of the IMEX method is only determined by the following condition:

$$\|I - \mathcal{L}^a\| \leq 1.$$

Since \mathcal{L}^a only depends on the Courant number (4.2), this condition implies that the IMEX method is stable independently of the diffusive Von Neumann number, and only the CFL condition has to be satisfied.

The fact that the IMEX method does not require the diffusive Von Neumann condition for stability makes it suitable to evaluate (by comparison) the effect of viscosity on the stability and performance of fully explicit methods. We can thus assess whether the combined EXI and EXV method effectively relieves the diffusive Von Neumann constraint. We will address this issue in the following section through numerical experiments.

4.3 Convergence results

In this section, three benchmark problems for the compressible Navier-Stokes equations are considered. First, we explain how to compute the local pseudo-time step size. Then, we compare the performance of the EXI method with the combined EXI and EXV method and the IMEX method to assess the effect of viscosity on the stability and performance of the explicit methods. Finally, we show the feasibility of the explicit methods for the flow around a 3D delta wing.

4.3.1 Local pseudo-time step-size

The space-time method is unconditionally stable in *physical* time, which allows us to take any desired physical time step Δt and solve the non-linear system using the pseudo-time stepping methods discussed in this paper. Since accuracy is not an issue in *pseudo-time* we can use local steps $(\Delta\tau)_K$, which are determined for each element K as:

$$(\Delta\tau)_K = \begin{cases} \min\{(\Delta\tau)_K^i, (\Delta\tau)_K^v\} & \text{for EXI and EXV,} \\ (\Delta\tau)_K^i & \text{for IMEX.} \end{cases}$$

The local inviscid and viscous pseudo-time steps are computed as:

$$\begin{aligned} (\Delta\tau)_K^i &= \frac{\sigma_{\Delta\tau} d_K}{\lambda_K^i} & \text{with } \lambda_K^i &= \max\{|u_K - a_K|, |u_K + a_K|\}, \\ (\Delta\tau)_K^v &= \frac{\delta_{\Delta\tau} (d_K)^2}{\lambda_K^v} & \text{with } \lambda_K^v &= \max\left\{\frac{1}{c_v} \frac{\kappa_K}{\rho_K}, \frac{4}{3} \frac{\mu_K}{\rho_K}\right\}, \end{aligned}$$

where $\sigma_{\Delta\tau}$ is the pseudo-time Courant number, $\delta_{\Delta\tau}$ the pseudo-time diffusive Von Neumann number (both from Table 4.1) and d_K the diameter of the circle inscribed in element K . The cell Reynolds number Re_K is defined as:

$$\text{Re}_K = \frac{\lambda_K^i d_K}{\lambda_K^v},$$

and λ^i represents the absolute maximum of the eigenvalues of the inviscid Jacobian and λ^v of the viscous Jacobian. These wave-speeds are computed at the element faces during the flux evaluation and we take the maximum over all faces belonging to element K , where u is the flow speed, a the speed

of sound and ρ the density. Note that the specific heat at constant volume c_v is constant throughout the domain but the dynamic viscosity μ and the thermal conductivity coefficient κ depend on the temperature in element K , see [29] and Chapter 2. Even though the stability analysis was only done for the advection-diffusion equation on a periodic domain, the resulting stability constraints proved also adequate in case of the compressible Navier-Stokes equations.

4.3.2 NACA0012 airfoil

Steady-state case. To test the effect of viscosity on the stability and performance of the explicit methods, we have chosen the A1 case described in [12] for the viscous flow past a NACA0012 airfoil. This case has become a standard benchmark for discontinuous Galerkin methods for the compressible Navier-Stokes equations as it was treated in the seminal paper by Bassi and Rebay [3]. The Prandtl number is fixed at $\text{Pr} = 0.72$, the far-field Mach and Reynolds numbers at $M_\infty = 0.8$ and $\text{Re}_\infty = 73$, respectively. With an angle of attack $\alpha = 12^\circ$ this gives a highly viscous flow. Since it is a steady flow, we take one huge dimensionless physical time step $\Delta t = 10^{21}$, see Chapter 2 for the non-dimensionalization. For laminar viscous flow, the boundary layer thickness at the nose of the airfoil is estimated as [37]:

$$\delta \approx 5/\sqrt{\text{Re}_\infty},$$

which means that $\delta \approx 0.6$ in this case. To compute the boundary layer with reasonable accuracy, we have chosen a C-type mesh with 112×38 elements, which is a coarsening of the 224×76 grid shown in Figure 4.6.

The local Mach number isolines and the convergence results are presented in Figure 4.7. The cell Reynolds number varies between 0.09 and 88 which explains why the convergence of the EXI method is very slow: one order of convergence in 80 000 pseudo-time iterations. For the combined EXI and EXV method, one order of convergence requires ten times fewer iterations, which is roughly what we expect from the stability analysis as the diffusive Von Neumann number for the EXV method is eight times larger (Table 4.1). Furthermore, the combined method achieves seven orders of convergence within 50 000 steps. The IMEX method performs much better in terms of iterations: six orders of convergence in 3 000 pseudo-time steps. Therefore, we conclude that combining the EXI method with the EXV

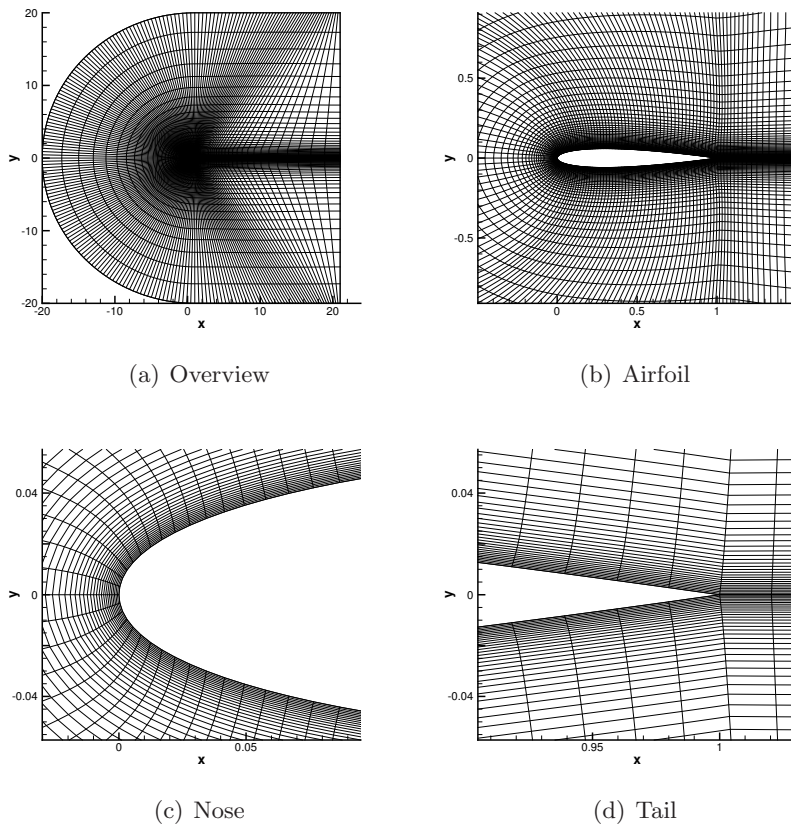


Figure 4.6: Details of the NACA0012 C-grid with 224×76 elements.

method significantly improves the convergence in pseudo-time but does not completely rule out the effect of viscosity on the stability restriction and thus on the convergence of explicit methods.

Remark. For the IMEX method, the implicit linear system must be solved at each Runge-Kutta stage. We do so using the block sparse GMRES solver with Jacobi preconditioner available in the PETSc package [36]. In terms of CPU-time, the IMEX method converges significantly slower than the combined EXI and EXV method in all considered cases and is only pursued to evaluate the effect of the viscosity on the fully explicit methods.

Time-dependent case. To test the performance of the explicit pseudo-time stepping methods for time-dependent simulations, we have chosen the A7 case described in [12]. The Prandtl number is fixed at $\text{Pr} = 0.72$, the far-field Mach and Reynolds numbers at $M_\infty = 0.85$ and $\text{Re}_\infty = 10^4$, respectively, and the angle of attack is $\alpha = 0^\circ$, which gives a time-dependent viscous flow with a shock and vortex shedding. To capture the vortex shedding, we have estimated the velocity of the vortices to be around $u = 0.8$ and the diameter to be around $D = 0.1$, which, together with a physical time step $\Delta t = 0.05$, gives a physical Courant number $u\Delta t/D = 0.4$: small enough for accuracy in time. To compute the boundary layer in the A7 case with reasonable accuracy, we use the C-type grid for viscous flows shown in Figure 4.6 with 224×76 elements which offers more than 30 elements in the boundary layer with thickness $\delta \approx 0.05$.

In the A7 case, for each physical-time step, the EXI method achieves three orders of convergence in 1000 pseudo-time steps, see Figure 4.8. The physical time-step is already fairly small in order to capture the vortex shedding, which explains the relatively small number of pseudo-time steps needed to solve the system. In this case, the cell Reynolds number varies between 2 and 14 000, which explains why the difference between the EXI and the combined EXI and EXV method is less than in the A1 case. Still, the convergence is twice as fast. The IMEX method requires 200 iterations, which is two and half times faster in terms of iterations than the combined EXI and EXV method. Therefore, as in the steady-state case A1, we conclude that combining the EXI method with the EXV method significantly improves the convergence in pseudo-time but viscosity still has some influence on the convergence of the combined EXI and EXV method.

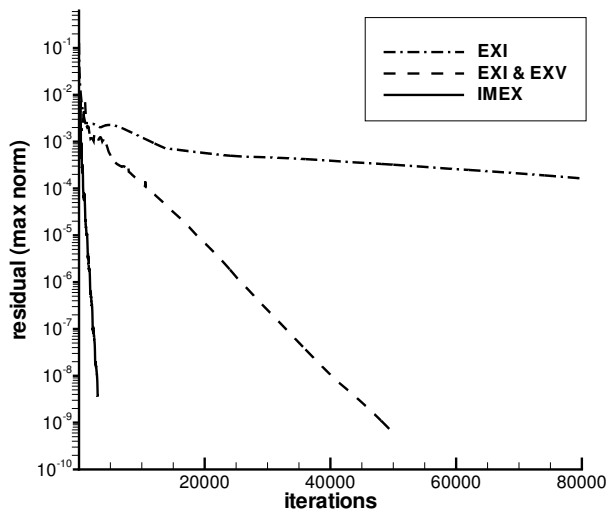
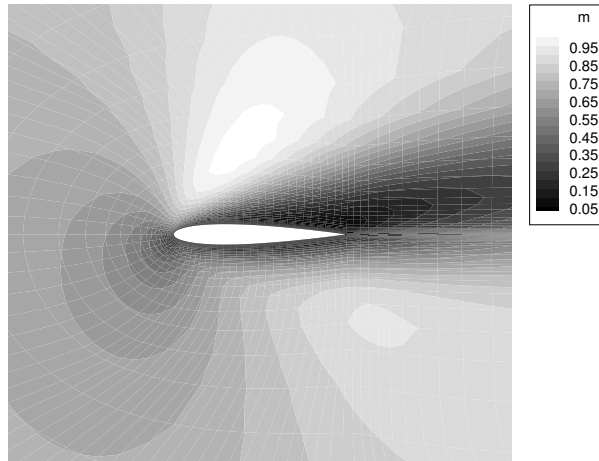


Figure 4.7: Local Mach numbers for the A1 case ($M_\infty = 0.8$, $Re_\infty = 73$, $\alpha = 12^\circ$) on the 112×38 grid and convergence to steady-state for the different pseudo-time stepping methods.

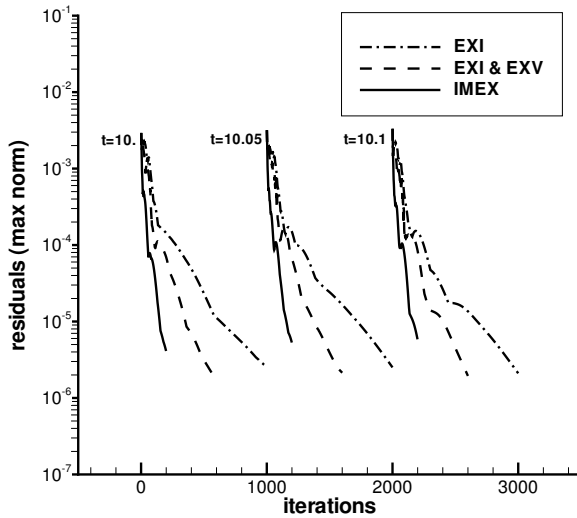
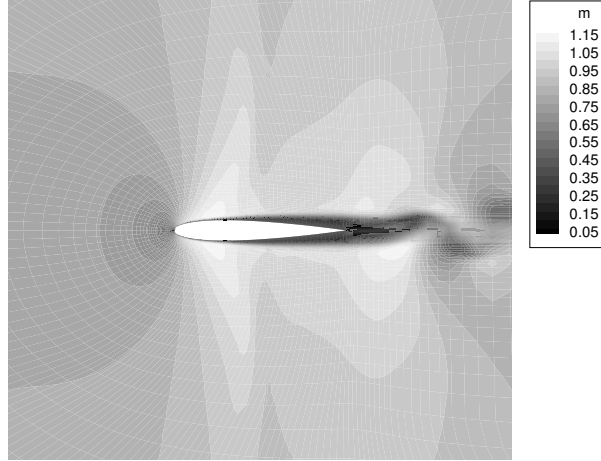


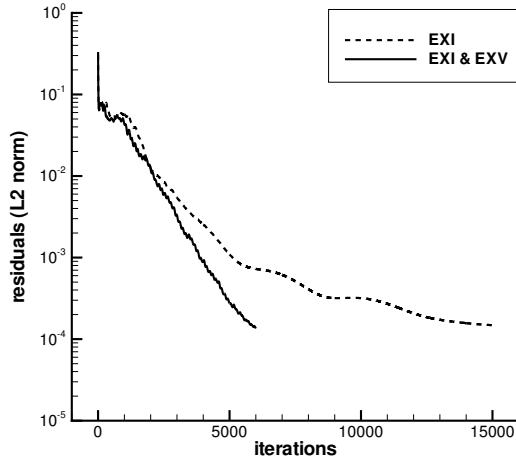
Figure 4.8: Local Mach numbers for the A7 case ($M_\infty = 0.85$, $Re_\infty = 10^4$, $\alpha = 0^\circ$) on the 224×76 grid at $t = 10$ and convergence in pseudo-time for three physical time steps with the different pseudo-time stepping methods.

4.3.3 Delta wing

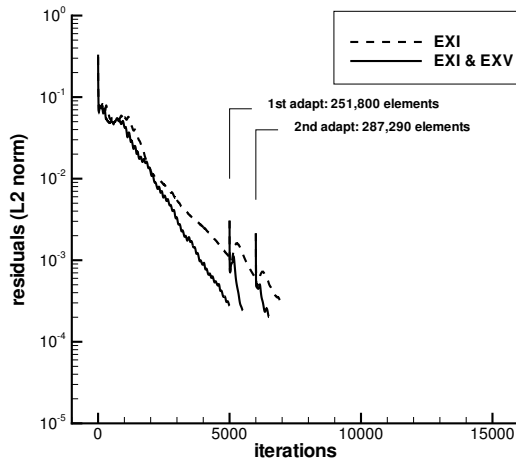
In the previous chapter, we presented results for the steady-state flow around a 3D delta wing at $M_\infty = 0.3$, $Re = 4 \cdot 10^4$ and $\alpha = 12.5^\circ$ on an adapted mesh. The mesh adaptation was based on the vorticity in order to capture the primary and secondary vortices along both sides of the sharp leading edge of the wing, see Chapter 3 for the wing geometry, an impression of the mesh and the flow field. The results for the pressure coefficient and the helicity showed good agreement with those obtained on a fine mesh (1 650 000 elements). Therefore, we concluded that the solution adaptive space-time method results in significant cost savings for this application. Here, we show the convergence for this case using the EXI and combined EXI and EXV method. The IMEX method was not attempted in this case due to the size of the linear system.

We start by computing the solution on the coarse mesh with 208 896 elements, with uniform flow as initial condition. Based on a comparison with the solution on the fine mesh, we consider the case solved after three orders of convergence. In Figure 4.9, we see that the combined EXI and EXV method meets our convergence criterion in five thousand pseudo-time steps while the EXI method takes fifteen thousand steps. Next, anisotropic mesh refinement is applied in vortex regions in such a way that eventually the mesh in the region around a vortex is as uniform as possible. A cell is refined in any of the three directions whenever the vorticity is greater than $2a_\infty/c$ and the mesh width in the given direction is greater than $0.01c$. In this way, the mesh around the primary vortices will eventually have a uniform resolution of $0.005c$. Mesh adaptation is based on the (more accurate) solution obtained with the combined EXI and EXV method. After each mesh adaptation, the residual has to be decreased again. We do so with the EXI and the combined EXI and EXV method and show the convergence results in Figure 4.9. The advantage of the combined EXI and EXV method becomes even more apparent, since, at each adaptation, the percentage of relatively small cells (and thus of small cell Reynolds numbers) increases. On the final adapted grid, the cell Reynolds number varies between 0.4 and 118 000, confirming the robustness of the combined EXI and EXV method for a wide range of cell Reynolds numbers.

The combined EXI and EXV method effectively relieves the viscous stability constraint and, in the next chapter, we investigate its potential as a smoother in a multigrid setting.



(a) coarse mesh with 208 896 elements



(b) mesh adaptations

Figure 4.9: Convergence to steady-state with the explicit pseudo-time stepping methods for the 3D delta wing ($M_\infty = 0.3$, $Re_\infty = 4 \cdot 10^4$, $\alpha = 12.5^\circ$) starting from uniform flow on a coarse mesh (top) and with two successive mesh adaptations (bottom).

Chapter 5

Multigrid analysis

The previously presented explicit Runge-Kutta methods are applied, in this chapter, as smoothers in a multigrid context. The scalar advection-diffusion equation is used as a model problem to study the multigrid convergence behavior with two-level Fourier analysis. A simple numerical example illustrates the multigrid method.

