

# Discontinuous Galerkin Methods for the Unsteady Compressible Navier-Stokes Equations

A thesis accepted by the Faculty  
of Aerospace Engineering and Geodesy of the Universität Stuttgart  
in partial fulfilment of the requirements for the degree of  
Doctor of Engineering Sciences (Dr.-Ing.)

by

**Gregor Gassner**

from Fussach

Main-referee : Prof. Dr. Claus-Dieter Munz  
Co-referee : Prof. Dr. Jan S. Hesthaven  
Date of defence : 19.1.2009

Institut für Aerodynamik und Gasdynamik  
Universität Stuttgart

2009



Für Annemarie, Josef, Veronika, Karolin, Johanna und Emanuel.

## **Preface**

This thesis was developed during my work as scientific employee at the Institut für Aero- und Gasdynamik of the Universität Stuttgart.

Many thanks to my doctoral supervisor Prof. Dr. Claus-Dieter Munz for the exceptional working conditions in his research group, especially for all the scientific freedom I was granted under his supervision.

The work was financed by Deutsche Forschungsgemeinschaft (DFG).

Many thanks also to all my colleagues at IAG for the good working atmosphere and all the fruitful scientific discussions they had with me in the last three years. A special thank at this place to my colleague Frieder Lörcher.

Stuttgart, 3. September 2008

Gregor Gassner

# Contents

<b>Symbols</b>	<b>vii</b>
<b>Abbreviations</b>	<b>x</b>
<b>Kurzfassung</b>	<b>xi</b>
<b>Abstract</b>	<b>xii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Objectives . . . . .	1
1.2 Discontinuous Galerkin for Compressible Navier-Stokes . . . . .	2
1.3 Tensor and Differentiation Operator Notations . . . . .	2
1.4 Governing Equations . . . . .	4
1.5 Outline . . . . .	7
<b>2 Numerics</b>	<b>9</b>
2.1 Polynomial Representation of Data on Unstructured Grids . . . . .	9
2.1.1 State of the Art . . . . .	9
2.1.2 Polymorphic Nodal Elements . . . . .	10
2.1.3 Discussion . . . . .	16
2.2 Discontinuous Galerkin Formulations . . . . .	17
2.2.1 State of the Art . . . . .	17
2.2.2 The Ultra Weak Formulation . . . . .	20
2.2.3 Discussion . . . . .	22
2.3 Efficient Numerical Approximation of the DG Formulations . . . . .	23
2.3.1 State of the Art . . . . .	23
2.3.2 The Modal Approach with Nodal Integration . . . . .	26
2.3.3 Discussion . . . . .	27
2.4 The Numerical Fluxes . . . . .	29
2.4.1 State of the Art . . . . .	32
2.4.2 The Diffusive Generalized Riemann Problem . . . . .	34
2.4.3 Discussion . . . . .	38

2.5	Time Discretization . . . . .	39
2.5.1	State of the Art . . . . .	39
2.5.2	The Predictor Corrector Approach . . . . .	41
2.5.3	Discussion . . . . .	48
<b>3</b>	<b>Computations</b>	<b>51</b>
3.1	Validation and Proof of Concept . . . . .	51
3.1.1	Accuracy and Applicability of the Polymorphic Nodal Elements . . . . .	51
3.1.2	Efficiency of the Numerical Method . . . . .	58
3.1.3	Unresolved Problems . . . . .	62
3.2	Application . . . . .	68
3.2.1	Direct Numerical Simulation 2D . . . . .	68
3.2.2	Direct Numerical Simulation 3D . . . . .	73
<b>4</b>	<b>Prospects</b>	<b>83</b>
<b>A</b>	<b>Lebesgue Constants</b>	<b>85</b>
<b>B</b>	<b>Coefficients for CERK schemes</b>	<b>89</b>
<b>C</b>	<b>A Hybrid Grid for h-Refinement Studies</b>	<b>93</b>
<b>D</b>	<b>The Linearized Euler Equations</b>	<b>95</b>
<b>E</b>	<b>An Analytical Solution for the Euler Equations</b>	<b>97</b>
<b>F</b>	<b>An Analytical Solution for the Navier-Stokes Equations</b>	<b>99</b>
	<b>Bibliography</b>	<b>101</b>
	<b>List of Tables</b>	<b>109</b>
	<b>List of Figures</b>	<b>111</b>
	<b>Lebenslauf</b>	<b>113</b>

## Symbols

$a_{ij}, b_j, b_{ij}^t$	continuous extension Runge-Kutta coefficients
$c$	speed of sound
$C_d, C_l$	drag/lift coefficient
$c_p, c_v$	specific heats
$d$	spatial dimension
$h$	typical mesh size
$\ell, \ell_{2D/3D}$	parameters for the definition of the interpolation points
$K$	Kelvin
$\mathcal{K}^{1,G}$	Gauss integration matrix
$\mathcal{K}^{1,M}$	modal stiffness matrix
$\mathcal{K}^{1,N}$	nodal stiffness matrix
$L_2$	space of square-integrable functions
$m$	number of unknowns in a system of PDEs
$Ma$	Mach number
$\mathcal{M}_r$	mapping for the interior interpolation points
$\vec{n}$	normal vector
$n_G$	number of Gauss points
$n_h$	refinement level of hybrid grid
$n_L$	number of Lagrange (nodal) basis functions
$n_M$	number of modal basis functions
$n_P$	number of processors
$n_{stages}$	number of continuous extension Runge-Kutta stages
$p$	maximal polynomial degree
$p^*$	maximal interpolation degree supported with pure surface points
$p_L$	maximal interpolation polynomial degree
$p_t$	polynomial degree in time
$p$	pressure
$Pr$	Prandtl number

$\mathcal{P}_L$	set of interpolation points
$\mathcal{P}_L^S$	set of interpolation points on the grid cell surface
$\vec{q}$	heat flux vector
$Q$	grid cell
$r$	recursion level
$R$	specific gas constant
$\mathcal{R}_{DG}$	spatial discontinuous Galerkin residual
$\mathcal{R}_G$	spatial Galerkin residual
$Re$	Reynolds number
$S$	parallel efficiency
$Str$	Strouhal number
$t$	time
$t^n, t^{n+1}$	time levels
$T$	temperature
$T_0$	period of the fundamental frequency
$T_{infly}$	freestream temperature
$T_s$	Sutherland temperature
$u, U$	approximative solution
$u^e, U^e$	exact solution
$\hat{\mathbf{u}}$	vector of modal degrees of freedom
$\tilde{\mathbf{u}}$	vector of nodal degrees of freedom
$v$	velocity magnitude
$\vec{v}$	velocity vector
$V$	approximative solution of the Cauchy problem
$\mathbf{V}$	Vandermonde matrix
$\vec{x}$	spatial coordinate Vector
$x, x_1, x_2, y$	spatial coordinates
$\alpha$	advection time step stability number
$\beta$	diffusion time step stability number
$\tilde{\beta}$	parameter for the viscous flux
$\delta_1$	displacement thickness at inflow
$\delta_{99}$	boundary layer thickness
$\delta_{ij}$	Kronecker symbol
$\Delta C_d, \Delta C_l$	amplitudes of drag/lift



$\Delta t$	timestep
$\Delta t^A$	advection time step
$\Delta t^D$	diffusion time step
$\Delta x$	typical grid cell size for time step
$\Delta \Theta$	phase shift
$\eta, \tilde{\eta}$	viscous flux constants
$\eta_{BR2}$	constant of the BR2 flux approximation
$\eta_{SIP}$	penalty constant
$\kappa$	ratio of specific heats
$\Lambda$	Lebesgue constant
$\mu$	physical viscosity
$\nu$	artificial viscosity
$\tilde{\xi}$	interpolation point
$\pi_i$	monomial basis functions
$\rho$	density
$\rho e$	total energy per mass unit
$\sigma$	filter function
$\underline{\tau}$	viscous stress tensor
$\Phi, \phi, \varphi_i$	modal basis and test function
$\tilde{\chi}^i$	Gauss points
$\Psi, \psi_i$	nodal basis function
$\omega_0$	fundamental frequency
$\omega_i$	Gauss weights
$\Omega$	domain
$\partial Q$	grid cell boundary
$\vec{\nabla}$	nabla operator

## Abbreviations

BR2	second method of Bassi and Rebay
CERK	Continuous Extension Runge-Kutta
CERKG	Continuous Extension Runge-Kutta Galerkin
CFD	Computational Fluid Dynamics
CFL	Courant-Friedrichs-Levy number
CPU	Central Processing Unit
DG	Discontinuous Galerkin
dGRP	diffusive Generalized Riemann Problem
DOF	Degree(s) Of Freedom
EOC	Experimental Order of Convergence
EU	Element Update
FD	Finite Difference
FE	Finite Element
FV	Finite Volume
GTS	Global Time Stepping
HLLC	Harten-Lax-van Leer Contact wave
(H)WENO	(Hermite) Weighted Essentially Non-Oscillatory
IMEX	IMplicit EXplicit
LDG	Local Discontinuous Galerkin
LEE	Linearized Euler Equations
LES	Large Eddy Simulation
LGL	Legendre-Gauss-Lobatto
LTS	Local Time Stepping
MHD	Magneto-HydroDynamics
MOL	Method Of Line
MPI	Message Passing Interface
ODE	Ordinary Differential Equation
PDE	Partial Differential Equation
RK	Runge-Kutta
RKDG	Runge-Kutta Discontinuous Galerkin
SIP	Symmetric Interior Penalty
SVD	Singular Value Decomposition
VMS	Variational Multi-Scale

## Kurzfassung

In dieser Arbeit wird ein neues explizites unstetiges Finite Element Verfahren von *beliebig* hoher Genauigkeit für die Lösung der zeitabhängigen kompressiblen Navier-Stokes Gleichungen entwickelt. Obwohl der Fokus dieser Arbeit auf dem Lösen der kompressiblen Navier-Stokes Gleichung liegt, kann man die entwickelte Methode auch direkt für die Lösung von anderen rein hyperbolischen, rein parabolischen oder gemischt hyperbolisch/parabolischen zeitabhängigen Erhaltungsgleichungen verwenden.

Die unstetige Finite Elemente Methode zeichnet sich durch eine Reihe von wichtigen Eigenschaften aus. So ist sie, obwohl sie zur Klasse der Finite Elemente Verfahren gehört, *lokal* konservativ. Zudem ist es möglich, allgemeine unstrukturierte Netze mit stark verzerrten anisotropen Gitterzellen zu verwenden, ohne die formale Genauigkeit zu verlieren. Die formale Genauigkeitsordnung des Verfahrens lässt sich beliebig einstellen, da man im Prinzip nur den Polynomgrad ändern muss. Die wichtigste Eigenschaft des unstetigen Finite Element Verfahrens ist, dass die Lösung einer Gitterzelle nur von den Lösungen der Nachbargitterzellen abhängt. Diese Kompaktheit erlaubt eine hohe Parallelisierbarkeit der Methode, was für große Rechnungen essentiell ist.

Basierend auf der Standard unstetigen Finite Elemente Methodik werden verschiedene Modifikationen vorgeschlagen, um die Recheneffizienz zu erhöhen. Zuerst wird die Konstruktion von neuartigen modalen und nodalen Basisfunktionen für beliebige Gitterzellen eingeführt. Für die Navier-Stokes Gleichungen wird eine neuartige schwache Formulierung entwickelt. Die neuartige Approximation der viskosen Flüsse basiert auf einer lokalen Lösung des Riemannproblems. Diese räumliche Diskretisierung wird zudem mit einer neuen Zeitdiskretisierung kombiniert, welche zeitgenaue lokale Zeitschritte erlaubt.

Für die Validierung des entwickelten Verfahrens werden verschiedene Testfälle betrachtet, mit Hilfe derer die hohe Genauigkeit und Effizienz des Verfahrens gezeigt wird. Dieses Verfahren wird dann zur Lösung von zwei- und drei-dimensionalen kompressiblen Strömungsproblemen verwendet.

## Abstract

In this work a new explicit arbitrary high order accurate discontinuous Galerkin finite element solver for the *unsteady compressible Navier-Stokes equations* is developed. Although the focus is on the compressible Navier-Stokes equations, the developed framework can directly be applied to other pure hyperbolic, pure parabolic or mixed hyperbolic/parabolic time dependent conservation laws.

Discontinuous Galerkin finite element based methods have several important properties. They are locally conservative schemes, despite their affiliation to the class of finite element methods. They allow to use arbitrary unstructured non-conforming meshes, while remaining their formal (high) order of accuracy, even with skewed and anisotropic grid cells. The resolution can be adapted locally by increasing or decreasing the local polynomial degree, without the difficulties of a conforming finite element approach. The (formal) order of accuracy is essential a parameter, as one only has to choose the polynomial degree of the approximation. The most important property of discontinuous Galerkin schemes is that the solution in a grid cell depends only on data from grid cells sharing a face, independent of the approximation order. This compactness yields an excellent parallelizability of the method, which is essential for large scale computations.

Based on the standard spatial discontinuous Galerkin framework several modifications to increase the computational efficiency are proposed. In a first step a novel construction guideline for *modal and nodal* basis functions on *arbitrary shaped* grid cells is introduced. For the discretization of problems with high order derivatives a novel *weak formulation* is introduced and applied to the second order compressible Navier-Stokes equations. For the approximation of the *viscous fluxes* a new approximation based on local *Riemann* solutions is used. This spatial discretization is combined with a new time discretization, which allows *time consistent local time stepping*.

In order to validate the high accuracy and efficiency of the developed method, several test cases including the linearized Euler equations, the non-linear Euler equations and the compressible Navier-Stokes equations are considered. Finally, this method is applied to solve two and three dimensional compressible unsteady flow problems.

# 1 Introduction

## 1.1 Objectives

The aim of this work is to enhance the computational efficiency of a discontinuous Galerkin based solver for the *time dependent* nonlinear *compressible* Navier-Stokes equations. Collis [22] and Collis and Ghayour [23] were the first who used a discontinuous Galerkin scheme for *the direct numerical simulation of turbulent flow*. They used hexahedral discontinuous finite elements with tensor product Legendre-Gauss-Lobatto integration, the Lax-Friedrichs approximation for the Euler fluxes [75], the first method of Bassi and Rebay [8] for the treatment of the viscous terms and a third order total variation diminishing Runge-Kutta method [36] for time advancement. They exploited their discontinuous Galerkin based discretization and used local *hp*-refinement to reduce the overall number of unknowns. Furthermore the effect of weakly imposed boundary conditions through numerical fluxes was investigated, indicating very accurate results even for slightly underresolved approximations. In [21], Collis combined the discontinuous Galerkin method with the variational multi-scale method [20, 43] creating a synergistic approach for large eddy simulations of turbulent flows. In the context of motivation for this work it is worth mentioning that this encouraging results were the starting point for this thesis with the aim to further exploit the possibilities of a discontinuous Galerkin based discretization for the unsteady compressible Navier-Stokes equations. The biggest advantage of a discontinuous Galerkin based algorithm is its extremely local character. Independent of the order of the scheme, only data from neighbors sharing a common face are needed, allowing a high efficiency for parallel computations. This high parallelizability of the discontinuous Galerkin scheme allows to effectively use modern massive parallel scalar clusters and thus gives access to raw processing power, which is desperately needed for large scale Navier-Stokes computations. If we are interested in enhancing the computational efficiency of our discontinuous Galerkin based solver, we should always keep in mind that with respect to parallelizability, no compromises can be made. Regarding the state of the art techniques used by Collis et al. it seems that there is some room left for improvement.

## 1.2 Discontinuous Galerkin for Compressible Navier-Stokes

The discontinuous Galerkin method was first proposed in the early 1970's in [69] for linear steady hyperbolic problems. Cockburn and Shu [14–17, 19] combined the spatial discontinuous Galerkin discretization with an explicit high order Runge-Kutta time discretization and generalized the method for the application to non-linear hyperbolic conservation laws, yielding a powerful computational tool for the solution of the Euler equations [9]. The parallelizability of the Runge-Kutta discontinuous Galerkin scheme was investigated by Biswas, Devine and Flaherty [12] showing a solution parallel efficiency of 99% in the NCUBE/2. The corresponding issue of dynamic load balancing was investigated by Flaherty et al. [32] and by Devine et al. [25, 26]. For problems with complex curved geometries, Bassi and Rebay showed the importance of an accurate discrete geometry representation [9]. The extension to problems of viscous gas dynamics was initiated by Bassi and Rebay in [8, 10] and this again has led to several related formulations [18, 37, 50, 53, 62] for the compressible Navier-Stokes equations. In the last decade several groups applied a spatial discontinuous Galerkin based discretization to solve the time dependent compressible Navier-Stokes equations, using either a method of line (MOL) type explicit time discretization [8, 18, 53, 65], a MOL type implicit time discretization [63], an implicit-explicit Runge-Kutta time discretization [46], a fully implicit space-time discontinuous Galerkin discretization [50] or an explicit space-time approach with time accurate local time stepping [33, 35].

## 1.3 Tensor and Differentiation Operator Notations

Throughout this thesis several tensor and differentiation operators are used. This subsection is used to define and clarify the mathematical notations.

We introduce  $d$  as the number of space dimensions and  $m$  as the number of physical variables in the state vector. Vectors of physical variables, such as the vector of conserved variables, are denoted with capital characters, scalar quantities with small characters, while all vector quantities with components in different space directions, as the velocity  $\vec{v}$ , are overset with an arrow. To indicate matrices the character is underlined. For vectors  $\vec{a} \in \mathbb{R}^d$ ,  $\vec{b} \in \mathbb{R}^d$ ,  $V \in \mathbb{R}^m$ ,  $W \in \mathbb{R}^m$ , the matrix  $\underline{D} \in \mathbb{R}^{m \times m}$  and the matrix  $\underline{\underline{G}} \in \mathbb{R}^{nm \times nm}$ , consisting of  $d^2$  blocks of matrices  $\underline{G}_{ij} \in \mathbb{R}^{m \times m}$  with  $i, j = 1, \dots, d$ , we define the following operations. The operation  $\cdot$  and  $\cdot$  denote the scalar product of the two types

of vectors

$$\begin{aligned}\mathbb{R} \ni \vec{a} \cdot \vec{b} &:= \sum_{j=1}^d \vec{a}[j] \vec{b}[j], \\ \mathbb{R} \ni V \cdot W &:= \sum_{j=1}^m V[j] W[j],\end{aligned}\tag{1.1}$$

while

$$\begin{aligned}\mathbb{R}^{d \times d} \ni \vec{a}\vec{b}, \text{ with } \vec{a}\vec{b}[i, j] &:= \vec{a}[i] \vec{b}[j], \quad i, j = 1, \dots, d, \\ \mathbb{R}^{d \times m} \ni \vec{a}V, \text{ with } \vec{a}V[i, j] &:= \vec{a}[i] V[j], \quad i = 1, \dots, d, \quad j = 1, \dots, m,\end{aligned}\tag{1.2}$$

denote dyadic products. We need the following products between the different tensors

$$\begin{aligned}\mathbb{R}^m \ni \vec{b} \cdot \vec{a}V, \text{ with } \vec{b} \cdot \vec{a}V[j] &:= \sum_{i=1}^d \vec{b}[i] \vec{a}[i] V[j], \quad j = 1, \dots, m; \\ \mathbb{R}^{d \times m} \ni \underline{\underline{G}}(\vec{a}V), \text{ with } \underline{\underline{G}}(\vec{a}V)[i, j] &:= \sum_{l=1}^m \vec{a}[l] \sum_{p=1}^m \underline{\underline{G}}_i[j, p] V[p], \\ i = 1, \dots, d, \quad j = 1, \dots, m; \\ \mathbb{R} \ni (\vec{a}V) : (\vec{b}W) &:= \sum_{i=1}^d \sum_{j=1}^m (\vec{a}V)[i, j] (\vec{b}W)[i, j].\end{aligned}\tag{1.3}$$

together with the identities

$$(\underline{\underline{D}}V) \cdot W = V \cdot (\underline{\underline{D}}^T W), \quad (\underline{\underline{G}}(\vec{a}V)) : (\vec{b}W) = (\vec{a}V) : (\underline{\underline{G}}^T(\vec{b}W)).\tag{1.4}$$

Using the nabla operator

$$\vec{\nabla} = \begin{pmatrix} \frac{\partial}{\partial x_1} \\ \vdots \\ \frac{\partial}{\partial x_d} \end{pmatrix}\tag{1.5}$$

the differential operators 'grad' and 'div' are defined by

$$\begin{aligned}\text{div}(\vec{a}V) = \vec{\nabla} \cdot \vec{a}V, \text{ with } \vec{\nabla} \cdot \vec{a}V[j] &:= \sum_{i=1}^d \frac{\partial}{\partial x_i} \vec{a}[i] V[j], \quad j = 1, \dots, m, \\ \text{grad}(V) = \vec{\nabla}V, \text{ with } \vec{\nabla}V[i, j] &:= \frac{\partial}{\partial x_i} V[j], \quad i = 1, \dots, d, \quad j = 1, \dots, m.\end{aligned}\tag{1.6}$$

Differentiation of a function with respect to a space or the time coordinate is abbreviated with a subscript of the coordinate, e.g.

$$u_{x_1} := \frac{\partial}{\partial x_1} u \quad \text{or} \quad u_t := \frac{\partial}{\partial t} u. \quad (1.7)$$

## 1.4 Governing Equations

Although we are interested in solving the compressible Navier-Stokes equations, the developed numerical method is not restricted to this particular application. The scheme is designed to solve advection-diffusion equations of a special type. The considered class of advection-diffusion problems has the following structure

$$U_t + \vec{\nabla} \cdot \vec{F}^A(U) - \vec{\nabla} \cdot \vec{F}^D(U, \vec{\nabla}U) = S, \quad (1.8)$$

where  $U \in \mathbb{R}^m$  is the vector of conservative variables,  $S \in \mathbb{R}^m$  the vector of source terms,  $\vec{F}^A = (F_1^A, \dots, F_d^A)^T \in \mathbb{R}^{d \times m}$  the advection fluxes and  $\vec{F}^D = (F_1^D, \dots, F_d^D)^T \in \mathbb{R}^{d \times m}$  the diffusion fluxes. We suppose that for smooth solutions  $U$  the advection fluxes could be rewritten as

$$\vec{\nabla} \cdot \vec{F}^A(U) = (F_1^A(U))_{x_1} + \dots + (F_d^A(U))_{x_d} = \underline{A}_1(U)U_{x_1} + \dots + \underline{A}_d(U)U_{x_d}, \quad (1.9)$$

where the matrices  $\underline{A}_i \in \mathbb{R}^{m \times m}$ ,  $i = 1, \dots, d$  have a full set of real eigenvectors and eigenvalues. We furthermore require that the diffusion fluxes are homogeneous with respect to the gradient

$$\vec{F}^D(U, \vec{\nabla}U) = \underline{\underline{D}}(U)\vec{\nabla}U, \quad (1.10)$$

where  $\underline{\underline{D}}$  is composed of  $d^2$  diffusion matrices

$$\underline{\underline{D}} := \begin{pmatrix} \underline{D}_{11} & \dots & \underline{D}_{1d} \\ \vdots & \vdots & \vdots \\ \underline{D}_{d1} & \dots & \underline{D}_{dd} \end{pmatrix}, \quad \underline{D}_{ij} \in \mathbb{R}^{m \times m}, \quad (1.11)$$

with full sets of eigenvectors and real eigenvalues *greater equal* zero, as such positive semi definite diffusion matrices guarantee physically reasonable diffusion effects.

We restrict ourselves to rotational invariant problems, which allows us to generalize one-dimensional numerical flux approximations to the multi-dimensional case. We therefore need a rotation matrix  $\underline{R} \in \mathbb{R}^{m \times m}$  which transforms the



state vector from one system into another system. If we consider  $d$  orthogonal and normalized vectors  $\vec{y}_1, \dots, \vec{y}_d$ , we can define the rotation matrix for the compressible Navier-Stokes equations as

$$\underline{R} := \begin{pmatrix} 1 & 0 & \dots & 0 & 0 \\ 0 & y_{11} & \dots & y_{1d} & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & y_{d1} & \dots & y_{dd} & 0 \\ 0 & 0 & \dots & 0 & 1 \end{pmatrix}, \quad (1.12)$$

which transforms the state vector of conservative variables  $U$  from the  $\vec{x}$ -system to the  $\vec{y}$ -system

$$\mathcal{U}(\vec{x}, t) := \underline{R}U(\vec{x}, t). \quad (1.13)$$

For the gradient of the state vector  $\vec{\nabla}U = (U_{x_1}, \dots, U_{x_d})^T$  the transformations read as

$$\mathcal{U}_{y_i} := \underline{R} \left( \vec{y}_i \cdot \vec{\nabla}U \right), \quad i = 1, \dots, d. \quad (1.14)$$

Using the definitions from above, the rotational invariance property for the advection fluxes can be expressed as

$$F_{\vec{y}_i}^A(U) := \vec{y}_i \cdot \vec{F}^A(U) = \underline{R}^{-1} F_i^A(\mathcal{U}), \quad (1.15)$$

and for the diffusion fluxes as

$$F_{\vec{y}_i}^D(U, \vec{\nabla}U) := \vec{y}_i \cdot \vec{F}^D(U, \vec{\nabla}U) = \underline{R}^{-1} F_i^D(\mathcal{U}, (\mathcal{U}_{y_1}, \dots, \mathcal{U}_{y_d})^T). \quad (1.16)$$

Some well known representatives of this class are the viscous MHD equations [81] and the compressible Navier-Stokes equations [35]. For the three dimensional compressible *Navier-Stokes* equations for Newtonian liquids the vector of conservative variables is given by  $U = (\rho, \rho v_1, \rho v_2, \rho v_3, \rho e)^T$ , the Euler fluxes by

$$F_l^A(U) = \begin{pmatrix} \rho v_l \\ \rho v_1 v_l + \delta_{1l} p \\ \rho v_2 v_l + \delta_{2l} p \\ \rho v_3 v_l + \delta_{3l} p \\ \rho e v_l + p v_l \end{pmatrix}, \quad l = 1, 2, 3, \quad (1.17)$$

and the diffusion fluxes by

$$F_l^D(U, \vec{\nabla}U) = \begin{pmatrix} 0 \\ \tau_{1l} \\ \tau_{2l} \\ \tau_{3l} \\ \tau_{lj}v_j - q_l \end{pmatrix}, \quad l = 1, 2, 3. \quad (1.18)$$

The viscous stress tensor depends on the symmetric part of the velocity gradient

$$\underline{\tau} := \mu(\vec{\nabla}\vec{v} + (\vec{\nabla}\vec{v})^T - \frac{2}{3}(\vec{\nabla} \cdot \vec{v})\underline{I}), \quad (1.19)$$

and the heat flux  $\vec{q} = (q_1, q_2, q_3)^T$  is proportional to the temperature gradient

$$\vec{q} := -k\vec{\nabla}T, \quad \text{with} \quad k = \frac{c_p\mu}{Pr}, \quad (1.20)$$

where we close the system with the equation of state of a perfect gas

$$p = \rho RT = (\kappa - 1)\rho\left(e - \frac{1}{2}\vec{v} \cdot \vec{v}\right), \quad \text{and} \quad e = \frac{1}{2}\vec{v} \cdot \vec{v} + c_v T. \quad (1.21)$$

The adiabatic exponent  $\kappa = \frac{c_p}{c_v}$  with the specific heats  $c_p, c_v$ , the specific gas constant  $R = c_p - c_v$ , the viscosity  $\mu$  and the Prandtl number  $Pr$  depend on the fluid. If we consider *air*, the temperature dependence of viscosity  $\mu$  is modeled using Sutherland's law [72]

$$\mu(T) = \mu(T_\infty) T^{3/2} \frac{1 + T_s}{T + T_s}, \quad (1.22)$$

with  $T_\infty = 280 \text{ K}$ ,  $\mu(T_\infty) = 1.735 \cdot 10^{-5} \frac{\text{kg}}{\text{ms}}$  and  $T_s = 110.4 \text{ K}$ .

The compressible Navier-Stokes equations are conservative, nonlinear and of mixed hyperbolic/parabolic type. Whereas the mass conservation is purely hyperbolic the conservation of momentum and energy is parabolic. If we neglect viscous and heat conduction effects ( $\mu = 0$ ) the diffusion fluxes  $\vec{F}^D$  vanish, reducing the Navier-Stokes equations to the famous pure hyperbolic *Euler* equations.

## 1.5 Outline

The remaining part of the thesis is structured in three chapters. Chapter 2 addresses the numerics. Starting with high order accurate representations of data on general unstructured grids in section 2.1 we discuss in section 2.2 how to define the approximative solution. The efficient implementation of the discontinuous Galerkin formulation is discussed in 2.3. For the approximation of the surface integrals suitable approximate Riemann solvers are used for both the advection and diffusion fluxes, section 2.4. In section 2.5 a new time integration is introduced yielding a compact explicit space-time discretization, which we reformulate into a predictor-corrector form. Chapter 3 is split into two sections. In the first section the developed scheme is validated and tested. In the second section the scheme is applied to solve two and three dimensional compressible unsteady flow problems. Future prospects are discussed in the last chapter of this thesis.



## 2 Numerics

### 2.1 Polynomial Representation of Data on Unstructured Grids

When constructing a discretization for partial differential equations, the first consideration is to determine the representation of the solution data in the domain of interest  $\Omega$ . In a discontinuous Galerkin discretization, the data representation splits in two steps – the computational grid and the polynomial representation inside a grid cell. In the first step the domain  $\Omega \subset \mathbb{R}^d$  is discretized with non-overlapping grid cells. The shape of the grid cells are nearly arbitrary, whereas grid generators usually use triangles and quadrilaterals for  $d = 2$  and tetrahedra, hexahedra, pentahedra (prisms) and pyramids for  $d = 3$ . In the second step, the polynomial representation inside the grid cell has to be determined. Mathematically a polynomial representation is unique and thus it seems that this second step is not important. However considering the numerical implementation of a polynomial based discretization, the proper choice of the representation(s) is essential, as it determines efficiency, universality and accuracy. Summing up, the approximative discontinuous Galerkin solution is represented as piecewise polynomial data – continuous inside a grid cell, but discontinuous across grid cell interfaces.

#### 2.1.1 State of the Art

For an extensive state of the art overview we refer to the books of Karniadakis and Sherwin [47] and Hesthaven and Warburton [41]. Whereas the first book mainly deals with modal polynomial expansions the latter focuses on nodal representations. Starting from the one dimensional case, a natural choice seems to be the orthogonal Legendre polynomials for a modal representation and Lagrange polynomials based on Legendre-Gauss-Lobatto interpolation points for the nodal representation. For triangles/tetrahedra and quadrilaterals/hexahedra grid cell shapes a tensor product strategy of one dimensional Jacobian polynomials is used for the definition of a orthogonal modal basis function set. For nodal basis functions, tensor products of one dimensional Lagrange polynomials are used for quadrilaterals/hexahedra and multivariate

Lagrange basis functions based on optimized interpolation points are used for triangles/tetrahedra, yielding a nearly optimal Lebesgue constant. A straight forward extension of these sets of basis functions to general grid cell shapes is not known.