5.1 Model problem

5.1.1 Weak form

In this section we introduce the space-time discretization for which we study multigrid convergence. For that purpose we introduce the open space-time domain:

$$\Omega = \{(x, t) \mid x \in \mathbb{R}, t \in \mathbb{R}^+\},$$

with boundary $\partial\Omega = \{(x, t) \mid t = 0, x \in \mathbb{R}\}$ and consider the advection-diffusion equation in generic form:

$$-\nabla \cdot A \nabla u + b \cdot \nabla u = 0 \quad \text{in } \Omega, \quad u = u_0 \quad \text{on } \partial\Omega, \quad (5.1)$$

where:

$$\nabla = [\partial_x \quad \partial_t]^T, \quad b = [a \quad 1], \quad A = \begin{bmatrix} d & 0 \\ 0 & 0 \end{bmatrix}, \quad (5.2)$$

with $a > 0$ and $d > 0$. Next, to arrive at a DG discretization, we partition the domain Ω in regular rectangular cells of identical shape:

$$\Omega_j^n = \{(x, t) \mid jh < x < (j+1)h, n\Delta t < t < (n+1)\Delta t\}, \quad (5.3)$$

with $h > 0$, $\Delta t > 0$, $j \in \mathbb{Z}$ and $n \in \mathbb{N}$. Then $\Omega^n = \bigcup_{j \in \mathbb{Z}} \overline{\Omega_j^n}$ is a space-time slab. To find the discrete algebraic equations we introduce the discrete function space:

$$\mathcal{S}_h^n = \left\{ v_h^n \in L^2(\Omega) \mid v_h^n|_{\Omega_j^n} \in \mathcal{P}^k(\Omega_j^n), \forall j \in \mathbb{Z}, n \in \mathbb{N} \right\},$$

the space of piecewise polynomials of degree at most k in the coordinate directions. Then the discrete weak DG form of (5.1) reads ([10, 38] and 3): find $u_h^n \in \mathcal{S}_h^n$ such that:

$$B(u_h^n, v_h^n) = 0, \quad \forall v_h^n \in \mathcal{S}_h^n, \forall n \in \mathbb{N}, \quad (5.4)$$

with

$$\begin{aligned} B(u_h^n, v_h^n) &= \sum_{j \in \mathbb{Z}} \int_{\Omega_j^n} (A \nabla u_h^n) \cdot \nabla v_h^n \, d\Omega - \sum_{j \in \mathbb{Z}} \int_{\Gamma_j^n} \{ \{ A \nabla u_h^n \} \} \cdot \llbracket v_h^n \rrbracket \, ds \\ &- \sum_{j \in \mathbb{Z}} \int_{\Gamma_j^n} \{ \{ A \nabla v_h^n \} \} \cdot \llbracket u_h^n \rrbracket \, ds + \sum_{e \in \mathbb{Z}} \sum_{j \in \mathbb{Z}} \int_{\Omega_j^n} \eta_e (r_e(u_h^n) A) \cdot r_e(v_h^n) \, d\Omega \\ &- \sum_{j \in \mathbb{Z}} \int_{\Omega_j^n} \nabla v_h^n \cdot \mathbf{b} u_h^n \, d\Omega + \sum_{j \in \mathbb{Z}} \int_{\Gamma_j^{n-}} u_h^{n-} \mathbf{n}_j \cdot \mathbf{b} v_h^{n+} \, ds \\ &+ \sum_{j \in \mathbb{Z}} \int_{\Gamma_j^{n+}} u_h^{n+} \mathbf{n}_j \cdot \mathbf{b} v_h^{n+} \, ds, \end{aligned} \quad (5.5)$$

and A and \mathbf{b} defined in (5.2). Here, the first four terms in the righthand side are associated with the diffusion part of (5.1), where the term with stabilization parameter $\eta_e > 0$ prevents the discrete system of being indefinite. In the next section, we show how to compute this stabilization term. For a complete overview of DG methods for elliptic problems we refer to [1]. The common cell interface between two adjacent cells Ω_{j-1}^n and Ω_j^n in the time slab Ω^n is $\Gamma_j^n = \partial\Omega_{j-1}^n \cap \partial\Omega_j^n$. On this interface the jump operator $\llbracket \cdot \rrbracket$ and the average operator $\{ \{ \cdot \} \}$ are defined by:

$$\begin{aligned} \llbracket u_h^n(x, t) \rrbracket &= u_h^n(x, t)|_{\partial\Omega_{j-1}^n} \mathbf{n}_{j-1} + u_h^n(x, t)|_{\partial\Omega_j^n} \mathbf{n}_j, \quad \text{for } u_h^n \in \mathcal{S}_h^n, \\ \{ \{ \tau_h^n(x, t) \} \} &= \frac{1}{2} \left(\tau_h^n(x, t)|_{\partial\Omega_{j-1}^n} + \tau_h^n(x, t)|_{\partial\Omega_j^n} \right), \quad \text{for } \tau_h^n \in [\mathcal{S}_h^n]^2, \end{aligned}$$

with $x \in \Gamma_j^n$ and with \mathbf{n}_j the unit outward normal of cell Ω_j^n . Furthermore we distinguish between inflow and outflow boundaries of $\partial\Omega_j^n = \Gamma_j^{n-} \cup \Gamma_j^{n+}$. With Γ_j^{n-} we denote the inflow boundary part. Here is $\mathbf{n}_j \cdot \mathbf{b} < 0$. The outflow boundary is denoted by Γ_j^{n+} , i.e., $\mathbf{n}_j \cdot \mathbf{b} \geq 0$. The traces $u_h^{n\pm}$ at $\partial\Omega_j^n$ are defined by:

$$u_h^{n\pm} = \lim_{\varepsilon \uparrow 0} u_h^n(x \pm \varepsilon \mathbf{n}_x, t \pm \varepsilon n_t),$$

with $\mathbf{n}_j = [n_x \quad n_t]^T$. Notice that, because of the causality in time, $u_h^{0-}|_{\partial\Omega} = u_0$ and that $u_h^{n-}(x, n\Delta t) = u_h^{n-1}(x, n\Delta t)$. So for each time slab Ω^n we have to solve a system of algebraic equations. To explicitly describe the iterative methods studied in this paper, in the next section, we provide \mathcal{S}_h^n with a polynomial space and we give the discrete stencils associated with (5.5).

5.1.2 Discrete system

Here, we describe the linear system that must be solved for each time slab. For sake of clarity, in this presentation we restrict ourselves to a second order discretization although the analysis can be extended to higher order and multiple dimensions.

On the unit square $(\xi, \eta) \in (0, 1)^2$ we take the following polynomial space:

$$\phi_0(\xi, \eta) = 1, \quad \phi_1(\xi, \eta) = 2\xi - 1, \quad \phi_2(\xi, \eta) = 2(\eta - 1),$$

yielding the approximation:

$$u_h^n = \sum_{j \in \mathbb{Z}} \sum_{k=0}^2 c_{j,k}^n \phi_{j,k}^n(x, t) \equiv \sum_{j \in \mathbb{Z}} \sum_{k=0}^2 c_{j,k}^n \phi_k \left(\frac{x - jh}{h}, \frac{t - n\Delta t}{\Delta t} \right). \quad (5.6)$$

This polynomial basis is of interest because of two reasons: the basis functions are chosen such that the test and trial function can be split into an element mean \bar{u}_h at $t = t_{n+1}$ and a fluctuating part \tilde{u}_h [44]:

$$u_h(x, t) = \bar{u}_h + \tilde{u}_h(x, t), \quad \forall x, t \in \Omega_j^n$$

with $\bar{u}_h = c_{j,0}$ and

$$\int_{x \in \Omega_j^n} \tilde{u}_h(x, t_{n+1}) dx = 0.$$

As a consequence the relation between DG and finite volume discretizations is exposed: the equations for the element mean in the space-time DG discretization are the same as those of a finite volume discretization. The second reason is that it suits the definition of the artificial dissipation operator used in [44] as an alternative for slope limiters to guarantee monotone solutions around discontinuities and sharp gradients.

To compute the penalty term in (5.5), we consider its definition in variational form [10]: *find* $r_e(v_h^n) \in [\mathcal{S}_h^n]^2$ *such that*:

$$\sum_{j \in \mathbb{Z}} \int_{\Omega_j} r_e(v_h^n) \cdot \tau_h^n d\Omega = \int_{\Gamma_e^n} \llbracket v_h^n \rrbracket \cdot \{\!\!\{ \tau_h^n \}\!\!\} ds, \quad \forall \tau_h^n \in [\mathcal{S}_h^n]^2,$$

and $e \in \mathbb{Z}$. Since $[\mathcal{S}_h^n]^2 = [\text{Span}\{\phi_{j,k}^n\}]^2$ and because $r_e = [(r_e)_x \ (r_e)_t]^T$ is a polynomial expansion we take:

$$(r_e)_* = \sum_{j \in \mathbb{Z}} \sum_{k=0}^2 (a_{j,k}^n)_* \phi_{j,k}^n, \quad * = x, t, \quad (5.7)$$

with $2 \times 3\mathbb{Z}$ unknowns $(a_{j,k}^n)_*$. Taking the same number of test functions:

$$\tau_h^n \in \left\{ [\phi_{j,k}^n \ 0]^T, [0 \ \phi_{j,k}^n]^T \right\}, \quad j \in \mathbb{Z}, \ 0 \leq k \leq 2,$$

we find $r_e(\phi_{j,\tilde{k}}^n)$ and hence $r_e(u_h^n)$ by solving the small linear system for the unknowns $(a_{j,k}^n)_x$:

$$\sum_{j=e-1}^e \sum_{k=0}^2 (a_{j,k}^n)_x \int_{\Omega_j} \phi_{j,k}^n \phi_{j,l}^n d\Omega = \int_{\Gamma_e^n} \llbracket \phi_{j,\tilde{k}}^n \rrbracket \cdot \{\!\!\{ [\phi_{j,k}^n \ 0]^T \}\!\!\} ds, \quad (5.8)$$

with $\tilde{j} \in \{e-1, e\}$ and $l, \tilde{k} \in \{0, 1, 2\}$, while $(a_{j,k}^n)_x = 0$ for all $j \in \mathbb{Z}/\{e-1, e\}$ and $(a_{j,k}^n)_t = 0$ for all $j \in \mathbb{Z}$. So with the approximation (5.6) and with the definition of the lifting functions (5.7) and (5.8), the discrete system (5.4) is $3\mathbb{Z}$ block-Toeplitz. The corresponding operator is given by the associated stencils:

$$\mathcal{L}_{d_h}^n \cong \frac{d\Delta t}{h} \times \begin{bmatrix} -2\eta_e & 1-2\eta_e & 2\eta_e & 4\eta_e & 0 & -4\eta_e & -2\eta_e & -1+2\eta_e & 2\eta_e \\ -1+2\eta_e & -2+2\eta_e & 1-2\eta_e & 0 & 4\eta_e & 0 & 1-2\eta_e & -2+2\eta_e & -1+2\eta_e \\ 2\eta_e & -1+2\eta_e & -\frac{13}{6}\eta_e & -4\eta_e & 0 & \frac{13}{3}\eta_e & 2\eta_e & 1-2\eta_e & -\frac{13}{6}\eta_e \end{bmatrix}$$

for the diffusion part. The space-time advection stencil is given by:

$$\mathcal{L}_{a_h}^n \cong \left[\begin{array}{ccc|ccc} -a\Delta t & -a\Delta t & a\Delta t & a\Delta t + h & a\Delta t & -a\Delta t & 0 & 0 & 0 \\ a\Delta t & a\Delta t & -a\Delta t & -a\Delta t & a\Delta t + \frac{1}{3}h & a\Delta t & 0 & 0 & 0 \\ a\Delta t & a\Delta t & -\frac{4}{3}a\Delta t & -a\Delta t - 2h & -a\Delta t & \frac{4}{3}a\Delta t + 2h & 0 & 0 & 0 \end{array} \right].$$

The stencil containing data of the previous time slab is given by:

$$\mathcal{L}_{a_h}^{n-1} \cong \left[\begin{array}{ccc|ccc} 0 & 0 & 0 & -h & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -\frac{1}{3}h & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2h & 0 & 0 & 0 & 0 & 0 \end{array} \right].$$

With $\mathcal{L}_h^n = \mathcal{L}_{d_h}^n + \mathcal{L}_{a_h}^n$ and $f_h^n = \mathcal{L}_{a_h}^{n-1} u_h^{n-1}$ we have to solve for each time slab Ω^n the $3\mathbb{Z} \times 3\mathbb{Z}$ linear system:

$$\mathcal{L}_h^n u_h^n = f_h^n. \quad (5.9)$$

This system will be solved by multigrid iteration combined with Runge-Kutta smoothers and the resulting convergence behavior will be analyzed.

5.1.3 Runge-Kutta smoothers

In order to reduce the computational costs when handling complex higher dimensional problems, we are interested in fully explicit iterative solvers. For that purpose, we write the system of equations (5.9) as a system of ordinary differential equations that we want to iterate towards steady state. Hence we consider the problem:

$$\frac{dc_h^n}{d\tau} = f_h^n - A_h^n c_h^n,$$

for expansion coefficients c_h^n of u_h^n with A_h^n the $3\mathbb{Z} \times 3\mathbb{Z}$ block-Toeplitz matrix associated with the operator \mathcal{L}_h^n in (5.9). The first Runge-Kutta method used for this purpose is:

Algorithm 5.1 (EXI). Explicit Runge-Kutta method for inviscid flow with Melson [33] correction and pseudo-time step $\Delta\tau$.

1. Set $(c_h^n)_0^k = (c_h^n)^k$.
2. For all stages $s = 1$ to 5 compute $(c_h^n)_s^k$ as

$$(c_h^n)_s^k = \frac{1}{1 + \alpha_s \Delta\tau} \left((c_h^n)_0^k + \alpha_s \Delta\tau \left((c_h^n)_{s-1}^k + f_h^n - A_h^n (c_h^n)_{s-1}^k \right) \right).$$

3. Return $(c_h^n)^{k+1} = (c_h^n)_5^k$.

Here the Runge-Kutta coefficients are $\alpha_1 = 0.0791451$, $\alpha_2 = 0.163551$, $\alpha_3 = 0.283663$, $\alpha_4 = 0.5$ and $\alpha_5 = 1.0$. In Chapter 4, the performance of the EXI method was analyzed for the space-time DG discretization of the advection-diffusion equation. When diffusion dominates, the stability condition proved quite restrictive. To alleviate this restriction, a member of a family of Runge-Kutta methods proposed by Kleb e.a. [28] was used. It has a stability domain which stretches much further along the negative real axis than classical Runge-Kutta schemes, making it ideal for diffusion dominated flow problems. The method is given by:

Algorithm 5.2 (EXV). Explicit Runge-Kutta method for viscous flows with pseudo-time step $\Delta\tau$.

1. Set $(c_h^n)_0^k = (c_h^n)^k$.

2. For all stages $s = 1$ to 4 compute $(c_h^n)_s^k$ as

$$(c_h^n)_s^k = (c_h^n)_0^k + \alpha_s \Delta\tau \left(f_h^n - A_h^n (c_h^n)_{s-1}^k \right).$$

3. Return $(c_h^n)^{k+1} = (c_h^n)_4^k$.

For this iteration scheme the Runge-Kutta coefficients are $\alpha_1 = 0.0178571$, $\alpha_2 = 0.0568106$, $\alpha_3 = 0.1745130$, $\alpha_4 = 1.0$. In Chapter 4, the EXI and EXV method were combined based on the cell Reynolds number: the EXV method is used for elements with low cell Reynolds numbers (i.e. boundary layers) and the EXI method for high cell Reynolds numbers (i.e. the far-field). This approach proved very effective for the 3D compressible Navier-Stokes equations. Therefore, we study the smoothing properties of these two explicit Runge-Kutta methods. The error amplification operators associated with these Runge-Kutta methods are needed to assess the smoothing property. These operators for the error $(e_h^n)^{k+1} \equiv (c_h^n)^{k+1} - (c_h^n)^k$ are given by:

$$\begin{aligned} M_h^{\text{EXI}} &= \frac{I_h}{1 + \alpha_5 \Delta\tau} + \frac{\alpha_5 \Delta\tau (I_h - A_h^n)}{(1 + \alpha_4 \Delta\tau)(1 + \alpha_5 \Delta\tau)} + \dots \\ &+ \frac{\alpha_2 \alpha_3 \dots \alpha_5 (\Delta\tau (I_h - A_h^n))^4}{(1 + \alpha_1 \Delta\tau)(\dots)(1 + \alpha_5 \Delta\tau)} + \frac{\alpha_1 \alpha_2 \dots \alpha_5 (\Delta\tau (I_h - A_h^n))^5}{(1 + \alpha_1 \Delta\tau)(\dots)(1 + \alpha_5 \Delta\tau)}, \end{aligned} \quad (5.10)$$

and

$$M_h^{\text{EXV}} = I_h - \alpha_4 \Delta \tau A_h^n + \alpha_3 \alpha_4 (\Delta \tau A_h^n)^2 - \dots + \alpha_1 \alpha_2 \alpha_3 \alpha_4 (\Delta \tau A_h^n)^4. \quad (5.11)$$

5.1.4 Multigrid

At the core of any multigrid algorithm is the two-level scheme. Multilevel algorithms are obtained by recursively applying the two-level scheme in, for example, a V-cycle. Therefore, we study the error amplification operator of the two-level algorithm M_h^{TLA} , which is given by [25, 49]:

$$M_h^{\text{TLA}} = M_h^{\text{CGC}} M_h^{\text{REL}},$$

with M_h^{REL} the error amplification operator associated with either the EXV or the EXI scheme, given in (5.11) and (5.10). The coarse grid correction operator is defined as:

$$M_h^{\text{CGC}} = I_h - P_{hH} (\mathcal{L}_H^n)^{-1} \bar{R}_{Hh} \mathcal{L}_h^n, \quad (5.12)$$

with \mathcal{L}_H^n the system obtained by the space-time DG discretization for the time slab Ω^n on the coarse grid with $H = 2h$. The prolongation and restriction operators P_{hH} and \bar{R}_{Hh} are based on the embedding of the spaces $S_H^n \subset S_h^n$ and will be given in the next section.

Remark. Contrary to the internal penalty method, the discretization of the second order term (based on the method by Brezzi e.a. [1, 10]) only satisfies the Galerkin property ($\mathcal{L}_H = \bar{R}_{Hh} \mathcal{L}_h P_{hH}$) if the stabilization parameter η_e on the coarse mesh is a factor H/h larger than on the fine mesh (5.3). In general this property does not hold, e.g., on non-uniform meshes. Therefore we take the same stabilization parameter on the fine and coarse mesh. For stability of the discretization we take $\eta_e = 2$; equal to the number of spatial faces per cell [41].

The convergence behavior of the two-level algorithm for the space-time DG discretization is given by the spectral radius of the error amplification operator $\rho(M_h^{\text{TLA}})$ which represents the expected convergence factor per iteration. In the next section, we will apply Fourier analysis to compute the eigenvalue spectra of the two-level algorithm.

5.2 Fourier analysis

5.2.1 Grid functions and the space-time block Toeplitz operator

To study the convergence of the various iterative methods we introduce two-level Fourier analysis tools for the unknowns in the cells Ω_j^n . The key part in this analysis is to associate the coefficients $\{c_{j,0}^n, c_{j,1}^n, c_{j,2}^n\}_{j \in \mathbb{Z}}$ of the approximation (5.6) in the system (5.9) with the nodal points jh . In this way we avoid the staggering problem of transferring cell data from coarse to fine cells [25, 49], while we keep the data in cell wise ordering. Hence, we introduce an elementary mode $e_h(\omega) = e^{ijh\omega}$, with $\omega \in \mathbb{T}_h = [-\pi/h, \pi/h)$ on the space-time grid:

$$\mathbb{Z}_h^n = \{(jh, n\Delta t) \mid j \in \mathbb{Z}, n \in \mathbb{N}, h > 0, \Delta t > 0\}. \quad (5.13)$$

If we decompose \mathcal{L}_h^n into a strict block-lower, a block-diagonal and a strict block-upper matrix, where:

$$\mathcal{L}_h^n \cong \begin{bmatrix} L_h^n & D_h^n & U_h^n \end{bmatrix},$$

we compute the Fourier transform by:

$$\widehat{\mathcal{L}}_h^n(\omega) = L_h^n e^{-i\omega h} + D_h^n + U_h^n e^{+i\omega h}. \quad (5.14)$$

Then, following [25, 49], we find in the eigenvalue-eigenvector decomposition:

$$\widehat{\mathcal{L}}_h^n(\omega) \mathbf{v} = \Lambda_h^n(\omega) \mathbf{v}, \quad \omega \in \mathbb{T}_h,$$

that $\Lambda_h^n(\omega)$ is a 3×3 diagonal matrix with the eigenvalues $\lambda_i(\omega)$ of \mathcal{L}_h^n as function of $\omega \in \mathbb{T}_h$. The columns of $\mathbf{v} = [v_0, v_1, v_2]$ are the coefficients of the eigenvector $v_i e^{ijh\omega}$ of \mathcal{L}_h^n . We see that this eigenvector is a three-valued grid function on the grid (5.13) in the coefficient ordering $\{c_{j,0}^n, c_{j,1}^n, c_{j,2}^n\}_{j \in \mathbb{Z}}$.

In the next section we introduce the grid transfer operators that are needed to construct the two-level algorithm.

5.2.2 Prolongation and restriction

Important ingredients in the two-level analysis are the flat prolongation and the flat restriction operator. Any constant coefficient grid transfer

operator is a combination of a Toeplitz and a flat grid transfer operator. In this section we introduce the grid transfer operators and the Fourier transforms for grid functions $\{c_{j,0}^n, c_{j,1}^n, c_{j,2}^n\}_{j \in \mathbb{Z}}$ that are needed in the two-level analysis.

To avoid the data staggering problem related to the grid transfer operators acting on cell-wise data [25], it is necessary for this analysis to associate the cell data with the nodal points x_j . In this way, we can obtain vector valued grid functions $\mathbf{u}_h^n(jh) = \{c_{j,0}^n, c_{j,1}^n, c_{j,2}^n\}_{j \in \mathbb{Z}}$ in the Hilbert space $[l^2(\mathbb{Z}_h^n)]^3$ for which the grid transfer operators are easily defined. For such a grid function the flat prolongation $P_{hH}^0 : [l^2(\mathbb{Z}_H^n)]^3 \rightarrow [l^2(\mathbb{Z}_h^n)]^3$ is defined by:

$$P_{hH}^0 \mathbf{u}_H^n(jH) = \begin{cases} \mathbf{u}_H^n(\frac{j}{2}H) & \text{if } j \text{ even,} \\ \mathbf{0} & \text{if } j \text{ odd.} \end{cases}$$

The flat restriction operator $R_{hH}^0 : [l^2(\mathbb{Z}_h^n)]^3 \rightarrow [l^2(\mathbb{Z}_H^n)]^3$ is defined by:

$$(R_{Hh}^0 \mathbf{u}_h^n)(jH) = \mathbf{u}_h^n(2jh).$$

Then the prolongation $P_{hH} : \mathcal{S}_H^n \rightarrow \mathcal{S}_h^n$ so that $P_{hH} u_H^n(x) = u_h^n(x)$ for all $x \in \mathbb{R} \setminus \mathbb{Z}_h^n$ is uniquely defined by $P_{hH} = P_h P_{hH}^0$, where:

$$P_h \cong \left[\begin{array}{ccc|ccc} 1 & \frac{1}{2} & 0 & 1 & -\frac{1}{2} & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \end{array} \right].$$

Another unique operator that is needed in the two-level analysis is the restriction on the residual. It is the adjoint of the prolongation operator and is given by:

$$\bar{R}_{Hh} = R_{Hh}^0 P_h^T.$$

Then the Fourier transforms are computed from [25, 49]:

$$\widehat{P_{hH} \mathbf{u}_H^n}(\omega) = \left(P_h \widehat{P_{hH}^0 \mathbf{u}_H^n} \right) (\omega) = \frac{1}{2} \left[\begin{array}{c} \widehat{P}_h(\omega) \\ \widehat{P}_h(\omega + \frac{\pi}{h}) \end{array} \right] \widehat{\mathbf{u}_H^n}(\omega), \quad (5.15)$$

and with $\bar{R}_h = P_h^T$:

$$\begin{aligned} \widehat{\bar{R}_{Hh} \mathbf{u}_h^n}(\omega) &= R_{Hh}^0 \widehat{\bar{R}_h \mathbf{u}_h^n}(\omega) \\ &= \left[\widehat{\bar{R}}_h(\omega) \quad \widehat{\bar{R}}_h(\omega + \frac{\pi}{h}) \right] \left[\begin{array}{c} \widehat{\mathbf{u}}_h^n(\omega) \\ \widehat{\mathbf{u}}_h^n(\omega + \frac{\pi}{h}) \end{array} \right], \end{aligned} \quad (5.16)$$

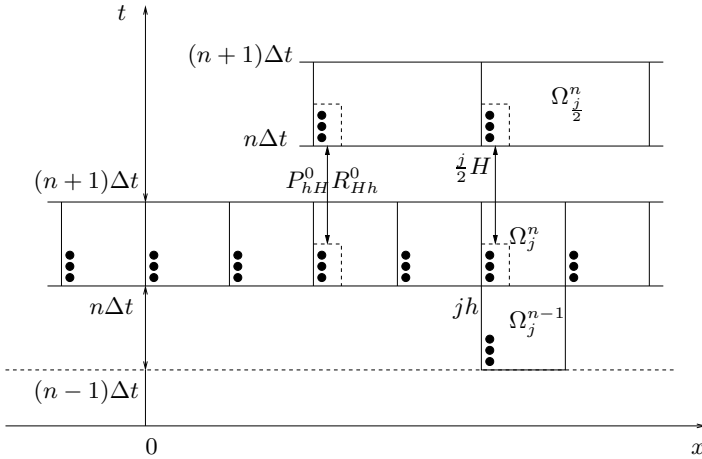


Figure 5.1: A sketch of the grid function $\mathbf{u}_h^n(jh)$ on the space-time grid (5.13) and a part of the coarse mesh. Pictorially the action (5.15) of the flat prolongation and the flat restriction (5.16) is shown.

with $\omega \in \mathbb{T}_H = [-\pi/H, \pi/H)$ and

$$\widehat{\mathbf{u}}_h^n(\omega) = \frac{h}{\sqrt{2\pi}} \sum_{j \in \mathbb{Z}} e^{-ijh\omega} \mathbf{u}_h^n(jh).$$

A sketch of the basic two-level set up is shown in Figure 5.1. Here we see the grid function $\mathbf{u}_h^n(jh)$ on the space-time grid; a part of the coarse grid and pictorially the action of the grid transfer operators. With these tools we construct the Runge-Kutta two-level analysis in the next section.

5.2.3 Two-level algorithm

The eigenvalue spectra of the two-level algorithm M_h^{TLA} is shown [49] to be $\{\lambda_i(\omega)\}$ with $i = 1, \dots, 6$ and $\lambda_i(\omega)$ the eigenvalues of the Fourier transform $\widehat{M}_h^{\text{TLA}}$ for $\omega \in \mathbb{T}_H$. The Fourier transform of the two-level operator reads:

$$\widehat{M}_h^{\text{TLA}}(\omega) = \widehat{M}_h^{\text{CGC}} \widehat{M}_h^{\text{REL}}(\omega),$$

with $\widehat{M}_h^{\text{REL}}$ the Fourier transform of the EXI (or EXV) Runge-Kutta method and $\widehat{M}_h^{\text{CGC}}$ of the coarse grid correction. For the operator $\widehat{\mathcal{L}}_h^n$ (see (5.14)),

the Fourier transforms of the EXI and EXV error amplification operators are:

$$\begin{aligned} \widehat{M}_h^{\text{EXI}}(\omega) &= \frac{I_h}{1 + \alpha_5 \Delta \tau} + \frac{\alpha_5 \Delta \tau \left(I_h - \widehat{\mathcal{L}}_h^n(\omega) \right)}{(1 + \alpha_4 \Delta \tau)(1 + \alpha_5 \Delta \tau)} + \dots \\ &+ \frac{\alpha_2 \alpha_3 \dots \alpha_5 \left(\Delta \tau \left(I_h - \widehat{\mathcal{L}}_h^n(\omega) \right) \right)^4}{(1 + \alpha_1 \Delta \tau)(\dots)(1 + \alpha_5 \Delta \tau)} + \frac{\alpha_1 \alpha_2 \dots \alpha_5 \left(\Delta \tau \left(I_h - \widehat{\mathcal{L}}_h^n(\omega) \right) \right)^5}{(1 + \alpha_1 \Delta \tau)(\dots)(1 + \alpha_5 \Delta \tau)}, \end{aligned}$$

and

$$\begin{aligned} \widehat{M}_h^{\text{EXV}}(\omega) &= I_h - \alpha_4 \Delta \tau \widehat{\mathcal{L}}_h^n(\omega) + \alpha_3 \alpha_4 \left(\Delta \tau \widehat{\mathcal{L}}_h^n(\omega) \right)^2 - \dots \\ &+ \alpha_1 \alpha_2 \alpha_3 \alpha_4 \left(\Delta \tau \widehat{\mathcal{L}}_h^n(\omega) \right)^4, \end{aligned}$$

with $\omega \in \mathbb{T}_h$. Using the Fourier transform of the block Toeplitz operator and of the prolongation and restriction operators the Fourier transform of the two-level error amplification operator is given by (see (5.12)):

$$\begin{aligned} \widehat{M}_h^{\text{TLA}}(\omega) &= \begin{bmatrix} I_h & 0 \\ 0 & I_h \end{bmatrix} - \begin{bmatrix} \widehat{P}_h(\omega) \\ \widehat{P}_h(\omega + \frac{\pi}{h}) \end{bmatrix} \left[\widehat{\mathcal{L}}_H(\omega)^{-1} \right] \times \\ &\begin{bmatrix} \widehat{R}_h(\omega) & \widehat{R}_h(\omega + \frac{\pi}{h}) \end{bmatrix} \begin{bmatrix} \widehat{\mathcal{L}}_h(\omega) & 0 \\ 0 & \widehat{\mathcal{L}}_h(\omega + \frac{\pi}{h}) \end{bmatrix} \begin{bmatrix} \widehat{M}_h^{\text{REL}}(\omega) & 0 \\ 0 & \widehat{M}_h^{\text{REL}}(\omega + \frac{\pi}{h}) \end{bmatrix}, \end{aligned}$$

with I_h the 3×3 identity matrix and $\omega \in \mathbb{T}_H$.

5.3 Results

5.3.1 Convergence of the two-level algorithm

In this section, the eigenvalue spectra and radii of the two-level algorithm are given for various situations, described by the Courant and Péclet numbers:

$$C_{\Delta t} = \frac{a \Delta t}{h} \quad \text{and} \quad \text{Pe}_h = \frac{ah}{d}.$$

The Courant number expresses the time-accuracy of the discretization and the Péclet number the importance of diffusion relative to advection. Since

Table 5.1: The TLA algorithm with EXI smoothing is preferable in the advection dominated case (denoted by *).

flow regime		stability	convergence	
$C_{\Delta t}$	Pe_h	$\Delta\tau/\Delta t$	$\rho(M_h^{\text{EXI}})$	$\rho(M_h^{\text{TLA}})$
100	100	$1.8 \cdot 10^{-4}$	0.991	0.623*
100	0.01	$1.0 \cdot 10^{-7}$	0.999	0.959
1	100	1.6	0.796	0.479*
1	0.01	$1.0 \cdot 10^{-3}$	0.999	0.957

Table 5.2: The TLA algorithm with EXV smoothing is preferable in the diffusion dominated case (denoted by *).

flow regime		stability	convergence	
$C_{\Delta t}$	Pe_h	$\Delta\tau/\Delta t$	$\rho(M_h^{\text{EXV}})$	$\rho(M_h^{\text{TLA}})$
100	100	$2.0 \cdot 10^{-5}$	0.999	0.914
100	0.01	$8.0 \cdot 10^{-7}$	0.999	0.744*
1	100	1.0	0.924	0.660
1	0.01	$8.0 \cdot 10^{-3}$	0.993	0.744*

the space-time DG discretization is *implicit* in physical time, the method is unconditionally stable [41] for any physical time step. This allows us to take the Courant number $C_{\Delta t} = 100$ for steady-state cases and $C_{\Delta t} = 1$ for time-dependent cases. We will consider Péclet numbers $Pe_h = 0.01$ and $Pe_h = 100$, which represent the diffusion and advection dominated cases, respectively. This defines the following four flow regimes:

1. steady, advection dominated: $C_{\Delta t} = 100$, $Pe_h = 100$
2. steady, diffusion dominated: $C_{\Delta t} = 100$, $Pe_h = 0.01$
3. unsteady, advection dominated: $C_{\Delta t} = 1$, $Pe_h = 100$
4. unsteady, diffusion dominated: $C_{\Delta t} = 1$, $Pe_h = 0.01$

The Runge-Kutta methods are *explicit* in pseudo time and their stability is governed by the ratio between the pseudo and physical time step $\Delta\tau/\Delta t$.

In Tables 5.1 and 5.2, we give the spectral radii of the smoothers and the two-level algorithm for these cases. Both the EXI and EXV smoother are stable but hardly converge (except in the unsteady advection dominated case) which shows the necessity of multigrid iteration. With the two-level algorithm, the situation is considerably improved. Clearly, the EXI method is preferable in the advection dominated case. We find convergence factors of 0.62 and 0.48 for the steady and unsteady case respectively (Table 5.1). The EXV method is preferable in the diffusion dominated case, where we find convergence factors of 0.74 both in the steady and unsteady situation (Table 5.2).

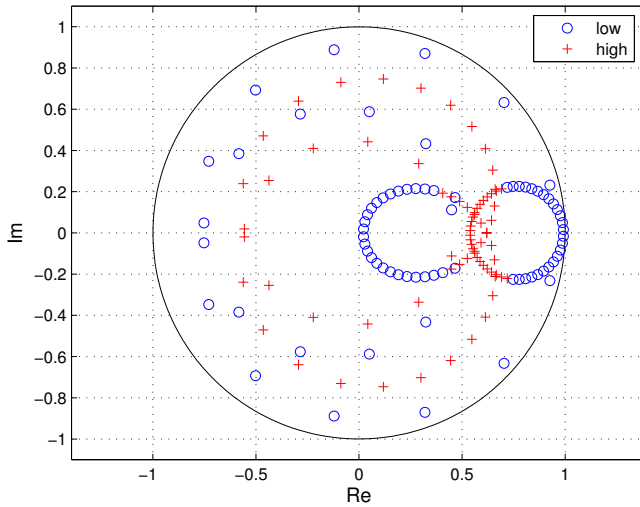
In Figures 5.2, 5.3, 5.4 and 5.5, we show the eigenvalue spectra of the preferable smoother and the two-level algorithm for each case. For the smoothers we have plotted the eigenvalues corresponding to a discrete series of low frequencies $\omega_i = -\pi/2h, -0.96\pi/2h, \dots, \pi/2h$ and associated high frequencies $\omega_i + \pi/h$. The eigenvalues corresponding to low frequencies are denoted by \circ ; those corresponding to high frequencies by $+$. The eigenvalue spectra of two-level algorithms are plotted for $\omega_i = -\pi/H, -0.96\pi/H, \dots, \pi/H$. Here we do not distinguish between low and high frequencies. The two-level algorithm must damp all frequencies.

Moreover, from these figures we see that the Runge-Kutta methods have the smoothing property, i.e., the high frequencies are damped. The observed smoothing factor of approximately 0.8 (which is often used as an estimate for the multigrid convergence [11]) is rather inaccurate in comparison to the true smoothing factor obtained with two-level analysis.