## 2.1.2 Polymorphic Nodal Elements

### 2.1.2.1 Modal and Nodal basis functions

We will first focus on defining different basis function sets for a given grid cell  $Q \subset \mathbb{R}^d$ . We introduce the *monomial* basis  $\{\pi_i\}_{i=1,\dots,n_M}$  for the space of polynomials with degree less or equal than  $p$ , where every basis function  $\pi_i$  can be written as

$$\pi_{i(j_1,\dots,j_d)}(\vec{x}) = x_1^{j_1} \cdot \dots \cdot x_d^{j_d} \text{ with } 0 \leq j_1 + \dots + j_d \leq p, \quad (2.1)$$

where  $i(j_1, \dots, j_d)$  is a hierarchical unique mapping of the exponents  $j_1, \dots, j_d$ . The dimension  $n_M$  of this space depends on the degree  $p$  and on the spatial dimension  $d$  of the grid cell  $Q$  and is given by

$$n_M = n_M(p, d) = \frac{(p+d)!}{d!p!}. \quad (2.2)$$

Based on the monomial basis  $\{\pi_i\}_{i=1,\dots,n_M}$  and the geometry of the grid cell  $Q$  the construction of an orthonormal basis  $\{\varphi_i\}_{i=1,\dots,n_M}$  using Gram-Schmidt orthogonalization is straight forward. This basis set is characterized by the property

$$\int_Q \varphi_i(\vec{x}) \varphi_j(\vec{x}) d\vec{x} = \delta_{ij}, \quad (2.3)$$

independent of the grid cell shape. For our discontinuous Galerkin scheme we choose such a modal orthonormal basis sets  $\{\varphi_i\}$  as a basis for the approximation as well as for the test functions. We will show later that for an efficient method, the use of a nodal representation is favorable. With the aid of the *modal* basis we may define an additional set of *nodal* basis functions. Given a set of interpolation points  $\{\xi_j\}_{j=1,\dots,n_L} \subset Q$ , we can construct the multivariate Lagrange basis  $\{\psi_j\}_{j=1,\dots,n_L}$  defined by the conditions

$$\begin{aligned} \psi_j(\vec{\xi}_i) &= \delta_{ij}, \\ u(\vec{x}) &:= \sum_{j=1}^{n_M} \hat{u}_j \varphi_j(\vec{x}) \stackrel{!}{=} \sum_{i=1}^{n_L} \tilde{u}_i \psi_i(\vec{x}). \end{aligned} \quad (2.4)$$

Combining these conditions yields the transformations

$$\mathbf{V} \hat{\mathbf{u}} = \tilde{\mathbf{u}} \text{ and } \mathbf{V}^T \boldsymbol{\psi} = \boldsymbol{\phi}, \quad (2.5)$$

where  $\hat{\mathbf{u}} \in \mathbb{R}^{n_M}$ ,  $\tilde{\mathbf{u}} \in \mathbb{R}^{n_L}$  are the vectors of modal/nodal coefficients and  $\boldsymbol{\phi} \in \mathbb{R}^{n_M}$ ,  $\boldsymbol{\psi} \in \mathbb{R}^{n_L}$  are the vectors of modal/nodal basis functions. We furthermore introduced the generalized Vandermonde matrix  $\mathbf{V}$  with entries

$$\mathbf{V}_{ij} = \varphi_j(\vec{\xi}_i), \quad i = 1, \dots, n_L; j = 1, \dots, n_M. \quad (2.6)$$

The inverse of the Vandermonde matrix is not uniquely defined in the case  $n_L \neq n_M$ . If one is interested in avoiding this problem, one has to extend the modal basis from dimension  $n_M$  to dimension  $n_L$ . We refer to Lörcher and Munz [57] for a strategy to find a basis extension for non-tensor product interpolation on a cartesian grid. However, the extension of this approach to the general case is not straightforward, as the non-singularity of the Vandermonde matrix is not guaranteed. To overcome this issue a singular value decomposition (SVD) based strategy is used to define the following pseudo-inverse transformations

$$\hat{\mathbf{u}} = \mathbf{V}^{-1} \tilde{\mathbf{u}} \text{ and } \boldsymbol{\psi} = \mathbf{V}^{-T} \boldsymbol{\phi}. \quad (2.7)$$

Using the pseudo-inverse Vandermonde matrix, condition (2.4) is only satisfied in the *least squares* sense. Thus, if we define the polynomial approximation of a function  $f$  as

$$f(\vec{x}) \approx f_L(\vec{x}) := \sum_{j=1}^{n_L} f(\vec{\xi}_j) \psi_j(\vec{x}) =: \boldsymbol{\psi}^T \tilde{\mathbf{f}}, \quad (2.8)$$

the nodal degree of freedom  $\tilde{f}_j = f(\vec{\xi}_j)$  is not the value of the interpolation  $f_L(\vec{x})$  at the node  $\vec{x} = \vec{\xi}_j$ , as  $\psi_j(\vec{\xi}_i) \neq \delta_{ij}$ . Furthermore, the modal approximation

$$f(\vec{x}) \approx f_M(\vec{x}) := \sum_{j=1}^{n_M} \hat{f}_j \phi_j(\vec{x}) \text{ with } \hat{\mathbf{f}} = \mathbf{V}^{-1} \tilde{\mathbf{f}} \quad (2.9)$$

is in the general case not equal to the nodal approximation

$$f_L(\vec{x}) \neq f_M(\vec{x}). \quad (2.10)$$

In the special case, where the function  $f$  belongs to the space of (complete order) polynomials (2.1), the nodal approximation (2.8) is identical

$$f(\vec{x}) = f_L(\vec{x}), \quad (2.11)$$

and consequently using (2.9), the modal approximations satisfies the uniqueness condition

$$f_M(\vec{x}) = f_L(\vec{x}) (= f(\vec{x})) \quad \forall \vec{x} \in Q. \quad (2.12)$$

Summarizing the impact of the pseudo-inverse, if we start with a nodal polynomial and convert it into a modal polynomial using (2.7), the two polynomials are in general not equal, as the nodal space is 'richer' than the modal space. However, if we start with a modal polynomial and use (2.5) to define the nodal interpolation, both polynomials are equal. A good measure of the quality of such an approximation is given by the Lebesgue constant  $\Lambda$ , defined as

$$\Lambda := \max_{\vec{x} \in Q} \sum_{j=1}^{n_L} |\psi_j(\vec{x})|. \quad (2.13)$$

With this definition one easily realizes that

$$\|f - f_L\|_\infty = \|f - f^* + f^* - f_L\|_\infty \leq \|f - f^*\|_\infty + \|f^* - f_L\|_\infty \leq (1 + \Lambda) \|f - f^*\|_\infty, \quad (2.14)$$

where  $\|\cdot\|_\infty$  is the usual maximum norm and  $f^*$  is the best approximating polynomial of  $f$ . As the nodal basis  $\{\psi_j\}_{j=1, \dots, n_L}$  depends only on the interpolation points  $\{\vec{\xi}_i\}_{i=1, \dots, n_L}$ , we next focus on the construction of nodal sets for different grid cell shapes which minimize the growth of the Lebesgue constant with respect to the polynomial degree.

### 2.1.2.2 Node distributions

For an interpolation with polynomials of degree  $p_L$ , we demand the following characteristics from the sets of interpolation points  $\mathcal{P}_L := \{\vec{\xi}_i\}_{i=1, \dots, n_L}$

- the points support a  $p_L$ -th order interpolation in the volume and on the surface of the grid cell, which guarantees that the basis separates into boundary and interior components,
- the distribution of the points reflects the possible symmetries of the grid cell,
- the size of the nodal set  $n_L \geq n_M$  depends on the interpolation order  $p_L$ , the dimension  $d$  and the *shape* of the grid cell.



**One-dimensional Case** For an interval,  $p_L + 1$  points have to be chosen. There may be a number of different distributions of the  $p_L + 1$  points with the restriction that the endpoints are included. For instance, one can choose equidistant (E) points, Tschebycheff-Gauss-Lobatto points or Legendre-Gauss-Lobatto (LGL) points. We choose for every side in 2D and edge in 3D the LGL node distribution, as its corresponding one dimensional Lagrange functions are known for a good Lebesgue constant  $\Lambda$ . An extended discussion of the one-dimensional case can be found in [38]. Based on the LGL node distribution we define the following warp function for  $x \in [0; 1]$

$$w_{p_L}(x) = \sum_{j=1}^{p_L+1} \xi_j^{LGL} \psi_j^E(x) \quad (2.15)$$

where  $\xi_j^{LGL}$  are the Legendre-Gauss-Lobatto points and  $\psi_j^E$  the Lagrange polynomials based on the equidistant points  $\{\xi_j^E\}_{j=1, \dots, p_L+1}$ . According to Warburton [80],  $w_{p_L}(x)$  is a  $(p_L + 1)$ -th order approximation to a function mapping 'bad' points (E) to 'good' points (LGL).

**Two-dimensional Case** In two space dimensions we split the set of interpolation points  $\mathcal{P}_L(p_L)$  into two parts: The set of points that live in the interior of the cell and the set of points that live on the surface, named  $\mathcal{P}_L^S(p_L)$ . The set  $\mathcal{P}_L^S(p_L)$  is defined such that it contains  $p_L + 1$  LGL points for each side of the grid cell surface. This guarantees that the nodal approximation on the whole surface is of order  $p_L + 1$  and a separated basis by polynomial uniqueness. We note that when using only these surface points for the approximation within the volume, the corresponding Vandermonde matrix is non-singular for  $p_L$  up to a value  $p^*$ , which depends on the shape of the grid cell. The value for  $p^*$  is 2 and 3 for triangles and quadrilaterals, respectively. Hence, for an interpolation with  $p_L > p^*$ , additional points in the interior of the grid cell are needed. The definition of these interpolation points can be done in the following recursive way

$$\mathcal{P}_L(p_L) := \begin{cases} \emptyset & \text{for } p_L < 0, \\ \{\vec{x}_{\text{barycenter}}\} & \text{for } p_L = 0, \\ \mathcal{M}_r(\mathcal{P}_L^S(p_L)) \cup (\mathcal{P}_L(p_L - (p^* + 1) + \ell_{2D})) & \text{for } p_L > 0. \end{cases} \quad (2.16)$$

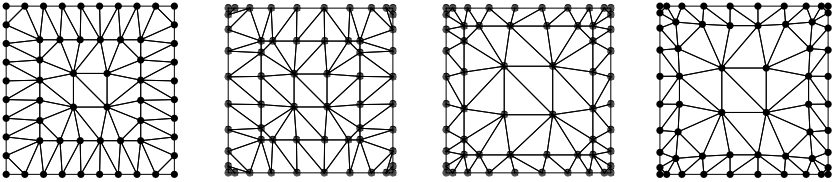
We notice that the interior nodes consist of nested and 'shrunk' surface points. The mapping  $\mathcal{M}_r$  determines how the point sets are nested and shrunk for every recursion step  $r$ , e.g., the mapping for the first recursion  $r = 1$  is the identity,

as the first points of the set  $\mathcal{P}_L^S(p_L)$  are lying on the real surface of the grid cell and thus will not be shrunk. A simple approach for the mappings  $\mathcal{M}_r$  for  $r > 1$  would be one which yields an equidistant nesting. However, it is well known that the Lebesgue constant of the corresponding nodal basis is improved when the node distribution is more dense close to the boundary of the grid cell. Thus, to improve the nodal set we propose to use a mapping which yields LGL-type nesting. In this work the warp function (2.15) is used to define the following mapping

$$\begin{aligned} \mathcal{M}_r(\vec{\xi}) &= (\vec{\xi} - \vec{x}_{bary})\alpha(r) + \vec{x}_{bary}, \\ \alpha(r) &= 1 - 2w_{p_L(r=0)}\left(\frac{r}{\max(1, \tilde{r} - 1)}\right) \end{aligned} \quad (2.17)$$

where  $\vec{\xi} \in \Omega_L^S(p_L)$ ,  $\vec{x}_{bary}$  denotes the barycenter of the grid cell and  $r$  the recursion level.  $\tilde{r}$  is two times the maximum number of recursions:  $\tilde{r} = 2r_{max}$ . We subtract one from  $\tilde{r}$ , if the innermost interpolation point set consists only of the grid cell barycenter:  $\tilde{r} = 2r_{max} - 1$ . Starting from this node distributions, it is also possible to further optimize the nodal sets with electrostatic considerations, as proposed by Hesthaven [38].

To illustrate these different strategies, we plot the corresponding node distributions of the  $p_L = 9$  ( $\ell_{2D} = 0$ ) quadrilateral in Figure 2.1. The set with a purely equidistant distribution yields a Lebesgue constant of  $\Lambda = 97$ , whereas the LGL points with equidistant nesting yields  $\Lambda = 44$ . Using LGL points and LGL-type nesting yields a Lebesgue constant of  $\Lambda = 21$ , which is slightly greater than  $\Lambda = 17$  for the electrostatic optimized points. Although the electrostatic optimized interpolation points yield the best Lebesgue constant, we use the LGL points with LGL-type nesting, as these point sets are simple and straight forward to implement for all grid cell shapes and could be computed in an initial phase of the simulation. An important parameter in the recursion formula (2.16) is the integer  $\ell_{2D}$  which can be used to tune the relation between the interpolation quality and number of points. For  $\ell_{2D} = 0$ , as considered up to now, algorithm (2.16) yields the smallest possible number of points and thus, the most efficient scheme according to the computational effort. However, we observed that in some cases, especially for quadrilaterals, the use of a few more points pays off in terms of a dramatically improved accuracy. The parameter  $\ell_{2D}$  with  $0 \leq \ell_{2D} \leq p^*$  can be used to control the number of recursions in (2.16). For this type of interpolation points with  $\ell_{2D} = 0$  for triangles and  $\ell_{2D} \in \{0, 1\}$  for quadrilaterals the corresponding Lebesgue constants  $\Lambda$  are listed in Table A.1.



**Figure 2.1:** Quadrilateral with  $p_L = 9$  ( $\ell_{2D} = 0$ ). From left to right: pure equidistant distribution, LGL points with equidistant nesting, LGL points with LGL-type nesting and optimized points.

**Three-dimensional Case** The definition of the three-dimensional set of interpolation points is done analogously to that of the two-dimensional case. Again, the set  $\mathcal{P}_L(p_L)$  is split into two parts, where  $\mathcal{P}_L^S(p_L)$  denotes the set of points on the surface. The recursion algorithm reads as follows

$$\mathcal{P}_L(p_L) := \begin{cases} \emptyset & \text{for } p_L < 0, \\ \{\vec{x}_{\text{barycenter}}\} & \text{for } p_L = 0, \\ \mathcal{M}_r(\mathcal{P}_L^S(p_L, \ell_{2D})) \cup \mathcal{P}_L(p_L - (p^* + 1) + \ell_{3D}) & \text{for } p_L > 0. \end{cases} \quad (2.18)$$

In this work the 3D *standard* shapes, namely tetrahedra, hexahedra, pentahedra (prisms) and pyramids are considered. The surfaces of this *standard* grid cells consist of triangles and quadrilaterals. Thus, for the definition of the surface point set  $\mathcal{P}_L^S(p_L, \ell_{2D})$  we can use the previously defined two-dimensional nodal points. Again, using surface points only yields non-singular interpolation up to a polynomial degree  $0 < p_L \leq p^*$ . The value of  $p^*$  is 3 for the tetrahedron, 5 for the hexahedron and 4 for the pentahedron and pyramid, respectively. We note that these values are independent of the choice of the parameter  $\ell_{2D}$ . Although the number of surface points increases with greater  $\ell_{2D}$ , the rank of the volume interpolation does not. We thus use the recursive nesting strategy (2.18) and introduce an additional parameter  $\ell_{3D}$ , which controls the number of recursions. The mapping  $\mathcal{M}_r$  (2.17) is again used to shrink the new nested surface points in a LGL-type manner. For this type of interpolation points the corresponding Lebesgue constants  $\Lambda$  are listed in Tables A.2 and A.3 for reasonable parameters  $\ell_{2D}$  and  $\ell_{3D}$ .