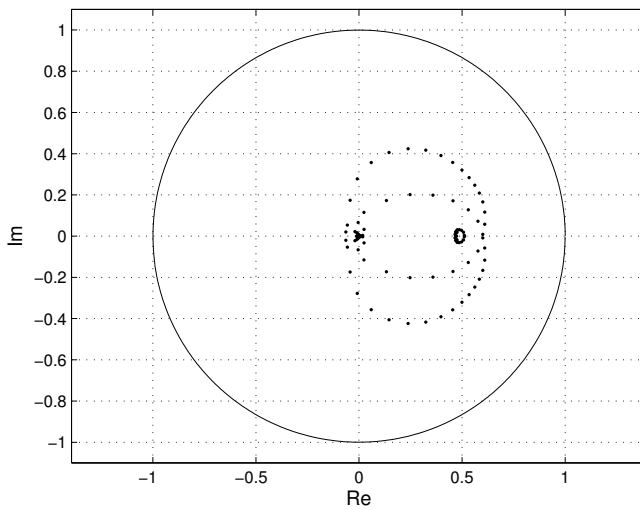
5.3.2 Numerical illustration

To illustrate the results of the multigrid analysis, we consider the space-time discretization of the scalar advection-diffusion equation for the following simple initial boundary value problem:

$$\begin{cases} u_t + au_x = du_{xx}, & x \in (0, 1), \quad t \in \mathbb{R}^+, \\ u(0, t) = 1, \quad u(1, t) = 0, & t \in \mathbb{R}^+, \\ u(x, 0) = 1 - x, & x \in (0, 1). \end{cases}$$

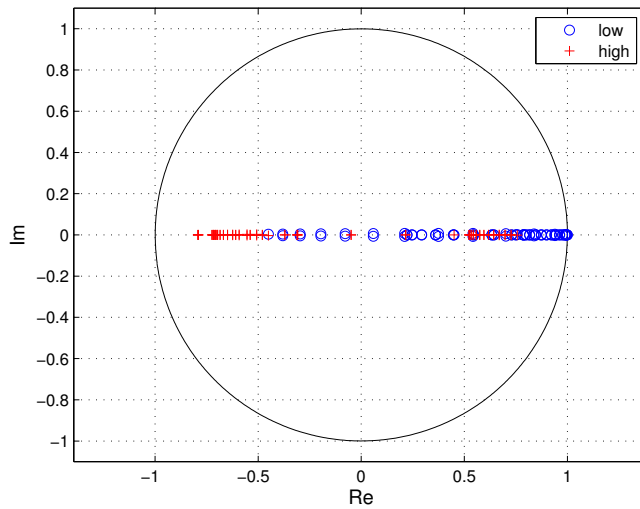


(a) EXI

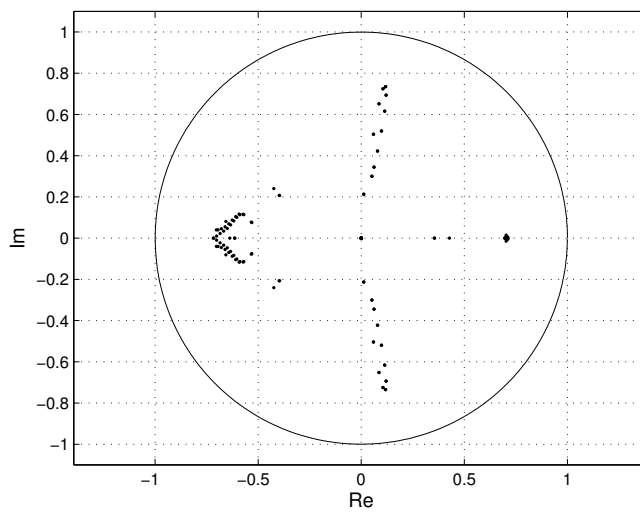


(b) TLA with EXI

Figure 5.2: Eigenvalue spectra of the EXI smoother and two-level algorithm in the steady advection dominated case ($C_{\Delta t} = 100$ and $Pe_h = 100$).

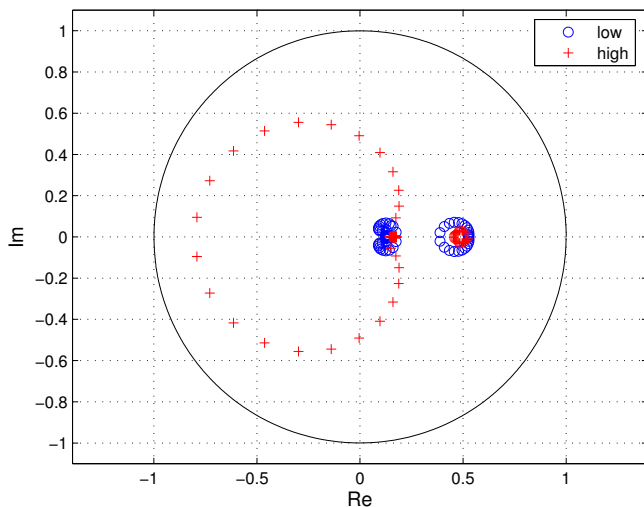


(a) EXV

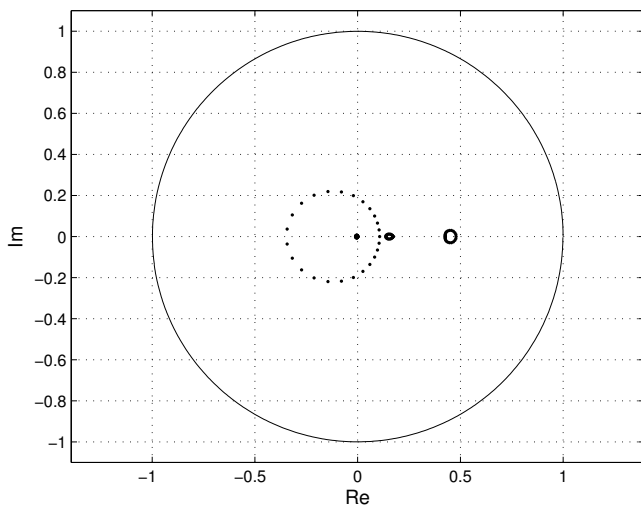


(b) TLA with EXV

Figure 5.3: Eigenvalue spectra of the EXV smoother and two-level algorithm in the steady diffusion dominated case ($C_{\Delta t} = 100$ and $Pe_h = 0.01$).

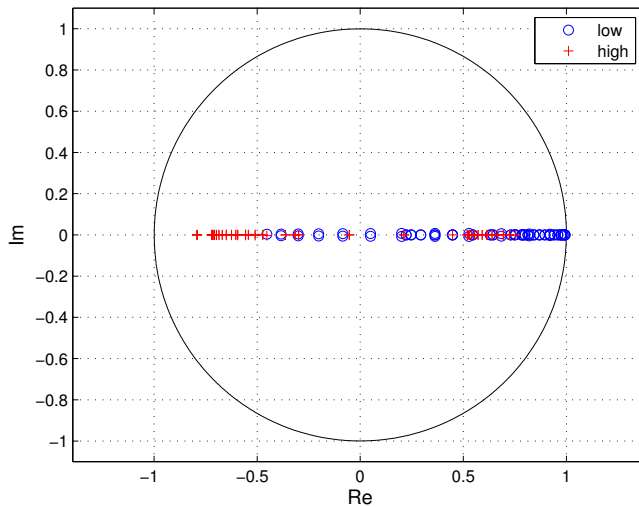


(a) EXI

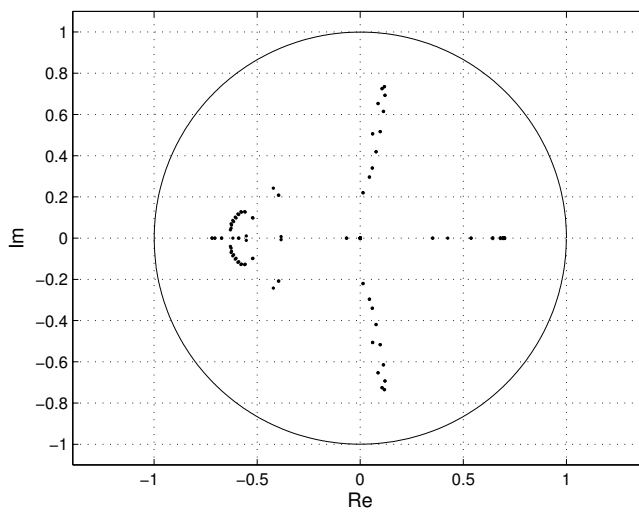


(b) TLA with EXI

Figure 5.4: Eigenvalue spectra of the EXI smoother and two-level algorithm in the unsteady advection dominated case ($C_{\Delta t} = 1$ and $Pe_h = 100$).



(a) EXV



(b) TLA with EXV

Figure 5.5: Eigenvalue spectra of the EXV smoother and two-level algorithm in the unsteady diffusion dominated case ($C_{\Delta t} = 1$ and $Pe_h = 0.01$).

The exact (steady state) solution is given by:

$$u(x) = \frac{e^{a/d} - e^{ax/d}}{e^{a/d} - 1},$$

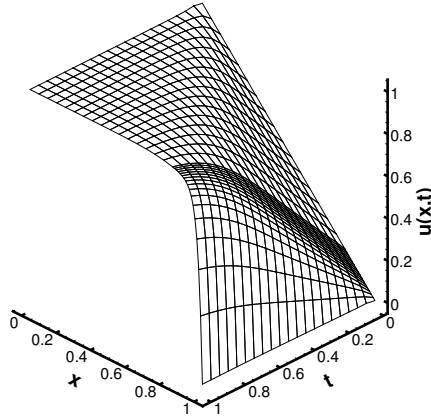
and features an exponential boundary layer near $x = 1$. Such a case is best solved on a so-called Shishkin mesh [40]. With N elements, this mesh is piecewise equidistant with nodes x_j given by:

$$x_j = \begin{cases} 2(1-c)j/N & \text{for } j = 0, 1, \dots, N/2, \\ 1-c + 2c/N(j - N/2) & \text{for } j = N/2, N/2 + 1, \dots, N, \end{cases}$$

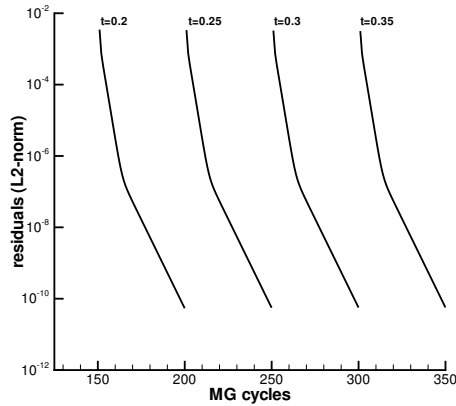
where $c = (2/a)d \ln(N)$. For our example, we take $a = 1$, $d = 0.025$ and $N = 32$. Advection dominates in the first part, so we use the EXI scheme there and the EXV scheme in the second part. We use three level multigrid in a V-cycle with two pre- and post-relaxations. The coarse grid problem is solved approximately with four relaxations, which is more realistic in view of applications to complex problems where the exact coarse grid solution cannot be attained.

The problem can be solved in two ways: time accurate with $\Delta t = 0.05$ which corresponds to $C_{\Delta t} \approx \mathcal{O}(1)$ or directly steady-state with $\Delta t = 5$ which corresponds to $C_{\Delta t} \approx \mathcal{O}(100)$. In Figure 5.6, the space-time solution and the convergence in pseudo-time for a few physical time steps are shown. With eight orders of convergence in fifty cycles, an effective damping factor of 0.7 is achieved. In Figure 5.7, the steady-state solution is shown. With a single time step the convergence in pseudo-time is ten orders in one hundred and fifty cycles which corresponds to a damping factor of 0.85. Despite the presence of boundary conditions and the inaccurate solution of the coarse grid problem, these convergence rates are in agreement with the rate obtained from the analysis. The latter being 0.74 when diffusion is dominating (Table 5.2), which is the case in the boundary layer.

These results show that the EXI and EXV methods can indeed be combined to form a cheap local smoother for a full multigrid setting as expected from the analysis. However, its application to the 3D compressible Navier-Stokes equations is complicated by the fact that the grid levels are generally not nested. Therefore, in the next chapter, we limit ourselves to a simpler multigrid algorithm which uses constant basis functions on the coarse mesh.

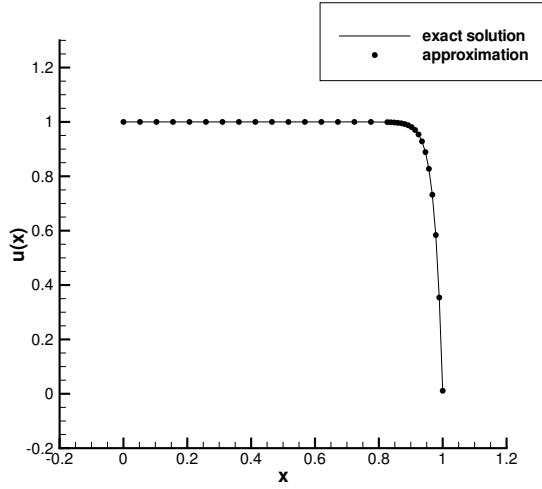


(a) Solution

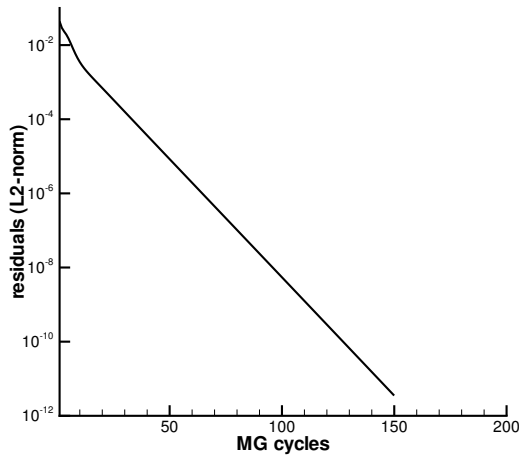


(b) Convergence

Figure 5.6: The space-time solution of the advection-diffusion equation ($a = 1$, $d = 0.025$) on a Shishkin mesh with 32 elements and the convergence in pseudo-time of the MG algorithm for a few physical time steps $\Delta t = 0.05$.



(a) Solution



(b) Convergence

Figure 5.7: The steady-state solution of the advection-diffusion equation ($a = 1, d = 0.025$) on a Shishkin mesh with 32 elements and the convergence in pseudo-time of the MG algorithm for a single physical time step $\Delta t = 5$.

Chapter 6

Multigrid application

In this chapter, we study the multigrid algorithm for the space-time discretization with linear basis functions on the fine grid and constant basis functions on the coarse grids. We first derive the grid-dependent stabilization parameter needed for the discretization with constant basis functions. We then study the multigrid convergence with the two-level Fourier analysis developed in the previous chapter and compare with the convergence observed for the compressible Navier-Stokes equations in various 2D and 3D situations.

6.1 Grid-dependent stabilization parameter

The basis functions used for the space-time DG discretization of the compressible Navier-Stokes equations (see Section 3.2.1) split the trial function U_h into an element mean at time t_{n+1} and a fluctuating part [44]:

$$U_h(t, \bar{x}) = \bar{U}_h(K_j(t_{n+1}^-)) + \tilde{U}_h(t, \bar{x}), \quad (6.1)$$

with

$$\bar{U}_h(K_j(t_{n+1}^-)) = \hat{U}_0, \quad \int_{K_j(t_{n+1}^-)} \tilde{U}_h(t, \bar{x}) dK = 0.$$

This splitting is useful for the multigrid algorithm because, on the coarse grid levels, we only use the equations for the element mean; the space-time discretization is then closely related to a first order finite volume method.

When constant basis functions are used, $U_h(t, \bar{x}) = \bar{U}_h$ and the terms related to the viscous flux F^v drop out of the weak form (3.13), except for the stabilization term. The stabilization parameter η_S then necessarily becomes grid-dependent. In this section, we derive η_S for hexahedral meshes. A similar approach gives the grid-dependent stabilization parameter proposed in [20] for 2D triangular meshes.

To derive the grid-dependent stabilization parameter, we consider the 3D Laplace equation for $U \in \mathbb{R}$ with a source term $F \in \mathbb{R}$:

$$(U_{,k})_{,k} = F, \quad \text{on } \mathcal{E}$$

with $k = 1, 2, 3$. Using constant basis functions, the stabilization term for internal faces reduces to:

$$\sum_{S \in \mathcal{S}_I^n} \int_S \llbracket W \rrbracket_k \{ \eta_S (\mathcal{R}^S)_k \} dS = \sum_{\mathcal{K} \in \mathcal{T}_h^n} \int_{\mathcal{K}} W F d\mathcal{K}.$$

Taking $W = 1$ in \mathcal{K} and zero elsewhere gives:

$$\sum_{S \in \partial \mathcal{K}} \int_S \bar{n}_k^L \{ \eta_S (\mathcal{R}^S)_k \} dS = \int_{\mathcal{K}} F d\mathcal{K}.$$

Suppose that the source term is defined by $F = (U_{,k})_{,k}$ and apply Gauss' theorem:

$$\sum_{S \in \partial \mathcal{K}} \int_S \bar{n}_k^L \{ \eta_S (\mathcal{R}^S)_k \} dS = \sum_{S \in \partial \mathcal{K}} \int_S (U_{,k})^L \bar{n}_k^L dS,$$

with U^L the trace from \mathcal{K}^L at $S = \bar{\mathcal{K}}^L \cap \bar{\mathcal{K}}^R$ and \bar{n}^L the spatial outward normal vector. A sufficient condition for this equation to hold is:

$$\{ \eta_S (\mathcal{R}^S)_k \} = (U_{,k})^L \quad \text{on } S \in \partial \mathcal{K}. \quad (6.2)$$

We now introduce a finite difference approximation of the derivatives on the face S :

$$(U_{,k})^L \approx \frac{\llbracket U \rrbracket_k}{d}, \quad (6.3)$$

with d the distance between the centers of gravity of elements \mathcal{K}^L and \mathcal{K}^R . For constant basis functions, the lifting operator \mathcal{R}^S in elements $\mathcal{K}^{L,R}$ at face S reduces to:

$$(\mathcal{R}^S)_k^{L,R} = \frac{|S| \llbracket U \rrbracket_k}{2|\mathcal{K}^{L,R}|},$$

Therefore, the l.h.s of (6.2) can be written as:

$$\begin{aligned} \{\{\eta_S(\mathcal{R}^S)_k\}\} &= \frac{1}{2}(\eta_S(\mathcal{R}^S)_k^L + \eta_S(\mathcal{R}^S)_k^R), \\ &= \frac{\eta_S|\mathcal{S}|}{4} \left(\frac{1}{|\mathcal{K}^L|} + \frac{1}{|\mathcal{K}^R|} \right) \llbracket U \rrbracket_k, \end{aligned}$$

and, with approximation (6.3) on the r.h.s. of (6.2), we obtain:

$$\eta_S = \frac{4}{d|\mathcal{S}|} \left(\frac{|\mathcal{K}^L||\mathcal{K}^R|}{|\mathcal{K}^L| + |\mathcal{K}^R|} \right).$$

The same reasoning can be followed for boundary faces which gives: $\eta_S = |\mathcal{K}^L|/(d|\mathcal{S}|)$, with d the distance between the element and the face centers of gravity.