### 2.1.3 Discussion

As an alternative to the approach described above, we can use *monomial* basis functions expanded in the barycenter of the grid cell for the modal representation and multivariate Lagrange interpolation based on *equidistant* distributed interpolation points for the nodal representation. Such polynomial representations can be constructed for arbitrary grid cell shapes. Unfortunately, it was shown, e.g. [41], [39], that a method based on this polynomial representations suffers from being ill-conditioned, especially when using high approximation orders, resulting in an inaccurate and even unstable discretization. Concluding, the main advantages of the new approach are

- stable and accurate modal and nodal basis functions for high order polynomial representations,
- minimum number of modal basis functions and near optimal number of nodal basis functions,
- generally applicable for arbitrary grid cell shapes.

## 2.2 Discontinuous Galerkin Formulations

After defining the representation of the approximative solution, the next step in constructing a discretization is to determine in which sense the approximate solution satisfies the partial differential equations. There are a lot of different possibilities, for instance one can choose a finite difference like method [52], a finite volume like method [71], [29], a finite element like method [8], [14] or a combination [27]. We choose in this work the discontinuous Galerkin finite element framework.

A discretization based on high order finite elements has the major drawback that it is rather difficult to implement with respect to coding. However, if one accepts the additional coding effort, the advantages offered by the discontinuous Galerkin method are convincing and unique. The *discontinuous* approximation space guarantees that the mass matrix of the finite element formulation is local for every grid cell, in contrast to the classic continuous finite element approach, where the mass matrix is globally coupled. This and the fact that the weak formulation of the grid cell depends only on data from its von Neumann neighbors gives the possibility to naturally introduce an explicit time integration scheme yielding a highly efficient discretization method for massive parallel large scale simulations [54]. Additionally, due to the *integration* based framework, we are able to introduce *non-conforming* approximation spaces, i.e., hanging nodes and different local polynomial degrees. This flexibility of the approximation spaces could be used for local *hp*-adaptation.

In the next section a review of the state of the art discontinuous Galerkin formulations for advection-diffusion problems is given.

### 2.2.1 State of the Art

To demonstrate the DG approach we restrict ourselves to a scalar non-linear conservation law

$$u_t^e + \vec{\nabla} \cdot \vec{f}(u^e) = 0, \quad (2.19)$$

where  $u^e$  is the *exact* solution and  $\vec{f}(u^e)$  the vector of fluxes. The base of the DG method is a local weak formulation of the problem (2.19). We therefore multiply equation (2.19) with a test function  $\phi(\vec{x})$  and integrate over the grid cell  $Q$

$$\int_Q \left( u_t^e + \vec{\nabla} \cdot \vec{f}(u^e) \right) \phi \, d\vec{x} = 0. \quad (2.20)$$

To derive the **weak** formulation of the problem (2.19), we proceed with a spatial integration by parts of the flux term

$$\int_Q u_t^e \phi \, d\vec{x} = - \int_{\partial Q} (\vec{f}(u^e) \cdot \vec{n}) \phi \, ds + \int_Q \vec{f}(u^e) \cdot \vec{\nabla} \phi \, d\vec{x}. \quad (2.21)$$

For the weak discontinuous Galerkin formulation we first introduce the following Ansatz for an approximate solution in a grid cell  $Q$

$$u^e(\vec{x}, t)|_Q \approx u(\vec{x}, t) = \sum_{i=1}^{n_M} \hat{u}_i^Q(t) \varphi_i(\vec{x}), \quad (2.22)$$

where we choose the piecewise polynomial approximation space with the modal basis functions  $\{\varphi_i\}_{i=1, \dots, n_M}$  from section 2.1. As the approximate solution is discontinuous at the grid cell interface  $\partial Q$ , the normal component of the flux  $\vec{f}(u) \cdot \vec{n}(\vec{x})$  is not uniquely defined and is consequently replaced by a numerical approximation  $\vec{f}(u) \cdot \vec{n}(\vec{x}) \approx g = g(u^-, u^+)$ , which depends on the value from the adjacent grid cell  $u^+$ , and on the value from the considered grid cell  $u^-$ . The numerical flux function, abbreviated with  $g(u^\pm)$ , determines the global stability and consistency of the DG method and depends on the problem to be solved. The proper choice of the numerical flux approximations is discussed in chapter 2.4. We assume now that an appropriate numerical flux is available and end up with the *weak* discontinuous Galerkin formulation

$$\int_Q u_t \phi \, d\vec{x} = - \int_{\partial Q} g(u^\pm) \phi^- \, ds + \int_Q \vec{f}(u) \cdot \vec{\nabla} \phi \, d\vec{x}. \quad (2.23)$$

An equivalent formulation is derived when we assume that the volume integral is determined with interior data only

$$\int_Q \vec{f}(u) \cdot \vec{\nabla} \phi \, d\vec{x} = \int_{\partial Q} (\vec{f}(u^-) \cdot \vec{n}) \phi^- \, ds - \int_Q \vec{\nabla} \cdot \vec{f}(u) \phi \, d\vec{x} \quad (2.24)$$

yielding the *strong* discontinuous Galerkin formulation

$$\int_Q (u_t + \vec{\nabla} \cdot \vec{f}(u)) \phi \, d\vec{x} = - \int_{\partial Q} (g(u^\pm) - (\vec{f}(u^-) \cdot \vec{n})) \phi^- \, ds. \quad (2.25)$$

For every grid cell  $Q$ ,  $n_M$  time dependent polynomial coefficients  $\hat{u}_i^Q(t)$  are unknown. If we choose  $\phi = \varphi_j$  for  $j = 1, \dots, n_M$  a system of ordinary differential

equations (ODE) in time results for every grid cell  $Q$ . As we focus in this section on the spatial discontinuous Galerkin formulation, we shift the discussion about the time discretization to section 2.5.

If we use the same approach for diffusive problems, where the flux  $\vec{f} = \vec{f}(u, \vec{\nabla}u)$  additionally depends on the gradient of the solution, the convergence order of the resulting discontinuous Galerkin discretization is non-optimal, causing the common opinion that discontinuous Galerkin for diffusion is not straightforward. To overcome this problem, Bassi and Rebay [8, 10] combined the DG framework with a mixed finite element approach, where they rewrote the second order problem into a first order system. To demonstrate the method of Bassi and Rebay, we assume now, that the flux vector depends not only on the solution but on the gradient of the solution as well

$$u_t^e + \vec{\nabla} \cdot \vec{f}(u^e, \vec{\nabla}u^e) = 0. \quad (2.26)$$

The idea is to introduce the auxiliary variables  $\vec{q}^e = \vec{\nabla}u^e$  and to rewrite (2.26) into

$$\begin{aligned} u_t^e &= \vec{\nabla} \cdot \vec{f}(u^e, \vec{q}^e), \\ \vec{q}^e &= \vec{\nabla}u^e. \end{aligned} \quad (2.27)$$

For this 'new' problem the usual discontinuous Galerkin approach as described above can be used. We introduce a piecewise polynomial approximation  $(u, \vec{q})$  for the unknowns  $(u^e, \vec{q}^e)$  and multiply (2.27) with test functions. After integration by parts we get

$$\begin{aligned} \int_Q u_t \phi \, d\vec{x} &= \int_{\partial Q} g(u^\pm, \vec{q}^\pm) \phi^- \, ds - \int_Q \vec{f} \cdot \vec{\nabla} \phi \, d\vec{x}, \\ \int_Q \vec{q} \phi \, d\vec{x} &= \int_{\partial Q} \tilde{u} \vec{n} \phi \, ds - \int_Q u \vec{\nabla} \phi \, d\vec{x}, \end{aligned} \quad (2.28)$$

where we introduced the numerical approximation of the flux normal component  $g(u^\pm, \vec{q}^\pm)$  and an approximation of the solution trace  $\tilde{u}$ . We note that not only the trace of the flux is necessary, but also the trace of the solution itself, which corresponds well to the nature of the continuity conditions of a second order problem. The discontinuous mixed finite element approach of Bassi and Rebay was furthermore applied to higher order problems by Yan and Shu [83, 84], showing the generality of this approach. However, the major drawback of the method is that for a system with  $m$  unknowns in  $d$  dimensions  $m \times d$  auxiliary variables (the gradients of the conservative variables) are needed.

### 2.2.2 The Ultra Weak Formulation

We derive now a novel discontinuous Galerkin formulation for advection diffusion systems. The base of the formulation is a straightforward extension of the weak formulation for first order problems (2.21) to problems with second order terms, which we name as 'ultra' weak formulation. To derive this formulation we first multiply the advection-diffusion equations (1.8) with a test function  $\Phi = \Phi(\vec{x}) \in \mathbb{R}^m$  and integrate over an arbitrary cell  $Q$ :

$$\int_Q \left( U_t^e + \vec{\nabla} \cdot \left( \vec{F}^A(U^e) - \vec{F}^D(U^e, \vec{\nabla} U^e) \right) \right) \cdot \Phi \, d\vec{x} = 0. \quad (2.29)$$

As usual we apply spatial integration by parts to the flux terms and obtain

$$\int_Q U_t^e \cdot \Phi \, d\vec{x} + \int_{\partial Q} \left( F_{\vec{n}}^A - F_{\vec{n}}^D \right) \cdot \Phi \, ds - \int_Q \left( \vec{F}^A - \vec{F}^D \right) : (\vec{\nabla} \Phi) \, d\vec{x} = 0, \quad (2.30)$$

where

$$F_{\vec{n}}^A := \vec{n} \cdot \vec{F}^A, \quad F_{\vec{n}}^D := \vec{n} \cdot \vec{F}^D \quad (2.31)$$

denote the fluxes into the direction  $\vec{n}$  normal to the edge. We take now into account that the dissipation terms contain second order derivatives, i.e., that the second volume integral in formulation (2.30) still depends on the gradient of the solution, as  $\vec{F}^D = \vec{F}^D(U^e, \vec{\nabla} U^e)$ . Thus we have another volume integral with derivatives and we may proceed by applying a **second** integration by parts, using the homogeneity property (1.10) of the viscous flux with respect to the gradient

$$\int_Q \vec{F}^D : (\vec{\nabla} \Phi) \, d\vec{x} = \int_Q (\underline{\underline{D}}(U^e) \vec{\nabla} U^e) : (\vec{\nabla} \Phi) \, d\vec{x} = \int_Q (\vec{\nabla} U^e) : (\underline{\underline{D}}(U^e)^T (\vec{\nabla} \Phi)) \, d\vec{x}, \quad (2.32)$$

yielding

$$\int_Q \vec{F}^D : (\vec{\nabla} \Phi) \, d\vec{x} = \int_{\partial Q} U^e \cdot F_{\vec{n}}^{D'}(U^e, \vec{\nabla} \Phi) \, ds - \int_Q U^e \cdot (\vec{\nabla} \cdot \vec{F}^{D'}(U^e, \vec{\nabla} \Phi)) \, d\vec{x}, \quad (2.33)$$



with the definition of the *adjoint* fluxes

$$\begin{aligned}
 \vec{F}^{D'}(A, \vec{\nabla}B) &:= \underline{\underline{D}}^T(A) \vec{\nabla}B, \\
 \vec{F}^{D'}[l](A, \vec{\nabla}B) &= \sum_{j=1}^n \underline{\underline{D}}_{jl}^T(A) B_{x_j}, \quad l = 1, \dots, n \\
 F_{\vec{n}}^{D'}(A, \vec{\nabla}B) &:= \vec{n} \cdot \vec{F}^{D'}(A, \vec{\nabla}B).
 \end{aligned} \tag{2.34}$$

We note that due to the second integration by part, an additional surface integral appears. This new term depends on the trace of the solution and closely corresponds to the additional surface term of the mixed discontinuous finite element formulation of Bassi and Rebay, (2.28). Inserting this into the equations (2.30) yields the **ultra weak** formulation of problem (1.8)

$$\begin{aligned}
 \int_Q U_t^e \cdot \Phi \, d\vec{x} &= - \int_{\partial Q} \left( F_{\vec{n}}^A(U^e) - F_{\vec{n}}^D(U^e, \vec{\nabla}U^e) \right) \cdot \Phi + U^e \cdot F_{\vec{n}}^{D'}(U^e, \vec{\nabla}\Phi) \, ds \\
 &+ \int_Q \vec{F}^A(U^e) : \vec{\nabla}\Phi + U^e \cdot (\vec{\nabla} \cdot \vec{F}^{D'}(U^e, \vec{\nabla}\Phi)) \, d\vec{x}.
 \end{aligned} \tag{2.35}$$

For our discontinuous Galerkin formulation we introduce again a piecewise polynomial approximation for every component of the solution vector  $U^e|_Q \approx U$ , yielding the *ultra weak* discontinuous Galerkin formulation

$$\begin{aligned}
 \int_Q U_t \cdot \Phi \, d\vec{x} &= - \int_{\partial Q} \left( G^A(U^\pm) - G^D(U^\pm, \vec{\nabla}U^\pm) \right) \cdot \Phi^- + \tilde{U} \cdot F_{\vec{n}}^{D'}(\tilde{U}, \vec{\nabla}\Phi^-) \, ds \\
 &+ \int_Q \vec{F}^A(U) : \vec{\nabla}\Phi + U \cdot (\vec{\nabla} \cdot \vec{F}^{D'}(U, \vec{\nabla}\Phi)) \, d\vec{x},
 \end{aligned} \tag{2.36}$$

where we introduced the numerical approximation of the normal fluxes and of the solution trace  $G^A$ ,  $G^D$  and  $\tilde{U}$ . We can again find an equivalent form of (2.36) when we assume that the volume integral depends only on interior data

$$\int_Q U \cdot (\vec{\nabla} \cdot \vec{F}^{D'}(U, \vec{\nabla}\Phi)) \, d\vec{x} = \int_{\partial Q} U^- \cdot F_{\vec{n}}^{D'}(U^-, \vec{\nabla}\Phi^-) \, ds - \int_Q \vec{F}^D : (\vec{\nabla}\Phi) \, d\vec{x} \tag{2.37}$$

yielding the *weak* discontinuous Galerkin formulation

$$\int_Q U_t \cdot \Phi \, d\vec{x} = - \int_{\partial Q} (G^A - G^D) \cdot \Phi^- + h \, ds + \int_Q (\vec{F}^A - \vec{F}^D) : (\vec{\nabla} \Phi) \, d\vec{x}, \quad (2.38)$$

with the abbreviation

$$h = h(U^\pm, \vec{\nabla} \Phi^-) := \tilde{U} \cdot F_n^{D'}(\tilde{U}, \vec{\nabla} \Phi^-) - U^- \cdot F_n^{D'}(U^-, \vec{\nabla} \Phi^-). \quad (2.39)$$

Analogous to (2.24) and (2.25) another form of (2.36) and (2.38) is the *strong* discontinuous Galerkin formulation

$$\int_Q (U_t + \vec{\nabla} \cdot (\vec{F}^A - \vec{F}^D)) \cdot \Phi \, d\vec{x} = - \int_{\partial Q} (F \cdot \Phi^- + h) \, ds, \quad (2.40)$$

with the abbreviation

$$F(U^\pm, \vec{\nabla} U^\pm) = (G^A(U^\pm) - F_n^A(U^-)) - (G^D(U^\pm, \vec{\nabla} U^\pm) - F_n^D(U^-, \vec{\nabla} U^-)). \quad (2.41)$$

We note that the strong, weak and ultra weak formulation are mathematically equivalent. Considering the numerical implementation of the weak formulation, i.e. the approximation of the surface and volume integrals, the strong formulation seems to be inferior because of the not guaranteed *exact* local conservation. Exact conservation is only achieved when the volume integral of the fluxes cancels with the surface integral of the fluxes, evaluated inside the grid cell. Especially for non-linear flux functions this property is not guaranteed. To overcome this issue a simple trick is introduced in section 2.3.3.1.