Remark. Note that on uniform grids, $|\mathcal{K}^L| = |\mathcal{K}^R| = d|\mathcal{S}|$ for internal faces and $|\mathcal{K}^L| = 2d|\mathcal{S}|$ for boundary faces. Therefore, on uniform grids, $\eta_S = 2$ for all faces.

6.2 Multigrid algorithm

In this section, we present the multigrid algorithm used to solve the non-linear system of algebraic equations (3.25) by pseudo-time integration. The combined EXI and EXV method (Chapter 4) is used as smoother. To gain insight in the multigrid convergence behavior, we analyze the convergence of the two-level algorithm for the model problem introduced in Section 5.1.

6.2.1 Two-level algorithm

Let the subscripts h and H denote quantities associated with the fine and coarse grid, respectively. Let \hat{U} denote an approximation of the steady-state solution \hat{U}^n of (3.25), R the restriction operator for the solution, \bar{R} the restriction operator for the residuals and P the prolongation operator. With this notation, the two-level algorithm can be written as:

Algorithm 6.1 (TLA). Two-level algorithm.

1. A pseudo-time step on the fine grid with the combined EXI and EXV method gives an approximation \hat{U}_h .

2. Restrict this approximation to the coarse grid: $\hat{U}_H = R(\hat{U}_h)$.
3. Compute the forcing:

$$F_H \equiv \mathcal{L}(\hat{U}_H; \hat{U}_H^{n-1}) - \bar{R}(\mathcal{L}(\hat{U}_h; \hat{U}_h^{n-1})).$$

4. Solve the coarse grid problem for \hat{U}_H^n :

$$\mathcal{L}(\hat{U}_H^n; \hat{U}_H^{n-1}) - F_H = 0,$$

5. Compute the coarse grid error $E_H = \hat{U}_H^n - \hat{U}_H$ and correct the fine grid approximation: $\hat{U}_h \leftarrow \hat{U}_h + P(E_H)$.

Solving the coarse grid problem at stage four of Algorithm 6.1 can again be done with the two-level algorithm. This recursively defines the multi-level algorithm. It is common practice to do ν_1 pseudo-time steps at stage one of Algorithm 6.1 and another ν_2 pseudo-time steps after stage five. In that case, ν_1 and ν_2 are called the number of pre- and post-relaxations, respectively.

In this multigrid algorithm, we use the linear basis functions (6.1) on the fine grid and the stabilization parameter η_S is constant and (at least) equal to the number of spatial faces of an element [10, 41]: four in 2D and six in 3D for hexahedra. For the coarse grid problems, we use the discretization with constant basis functions and the grid-dependent stabilization parameter η_S derived in Section 6.1.

The reason we prefer constant basis functions on the coarse grids stems from the construction of the different grid levels. The coarse grid is obtained from the fine grid by abandoning the nesting of spaces (see Figure 6.1). This keeps the coarse grid hexahedral. Without the nesting, the two grids overlap only partially, therefore the inter-grid transfer operators must take care of the 'gaps'. Such gaps also occur when local mesh refinement is applied near curvilinear boundaries. The splitting of the basis functions in element means and fluctuations now comes in: the restriction and prolongation work directly on the element mean of U , which is constant and therefore easily extrapolated to the gaps. The restriction and prolongation operators for the solution are defined as [44]:

$$R(\hat{U})|_{\mathcal{K}_H} = \frac{\sum \hat{U}_0(\mathcal{K}_h)|\mathcal{K}_h|}{\sum |\mathcal{K}_h|}, \quad P(\hat{U})|_{\mathcal{K}_h} = \hat{U}_0(\mathcal{K}_H), \quad (6.4)$$

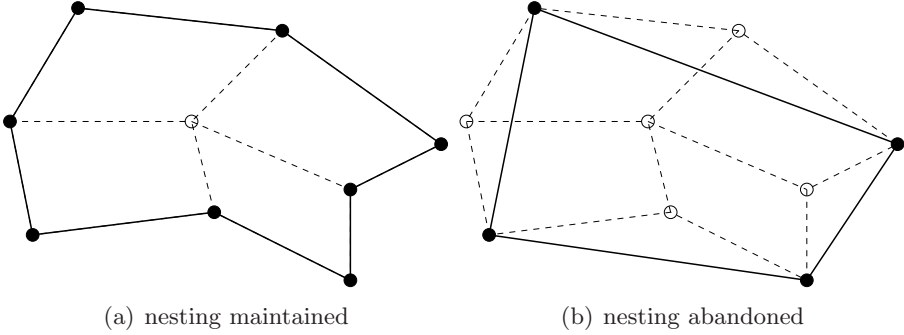


Figure 6.1: Hierarchy between the (curvilinear) fine and coarse grid.

where the summation in the coarse grid element \mathcal{K}_H is over the set $\{\mathcal{K}_h\}$ of corresponding fine grid elements. The restriction operator \bar{R} for the residual is the same as for the solution ($\bar{R} = R$). Note that in the definition of the restriction operator, we divide by $\sum |\mathcal{K}_h|$ instead of $|\mathcal{K}_H|$ to accommodate non-uniform grids.

6.2.2 Fourier analysis

The convergence behavior of Algorithm 6.1 is studied with Fourier analysis. Here, we apply the analysis technique presented in Chapter 5 for the space-time discontinuous Galerkin discretization of the advection-diffusion equation, in the special case when constant basis functions are used for the coarse grid problem.

The space-time discontinuous Galerkin discretization of the advection-diffusion equation with linear basis functions on a uniform mesh with elements $\mathcal{K}_j^n = (x_j, x_{j+1}) \times (t^n, t^{n+1})$ yields a discrete system for the vector of expansion coefficients \hat{u} of u at time level n :

$$\mathcal{L}_h(\hat{u}^n; \hat{u}^{n-1}) = 0,$$

with

$$\mathcal{L}_h(\hat{u}^n; \hat{u}^{n-1}) \equiv (\mathcal{L}_h^a + \mathcal{L}_h^d)\hat{u}^n + \mathcal{L}_h^t\hat{u}^{n-1},$$

and $h = x_{j+1} - x_j$. The stencils associated with \mathcal{L}_h^a , \mathcal{L}_h^d and \mathcal{L}_h^t are given in Section 5.1.2. Remember that the total $3\mathbb{Z} \times 3\mathbb{Z}$ system has a block

Toeplitz structure with 3×3 blocks, with associated stencil of the form:

$$\mathcal{L}_h \cong [L_h \quad D_h \quad U_h],$$

where L_h represents the left block, D_h the diagonal block and U_h the right block. For the viscous part \mathcal{L}_h^d of the stencil we have $U_h = L_h^T$, which reflects the symmetry in the discretization of the diffusive term. The stabilization parameter in \mathcal{L}_h^d is taken equal to the number of spatial faces per element: $\eta_S = 2$ in the one-dimensional case.

The discretization on the coarse grid ($H = 2h$) is obtained by using constant basis functions. In this case, the discretization only involves the element means $\bar{u} = \hat{u}_0$ and reduces to:

$$\mathcal{L}_H(\bar{u}^n; \bar{u}^{n-1}) \equiv (\mathcal{L}_H^a + \mathcal{L}_H^d)\bar{u}^n + \mathcal{L}_H^t\bar{u}^{n-1},$$

with

$$\begin{aligned} \mathcal{L}_H^a &\cong [-a\Delta t \quad a\Delta t + H \quad 0], \\ \mathcal{L}_H^d &\cong \frac{d\Delta t}{H} [-2\eta_S \quad 4\eta_S \quad -2\eta_S], \quad \mathcal{L}_H^t \cong [0 \quad -H \quad 0]. \end{aligned}$$

This $\mathbb{Z} \times \mathbb{Z}$ system has a block Toeplitz structure with 1×1 blocks, with associated stencil:

$$\mathcal{L}_H \cong [L_H \quad D_H \quad U_H].$$

Since the grid is uniform, the stabilization parameter is $\eta_S = 2$, see Section 6.1.

In the two-level analysis of the model problem, we aim at computing the eigenvalue spectra of the two-level algorithm given by:

$$M_h^{\text{TLA}} = M^{\text{CGC}} M_h^{\text{REL}},$$

because the spectral radius $\rho(M_h^{\text{TLA}})$ represents the expected convergence factor per iteration. Here, M_h^{REL} is the fine grid relaxation matrix corresponding to the EXI or EXV Runge-Kutta smoother and the coarse grid correction (CGC) of Algorithm 6.1 is given by:

$$M^{\text{CGC}} = I - P\mathcal{L}_H^{-1}\bar{R}\mathcal{L}_h.$$

On this uniform grid, the $3\mathbb{Z} \times \mathbb{Z}$ system associated with the prolongation P defined in (6.4) has a block Toeplitz structure with 3×1 blocks:

$$P = \left[\begin{array}{c|c|c} 1 & 1 & 0 \\ \hline 0 & 0 & 0 \\ \hline 0 & 0 & 0 \end{array} \right],$$

and the restriction operator for the residual is $\bar{R} = P^T$. The Fourier transform $\widehat{\mathcal{L}}_h$ of a block Toeplitz operator \mathcal{L}_h for a frequency ω is given by:

$$\widehat{\mathcal{L}}_h(\omega) = L_h e^{-i\omega h} + D_h + U_h e^{+i\omega h},$$

with $i = \sqrt{-1}$. The eigenvalue spectra of the two-level algorithm M_h^{TLA} is $\{\lambda_i(\omega)\}$ with $i = 1, \dots, 6$ and $\lambda_i(\omega)$ the eigenvalues of the Fourier transform $\widehat{M}_h^{\text{TLA}}$ for $\omega \in [-\pi/H, \pi/H)$, as shown in [25, 49]. The Fourier transform of the two-level error amplification operator (Section 5.2.3) is given by:

$$\begin{aligned} \widehat{M}_h^{\text{TLA}}(\omega) &= \begin{bmatrix} I_h & 0 \\ 0 & I_h \end{bmatrix} - \begin{bmatrix} \widehat{P}(\omega) \\ \widehat{P}(\omega + \frac{\pi}{h}) \end{bmatrix} \left[\widehat{\mathcal{L}}_H(\omega)^{-1} \right] \times \\ &\quad \begin{bmatrix} \widehat{R}(\omega) & \widehat{R}(\omega + \frac{\pi}{h}) \end{bmatrix} \begin{bmatrix} \widehat{\mathcal{L}}_h(\omega) & 0 \\ 0 & \widehat{\mathcal{L}}_h(\omega + \frac{\pi}{h}) \end{bmatrix} \begin{bmatrix} \widehat{M}_h^{\text{REL}}(\omega) & 0 \\ 0 & \widehat{M}_h^{\text{REL}}(\omega + \frac{\pi}{h}) \end{bmatrix}, \end{aligned}$$

with I_h the 3×3 identity matrix. Here, $\omega \in [-\pi/H, \pi/H)$ corresponds to the low frequencies and $\omega + \pi/h$ to the associated high frequencies. The Fourier transforms of the EXI and EXV error amplification operators are given in Section 5.2.3. For further details on the two-level analysis, the error amplifications operators and the Fourier transforms, we refer to Chapter 5.

We are interested in the multigrid behavior in both steady and unsteady flow regimes which are either advection or diffusion dominated. These situations are described by the Courant number $C_{\Delta t} = a\Delta t/h$ and cell Reynolds number $Re_h = ah/d$ as follows:

1. steady, advection dominated: $C_{\Delta t} = 100$, $Re_h = 100$
2. steady, diffusion dominated: $C_{\Delta t} = 100$, $Re_h = 0.01$
3. unsteady, advection dominated: $C_{\Delta t} = 1$, $Re_h = 100$

4. unsteady, diffusion dominated: $C_{\Delta t} = 1$, $Re_h = 0.01$

Being implicit in physical time, the space-time DG discretization is unconditionally stable [41] for any physical time step. However, the Runge-Kutta methods are explicit in pseudo time and their stability depends on the ratio between the pseudo and physical time step $\Delta\tau/\Delta t$.

In the advection dominated cases, the EXI method is preferable as smoother (see Section 5.3) and we show the eigenvalue spectra of the two-level algorithm for advection dominated cases in Figures 6.2 and 6.4. The results for diffusion dominated cases with the EXV smoother are shown in Figures 6.3 and 6.5. We have plotted the eigenvalues corresponding to frequencies ω_i and $\omega_i + \pi/h$ with $\omega_i = -\pi/H, -0.96\pi/H, \dots, \pi/H$.

The spectral radii of the smoothers and the two-level algorithm are given in Tables 6.1 and 6.2. For steady-state cases, the spectral radius of the smoother is typically 0.99 and the TLA hardly improves the situation: only in the advection dominated case the spectral radius of the TLA is 0.98. Since $0.99^N \approx 0.98^{N/2}$, this leads us to expect that the TLA will converge twice as fast as the single-grid algorithm for this case. For the unsteady diffusion dominated case a similar conclusion is reached. For the other cases, the TLA does not improve the convergence factor, but note that the smoother is already very efficient for the time accurate advection dominated cases: its spectral radius is 0.79.

6.3 Numerical simulations

Definition of work units. To measure the efficiency of the multigrid algorithm, we have to define a basic work unit. The CPU time does not reflect the true work load as it is greatly affected by the implementation, optimization and the machine the code runs on. Therefore, we propose a more transparent definition: one work unit corresponds to one Runge-Kutta step on the fine grid.

To account for the work done on the coarse grids in terms of this work unit, we make use of the following observation. In a well written code, the computational effort of an explicit Runge-Kutta step is proportional to the number of degrees of freedom (DoF). The number of DoF on the fine mesh is $N_e N_q N_c$ with N_e the number of elements, N_q the number of equations and N_c the number of expansion coefficients. On the coarse mesh,

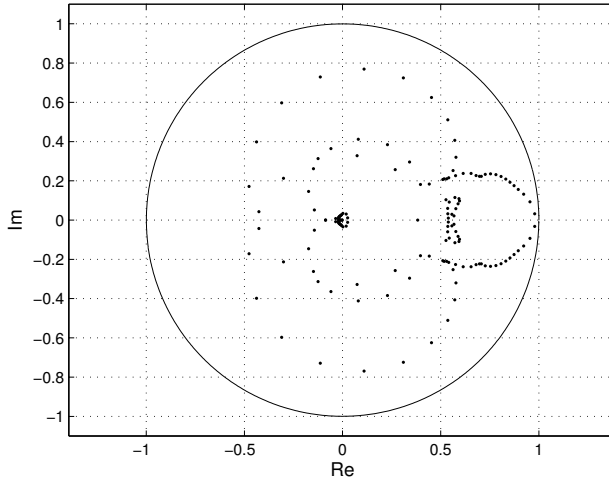


Figure 6.2: Eigenvalue spectra of the two-level algorithm with EXI smoothing in the steady advection dominated case ($C_{\Delta t} = 100$ and $Re_h = 100$).

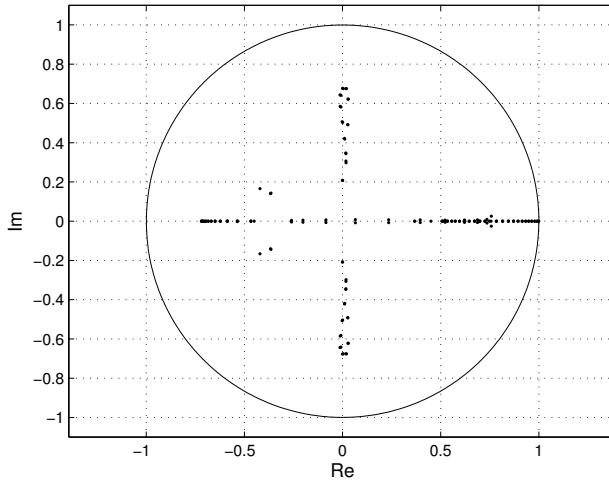


Figure 6.3: Eigenvalue spectra of the two-level algorithm with EXV smoothing in the steady diffusion dominated case ($C_{\Delta t} = 100$ and $Re_h = 0.01$).

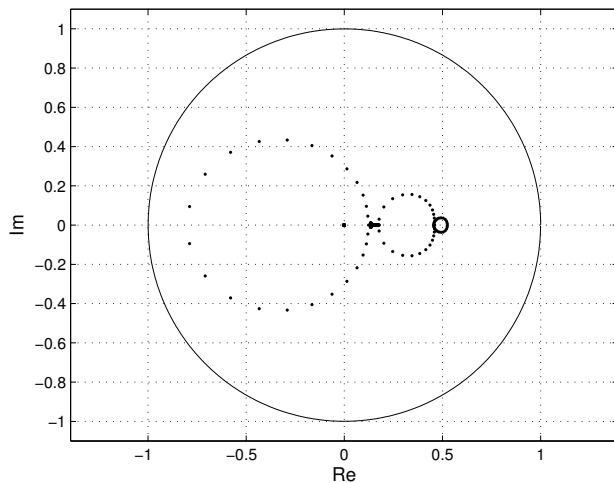


Figure 6.4: Eigenvalue spectra of the two-level algorithm with EXI smoothing in the unsteady advection dominated case ($C_{\Delta t} = 1$ and $Re_h = 100$).