### 2.2.3 Discussion

Comparing to the discontinuous mixed finite element approach of Bassi and Rebay, the advantage of the ultra weak discontinuous Galerkin formulation is that **no auxiliary variables** are needed. The application of the ultra weak formulation to higher order problems was recently given by Cheng and Shu [13], where they used multiple integration by parts. The drawback of multiple integration by parts is that some sort of homogeneity property with respect to the high order derivatives is needed. Although this is the natural structure for second order problems [56], this property can not be guaranteed for general high order problems.

## 2.3 Efficient Numerical Approximation of the DG Formulations

When solving the discrete weak formulation derived in the previous section, we face the challenge of 'integration of non-linear fluxes over arbitrary domains'. Regarding accuracy, we are interested in an exact evaluation of the integrals. However, in practice exact evaluation is at least inefficient and often impossible. A common alternative is that the evaluation of the integrals is exact only for flux functions with constant Jacobi matrices, which guarantees that the implementation is (asymptotically) high order accurate, even in case of non-linear flux functions.

### 2.3.1 State of the Art

To keep matters simple we restrict ourselves to a scalar non-linear conservation law of the form

$$u_t^e + \vec{\nabla} \cdot \vec{f}(u^e) = 0, \quad (2.42)$$

where the weak discontinuous Galerkin formulation reads as

$$\int_Q u_t \phi \, d\vec{x} + \int_{\partial Q} g(u^\pm) \phi^- \, ds - \int_Q \vec{f}(u) \cdot \vec{\nabla} \phi \, d\vec{x} = 0. \quad (2.43)$$

We concentrate the following discussion on the first term of the volume integral

$$\int_Q f_1(u) \frac{\partial \phi}{\partial x_1}(\vec{x}) \, d\vec{x}. \quad (2.44)$$

#### 2.3.1.1 The Modal Approach

In the modal approach we expand the approximative DG solution in terms of the modal basis function as introduced in section 2.1

$$u(x, t) = \sum_{i=1}^{n_M} \hat{u}_i(t) \varphi_i(\vec{x}), \quad (2.45)$$

and choose in (2.43) a modal test function  $\phi = \varphi_i$  for  $i = 1, \dots, n_M$ . The integral is then approximated with a proper Gaussian cubature rule. We denote with  $\{\vec{\chi}_j\}_{j=1, \dots, n_G}$  the position of the Gauss points and with  $\{\omega_j\}_{j=1, \dots, n_G}$

the corresponding Gauss weights, yielding the Gaussian approximation of the volume integral (2.44) as

$$\int_Q f_1(u) \frac{\partial \varphi_i}{\partial x_1}(\vec{x}) d\vec{x} \approx \sum_{j=1}^{n_G} f_1(u(\vec{\chi}_j)) \frac{\partial \varphi_i}{\partial x_1}(\vec{\chi}_j) \omega_j, \quad i = 1, \dots, n_M. \quad (2.46)$$

With the definition of the integration matrix  $\mathcal{K}^{1,G} \in \mathbb{R}^{n_M \times n_G}$  with

$$\mathcal{K}_{ij}^{1,G} := \frac{\partial \varphi_i}{\partial x_1}(\vec{\chi}_j) \omega_j, \quad i = 1, \dots, n_M \quad j = 1, \dots, n_G, \quad (2.47)$$

the approximation could be reformulated in more compact form as

$$\sum_{j=1}^{n_G} f_1(u(\vec{\chi}_j)) \frac{\partial \varphi}{\partial x_1}(\vec{\chi}_j) \omega_j = \mathcal{K}^{1,G} \mathbf{f}, \quad (2.48)$$

where  $(\mathbf{f})_j := f_1(u(\vec{\chi}_j))$ ,  $j = 1, \dots, n_G$ . Thus, for the evaluation of the volume integral  $n_G$  flux evaluations are needed and a matrix vector multiplication of the size  $n_M \times n_G$ . The number  $n_G$  depends on the order of the approximation space. In the one dimensional case for an approximation with polynomial degree  $p$ ,  $n_G = p$  Gauss evaluations are needed. As we made the assumption that the integration rule is exact only for linear flux functions, the integration rule has to integrate a polynomial of degree  $p$  times a polynomial of degree  $p-1$  (due to the differentiation of the testfunction) exactly. In the general multidimensional case  $n_G = dp(p+1)^{d-1}$  Gauss evaluations are needed. We note that when the grid cell has curved boundaries, the number of needed Gauss points furthermore drastically increases. For special grid cell types such as straight bounded simplices, optimized cubature rules exist. Numerical experiments and investigations confirm that for high order accurate three dimensional calculations a major cost is found in this traditional approach, yielding a computational inefficient scheme compared to other high order accurate unstructured methods [52], [29]. When choosing an in-exact integration rule, additional errors due to the non-linearity occur. For strong non-linear problems such as turbulence simulations and simulations involving strong discontinuities, these errors even cause an instability, known as *aliasing instability*. To circumvent the aliasing issues, exact integration is needed. However, as mentioned above, besides being impractical costly, general non-linear flux functions can easily exceed the space of polynomials. For instance in the Euler fluxes some terms are rational functions with respect to the conservative variables, making the task of exact numerical integration at least challenging. How to control the aliasing issue is discussed in the next subsection.

### 2.3.1.2 The Nodal Approach

Recently, Hesthaven and Warburton [40] introduced the nodal DG scheme. In this formulation, the approximation  $u$  is represented using the nodal basis functions  $\{\psi_j\}_{j=1,\dots,n_L}$

$$u(\vec{x}, t) = \sum_{j=1}^{n_L} \tilde{u}_j(t) \psi_j(\vec{x}), \quad (2.49)$$

which are also used as test functions. Using the nodal basis, we get for the volume integral (2.44)

$$\int_Q f_1(u) \frac{\partial \psi}{\partial x_1}(\vec{x}) d\vec{x} \approx \mathcal{K}^{1,N} \tilde{\mathbf{f}}_1, \quad (2.50)$$

where we approximated the non-linear flux function with  $f_1(u) \approx \boldsymbol{\psi}^T \tilde{\mathbf{f}}_1$  with the nodal values of the flux defined as  $(\tilde{\mathbf{f}}_1)_j := f_1(u(\vec{\xi}_j))$ . Hence, the nodal stiffness matrix is given by

$$\begin{aligned} \mathcal{K}^{1,N} &:= \int_Q \frac{\partial \psi}{\partial x_1}(\vec{x}) \boldsymbol{\psi}^T(\vec{x}) d\vec{x} = \mathbf{V}^{-T} \int_Q \frac{\partial \phi}{\partial x_1}(\vec{x}) \phi^T(\vec{x}) d\vec{x} \mathbf{V}^{-1} \\ &=: \mathbf{V}^{-T} \mathcal{K}^{1,M} \mathbf{V}^{-1}. \end{aligned} \quad (2.51)$$

The modal stiffness matrix  $\mathcal{K}^{1,M}$  is computed exactly with Gauss integration and stored in an initial phase of the simulation. The evaluation of the surface integrals can be done in a similar manner, we refer to [41] for a detailed description of the nodal DG scheme. If we consider a hexahedron with a  $p = 5$  approximation, we get  $(p+1)^d = 216$  evaluation points with the Gaussian strategy for the first volume integral, and, following section 2.1.2.2 with  $\ell = (0, 0)$ ,  $n_L = 80$  evaluation points for the nodal DG scheme. An advantage of the nodal approach is that for the evaluation of the surface integral, no additional evaluations are needed, as the interpolation points for the volume share the interpolation points for the surface and could be re-used. Similar to the inexact modal approach discussed above, this approach causes aliasing errors. This issue is well-known from spectral methods, e.g. [39], and can be controlled by use of a weak modal filter [41]. To apply the filter, the DG polynomial(s) are transformed from the nodal representation into the modal representation

by multiplying the DOF with the Vandermonde matrix. The modal degrees of freedom are then filtered

$$u(\vec{x}, t) = \sum_{j=1}^{n_M} \hat{u}_j(t) \varphi_j(\vec{x}) \approx \sum_{j=1}^{n_M} \sigma(j) \hat{u}_j(t) \varphi_j(\vec{x}), \quad (2.52)$$

where the filter function  $\sigma$  is given by

$$\sigma(j) = \begin{cases} 1 & \text{for } 0 \leq n_M^{-1}(j) \leq p_c, \\ \exp\left(e \left(\frac{n_M^{-1}(j) - p_c}{1 - p_c}\right)^s\right) & \text{for } p_c < n_M^{-1}(j) \leq p, \end{cases} \quad (2.53)$$

where  $n_M^{-1}(j)$  maps the hierarchical sorted degrees of freedom counter  $j$  to the corresponding polynomial order. We furthermore introduced the positive integer parameter  $e$  and the modal cutoff  $p_c$ , in the sense that the polynomial part with degree smaller or equal  $p_c$  is not filtered. The parameter  $s$  is referred to as the order of the filter and is a positive *even* integer number. Increase  $s$  and/or  $p_c$  corresponds to a decrease of filter dissipation, whereas increasing  $e$  increases dissipation. According to [41] the best answer to the question how to choose the filter parameters  $e$ ,  $p_c$  and  $s$  is given by *filter as little as possible...but as much as is needed*. A good set of starting parameters for a reasonable resolved computation is given by  $e = 36$ ,  $p_c = 0$  and  $s = 16$ .

### 2.3.2 The Modal Approach with Nodal Integration

Using the above presented ideas from the nodal framework we now discuss how to enhance the efficiency of *modal DG* implementations. Instead of using Gauss integration, we use the nodal basis functions to define a high order interpolation of the flux  $f_1$

$$f_1(u) \approx \boldsymbol{\psi}^T \tilde{\mathbf{f}}_1, \quad (2.54)$$

where again  $(\tilde{\mathbf{f}}_1)_j := f_1(u(\xi_j^{\vec{x}}))$ . Inserting this into the volume integral yields

$$\int_Q f_1(u) \frac{\partial \phi}{\partial x_1}(\vec{x}) d\vec{x} \approx \boldsymbol{\kappa}^1 \tilde{\mathbf{f}}_1, \quad (2.55)$$

where the general stiffness matrix is given by

$$\boldsymbol{\kappa}^1 := \int_Q \frac{\partial \phi}{\partial x_1}(\vec{x}) \boldsymbol{\psi}^T(\vec{x}) d\vec{x} = \boldsymbol{\kappa}^{1,M} \boldsymbol{\mathcal{V}}^{-1}. \quad (2.56)$$

The evaluation of the stiffness matrix can again be done with Gauss integration in an initial phase of the simulation, yielding a quadrature free approach. The surface integrals are treated in a similar manner. De-aliasing with a weak filter (2.52) is straight forward, as the DG polynomial is already in modal representation. However, in all computations shown below no need for modal filtering is found.

### 2.3.3 Discussion

#### 2.3.3.1 Strong, but Conservative

As mentioned above, the approximate strong formulation suffers from being non-conservative in the general non-linear case. As we show in section 2.5 how to make use of the strong formulation to enhance efficiency of the fully discrete scheme, we are interested in ensuring exact conservation. An advantage of the *modal* DG scheme is the hierarchical structure of the data representation. The first degree of freedom  $\hat{u}_1$  corresponds to the mean value of the approximate solution. Thus, to ensure conservation, we only have to modify the DG equation of the first degree of freedom

$$\begin{aligned} \int_Q u_t \varphi_1 d\vec{x} &= - \int_{\partial Q} (g(u^\pm) \cdot \vec{n}) \varphi_1^- ds, \\ \int_Q \left( u_t + \vec{\nabla} \cdot \vec{f}(u) \right) \varphi_j d\vec{x} &= - \int_{\partial Q} \left( g(u^\pm) - (\vec{f}(u^-) \cdot \vec{n}) \right) \varphi_j^- ds, \quad j = 2, \dots, n_M, \end{aligned} \quad (2.57)$$

where we simple set

$$\int_Q \vec{\nabla} \cdot \vec{f}(u) \varphi_j d\vec{x} = \int_{\partial Q} \left( \vec{f}(u^-) \cdot \vec{n} \right) \varphi_j^- ds, \quad (2.58)$$

for  $j = 1$ , as  $\varphi_1(\vec{x}) = \text{constant}$ .

#### 2.3.3.2 Modal versus Nodal

Let us briefly consider a comparison of the pure nodal DG approach and the modal DG with nodal integration. We start with a reformulation of the modal DG scheme

$$\hat{\mathbf{u}}_t = - \int_{\partial Q} g(u^\pm) \phi ds + \int_Q \vec{f}(u) \cdot \vec{\nabla} \phi d\vec{x} =: \mathbf{r}(u, \phi), \quad (2.59)$$

where we used the fact, that the modal mass matrix is the identity. For the nodal DG scheme we get analogously

$$\mathcal{M}\tilde{\mathbf{u}}_t = \mathbf{r}(u, \psi) = \mathbf{V}^{-T} \mathbf{r}(u, \phi), \quad (2.60)$$

where we used the linearity of the residual with respect to the test functions and we introduced the nodal mass matrix

$$\mathcal{M} := \int_Q \psi \psi^T d\vec{x} = \mathbf{V}^{-T} \int_Q \phi \phi^T d\vec{x} \mathbf{V}^{-1} = \mathbf{V}^{-T} \mathbf{V}^{-1}. \quad (2.61)$$

If we use the fact that product  $\mathbf{V}^T \mathbf{V}^{-T}$  is the  $n_M \times n_M$  identity, equation (2.60) can be simplified to

$$\begin{aligned} \mathbf{V}^{-T} \mathbf{V}^{-1} \tilde{\mathbf{u}}_t &= \mathbf{V}^{-T} \mathbf{r}(u, \phi), \\ \mathbf{V}^{-1} \tilde{\mathbf{u}}_t &= \mathbf{r}(u, \phi). \end{aligned} \quad (2.62)$$

According to the inverse transformation (2.7) the term  $\mathbf{V}^{-1} \tilde{\mathbf{u}}_t$  simplifies to  $\hat{\mathbf{u}}_t^N$  and thus

$$\hat{\mathbf{u}}_t^N = \mathbf{r}(u, \phi). \quad (2.63)$$

If we make the assumption that we use the same discrete initial condition at  $t = 0$  for the modal and the nodal scheme and if we remind that the evaluations of the integrals are the same because of the nodal integration, we get that

$$\hat{\mathbf{u}}_t = \hat{\mathbf{u}}_t^N \quad (2.64)$$

and hence the equality of both approximate solutions for all  $t$ . Based on this results the following discussion about the different approaches only concerns the computational complexity and not the solution quality. Considering first order problems the main advantage of the pure nodal variant is that for the evaluation of the integrals, i.e. fluxes, the values of the (nodal) solution could directly be used, whereas for the modal variant a matrix vector multiplication (with the Vandermonde matrix, see (2.5)) is needed to convert the modal DOF into nodal DOF. However for strong non-linear problems the pure nodal approach needs an additional transformation with the Vandermonde monde matrix to apply the modal filter. Considering second order problems, where in addition the gradient of the solution is needed, the advantage of the pure nodal variant decreases as the interpolation of the gradient at the nodal points needs a matrix vector multiplication for both variants. Keeping in mind that the number of DOF in the pure nodal variant of the scheme is  $n_L$  compared to  $n_G \leq n_L$  for the modal-nodal variant, it seems that both schemes are nearly equal in terms of computational complexity.