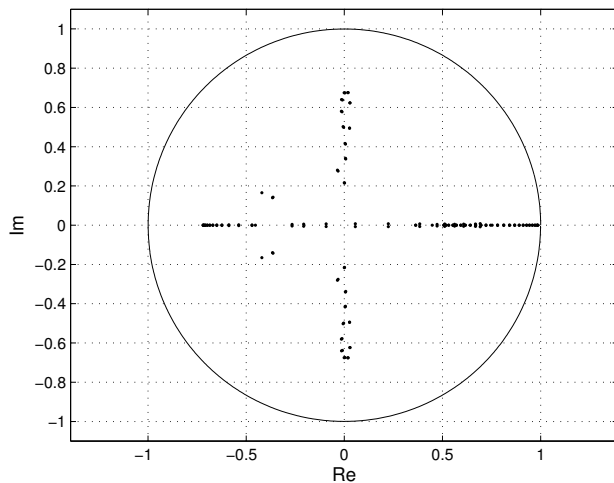


Figure 6.5: Eigenvalue spectra of the two-level algorithm with EXV smoothing in the unsteady diffusion dominated case ($C_{\Delta t} = 1$ and $Re_h = 0.01$).

Table 6.1: Spectral radii in the advection dominated flow regimes.

flow regime		stability	convergence	
$C_{\Delta t}$	Pe_h	$\Delta\tau/\Delta t$	$\rho(M_h^{\text{EXI}})$	$\rho(M_h^{\text{TLA}})$
100	100	$1.8 \cdot 10^{-4}$	0.991	0.979
1	100	1.6	0.796	0.794

Table 6.2: Spectral radii in the diffusion dominated flow regimes.

flow regime		stability	convergence	
$C_{\Delta t}$	Pe_h	$\Delta\tau/\Delta t$	$\rho(M_h^{\text{EXV}})$	$\rho(M_h^{\text{TLA}})$
100	0.01	$8.0 \cdot 10^{-7}$	0.999	0.998
1	0.01	$8.0 \cdot 10^{-3}$	0.993	0.985

the number of elements is N_e/f_e with f_e the mesh coarsening factor and $N_c = 1$ because we use constant basis functions. Therefore, the number of DoF on the coarse mesh is $1/(f_e N_c)$ with respect to fine mesh. For example, in 3D, $f_e = 8$ and $N_c = 5$ so forty coarse grid Runge-Kutta steps are counted as one work unit. A similar counting is done for multiple levels. The prolongation and restriction are trivial and this effort is neglected.

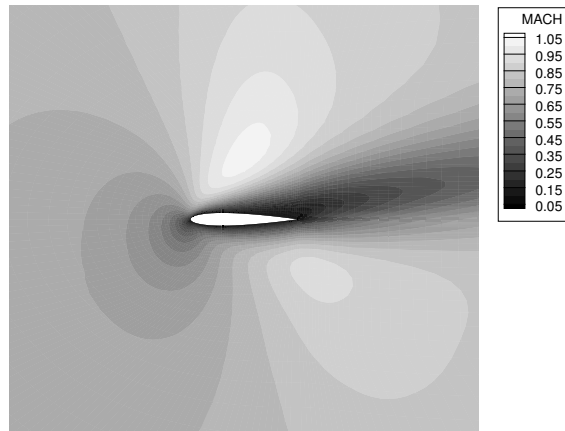
NACA 0012 airfoil, steady-state case. To test the multigrid performance for the space-time DG discretization of the compressible Navier-Stokes equations, we first consider the A1 case described in [12]. In this test case the far-field Reynolds number is $Re_\infty = 73$, the far-field Mach number is $M_\infty = 0.8$ and the angle of attack is $\alpha = 10^\circ$. This gives a steady-state solution with a relatively thick (diffusion dominated) boundary layer; advection dominates outside the boundary layer. We solve the problem on a C-type grid with 224×76 elements (see Figure 4.6) and start by comparing the solution obtained with constant and linear basis functions in Figure 6.6. Clearly, with constant basis functions the discretization is more dissipative than with linear basis functions. However, as can be seen in Figure 6.7, constant basis functions for the coarse grid correction is enough to reduce the work load. Here, the multigrid iteration consists of a V-cycle with three levels and 3 pre- and post-relaxations on each level. The number of work units needed to obtain four orders of convergence with multigrid is less than

half of the number needed for single-grid computations.

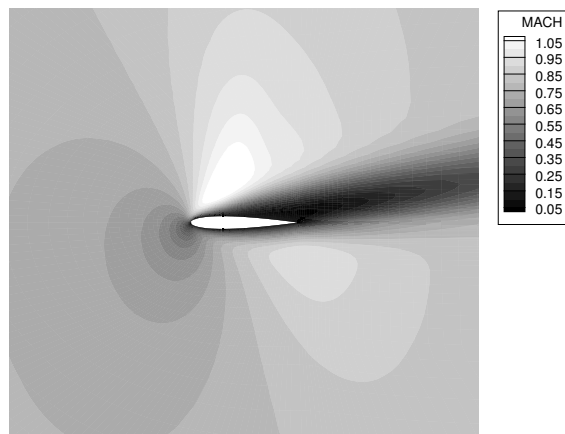
NACA 0012 airfoil, time-dependent case. Second, we consider the A7 case described in [12]. In this test case, the far-field Reynolds number is $Re_\infty = 10^4$, the far-field Mach number is $M_\infty = 0.85$ and the angle of attack is $\alpha = 0^\circ$. The case is time-dependent with a much thinner boundary layer than the A1 case (see Figure 4.8). We use three level multigrid with 3 pre- and post-relaxations on the fine C-type mesh with 224×76 elements. The physical time step needed to accurately capture the vortex shedding is $\Delta t = 0.05$, see Section 4.3.2. In Figure 6.8, we show the convergence in pseudo-time for three physical time steps corresponding to $t = 10, 10.05$ and 10.1 . As expected from the analysis, the convergence is much better than in the steady-state case: three orders of convergence are achieved within 600 work units. In this time-dependent case, where advection dominates in most of the domain, multigrid iteration does not improve the convergence.

3D delta wing. Finally, we consider the steady state flow around the 75° delta wing used in the experiments by Riley and Lawson [39], see Chapter 3 for the solution and Chapter 4 for the convergence results with the (single-grid) combined EXI and EXV method. In this case, the far-field Reynolds number is $Re_\infty = 4 \cdot 10^4$, the far-field Mach number $M_\infty = 0.3$ and the angle of attack is $\alpha = 12.5^\circ$. The solution on a coarse mesh with 209 000 elements is computed first. Then, the grid is refined in vortex regions as follows. An element is subdivided in any of the three directions if the vorticity is greater than $2a_\infty/c$ (with c the length of the wing) unless the mesh width in that direction is already $0.005c$ or less. This gives a uniform resolution around the primary vortices.

In Figure 6.9, we show the difference between single and multigrid convergence. In this case, the multigrid algorithm consists of a V-cycle with two levels and three pre- and post-relaxations on each level. On the original grid, multigrid iteration converges twice as fast as single grid iteration: three orders of convergence in two thousand work units. After each mesh refinement, the residual has to be decreased again. With multigrid iteration this costs a few hundred work units, considerably less than with single-grid iteration.



(a) Constant basis functions



(b) Linear basis functions

Figure 6.6: Solution of the A1 case ($Re_\infty = 73$, $M_\infty = 0.8$, $\alpha = 10^\circ$) on the C-type grid with 224×76 elements.

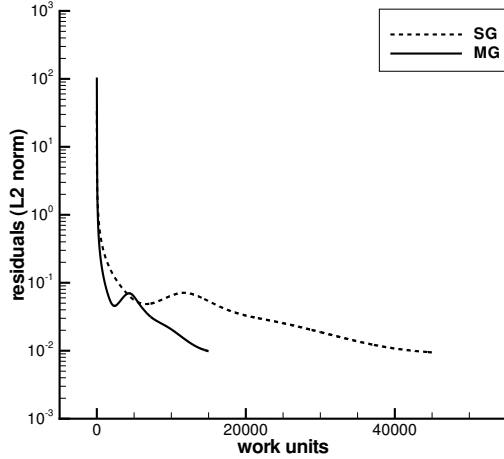


Figure 6.7: Single-grid (SG) and multigrid (MG) convergence for the A1 case ($Re_\infty = 73$, $M_\infty = 0.8$, $\alpha = 10^\circ$) on the C-type grid with 224×76 elements.

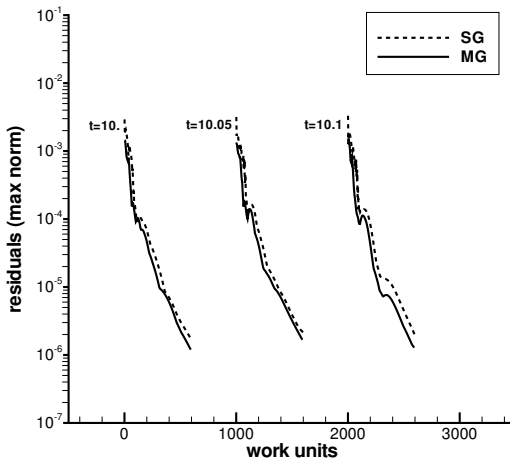
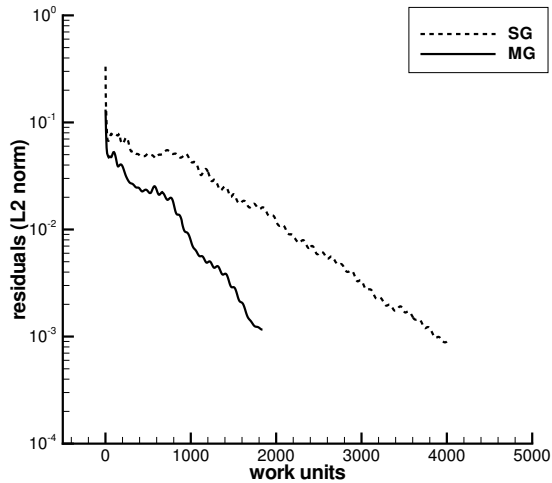
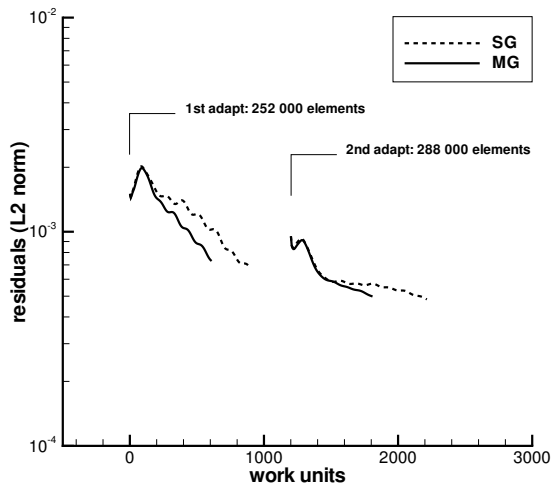


Figure 6.8: Single-grid (SG) and multigrid (MG) convergence in pseudo-time for three physical time steps of the A7 case ($Re_\infty = 10^4$, $M_\infty = 0.85$, $\alpha = 0^\circ$) on the C-type grid with 224×76 elements.



(a) original grid (209 000 elements)



(b) original grid adapted twice

Figure 6.9: Single-grid (SG) and multigrid (MG) convergence for the delta wing ($Re_\infty = 4 \cdot 10^4$, $M_\infty = 0.3$, $\alpha = 12.5^\circ$).

Chapter 7

Conclusions and recommendations

The performance of the space-time discontinuous Galerkin discretization in combination with multigrid pseudo-time integration was tested for various 2D and 3D simulations of compressible flow. In this chapter, we draw conclusions from these numerical experiments and give recommendations for further research.

Space-time method. We presented a space-time discontinuous Galerkin method for the compressible Navier-Stokes equations aimed at the accurate solution of time dependent problems on moving and deforming grids. The method does not distinguish between space and time, thereby providing optimal flexibility to accommodate time dependent boundaries and element deformation. We have discussed our choices for the space-time numerical fluxes and emphasized the treatment of the viscous part of the Navier-Stokes equations needed to maintain locality of the stencil and optimal order of accuracy.

The method was implemented in the NLR program HEXADAP, parallelized using OpenMP and typically runs at 6.4 Gflops/s on 8 processors of an SGI Altix supercomputer. It accurately handles complex aerodynamical problems, which we demonstrated by computing the flow around a 3D delta wing and around a NACA0012 airfoil in rapid pitch-up maneuver. We found that the results for the 3D delta wing on an adapted mesh with

300 000 elements are comparable with those on a (costly) fine mesh with 1 600 000 elements. The method remains accurate even in the case of significant mesh movement and deformation as required by the NACA0012 airfoil in rapid pitch-up maneuver.

Pseudo-time integration. When applying the space-time discontinuous Galerkin method to the compressible Navier-Stokes equations one obtains a system of non-linear algebraic equations. To solve this system we developed a fully explicit pseudo-time stepping method suitable for a wide range of applications. This was achieved by combining two explicit Runge-Kutta methods: the EXI method for the inviscid part of the flow domain and the EXV method for the viscous part.

We showed that these methods are stable if either the pseudo-time CFL condition or the Von Neumann condition is satisfied, depending on the cell Reynolds number. In the inviscid flow regime, the cell Reynolds numbers are large and the CFL condition is the most restrictive, while in the viscous flow regime small cell Reynolds numbers occur so that the Von Neumann condition is the most restrictive. To assess the influence of viscosity on the performance of the explicit methods, we also considered the implicit-explicit version of the EXI method where the viscous terms are treated implicitly and the inviscid terms explicitly. We showed that the stability of the IMEX method only depends on the inviscid part, thereby effectively relieving us of the Von Neumann condition, providing a method against which we could test our explicit algorithms.

The combination of the EXI and EXV method significantly improves the convergence in pseudo-time compared to the EXI method, yet viscosity is still limiting the convergence when compared to the IMEX results in terms of number of iterations. However, in terms of CPU-time, the IMEX method is significantly slower due to the linear system which must be solved at each iteration. The main advantage of the combined EXI and EXV method is its locality, which matches the locality of the space-time discontinuous Galerkin discretization and dispenses with the assemblage and storage of a global (sparse) matrix.

Multigrid. To further reduce the computational effort, we applied the combined EXI and EXV method as a smoother in a multigrid algorithm.

We analyzed the convergence of Runge-Kutta multigrid iteration for solving the system of algebraic equations resulting from the space-time DG discretization of the scalar advection-diffusion equation. In this two-level analysis we avoid the cell-staggering problem of transferring cell data from coarse to fine cells, by associating the data in a cell with a nodal point.

The analysis shows that explicit Runge-Kutta methods can be applied as smoothers in an h -multigrid setting. Depending on the case under consideration, two-level convergence factors between 0.48 and 0.74 are obtained. With a numerical illustration we showed the multigrid convergence behavior in practice and we found convergence factors that are in agreement with those obtained from the analysis. The advantage of using Runge-Kutta smoothing is that it results in a fully explicit approach, which can be extended to complex multidimensional problems where implicit smoothing may be too costly.

For the pseudo-time integration of the system of algebraic equations arising from the space-time DG discretization of the compressible Navier-Stokes equations, h -multigrid is complicated and we have limited ourselves to an easier approach. The main idea is to exploit the construction of the basis functions which separates element means from fluctuations and only use the equations for the element means on the coarse grids in the multigrid method. This facilitates the definition of the inter-grid transfer operators for non-uniform grids where the embedding of spaces does not hold.

We applied two-level Fourier analysis to the multigrid algorithm with constant basis functions on the coarse grid and found two-level convergence factors ranging between 0.79 and 0.98, depending on the case. Based on the analysis, we expect the multigrid algorithm to be twice as fast as the single-grid algorithm for steady-state cases and this was confirmed by 2D and 3D numerical simulations of the compressible Navier-Stokes equations. This leads us to conclude that the two-level analysis of the scalar advection diffusion equation gives useful predictions for the multigrid behavior of the 3D compressible Navier-Stokes equations.

Further research. The space-time DG method gives accurate results on moving, deforming and locally refined meshes needed in many flow problems. However, high computational costs currently prevent its application in industry. These costs are related to the number of iterations required by

the explicit pseudo-time integration which solves the system of algebraic equations. For time-dependent flows at moderate Reynolds numbers ten to twenty pseudo-time steps per physical time step would be acceptable; currently we still need hundreds. We have shown that, in principle, h -multigrid can achieve the necessary cost reduction. Further research should therefore primarily aim at bringing h -multigrid into practice, which implies developing inter-grid transfer operators suitable for situations where the coarse and fine grid are not nested, for instance near curvilinear boundaries.

Appendix A

Computational details

A.1 Homogeneity tensor

The elements of the homogeneity tensor (A_{ikrs}) are calculated by applying the definition:

$$A_{ikrs}(U) := \frac{\partial F_{ik}^v(U, \bar{\nabla}U)}{\partial(U_{r,s})},$$

for $i, r = 1, \dots, 5$ and $k, s = 1, \dots, 3$ and by using the Stokes hypothesis $3\lambda + 2\mu = 0$ to eliminate λ . For clarity's sake, the elements are grouped in the following matrices:

$$\begin{aligned} A_{11} &:= A|_{s=1}^{k=1}, & A_{12} &:= A|_{s=2}^{k=1}, & A_{13} &:= A|_{s=3}^{k=1}, \\ A_{21} &:= A|_{s=1}^{k=2}, & A_{22} &:= A|_{s=2}^{k=2}, & A_{23} &:= A|_{s=3}^{k=2}, \\ A_{31} &:= A|_{s=1}^{k=3}, & A_{32} &:= A|_{s=2}^{k=3}, & A_{33} &:= A|_{s=3}^{k=3}. \end{aligned}$$

which are given by:

$$A_{11} = \frac{1}{\rho} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ -\frac{4}{3}\mu u_1 & \frac{4}{3}\mu & 0 & 0 & 0 \\ -\mu u_2 & 0 & \mu & 0 & 0 \\ -\mu u_3 & 0 & 0 & \mu & 0 \\ A_{5111} & (\frac{4}{3}\mu - \frac{\kappa}{c_v})u_1 & (\mu - \frac{\kappa}{c_v})u_2 & (\mu - \frac{\kappa}{c_v})u_3 & \frac{\kappa}{c_v} \end{bmatrix},$$

with

$$A_{5111} = -\frac{1}{3}\mu u_1^2 - \mu \|\vec{u}\|^2 - \frac{\kappa}{c_v}(E - \|\vec{u}\|^2),$$

$$\begin{aligned}
A_{12} &= \frac{1}{\rho} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ \frac{2}{3}\mu u_2 & 0 & -\frac{2}{3}\mu & 0 & 0 \\ -\mu u_1 & \mu & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ -\frac{1}{3}\mu u_1 u_2 & \mu u_2 & -\frac{2}{3}\mu u_1 & 0 & 0 \end{bmatrix}, \\
A_{13} &= \frac{1}{\rho} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ \frac{2}{3}\mu u_3 & 0 & 0 & -\frac{2}{3}\mu & 0 \\ 0 & 0 & 0 & 0 & 0 \\ -\mu u_1 & \mu & 0 & 0 & 0 \\ -\frac{1}{3}\mu u_1 u_3 & \mu u_3 & 0 & -\frac{2}{3}\mu u_1 & 0 \end{bmatrix}, \\
A_{21} &= \frac{1}{\rho} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ -\mu u_2 & 0 & \mu & 0 & 0 \\ \frac{2}{3}\mu u_1 & -\frac{2}{3}\mu & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ -\frac{1}{3}\mu u_1 u_2 & -\frac{2}{3}\mu u_2 & \mu u_1 & 0 & 0 \end{bmatrix}, \\
A_{22} &= \frac{1}{\rho} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ -\mu u_1 & \mu & 0 & 0 & 0 \\ -\frac{4}{3}\mu u_2 & 0 & \frac{4}{3}\mu & 0 & 0 \\ -\mu u_3 & 0 & 0 & \mu & 0 \\ A_{5212} & (\mu - \frac{\kappa}{c_v})u_1 & (\frac{4}{3}\mu - \frac{\kappa}{c_v})u_2 & (\mu - \frac{\kappa}{c_v})u_3 & \frac{\kappa}{c_v} \end{bmatrix},
\end{aligned}$$

with

$$A_{5212} = -\frac{1}{3}u_2^2 - \mu\|\vec{u}\|^2 - \frac{\kappa}{c_v}(E - \|\vec{u}\|^2),$$

$$\begin{aligned}
A_{23} &= \frac{1}{\rho} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ \frac{2}{3}\mu u_3 & 0 & 0 & -\frac{2}{3}\mu & 0 \\ -\mu u_2 & 0 & \mu & 0 & 0 \\ -\frac{1}{3}\mu u_2 u_3 & 0 & \mu u_3 & -\frac{2}{3}\mu u_2 & 0 \end{bmatrix}, \\
A_{31} &= \frac{1}{\rho} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ -\mu u_3 & 0 & 0 & \mu & 0 \\ 0 & 0 & 0 & 0 & 0 \\ \frac{2}{3}\mu u_1 & -\frac{2}{3}\mu & 0 & 0 & 0 \\ -\frac{1}{3}\mu u_1 u_3 & -\frac{2}{3}\mu u_3 & 0 & \mu u_1 & 0 \end{bmatrix}, \\
A_{32} &= \frac{1}{\rho} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ -\mu u_3 & 0 & 0 & \mu & 0 \\ \frac{2}{3}\mu u_2 & 0 & -\frac{2}{3}\mu & 0 & 0 \\ -\frac{1}{3}\mu u_2 u_3 & 0 & -\frac{2}{3}\mu u_3 & \mu u_2 & 0 \end{bmatrix},
\end{aligned}$$

$$A_{33} = \frac{1}{\rho} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ -\mu u_1 & \mu & 0 & 0 & 0 \\ -\mu u_2 & 0 & \mu & 0 & 0 \\ -\frac{4}{3}\mu u_3 & 0 & 0 & \frac{4}{3}\mu & 0 \\ A_{5313} & (\mu - \frac{\kappa}{c_v})u_1 & (\mu - \frac{\kappa}{c_v})u_2 & (\frac{4}{3}\mu - \frac{\kappa}{c_v})u_3 & \frac{\kappa}{c_v} \end{bmatrix},$$

with

$$A_{5313} = -\frac{1}{3}\mu u_3^2 - \mu \|\vec{u}\|^2 - \frac{\kappa}{c_v}(E - \|\vec{u}\|^2).$$

A.2 IMEX method

The system of non-linear algebraic equations for the expansion coefficients \hat{U}^n in space-time slab I_n can be written as:

$$\mathcal{L}(\hat{U}^n, \hat{U}^{n-1}) = 0,$$

with $\mathcal{L} = \mathcal{L}^e + \mathcal{L}^v$. The term \mathcal{L}^e corresponds to the inviscid part of the residuals and \mathcal{L}^v to the viscous part of the residual, see Chapter 3 for details of the derivation. The IMEX method uses a Newton matrix \mathcal{D} which is an approximation of the Jacobian of \mathcal{L}^v . In this Appendix, we give the necessary details to compute \mathcal{D} .

A.2.1 Linearization of the \mathcal{L}^v

The viscous part \mathcal{L}^v of the residual is defined as:

$$\mathcal{L}_{il}^v = \sum_{\mathcal{K} \in \mathcal{T}_h^n} D_{il} + \sum_{\mathcal{S} \in \mathcal{S}_{I,B}^n} (-F_{il} - G_{il} + H_{il}),$$

with $i = 1, \dots, N_p$ the physical equation index, $l = 0, \dots, N_c$ the element expansion coefficient index. The coefficient matrices are defined as:

$$D_{il} = \int_{\mathcal{K}} \psi_{l,k} A_{ikrs} U_{r,s} d\mathcal{K},$$

$$F_{il} = \begin{cases} \int_{\mathcal{S}} \{\psi_{l,k} A_{ikrs}\} \llbracket U_r \rrbracket_s d\mathcal{S} & \text{for } \mathcal{S} \in \mathcal{S}_I, \\ \int_{\mathcal{S}} \psi_{l,k}^L A_{ikrs}^L (U_r^L - U_r^b) \bar{n}_s^L d\mathcal{S} & \text{for } \mathcal{S} \in \mathcal{S}_B, \end{cases}$$

$$G_{il} = \begin{cases} \int_{\mathcal{S}} \llbracket \psi_l \rrbracket_k \{\{A_{ikrs} U_{r,s}\}\} d\mathcal{S} & \text{for } \mathcal{S} \in \mathcal{S}_I, \\ \int_{\mathcal{S}} \psi_l^L (A_{ikrs}^b U_{r,s}^b) \bar{n}_k^L d\mathcal{S} & \text{for } \mathcal{S} \in \mathcal{S}_B, \end{cases}$$

$$H_{il} = \begin{cases} \eta \int_{\mathcal{S}} \llbracket \psi_l \rrbracket_k \{ \mathcal{R}_{ik}^{\mathcal{S}} \} d\mathcal{S} & \text{for } \mathcal{S} \in \mathcal{S}_I, \\ \eta \int_{\mathcal{S}} \psi_l^L \mathcal{R}_{ik}^{\mathcal{S}} \bar{n}_k^L d\mathcal{S} & \text{for } \mathcal{S} \in \mathcal{S}_B. \end{cases}$$

Since the terms D , F , G and H depend on the non-linear homogeneity tensor A , the viscous part is non-linear in \hat{U} . The terms are linearized by freezing the homogeneity tensor A at the element mean of the previous Runge-Kutta stage: $A \cong A(\bar{V}^{s-1})$ so that A can be treated as a constant. With this approximation, the terms D , F , G and H of \mathcal{L}^v become linear in \hat{U} and can be expressed using the following basic integrals:

$$\begin{aligned} I_{klms}^{\mathcal{K}} &= \int_{\mathcal{K}} \psi_{l,k} \psi_{m,s} d\mathcal{K}, & N_{klms}^{LR} &= \int_{\mathcal{S}} \psi_{l,k}^L \psi_m^R \bar{n}_s^R d\mathcal{S}, \\ P_{klms}^{LR} &= \int_{\mathcal{S}} \psi_l^L \psi_{m,s}^R \bar{n}_k^L d\mathcal{S}, & S_{lkj}^{LR} &= \int_{\mathcal{S}} \psi_l^L \psi_j^R \bar{n}_k^L d\mathcal{S}, \\ L_{nms}^{LR} &= \int_{\mathcal{S}} \psi_n^L \psi_m^R \bar{n}_s^R d\mathcal{S}. \end{aligned}$$

As an example, consider the approximation of the coefficient matrix D . The linearization of D for an element \mathcal{K} in terms of the basic integral $I^{\mathcal{K}}$ and the frozen homogeneity tensor A is given by:

$$D_{il} \equiv \int_{\mathcal{K}} \psi_{l,k} A_{ikrs} U_{r,s} d\mathcal{K} = \int_{\mathcal{K}} \psi_{l,k} A_{ikrs} \psi_{m,s} \hat{U}_{rm} d\mathcal{K} \cong A_{ikrs}^{\mathcal{K}} I_{klms}^{\mathcal{K}} \hat{U}_{rm}^{\mathcal{K}}.$$

In a similar way, we obtain the linearization of the coefficient matrices F , G and H for an internal face $\mathcal{S} \in \mathcal{S}_I$ in terms of the basic integrals and the homogeneity tensor:

$$\begin{aligned} F_{il}^{\mathcal{S}} &\cong \left(\frac{1}{2} A_{ikrs}^L N_{klms}^{LL} + \frac{1}{2} A_{ikrs}^R N_{klms}^{RL} \right) \hat{U}_{rm}^L \\ &\quad + \left(\frac{1}{2} A_{ikrs}^L N_{klms}^{LR} + \frac{1}{2} A_{ikrs}^R N_{klms}^{RR} \right) \hat{U}_{rm}^R, \\ G_{il}^{\mathcal{S}} &\cong \left(\frac{1}{2} A_{ikrs}^L P_{klms}^{LL} + \frac{1}{2} A_{ikrs}^L P_{klms}^{RL} \right) \hat{U}_{rm}^L \\ &\quad + \left(\frac{1}{2} A_{ikrs}^R P_{klms}^{LR} + \frac{1}{2} A_{ikrs}^R P_{klms}^{RR} \right) \hat{U}_{rm}^R, \\ H_{il}^{\mathcal{S}} &\cong \frac{1}{4} \eta (M^{-1})_{jn}^L L_{nms}^{LL} A_{ikrs}^L (S_{lkj}^{LL} + S_{lkj}^{RL}) \hat{U}_{rm}^L \\ &\quad + \frac{1}{4} \eta (M^{-1})_{jn}^R L_{nms}^{RL} A_{ikrs}^R (S_{lkj}^{LR} + S_{lkj}^{RR}) \hat{U}_{rm}^R \\ &\quad + \frac{1}{4} \eta (M^{-1})_{jn}^L L_{nms}^{LR} A_{ikrs}^L (S_{lkj}^{LL} + S_{lkj}^{RL}) \hat{U}_{rm}^R \\ &\quad + \frac{1}{4} \eta (M^{-1})_{jn}^R L_{nms}^{RR} A_{ikrs}^R (S_{lkj}^{LR} + S_{lkj}^{RR}) \hat{U}_{rm}^R. \end{aligned}$$

These approximations can now be used to construct the matrix \mathcal{D} , needed in the IMEX-RK5 method.

A.2.2 Construction of the Newton matrix \mathcal{D}

Note that the different contributions to the viscous part of the residual are all of the same form:

$$\Xi_{il} = \Lambda_{ilrm} \hat{U}_{rm},$$

where Λ depends on the basic integrals I , N , P , S and L and on the frozen homogeneity tensor A . These contributions can be expressed as matrix vector products of the form:

$$\Xi_q = \hat{\Lambda}_{qp} \hat{U}_p,$$

by applying the transformations $q = (i-1)N_c + l$ and $p = (r-1)N_c + m$ with $i, r \in \{1, \dots, N_p\}$ the physical equation index and $l, m \in \{1, \dots, N_c\}$ the element expansion coefficient index. The implicit matrix $\mathcal{D} \in \mathbb{R}^{N_n N_p N_c \times N_n N_p N_c}$ consists of $N_n \times N_n$ blocks of size $N_p N_c \times N_p N_c$ and is constructed as follows.

Algorithm A.1 (Matrix \mathcal{D}). The construction of the matrix \mathcal{D} .

1. Initialize $\mathcal{D} = 0$.
2. For all elements $\mathcal{K} \in \mathcal{T}_h^n$ do:

$$[\mathcal{D}_{qp}]^{\mathcal{K}\mathcal{K}} \leftarrow [\mathcal{D}_{qp}]^{\mathcal{K}\mathcal{K}} + A_{ikrs} I_{klms}^{\mathcal{K}}.$$

3. For all faces $\mathcal{S} \in \mathcal{S}_I^n$ do:

$$\begin{aligned} [\mathcal{D}_{qp}]^{LL} \leftarrow [\mathcal{D}_{qp}]^{LL} &- \frac{1}{2} A_{ikrs}^L N_{klms}^{LL} - \frac{1}{2} A_{ikrs}^L P_{klms}^{LL} \\ &+ \frac{1}{4} \eta (M^{-1})_{jn}^L L_{nms}^{LL} A_{ikrs}^L S_{lkj}^{LL} \\ &+ \frac{1}{4} \eta (M^{-1})_{jn}^R L_{nms}^{RL} A_{ikrs}^R S_{lkj}^{LR}, \end{aligned}$$

$$\begin{aligned} [\mathcal{D}_{qp}]^{LR} \leftarrow [\mathcal{D}_{qp}]^{LR} &- \frac{1}{2} A_{ikrs}^L N_{klms}^{LR} - \frac{1}{2} A_{ikrs}^R P_{klms}^{LR} \\ &+ \frac{1}{4} \eta (M^{-1})_{jn}^L L_{nms}^{LR} A_{ikrs}^L S_{lkj}^{LL} \\ &+ \frac{1}{4} \eta (M^{-1})_{jn}^R L_{nms}^{RR} A_{ikrs}^R S_{lkj}^{LR}, \end{aligned}$$

$$\begin{aligned}
[\mathcal{D}_{qp}]^{RL} \leftarrow [\mathcal{D}_{qp}]^{RL} &- \frac{1}{2}A_{ikrs}^R N_{klms}^{RL} - \frac{1}{2}A_{ikrs}^L P_{klms}^{RL} \\
&+ \frac{1}{4}\eta(M^{-1})_{jn}^L L_{nms}^{LL} A_{ikrs}^L S_{lkj}^{RL} \\
&+ \frac{1}{4}\eta(M^{-1})_{jn}^R L_{nms}^{RL} A_{ikrs}^R S_{lkj}^{RR},
\end{aligned}$$

$$\begin{aligned}
[\mathcal{D}_{qp}]^{RR} \leftarrow [\mathcal{D}_{qp}]^{RR} &- \frac{1}{2}A_{ikrs}^R N_{klms}^{RR} - \frac{1}{2}A_{ikrs}^L P_{klms}^{RR} \\
&+ \frac{1}{4}\eta(M^{-1})_{jn}^L L_{nms}^{LR} A_{ikrs}^L S_{lkj}^{RL} \\
&+ \frac{1}{4}\eta(M^{-1})_{jn}^R L_{nms}^{RR} A_{ikrs}^R S_{lkj}^{RR}.
\end{aligned}$$

4. Return \mathcal{D} .

Since the Algorithm A.1 consists of loops over elements and faces, it is directly suitable for structured and unstructured meshes, with or without grid adaptation. It also becomes clear that the matrix \mathcal{D} is a sparse block matrix. For each block row, \mathcal{D} contains one block on the diagonal and the number of blocks off the diagonal is equal to the number of faces of the element. The homogeneity tensor A is neither symmetric nor positive definite, so neither is the matrix \mathcal{D} . We therefore solve the system with the robust sparse iterative GMRES solver with Jacobi preconditioning, available from the PETSc package.

A.3 EXV method

In [28] a family of Runge-Kutta schemes for efficient time-marching of viscous flow problems is presented. We used a member of this family, the EXV method, for local pseudo-time stepping in flow regions with low cell Reynolds numbers. In this appendix, we summarize the derivation of the entire family.

Consider the following N stage Runge-Kutta scheme:

1. Initialize $\hat{v}^0 = \hat{u}$.
2. For all stages $s = 1$ to N compute \hat{v}^s as:

$$\hat{v}^s = \hat{v}^0 - \alpha_s \lambda \mathcal{L}(\hat{v}^{s-1}; \hat{u}^{n-1}).$$

3. Return $\hat{u} = \hat{v}^N$.

When applied to the simple model problem:

$$\frac{\partial u}{\partial \tau} = -\frac{\mu}{\Delta t}u,$$

the stages s are updated according to: $v^s = v^0 - \alpha_s \lambda \mu v^{s-1}$ and therefore the amplification factor G is of the form:

$$G_N(z) = 1 + \alpha_N z + \alpha_N \alpha_{N-1} z^2 + \cdots + \alpha_N \cdots \alpha_1 z^N, \quad (\text{A.1})$$

with $z = -\lambda \mu \in \mathbb{C}$. The family of Runge-Kutta schemes proposed in [28] can be derived by choosing the coefficients α_s in such a way that the amplification factor equals Manteuffel's transformation of Tchebyshev polynomials:

$$G_N(z) = \frac{T_N((d-z)/\epsilon)}{T_N(d/\epsilon)},$$

where T_N denotes the N -th Tchebyshev polynomial defined recursively as:

$$T_{n+1}(z) = 2zT_n(z) - T_{n-1}, \quad n \in \mathbb{N},$$

with $T_0(z) = 1$ and $T_1(z) = z$. Here, the parameter d defines the family of N stage Runge-Kutta schemes and the parameter ϵ is chosen such that:

$$G_N(0) = 1 \quad \text{and} \quad \left. \frac{dG_N}{dz} \right|_{z=0} = 1,$$

which ensures that the stability region touches the imaginary axis and is symmetric w.r.t. the real axis. The parameter d controls the scaling of the stability region.