## 2.4 The Numerical Fluxes

The choice of the numerical fluxes  $G^A$  and  $G^D$  determines the consistency, stability and the accuracy of the discontinuous Galerkin method. We recall that the fundamental difficulty lies in the fact, that the solution space is global discontinuous, i.e. it is polynomial inside a grid cell but discontinuous across grid cell interfaces. Thus the normal component of the flux on the surface, which is needed in the surface integral of the DG formulation, is not uniquely defined and has to be approximated. This problem is well known in the finite volume community where the framework of (approximative) Riemann solvers was developed. To demonstrate the idea, we restrict ourselves in a first step to the one dimensional scalar linear advection case. The Riemann problem is defined as an initial value problem with piecewise constant data

$$\begin{aligned} u_t + a u_x &= 0, \\ u(x, t = 0) &= \begin{cases} u^- & \text{for } x < 0, \\ u^+ & \text{for } x > 0, \end{cases} \end{aligned} \quad (2.65)$$

where  $a$  is the constant advection speed. We recall that for the discontinuous Galerkin discretization the approximation of the flux  $f^A(u) = a u$  at the grid cell interface  $\partial Q$  is needed, as the DG solution  $u$  itself is double valued, i.e. the value  $u^-$  from inside the grid cell and  $u^+$  from the neighbor grid cell, showing the close connection between the solution of the Riemann problem (2.65) and the approximation of the DG solution trace. In fact the idea of Riemann fluxes is to use the exact solution of the Riemann problem evaluated at  $x = 0$  for  $t > 0$  as an approximation of the flux at the grid cell interface

$$\begin{aligned} u_{Riemann} &= u^-, \quad \text{for } a > 0, \\ u_{Riemann} &= u^+, \quad \text{for } a < 0, \end{aligned} \quad (2.66)$$

yielding the well known upwind flux

$$\begin{aligned} f^A(u)|_{\partial Q} &= a u^-, \quad \text{for } a > 0, \\ f^A(u)|_{\partial Q} &= a u^+, \quad \text{for } a < 0, \\ \Rightarrow f^A(u)|_{\partial Q} &= \frac{1}{2}(a + |a|)u^- + \frac{1}{2}(a - |a|)u^+, \quad \forall a \in \mathbb{R}, \end{aligned} \quad (2.67)$$

which could be further reformulated, for the case  $a > 0$

$$\begin{aligned}
 a u^- &= a \left( \frac{1}{2}(u^- + u^+) - \frac{1}{2}(u^+ - u^-) \right), \\
 &= a \left( \{u\} - \frac{1}{2} \llbracket u \rrbracket \right), \\
 &= \frac{1}{2}(a u^- + a u^+) - a \frac{1}{2}(u^+ - u^-), \\
 &= \frac{1}{2}(f^A(u^-) + f^A(u^+)) - a \frac{1}{2}(u^+ - u^-), \\
 &= \{f^A(u)\} - a \frac{1}{2} \llbracket u \rrbracket,
 \end{aligned} \tag{2.68}$$

or for the general case  $a \in \mathbb{R}$

$$\begin{aligned}
 f^A(u)|_{\partial Q} &= a \{u\} - \frac{1}{2}|a| \llbracket u \rrbracket, \\
 &= \{f^A(u)\} - \frac{1}{2}|a| \llbracket u \rrbracket,
 \end{aligned} \tag{2.69}$$

with the trace operators *mean value*  $\{\cdot\}$  and *jump*  $\llbracket \cdot \rrbracket$ .

This approach could be generalized to non-linear hyperbolic systems with either exact or approximative Riemann fluxes. A straightforward way of constructing approximative Riemann fluxes is to use the solution of the one dimensional scalar linear equation (2.65). We start with a one dimensional non-linear hyperbolic system

$$U_t + F(U)_x = 0 \quad \text{or} \quad U_t + \underline{A}(U)U_x = 0, \tag{2.70}$$

where we make the assumption that the matrix  $\underline{A}$  has a full set of real eigenvalues and corresponding eigenvectors. We then introduce a linearization state  $U|_{\partial Q} \approx \tilde{U} = \tilde{U}(U^-, U^+)$  and solve the following Riemann problem for the linearized system

$$U_t + \underline{A}(\tilde{U})U_x = 0. \tag{2.71}$$

As the matrix  $\underline{A}(\tilde{U})$  has a full set of right eigenvectors  $\underline{T}$ , the system (2.71) could be diagonalized with the transformation to characteristic variables  $W := \underline{T}U$ , yielding  $m$  decoupled equation of the type (2.65), where the eigenvalues of  $\underline{A}(\tilde{U})$  are the advection speeds. For each decoupled equation, the solution of the Riemann problem is given by (2.69). After back-transformation from the

characteristic variables  $W$ , we get the exact solution of the Riemann problem for the linearized system (2.71) and thus the approximative Riemann flux for the non-linear problem (2.70)

$$F^A(U)|_{\partial Q} \approx \underline{A}(\tilde{U})\{U\} - \frac{1}{2}\underline{T}^{-1}|\underline{\Delta}|\underline{T} \llbracket U \rrbracket, \quad (2.72)$$

where  $\underline{\Delta}$  is a diagonal matrix with the eigenvalues of the matrix  $\underline{A}(\tilde{U})$ . An overview of different choices for the linearization state  $\tilde{U}$  and other different approximative Riemann solver strategies for a wide range of hyperbolic problems can be found in [75]. Qiu [66] and Qiu et al. [67] investigated different numerical fluxes for an explicit discontinuous Galerkin discretization for the inviscid Euler equations. They conclude that the influence on accuracy of different numerical fluxes decrease when the polynomial order of the approximation space increases. However in their investigation it seems that the HLLC flux [76] yields the best results with respect to accuracy and efficiency. Thus, in the simulations presented below, we choose the HLLC flux for the approximation of the Euler flux

$$G_{HLLC}^A := \begin{cases} F(U^-) & \text{for } s^- > 0, \\ F(U^-) + s^- (U_*^- - U^-) & \text{for } s^- \leq 0 < s^*, \\ F(U^+) + s^+ (U_*^+ - U^+) & \text{for } s^* \leq 0 \leq s^+, \\ F(U^+) & \text{for } s^+ < 0, \end{cases} \quad (2.73)$$

where

$$\begin{aligned} s^\pm &:= v^\pm \pm c^\pm, \\ s^* &:= \frac{\rho^+ v^+ (s^+ - v^+) - \rho^- v^- (s^- - v^-) + p^- - p^+}{\rho^+ (s^+ - v^+) - \rho^- (s^- - v^-)}, \end{aligned} \quad (2.74)$$

and

$$U_*^\pm := \rho^\pm \frac{s^\pm - v^\pm}{s^\pm - s^*} \left( \begin{array}{c} 1 \\ s^* \\ e^\pm + (s^* - v^\pm) \left( s^* + \frac{p^\pm}{\rho^\pm (s^\pm - v^\pm)} \right) \end{array} \right). \quad (2.75)$$

Up to now, we have only considered the one dimensional case and only the pure hyperbolic case. Due to the rotational invariance property of the Euler equations (1.15), the extension to the multidimensional case of the approximative Riemann fluxes is straightforward, as only the numerical flux in the normal

direction of the surface is needed. Thus, using the finite volume framework we are able to introduce accurate and stable numerical flux approximations for the DG discretization of hyperbolic problems. For the approximation of the diffusion fluxes it seems that other techniques are needed, techniques for instance introduced in the finite element community, such as the symmetric interior penalty (SIP) scheme of Nitsche [59]. We discuss in the next section an alternative technique developed in the DG community.

### 2.4.1 State of the Art

For the approximation of the diffusion fluxes a lot of different choices exist. As for the pure hyperbolic case, they differ in terms of accuracy, stability and efficiency. In [3] and [4] Arnold et al. give an overview of state of the art numerical flux approximations for the pure elliptic case, i.e. the Laplace problem. They found out that only a few numerical flux approximations yield optimal order of convergence for this simple test case. Namely, the SIP scheme [59] and [2], the local discontinuous Galerkin (LDG) scheme of Cockburn and Shu [18] and the second method of Bassi and Rebay (BR2) [10]. Regarding efficiency, the local DG method suffers from an enlarged stencil compared to the other two schemes, which use only data from the von Neumann neighbor grid cells. Comparing the SIP method to the second method of Bassi and Rebay, the SIP method is more efficient regarding the computational effort. However, we found out that the straightforward extension of the SIP method to the compressible Navier-Stokes equations [37] yields non-optimal convergence orders for approximation spaces with even polynomial degree. Thus, up to now, the BR2 method is the only method yielding optimal order of convergence for the compressible Navier-Stokes equations while remaining a compact stencil. To demonstrate the BR2 scheme we restrict ourselves again to the scalar diffusion problem (2.26) and use the mixed discontinuous finite element formulation of Bassi and Rebay (2.28)

$$\begin{aligned} \int_Q u_t \phi \, d\vec{x} &= \int_{\partial Q} g(u^\pm, \bar{q}^\pm) \phi^- \, ds - \int_Q \vec{f} \cdot \vec{\nabla} \phi \, d\vec{x}, \\ \int_Q \bar{q} \phi \, d\vec{x} &= \int_{\partial Q} \tilde{u} \vec{n} \phi \, ds - \int_Q u \vec{\nabla} \phi \, d\vec{x}, \end{aligned} \tag{2.76}$$

where we have to define the numerical approximation of the flux  $g(u^\pm, \bar{q}^\pm)$  and the solution  $\tilde{u}$ . Bassi and Rebay choose for the solution trace the arithmetic mean value

$$\tilde{u} \approx \{u\}, \tag{2.77}$$

which seems to be natural for diffusion problems. They furthermore reformulate the second equation in (2.76) using back integration by parts, where the new fluxes are evaluated from inside the grid cell

$$\begin{aligned} \int_Q \vec{q}\phi \, d\vec{x} &= \int_{\partial Q} (\{u\} - u^-) \vec{n} \phi^- \, ds + \int_Q \vec{\nabla} u \phi \, d\vec{x}, \\ &= \int_{\partial Q} \frac{1}{2} \llbracket u \rrbracket \vec{n} \phi^- \, ds + \int_Q \vec{\nabla} u \phi^- \, d\vec{x}. \end{aligned} \quad (2.78)$$

It is now possible to introduce the so called global lifting operator  $\vec{r} = \vec{r}(\llbracket u \rrbracket)$

$$\int_Q \vec{r}\phi \, d\vec{x} \stackrel{!}{=} \int_{\partial Q} \frac{1}{2} \llbracket u \rrbracket \vec{n} \phi^- \, ds, \quad (2.79)$$

where we choose for the global lifting operator an approximation of the same order to get a polynomial expanded in the modal basis functions  $\{\varphi_i\}_{i=1, \dots, n_M}$ . When we re-substitute, we get in the weak sense the following relationship

$$\vec{q} = \vec{\nabla} u + \vec{r}(\llbracket u \rrbracket), \quad (2.80)$$

which reveals that the BR2 DG derivative is composed of the local derivative plus a term depending on the jump of the DG solution. For the definition of the numerical flux  $g(u^\pm, \vec{q}^\pm)$  Bassi and Rebay furthermore introduced the local lifting operator  $\vec{l} = \vec{l}(\llbracket u \rrbracket)$ . For the construction of  $\vec{l}$  we have to split the surface in piecewise polygonal/polyhedral sub-surfaces  $\partial Q = \bigcup_j \partial Q_j$  to define

$$\int_Q \vec{l}_j \phi \, d\vec{x} \stackrel{!}{=} \int_{\partial Q_j} \frac{1}{2} \llbracket u \rrbracket \vec{n} \phi^- \, ds, \quad (2.81)$$

with the relation

$$\vec{r} = \sum_j \vec{l}_j. \quad (2.82)$$

According to Bassi and Rebay the flux for a position  $\vec{x} \in \partial Q_j$  is given by

$$g(u^\pm, \vec{q}^\pm) = \{\vec{f}(u, \vec{\nabla} u + \eta_{BR2} \vec{l}_j)\} \cdot \vec{n}, \quad (2.83)$$

where  $\eta_{BR2}$  is a positive constant, which should be chosen equal to the number of faces to guarantee the stability of the viscous operator. The extension of

this approach to diffusion systems is straightforward. The BR2 approximations could be used in the ultra weak formulation as well, yielding a compact scheme with optimal order. We refer to this approach as the DG community approach. In the next section we will show a novel approach for the derivation of the diffusion fluxes in the spirit of the finite volume community. A surprising result of this approach is a strong connection to the finite element techniques and a natural generalization of the SIP method to non-linear diffusion problems, such as the compressible Navier-Stokes equations.

### 2.4.2 The Diffusive Generalized Riemann Problem

We will derive now a framework for the approximation of diffusion fluxes analogous to the derivation of the hyperbolic fluxes as discussed above. In a first step we focus on the simple one dimensional scalar linear example

$$u_t - (du_x)_x = 0, \tag{2.84}$$

where  $d$  is the constant and positive diffusion coefficient. We proceed and consider the (diffusive) Riemann problem with piecewise constant initial data

$$\begin{aligned} u_t - (du_x)_x &= 0, \\ u(x, t = 0) &= \begin{cases} u^- & \text{for } x < 0, \\ u^+ & \text{for } x > 0, \end{cases} \end{aligned} \tag{2.85}$$

A general bounded solution [49] is given by

$$u(x, t) = \frac{1}{2\sqrt{d\pi t}} \int_{-\infty}^{\infty} u(\xi, t = 0) e^{-\frac{(x-\xi)^2}{4dt}} d\xi. \tag{2.86}$$

Inserting the discontinuous initial data yields the solution at  $x = 0$  and  $t > 0$

$$u(x = 0, t) = \frac{1}{2}(u^- + u^+) = \{u\}, \tag{2.87}$$

which corresponds very well to the intuitive choice of Bassi and Rebay. We can furthermore differentiate the solution with respect to  $x$  and get the flux at  $x = 0$  and  $t > 0$  as

$$u_x(0, t) = \frac{u^+ - u^-}{2\sqrt{\pi dt}} = \frac{[[u]]}{2\sqrt{\pi dt}}. \tag{2.88}$$

We note that the diffusion flux depends explicitly on time  $t$ , indicating a strong relation to the discretization in time. Unfortunately, numerical investigations

with different explicit time integration methods revealed that this approach is inconsistent. The main problem is that the approach only takes the local values  $u^\pm$  into account, but not the local derivatives  $u_x^\pm$ . We therefore generalize the above presented approach and extend the initial data to piecewise linear data

$$\begin{aligned} u_t - (d u_x)_x &= 0, \\ u(x, t = 0) &= \begin{cases} u^- + x u_x^- & \text{for } x < 0, \\ u^+ + x u_x^+ & \text{for } x > 0. \end{cases} \end{aligned} \quad (2.89)$$

We call this initial value problem the diffusive generalized Riemann problem (dGRP) motivated by the name in the hyperbolic case, which was introduced by Ben-Artzi and Falcovitz [11] for piecewise linear initial data, constructing a second order accurate finite volume scheme for the Euler equations.