The family member used in this paper is the 4-stage Runge-Kutta scheme with $d = -14$. For this scheme, we use the fourth order Tchebyshev polynomial: $T_4(z) = 8z^4 - 8z^2 + 1$ and obtain the following stability region:

$$G_4(z) = 1 + \frac{16\epsilon^2 d - 32d^3}{D} z + \frac{48d^2 - 8\epsilon^2}{D} z^2 - \frac{32d}{D} z^3 + \frac{8}{D} z^4, \quad (\text{A.2})$$

with $D = 8d^4 - 8d^2\epsilon^2 + \epsilon^4$. The coefficients α_s can now be computed by equating (A.2) with (A.1) which gives:

$$\alpha_1 = -\frac{1}{4d}, \quad \alpha_2 = \frac{4d}{\epsilon^2 - 6d^2}, \quad \alpha_3 = \frac{6d^2 - \epsilon^2}{2d(\epsilon^2 - 2d^2)}, \quad \alpha_4 = \frac{16d(\epsilon^2 - 2d^2)}{D}.$$

The condition $G_4(0) = 1$ is already satisfied and the condition on the derivative of G becomes:

$$\frac{16\epsilon^2 d - 32d^3}{D} = 1,$$

which has four solutions for ϵ from which we choose the following:

$$\epsilon = \sqrt{4d(d+2) - 2\sqrt{16d^2 + 8d^3 + 2d^4}}.$$

In the same way, we can derive the other members of the family. Note, however, that only even N produces consistent schemes.

Bibliography

- [1] D. Arnold, F. Brezzi, B. Cockburn, and D. Marini. Unified analysis of discontinuous Galerkin methods for elliptic problems. *SIAM J. Numer. Anal.*, 39:1749–1779, 2002.
- [2] F. Bassi, A. Crivellini, S. Rebay, and M. Savini. Discontinuous Galerkin solution of the Reynolds-averaged Navier-Stokes and $k - \omega$ turbulence model equations. *Computers & Fluids*, 34(4-5):507–540, 2004.
- [3] F. Bassi and S. Rebay. A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier-Stokes equations. *J. Comput. Phys.*, 131:267–279, 1997.
- [4] F. Bassi and S. Rebay. GMRES discontinuous Galerkin solution of the compressible Navier-Stokes equations. In B. Cockburn, G.Karniadakis, and C.-W. Shu, editors, *Discontinuous Galerkin Methods*, volume 11, pages 197–208. Springer, 1999.
- [5] F. Bassi and S. Rebay. Numerical evaluation of two discontinuous Galerkin methods for the compressible Navier-Stokes equations. *Int. J. Num. Meth. Fluids*, 40:197–207, 2002.
- [6] F. Bassi, S. Rebay, G.Mariotti, S. Pedinotti, and M. Savini. A high-order accurate discontinuous finite element method for inviscid and viscous turbomachinery flow. In R. Decuyper and G. Dibelius, editors, *2nd European conference on turbomachinery, fluid dynamics and thermodynamics*, pages 99–108. Technologisch Instituut, Antwerpen, 1997.

-
- [7] P. Batten, N. Clarke, C. Lambert, and D. Causon. On the choice of wavespeeds for the HLLC Riemann solver. *SIAM J. Sci. Comput.*, 18(6):1553–1570, 1997.
- [8] C.E. Baumann and J.T. Oden. A discontinuous hp finite element method for the Euler and Navier-Stokes equations. Tenth International Conference on Finite Elements in Fluids (Tucson, AZ, 1998). *Internat. J. Numer. Methods Fluids*, 31(1):79–95, 1999.
- [9] O.J. Boelens, H. van der Ven, B. Oskam, and A.A. Hassan. The boundary conforming discontinuous Galerkin finite element approach for rotorcraft simulations. *J. of Aircraft*, 39(5):776–785, 2002.
- [10] F. Brezzi, G. Manzini, D. Marini, P. Pietra, and A. Russo. Discontinuous Galerkin approximations for elliptic problems. *Numer. Meth. Part. Diff. Eq.*, 16(4):365–378, 2000.
- [11] W.L. Briggs, Van Emden Henson, and S.F. McCormick. *A Multigrid tutorial*. SIAM, 2000.
- [12] M.O. Bristeau, R. Glowinski, J. Periaux, and H. Viviand (Eds.). *Numerical simulation of compressible Navier-Stokes flows: A GAMM workshop, 4-6 December 1985, Nice, France*. Braunschweig, Vieweg, 1987.
- [13] B. Cockburn. Discontinuous Galerkin methods for convection-dominated problems. In T.J. Barth and H. Deconinck, editors, *Lect. Notes in Comp. Sci. and Eng.*, volume 9. Springer Verlag, 1999.
- [14] B. Cockburn. Discontinuous Galerkin methods. *ZAMM Z. Angew. Math. Mech.*, 11:731–754. 65–02, 2003.
- [15] B. Cockburn, G.E. Karniadakis, and C.-W. Shu (Eds.). Discontinuous Galerkin methods. Theory, computation and applications. *Lect. Notes in Comp. Sci. and Eng. (Springer Verlag, 2000)*, 11, 2000.
- [16] B. Cockburn and C.-W. Shu. The local discontinuous Galerkin method for time-dependent convection-diffusion systems. *SIAM J. Numer. Anal.*, 35:2240–2463, 1998.

- [17] B. Cockburn and C.-W. Shu. Runge-Kutta discontinuous Galerkin methods for convection-dominated problems. *J. Sci. Comput.*, 16(3):173–261, 2001.
- [18] V. Dolejší. On the discontinuous Galerkin method for the numerical solution of the Navier-Stokes equations. *Int. J. Numer. Meth. Fluids*, 45:1083–1106, 2004.
- [19] C. Farhat, P. Geuzaine, and C. Grandmont. The discrete geometric conservation law and the nonlinear stability of ALE schemes for the solution of flow problems on moving grids. *J. Comput. Phys.*, 174(2):669–694, 2001.
- [20] K.J. Fidkowski, T.A. Oliver, J. Lu, and D.L. Darmofal. p -Multigrid solution of high-order discontinuous Galerkin discretizations of the compressible Navier-Stokes equations. *J. Comput. Phys.*, 207(1):92–113, 2005.
- [21] J. Gopalakrishnan and G. Kanschat. A multilevel discontinuous Galerkin method. *Numerische Mathematic*, 95:527–550, 2003.
- [22] E. Hairer and G. Wanner. *Solving ordinary differential equations II. Stiff and differential-algebraic problems, Second revised edition*. Springer Verlag, Berlin, 2002.
- [23] R. Hartmann and P. Houston. Adaptive discontinuous Galerkin finite element methods with interior penalty for the compressible Navier-Stokes equations. In M. Feistauer, V. Dolejší, P. Knobloch, and K. Najzar, editors, *Numerical Mathematics and Advanced Applications, ENUMATH 2003*, pages 410–419. Springer, 2004.
- [24] R. Hartmann and P. Houston. Symmetric interior penalty DG methods for the compressible Navier-Stokes equations I: Method formulation. *Int. J. Num. Anal. Model.*, 3(1):1–20, 2006.
- [25] P.W. Hemker, W. Hoffmann, and M.H. van Raalte. Two-level Fourier analysis of a multigrid approach for discontinuous Galerkin discretisation. *SIAM Journal on Scientific Computing*, 25:1018–1041, 2004.

-
- [26] P.W. Hemker and M.H. van Raalte. Fourier two-level analysis for higher dimensional discontinuous Galerkin discretisation. *Computing and Visualization in Science*, 7:159–172, 2004.
- [27] K. Johannsen. Multigrid Methods for NIPG. Technical Report ICES 05-32, University of Texas, 2005.
- [28] W.L. Kleb, W.A. Wood, and B. van Leer. Efficient Multi-Stage Time Marching for Viscous Flows via Local Preconditioning. *AIAA J.*, 99-3267:181–194, 1999.
- [29] J. Kok. *An Industrially Applicable Solver for Compressible, Turbulent Flows*. PhD thesis, Delft University of Technology, 1998.
- [30] J. C. Kok and S. P. Spekreijse. Efficient and accurate implementation of the k - ω turbulence model in the NLR multi-block Navier-Stokes system. NLR TP-2000-144 (presented at ECCOMAS 2000, Barcelona, Spain, 11-14 September, 2000), 2000.
- [31] M. Lesoinne and C. Farhat. Geometric conservation laws for flow problems with moving boundaries and deformable meshes, and their impact on aeroelastic computations. *Comput. Methods. Appl. Mech. Eng.*, 134:71–90, 1996.
- [32] B.S. Massey. *Units, dimensional analysis and physical similarity*. Van Nostrand Reinhold company, 1971.
- [33] N.D. Melson, M.D. Sanetrik, and H.L. Atkins. Time-accurate Navier-Stokes calculations with multigrid acceleration. In *Proc. 6th Copper Mountain Confer. on Multigrid Methods*, 1993.
- [34] G.A. Osswald, K.N. Ghia, and U. Ghia. Simulation of dynamic stall phenomenon using unsteady Navier-Stokes equations. *Computer Physics Communications*, 65:209–218, 1991.
- [35] A. Ostermann. Stability of W-methods with applications to operator splitting and to geometric theory. *Appl. Num. Math.*, 42:353–366, 2002.
- [36] PETSc package. <http://www-unix.mcs.anl.gov/petsc/petsc-2>.

- [37] S.B. Pope. *Turbulent flows*. Cambridge University Press, 2000.
- [38] W. Reed and T. Hill. Triangular mesh methods for the neutron transport equation. Technical Report LA-UR 73-479, LANL, 1973.
- [39] A.J. Riley and M.V. Lowson. Development of a three dimensional free shear layer. *J. Fluid Mech.*, 369:49–89, 1998.
- [40] M. Stynes and E. O’Riordan. A Uniformly Convergent Galerkin Method on a Shishkin Mesh for Convection-Diffusion Problem. *J. Math. Anal. Appl.*, 214:36–54, 1997.
- [41] J.J. Sudirham, J.J.W. van der Vegt, and R.M.J. van Damme. Space-time discontinuous Galerkin method for advection-diffusion problems. Application to wet-chemical etching processes. *Appl. Numer. Mathematics (in press)*, 2006.
- [42] E.F. Toro. *Riemann solvers and numerical methods for fluid dynamics. A practical introduction*. Springer Verlag, 1997.
- [43] U. Trottenberg, C.W. Oosterlee, and A. Schüller. *Multigrid*. Academic Press, London, 2001.
- [44] J.J.W. van der Vegt and H. van der Ven. Space-time discontinuous Galerkin finite element method with dynamic grid motion for inviscid compressible flows. I. General formulation. *J. Comput. Phys.*, 182:546–585, 2002.
- [45] J.J.W. van der Vegt and H. van der Ven. Space-time discontinuous Galerkin finite element method with dynamic grid motion for inviscid compressible flows. In *33rd Computational Fluid Dynamics Course - Novel methods for solving convection dominated systems*, volume 1. VKI Lectures series Monographs: computational fluid dynamics, 2003.
- [46] H. van der Ven and O.J. Boelens. A framework for aeroelastic simulations of trimmed rotor systems in forward flight. In *Proceedings of the 30th European Rotorcraft Forum, Marseille, France, September 14-16, 2004*.

-
- [47] H. van der Ven and J.J.W. van der Vegt. Space-time discontinuous Galerkin finite element method with dynamic grid motion for inviscid compressible flows. II. Efficient flux quadrature. *Comput. Meth. Appl. Mech. Engrg.*, 191:4747–4780, 2002.
- [48] H. van der Ven, J.J.W. van der Vegt, and E.G. Bouwman. Space-time discontinuous Galerkin finite element method for inviscid gas dynamics. In *Computational fluid and solid mechanics 2003 (MIT Boston)*, volume 1, pages 1181–1184. Elsevier Science, Oxford, UK., 2003.
- [49] M. H. van Raalte. *Multigrid analysis and embedded boundary conditions for discontinuous Galerkin discretization*. PhD thesis, Korteweg-de Vries institute, University of Amsterdam, 2004.
- [50] M.R. Visbal and J.S. Shang. Investigation of the Flow Structure Around a Rapidly Pitching Airfoil. *AIAA J.*, 27(8):1044–1051, 1989.
- [51] P. Wesseling. A robust and efficient multigrid method. In W. Hackbush and U. Trottenberg, editors, *Multigrid Methods*, pages 614–630. Springer-Verlag, New York, 1982.
- [52] P. Wesseling. *Principles of Computational Fluid Dynamics*. Springer, 2000.

Summary

In this thesis, a space-time discontinuous Galerkin (DG) method for the compressible Navier-Stokes equations is presented. We explain the space-time setting, derive the weak formulation and discuss our choices for the numerical fluxes. The resulting numerical method allows local grid adaptation as well as moving and deforming boundaries, which we illustrate by computing the flow around a 3D delta wing on an adapted mesh and by simulating the dynamic stall phenomenon of a 2D airfoil in rapid pitch-up maneuver.

The space-time DG discretization results in a (non-linear) system of algebraic equations, which we solve with pseudo-time stepping methods. We show that explicit Runge-Kutta methods developed for the Euler equations suffer from a severe stability constraint linked to the viscous part of the equations and propose an alternative to relieve this constraint while preserving locality. To evaluate its effectiveness, we compare with an implicit-explicit Runge-Kutta method which does not suffer from the viscous stability constraint. We analyze the stability of the methods and show their performance for 2D and 3D simulations.

To improve the efficiency of the method, we apply fully explicit multigrid pseudo-time integration with Runge-Kutta smoothing. We analyze the convergence of multigrid iteration for solving the algebraic equations arising from a space-time DG discretization of the advection-diffusion equation. Depending on the Péclet number, we find multigrid convergence factors between 0.50 and 0.74 with Fourier two-level analysis. We illustrate the analysis with a numerical example.

For the Navier-Stokes equations, we consider multigrid with linear basis functions on the fine grid and constant basis functions on the coarse grids. This facilitates the definition of inter-grid transfer operators on non-uniform

grids. Two-level Fourier analysis shows that this multigrid iteration converges twice as fast as single-grid iteration for steady-state cases. This prediction is confirmed by the 2D and 3D simulations.

Samenvatting

In deze thesis wordt een ruimte-tijd discontinue Galerkin (DG) methode voor de compressibele Navier-Stokes vergelijkingen gepresenteerd. We geven uitleg over de ruimte-tijd, de afleiding van de zwakke vorm en onze keuzes voor de numerieke fluxen. De uiteindelijke numerieke methode maakt lokale grid verfijning mogelijk alsmede bewegende en vervormende grids, hetgeen we illustreren door de stroming rond een 3D delta vleugel uit te rekenen en het dynamische overtrekken van een 2D vleugelprofiel in een snelle opwaardse manoeuvre.

De ruimte-tijd DG discretizatie geeft een (niet-lineair) systeem van algebraïsche vergelijkingen, dat we oplossen met pseudo-tijdstaps methoden. We laten zien dat de expliciete Runge-Kutta methoden ontwikkeld voor de Euler vergelijkingen te lijden hebben onder een strenge stabiliteits voorwaarde gerelateerd aan het viskeuze deel van de vergelijkingen. We presenteren een alternatief om, onder behoud van lokaliteit, deze voorwaarde milder te maken. Om uit te zoeken hoe effectief dit is, vergelijken we voorgaande procedure met een impliciet-expliciete Runge-Kutta methode zonder viskeuze stabiliteits voorwaarde. We analyseren de stabiliteit van de methoden en laten hun prestaties zien voor 2D en 3D simulaties.

Om de efficiëntie van de methode verder te verbeteren, passen we volledige expliciete multigrid pseudo-tijds integratie toe, met Runge-Kutta damping. We analyseren de convergentie van multigrid iteratie voor het oplossen van de algebraïsche vergelijkingen voortgebracht door een ruimte-tijd DG discretisatie van de advection-diffusie vergelijking. Afhankelijk van het Péclet getal, vinden we multigrid convergentie factoren tussen 0.50 en 0.74 met Fourier twee-level analyse. We illustreren de analyse met een numeriek voorbeeld.

Voor de Navier-Stokes vergelijkingen bekijken we multigrid met lineai-

re basis functies op het fijne grid en constante basis functies op de grove grids. Dit maakt de definitie van inter-grid overdrachts operatoren op niet-uniforme grids eenvoudiger. Twee-level Fourier analyse wijst uit dat multigrid iteratie twee keer zo snel convergeert als enkel-grid iteratie voor tijdsonafhankelijke situaties. Deze voorspelling wordt bevestigd door de 2D en 3D simulaties.

Curriculum Vitae

Christiaan Marijn Klaij was born on the 3rd of November 1977 in Hyderabad, India. He grew up in India, the Netherlands, Niger, France and Ethiopia mainly being educated within the French school system. He got his scientific baccalaureate in 1996 at the French Lycée Guebre Mariam in Addis Ababa, Ethiopia, before moving to the Netherlands.

In 1996, he started studying Applied Mathematics at the University of Twente. In the final phase of his MSc he received a technical training from the Institut de Physique du Globe de Paris (IPGP) at their dependency in Poitiers. Under the supervision of dr. J.-F. Thovert, he worked on *Simulating well tests*; the simulation of compressible flow to a well in underground reservoirs for code debugging and validation purposes. Then, back in Twente, he worked under the supervision of prof.dr.ir. J.J.W. van der Vegt on *The effect of low Mach number on the accuracy of the discontinuous Galerkin method for compressible flow* and received his Master's degree in Applied Mathematics in 2002.

In 2002, he became a PhD student at the University of Twente under supervision of prof.dr.ir. J.J.W van der Vegt. His research was conducted under the umbrella of the STW involving a number of Institutions among which the NLR and CWI. This thesis contains the result of his research.