Inserting the initial data into (2.86) yields the exact dGRP solutions at  $x = 0$  for  $t > 0$

$$\begin{aligned} u(x = 0, t) &= \{u\} + \sqrt{\frac{dt}{\pi}} \llbracket u_x \rrbracket, \\ u_x(x = 0, t) &= \frac{\llbracket u \rrbracket}{2\sqrt{\pi dt}} + \{u_x\}. \end{aligned} \quad (2.90)$$

Numerical investigations indicate that the ultra weak discontinuous Galerkin formulation combined with this numerical approximations yields a scheme with optimal order of convergence [34], when a time integration scheme with corresponding time order is chosen. Before analyzing the numerical fluxes, we introduce a simplification of the exact solutions to get rid of the explicit time dependence. We integrate (2.90) exactly in time and average with the time interval  $\Delta t$

$$\begin{aligned} \frac{1}{\Delta t} \int_0^{\Delta t} u(x = 0, t) dt &= \{u\} + \frac{2}{3} \sqrt{\frac{d \Delta t}{\pi}} \llbracket u_x \rrbracket, \\ \frac{1}{\Delta t} \int_0^{\Delta t} u_x(x = 0, t) dt &= \frac{\llbracket u \rrbracket}{\sqrt{\pi d \Delta t}} + \{u_x\}, \end{aligned} \quad (2.91)$$

and substitute the explicit diffusive time step (2.102), derived in section 2.5

$$\Delta t = \beta(p) \frac{\Delta x^2}{(2p + 1)^2 d}, \quad (2.92)$$

where  $\Delta x$  is the grid cell width and  $\beta$  depends on the time integration method and on the order  $p$  of the DG approximation space. Inserting this definition yields

$$\begin{aligned} u|_{\partial Q} &:= \{u\}, \\ du_x|_{\partial Q} &:= d\left(\eta \frac{[[u]]}{\Delta x} + \{u_x\}\right), \end{aligned} \tag{2.93}$$

where we introduce the constant  $\eta := \frac{2p+1}{\sqrt{\beta(p)\pi}}$  and furthermore skipped the derivative dependent term in the first equation, as numerical investigations indicate that this term has very little influence on the behavior of the discretization. When combining this discretization with the presented time integration method, section 2.5, and its local time stepping feature, the choice of the time step constant  $\beta$  for the constant  $\eta$  yields non-optimal results, as the derivation of the flux assumes a common time line, i.e. global time stepping. However it is possible to separate the calculation of the time step  $\Delta t$  and the definition of the constant  $\eta \approx \tilde{\eta} := \frac{2p+1}{\sqrt{\beta\pi}}$  with the introduction of the new constant  $\tilde{\beta} = \tilde{\beta}(p)$ . A stability analysis [56] yields the optimal dGRP constant  $\tilde{\beta}_{max}$  which we choose for all simulations presented below

$p$	1	2	3	4	5	6	7
$(2p+1)^2 \tilde{\beta}_{max}$	2.8	0.86	0.40	0.24	0.16	0.12	0.09

**Table 2.1:** Stability numbers for the diffusion flux approximation

If we analyze the structure of this new numerical flux approximations, we get the arithmetic mean for the solution at the interface, which is not only intuitive but corresponds very well to the choice of Bassi and Rebay and to the approximation used in the SIP method. A surprising result is that the approximation of the flux term is identical to the approximation used in the SIP method. The only difference is the definition of the constant  $\tilde{\eta}$ , which is called *penalty constant* in the SIP framework and has the structure

$$\eta_{SIP} = \frac{C}{p^2}, \text{ with } C > 0 \text{ large enough.} \tag{2.94}$$

Within the framework of the dGRP flux approximation the (penalty) constant  $\eta_{SIP}$  could be chosen as  $\tilde{\eta}$ . The condition 'choose  $C > 0$  large enough' corresponds to the magnitude of the parameter  $\tilde{\beta}_{max}$ . When we choose  $\beta > \tilde{\beta}_{max}$ ,



the constant  $\tilde{\eta}$  decreases resulting in positive real parts of some eigenvalues of the semi discrete discontinuous Galerkin scheme and thus in an unstable discretization. Comparing the numerical flux approximation to the BR2 approximation, both have the structure *mean value of the local gradient* plus a term which depends on the *jump of the solution values*.

Analogous to the hyperbolic case, this approach could be used to find an approximate Riemann flux for non-linear diffusion systems

$$U_t - F^D(U, U_x)_x = 0 \quad \text{or} \quad U_t - (\underline{D}(U)U_x)_x = 0, \quad (2.95)$$

with the positive (semi) definite diffusion matrix  $\underline{D}$ . We introduce a proper linearization state, which we choose intuitively as  $\tilde{U} = \{U\}$ . We additionally assume that the resulting linear diffusion matrix  $\underline{D}(\tilde{U})$  can be diagonalized. We denote the 'characteristic' variables again with  $W := \underline{T}U$ , where  $\underline{T}$  denotes again the right eigenvector matrix, and end up with  $m$  decoupled linear diffusion equations, where the eigenvalues of the diffusion matrix are the scalar diffusion coefficients. For every scalar equation the numerical fluxes in (2.91) can be used. After back-transformation we get the approximative Riemann flux for the non-linear problem (2.95)

$$\begin{aligned} D(U)U_x|_{\partial Q} &\approx \underline{D}(\tilde{U})\{U_x\} + \underline{D}(\tilde{U})\frac{[[U]]}{\Delta x}\tilde{\eta}, \\ &\approx \underline{D}(\tilde{U})\left(\{U_x\} + \frac{[[U]]}{\Delta x}\tilde{\eta}\right). \end{aligned} \quad (2.96)$$

The extension to the multidimensional compressible Navier-Stokes equations case [35] is straightforward, as analogous to the hyperbolic approximation, the rotational invariance property of the diffusion fluxes could be used, see (1.16). For the approximation of the solution at the interface, we use again the idea of arithmetic mean. Inserting this into (2.39) yields

$$\begin{aligned} h(U, \vec{n}, \vec{\nabla}\Phi^-) &= \tilde{U} \cdot F_{\vec{n}}^{D'}(\tilde{U}, \vec{\nabla}\Phi^-) - U^- \cdot F_{\vec{n}}^{D'}(U^-, \vec{\nabla}\Phi^-), \\ &\approx \{U\} \cdot F_{\vec{n}}^{D'}(\{U\}, \vec{\nabla}\Phi^-) - U^- \cdot F_{\vec{n}}^{D'}(U^-, \vec{\nabla}\Phi^-). \end{aligned} \quad (2.97)$$

We furthermore introduce an additional simplification for the second term, which allows a more efficient implementation

$$\begin{aligned} h(U, \vec{n}, \vec{\nabla}\Phi^-) &\approx \{U\} \cdot F_{\vec{n}}^{D'}(\{U\}, \vec{\nabla}\Phi^-) - U^- \cdot F_{\vec{n}}^{D'}(\{U\}, \vec{\nabla}\Phi^-), \\ &= \frac{1}{2} [[U]] \cdot F_{\vec{n}}^{D'}(\{U\}, \vec{\nabla}\Phi^-). \end{aligned} \quad (2.98)$$

An interesting property of this numerical fluxes is that they are independent of the underlying problem, yielding a generic approximation for non-linear diffusion systems.

### 2.4.3 Discussion

In this chapter a novel approach for the definition of diffusion fluxes was introduced. We used ideas of the finite volume community, namely the framework of Riemann problems, and get stable and accurate approximations. We generalized the exact solution of the scalar linear one dimensional diffusive generalized Riemann problem in the spirit of approximative hyperbolic Riemann solvers to find appropriate parameter free approximations for the compressible Navier-Stokes diffusion fluxes. Comparing the scheme to the SIP-like discretization of the compressible Navier-Stokes equation introduced by Hartmann and Houston [37], the dGRP flux yields optimal order of convergence, whereas the SIP scheme yields only sub-optimal order for approximation spaces with even order polynomials. Starting from the dGRP flux (2.96) we get a similar SIP-like method, when we simplify the term involving the jump term. Substituting the diffusion matrix with the largest eigenvalue times the  $m \times m$  identity  $I$ , we get

$$\begin{aligned} D(U)U_x|_{\partial Q} &= \underline{D}(\tilde{U})\{U_x\} + \underline{D}(\tilde{U})\frac{[[U]]}{\Delta x}\tilde{\eta}, \\ &\approx \underline{D}(\tilde{U})\{U_x\} + \lambda_{max}\frac{[[U]]}{\Delta x}\tilde{\eta}. \end{aligned} \tag{2.99}$$

The problem with this approximation is that the jump term acts in every equation. When the diffusion matrix is only *semi* positive definite, which is the case for the compressible Navier-Stokes equation, the penalization in the corresponding characteristic variable results in a loss of optimal convergence. In the dGRP approximation, the jump term is multiplied with the diffusion matrix, yielding zero penalization when the corresponding eigenvalue is zero. Numerical experiments show that the SIP discretization and the dGRP discretization yields similar results for positive definite diffusion problems, but the latter yields improved convergence order for semi positive definite diffusion problems. Thus only the dGRP and the BR2 approximation are discretizations with optimal order of convergence, while preserving the small von Neumann stencil. Comparing the dGRP approach to the BR2 scheme, numerical experiments reveal a slight performance advantage of the dGRP flux of about 30% in our implementation, as no additional lifting operators have to be calculated and stored.

## 2.5 Time Discretization

So far, only spatial aspects of the discretization were considered. The semi discrete discontinuous Galerkin formulation yields a system of ODE's for the time dependent polynomial coefficients

$$\hat{U}_t(t) = \mathcal{R}_{DG}(\hat{U}(t)). \quad (2.100)$$

The residual  $\mathcal{R}_{DG}$  of the DG semi discretization has the favorable property of a dense structure, as the degrees of freedom of a given grid cell only depend on the degrees of freedom from the direct (von Neumann) neighbors. In this thesis, we focus on the development of a discretization for the simulation of (highly) unsteady problems. Thus to resolve the time scale of the problem, physical restrictions on the time step are needed. This is in contrast to simulations of steady problems, where the time step could be chosen arbitrarily large, with respect to the resolution of the time scale. A typical approach for the time discretization is to subdivide the time axis  $t \in [0, t_{end}]$  in (non-equidistant) time-intervals

$$0 = t^0 < t^1 < \dots < t^n < \dots t^{nt} = t_{end}, \quad (2.101)$$

and solve the ODE's (2.100) from time level to time level. The DG solution at a given time level  $t^n$  is denoted with  $U(\vec{x}, t = t^n) = U^n(\vec{x})$ .

### 2.5.1 State of the Art

In a series of paper [14–17, 19] Cockburn and Shu developed the Runge-Kutta discontinuous Galerkin (RKDG) framework. They used a high order accurate Runge-Kutta scheme to integrate the ODE (2.100) in time. The advantage of the RKDG scheme is that its explicit and local nature results in a very efficient method for massive parallel computations [12]. A characteristic of explicit time integration is that the maximum allowable time step is restricted to guarantee stability. For a discontinuous Galerkin scheme the time step restriction has the structure

$$\begin{aligned} \Delta t^A &\leq \frac{\alpha(p)}{\lambda_{max}^A} \left( \frac{\Delta x}{2p+1} \right) \text{ for advection,} \\ \Delta t^D &\leq \frac{\beta(p)}{\lambda_{max}^D} \left( \frac{\Delta x}{2p+1} \right)^2 \text{ for diffusion,} \\ \Delta t &\leq \left( (\Delta t^A)^{-2} + (\Delta t^D)^{-2} \right)^{-\frac{1}{2}} \text{ for advection-diffusion,} \end{aligned} \quad (2.102)$$

where  $\lambda_{max}^{A/D}$  denote the maximum eigenvalue of the advection Jacobians and diffusion matrices. The stability numbers  $\alpha$  and  $\beta$  depend on the order of the underlying approximation space and on the explicit time discretization method. We note that the time step is dependent on the spatial resolution  $\Delta x/(2p + 1)$  analogous to the *CFL* condition in finite volume schemes. This relates the maximum possible time step (time resolution) to the spatial resolution, which seems natural and physical meaningful for advection dominated problems. Actually it turns out that the physical meaningful time step determined for an unsteady problem is in the range of the explicit time step. For an approximation space with uniform spatial resolution and uniform distribution of maximal eigenvalues this yields an efficient method. However, practical problems of interest often have an extreme inhomogeneous distribution of maximal eigenvalues. Furthermore to save computational cost, the spatial resolution is non-uniformly distributed to take advantage of different solution behaviors during a simulation. Either constructing the approximation space in an initial phase of the simulation or in an adaptive fashion during the simulation, the magnitude of the local resolutions can vary drastically over the computational domain. In relation to this, a major drawback of an explicit time discretization is that the minimum time step over the whole computational domain has to be used as a global time step to advance the solution in time. Thus advancing with the minimum time step may result in a drastic decrease in efficiency in such simulations.

An alternative to explicit time stepping is to use implicit time integration methods, such as *backward difference formulae* or *implicit Runge Kutta* methods [79]. The advantage of these methods is that no theoretical stability limits for the maximum allowable time step exists. Thus the time step is only restricted due to physical considerations and not due to stability limits. The drawback however is that large (non-linear) algebraic systems have to be solved with Newton iteration methods [63], which cause a high computational cost per time step. An implicit time discretization therefore only pays off if the maximal global explicit time step is 'small' compared to the physical meaningful time step. The definition of 'small' is highly dependent on the problem to be solved.

If we focus on massive parallel large scale computations with  $\mathcal{O}(1000)$  processors, it is clear that up to now the algorithms and methods to solve large non-linear algebraic systems are under-developed. An interesting alternative solution strategy for the large non-linear system is to use a pseudo time approach, see e.g. van der Vegt and van der Ven [78], where the system is solved in pseudo time with an explicit scheme. The advantage of this method is that the

algorithm retains its explicit character which is good for the parallelizability, the drawback is that we have again some sort of (pseudo) time step restriction. An open topic is the use of (local) implicit convergence acceleration methods to increase the maximal allowed pseudo time step, which could be arbitrarily large as no accuracy in pseudo time is needed. Another approach that combines explicit and implicit techniques was recently introduced in Kanevsky et al. [46], where the ODE for the time dependent DG degrees of freedom are solved with an *implicit-explicit Runge-Kutta* method.

In the next section a new purely explicit time discretization is introduced, which gives us the possibility to use time accurate local time stepping as introduced in Lörcher et al. [55] to alleviate the (global) explicit time step restriction.

### 2.5.2 The Predictor Corrector Approach

In contrast to the method of line (MOL) approach discussed above, we introduce now an approach where the spatial and temporal discretization is *not* separated. We recall the strong discontinuous Galerkin formulation (2.40)

$$\int_Q U_t \cdot \Phi \, d\vec{x} = - \int_Q \left( \vec{\nabla} \cdot \left( \vec{F}^A - \vec{F}^D \right) \right) \cdot \Phi \, d\vec{x} - \int_{\partial Q} \left( F \cdot \Phi^- + h \right) \, ds, \quad (2.103)$$

which was the starting point for the MOL approach. In our approach we start with the fully discrete formulation, which results when we simply integrate (2.103) from time level  $t^n$  to time level  $t^{n+1}$

$$\begin{aligned} \int_Q \left( U^{n+1} - U^n \right) \cdot \Phi \, d\vec{x} = & - \int_{t^n}^{t^{n+1}} \int_Q \left( \vec{\nabla} \cdot \left( \vec{F}^A(U) - \vec{F}^D(U, \vec{\nabla}U) \right) \right) \cdot \Phi \, d\vec{x} \, dt \\ & - \int_{t^n}^{t^{n+1}} \int_{\partial Q} \left( F(U^\pm, \vec{\nabla}U^\pm) \cdot \Phi^- + h(U^\pm, \vec{\nabla}\Phi^-) \right) \, ds \, dt. \end{aligned} \quad (2.104)$$

The time integrals are approximated with a one dimensional Gauss quadrature rule with  $n_G = (p_t + 1)/2$  evaluation points, where  $p_t + 1$  is the desired order of accuracy in time. As the time Gauss positions are located in-between the time levels  $t^n$  and  $t^{n+1}$ , a value of the solution at these intermediate time levels is needed to evaluate the right hand side of (2.104). We discuss in the next subsection the construction of these solution values.

### 2.5.2.1 The Local Space-Time Solution

To get a high order scheme in space and time, we need a high order accurate approximation of the values at the time Gauss points. We therefore construct a local space-time approximation  $V = V(\vec{x}, t)$  for  $\vec{x} \in Q$  and  $t \in [t^n, t^{n+1}]$ . We search the space-time solution for the following Cauchy problem

$$\begin{aligned} V_t + \vec{\nabla} \cdot \left( F^A(V) - F^D(V, \vec{\nabla}V) \right) &= 0, \\ V(\vec{x}, t = 0) &= U_*^n(\vec{x}), \forall \vec{x} \in \mathbb{R}^d, \end{aligned} \quad (2.105)$$

where  $U_*^n$  is the DG polynomial in the grid cell  $Q$  at time level  $t^n$  extended in  $\mathbb{R}^d$ . We note that the calculation of the exact solution for this problem is due to the non-linearity impractically costly. However, considering the 'limited' accuracy of the resulting DG discretization, only a high order accurate approximation of this problem is needed. An efficient way to construct such an approximation is to use a Taylor expansion in space and time with appropriate order [1, 35, 48, 55]. The derivatives of the space-time expansion are approximated using the so-called Cauchy-Kovalevskaya procedure, see [31] for an efficient way of implementing a Cauchy-Kovalevskaya procedure for non-linear problems. Recently Dumbser et al. [27, 28] introduced another approach to find an approximation of this problem. They used either a discontinuous Galerkin or a continuous Galerkin discretization in time and solved the resulting *local* (non-linear) system with an iterative procedure. In practice, every technique for the solution of an initial value problem could be chosen, as long as the result is an analytical space-time solution. We will introduce now an alternative approach to find an approximation of the Cauchy problem (2.105). In [60, 61] a new Runge-Kutta based framework for the solution of initial value problems was introduced. The main feature of this approach is that the approximate solution could be naturally extended to a polynomial. We construct the approximative solution using a continuous extension Runge-Kutta Galerkin (CERKG) method. We start with the semi discrete Galerkin formulation

$$\begin{aligned} \int_Q \left( V_t + \vec{\nabla} \cdot \left( F^A(V) - F^D(V, \vec{\nabla}V) \right) \right) \cdot \Phi \, d\vec{x} &= 0, \\ \Rightarrow \hat{V}_t &= \mathcal{R}_G(V) \text{ with } \hat{V}^n = \hat{U}^n. \end{aligned} \quad (2.106)$$

We note that this Galerkin formulation does not need neighbor data. We get this formulation, when we either *don't* use integration by parts or equivalently use integration by parts, formulation (2.103), and choose the numerical fluxes

from *inside* the grid cell, as in this case the surface terms  $F$  and  $h$  vanish. For the evolution in time, we proceed with a continuous extension Runge-Kutta method (CERK)

$$\begin{aligned}
 \hat{\mathbf{V}}^{n+1} &= \hat{\mathbf{U}}^n + \Delta t \sum_{j=1}^{n_{stages}} b_j \hat{\mathbf{K}}_j, \\
 \hat{\mathbf{K}}_j &= \mathcal{R}_G(\hat{\mathbf{V}}^{n,j}), \\
 \hat{\mathbf{V}}^{n,j} &= \hat{\mathbf{U}}^n + \sum_{l=1}^{n_{stages}} a_{jl} \hat{\mathbf{K}}_l, \\
 \hat{\mathbf{V}}(t) &= \sum_{j=1}^{p_t+1} c_j^t t^j, \\
 c_j^t &= \sum_{l=1}^{n_{stages}} b_{jl}^t \hat{\mathbf{K}}_l,
 \end{aligned} \tag{2.107}$$

where  $n_{stages}$  and  $a_{jl}$ ,  $b_j$ ,  $b_{jl}^t$  are the CERK coefficients depending on the time order, see Appendix B. We note that for a discontinuous Galerkin discretization with time order  $p_t + 1$ , the time order of the local space-time solution approximation has to be  $p_t$ . Thus, we need for instance a second order accurate CERK scheme to get an overall temporal accuracy of third order. This approximative solution is calculated for every grid cell  $Q$  and can be used to evaluate the right hand side of (2.104)

$$\begin{aligned}
 \int_Q (U^{n+1} - U^n) \cdot \Phi \, d\vec{x} &= - \int_{t^n}^{t^{n+1}} \int_Q \left( \vec{\nabla} \cdot \left( \vec{F}^A(V) - \vec{F}^D(V, \vec{\nabla}V) \right) \right) \cdot \Phi \, d\vec{x} \, dt \\
 &\quad - \int_{t^n}^{t^{n+1}} \int_{\partial Q} \left( F(V^\pm, \vec{\nabla}V^\pm) \cdot \Phi^- + h(V^\pm, \vec{\nabla}\Phi^-) \right) \, ds \, dt,
 \end{aligned} \tag{2.108}$$

yielding the fully discrete scheme. Due to the explicit nature of our temporal discretization, a time step restriction is needed to guarantee stability. We list the stability numbers, based on a matrix analysis [54–56], in Tables 2.2 and 2.3. We note that in the pure diffusion case, the time order  $p_t + 1$  gets doubled with respect to the discretization parameter  $\Delta x$  as for pure diffusion  $\Delta t \approx \Delta x^2$ .

$p_t/p$	0	1	2	3	4	5	6	7
0	1.0	<0.05	<0.05	<0.05	<0.05	<0.05	<0.05	<0.05
1		1.0	0.9	0.4	0.3	0.3	0.3	0.25
2			0.8	0.7	0.6	0.55	0.48	0.42
3				0.7	0.6	0.55	0.48	0.42
4					0.6	0.55	0.48	0.42
5						0.55	0.48	0.42
6							0.48	0.42
7								0.42

**Table 2.2:** Stability numbers  $\alpha$  for the pure one dimensional scalar and linear advection case.

$p_t/p$	1	2	3	4	5	6	7
0	1.5	0.8	0.57	0.42	0.32	0.26	0.20
1		0.8	0.54	0.38	0.28	0.21	0.17
2				0.38	0.28	0.21	0.17
3						0.21	0.17

**Table 2.3:** Stability numbers  $\beta$  for the pure one dimensional scalar and linear diffusion case.

Up to now, the presented time integration suffers from the same problems as for instance the RKDG method. We discuss in the following section how to make use of the space time approximate solution to enhance the efficiency of the fully discrete scheme.

### 2.5.2.2 The Predictor Corrector Formulation

We recall the derivation of the approximate Cauchy solution (2.106)

$$\int_Q V_t \cdot \Phi \, d\vec{x} = - \int_Q \vec{\nabla} \cdot \left( F^A(V) - F^D(V, \vec{\nabla}V) \right) \cdot \Phi \, d\vec{x}. \quad (2.109)$$



Inserting this expression into the fully discrete scheme (2.108) yields

$$\begin{aligned} \int_Q (U^{n+1} - U^n) \cdot \Phi \, d\vec{x} &= \int_{t^n}^{t^{n+1}} \int_Q V_t \cdot \Phi \, d\vec{x} \, dt \\ &\quad - \int_{t^n}^{t^{n+1}} \int_{\partial Q} \left( F(V^\pm, \vec{\nabla} V^\pm) \cdot \Phi^- + h(V^\pm, \vec{\nabla} \Phi^-) \right) \, ds \, dt. \end{aligned} \quad (2.110)$$

We can now integrate exactly in time

$$\int_{t^n}^{t^{n+1}} \int_Q V_t \cdot \Phi \, d\vec{x} \, dt = \int_Q (V^{n+1} - V^n) \cdot \Phi \, d\vec{x}, \quad (2.111)$$

and use the initial condition  $V^n = U^n$  to get the *predictor corrector* formulation

$$\hat{U}^{n+1} = \hat{V}^{n+1} - \int_{t^n}^{t^{n+1}} \int_{\partial Q} \left( F(V^\pm, \vec{\nabla} V^\pm) \cdot \Phi^- + h(V^\pm, \vec{\nabla} \Phi^-) \right) \, ds \, dt, \quad (2.112)$$

which reveals that the DG degrees of freedom at the new time level  $t^{n+1}$  are the degrees of freedom predicted with the local Cauchy problem, corrected with the surface terms involving information from the neighbor grid cells. The advantage of the predictor corrector formulation is that the calculation of the volume integral in the predictor can be re-used to compute the volume integral in the DG formulation. As mentioned above the use of the strong formulation is not unproblematic with respect to exact local conservation. Hence, to overcome this issue, we use the modification described in subsection 2.3.3.1, yielding the following scheme

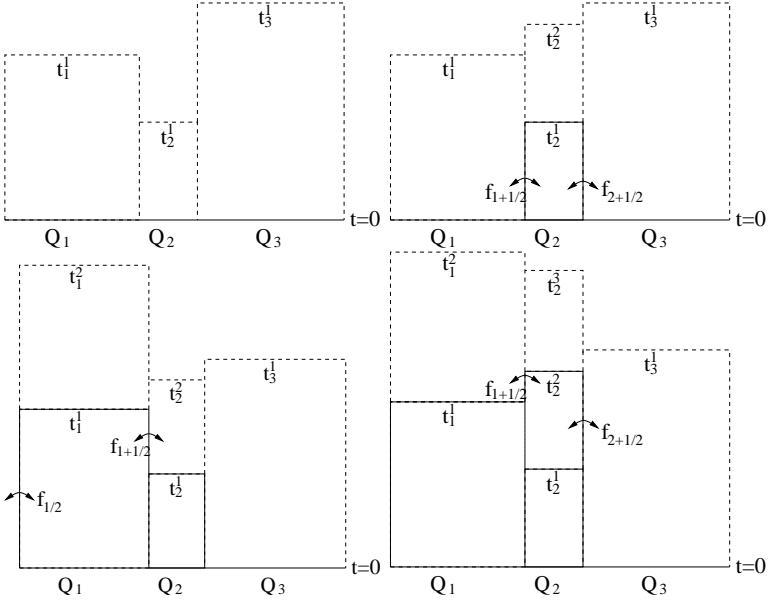
$$\begin{aligned} \hat{U}_1^{n+1} &= \hat{U}_1^n - \int_{t^n}^{t^{n+1}} \int_{\partial Q} \left( G^A(V^\pm) - G^D(V^\pm, \vec{\nabla} V^\pm) \right) \Phi_1^- \, ds \, dt, \\ \hat{U}_j^{n+1} &= \hat{V}_j^{n+1} - \int_{t^n}^{t^{n+1}} \int_{\partial Q} \left( F(V^\pm, \vec{\nabla} V^\pm) \Phi_j^- + h(V^\pm, \vec{\nabla} \Phi_j^-) \right) \, ds \, dt, \quad j = 2, \dots, n_M. \end{aligned} \quad (2.113)$$

### 2.5.2.3 Time Accurate Local Time Stepping

As discussed above, the main disadvantage of an explicit time discretization for an advection dominated problem is that a global time step is needed to ensure stability. We will show now how to make use of the locality of the discontinuous Galerkin scheme and of the space-time nature of our predictor-corrector approach to enhance the efficiency of our explicit time discretization by overcoming the global time step restriction. The predictor corrector approach allows us to introduce a time accurate fully conservative local time stepping, similar to the algorithms presented in [30, 35, 55, 56]. We first give up the assumption that all grid cells run with the same time step. We calculate for every grid cell  $Q_i$  the time step according to the local stability restriction (2.102) and introduce the local time levels  $t_i^n$  and the local time steps  $\Delta t_i^n$ . We can use the same corrector-predictor formulation (2.112) where we simply exchange the global time integration limits to the local one. The prediction  $\hat{\mathbf{V}}^{n+1}$  depends only on the local DG data, thus could be calculated independent of the neighbor grid cell data. For the correcting surface integral, data from grid cells sharing a side in two dimensions or a face in three dimensions, which we call henceforth *von Neumann neighbors*, are needed. To ensure the stability, accuracy and consistency of the algorithm the following *evolve condition* has to be satisfied for a grid cell  $Q_i$  at time level  $t_i^n$

$$t_i^{n+1} \leq t_{\text{von Neumann neighbors}}^{n+1} \quad (2.114)$$

As an example we sketch a sequence of four time steps with three adjacent grid cells in Figure 2.2 and start from a common time level  $t^0 = 0$ . After the determination of the local time steps, which are assumed to be different in our example, the space-time predictions are calculated in each element. After this step the degrees of freedom  $\hat{\mathbf{V}}^{n+1}$  are used as a container variable and are subsequently updated with contributions from the surface integrals. When all contributions are gathered, the prediction is completely corrected and becomes the new DG solution  $\hat{\mathbf{U}}^{n+1}$ . A grid cell can only update when the evolve condition (2.114) is satisfied. In our example, the only grid cell satisfying this condition is  $Q_2$ . So  $Q_2$  is evolved to  $t_2^1$ . To do so, the flux contributions at the right and left cell interface have to be computed and added to  $\hat{\mathbf{V}}_2^1$ . The flux integrals are calculated from  $t = t_2^0$  to  $t = t_2^1$  at the right interface  $\partial Q_{2+\frac{1}{2}}$  and the left interface  $\partial Q_{2-\frac{1}{2}}$ . The arguments for the numerical flux functions at time Gaussian points are obtained from the left and right space-time prediction.



**Figure 2.2:** Sequence of steps 1-4 of a computation with 3 different elements and local time-stepping

In order to get a conservative and efficient scheme, the flux contributions computed for the evolution of  $Q_2$  are added to  $\hat{V}_2^1$  and simultaneously to the corresponding neighbors  $\hat{V}_1^1$  and  $\hat{V}_3^1$ . If both flux contributions are added to  $\hat{V}_2^1$ , the update is completed and the DOF of  $Q_2$  at the new time level  $t_2^1$  are known. We can now proceed as in the first time step and construct a new space-time prediction. We are then in the situation sketched in the upper right corner of Figure 2.2. Now  $Q_1$  satisfies the evolve condition (2.114) and can be advanced to  $t_1^1$ . We note that a flux contribution from the evolution of  $Q_2$  has already been added to  $\hat{V}_1^1$ . Consequently only the missing flux contributions, which are sketched in the lower left corner of Figure 2.2, have to be added to the  $\hat{V}_1^1$  in order to get the  $\hat{U}_1^1$ . Namely, on the interface  $\partial Q_{1+\frac{1}{2}}$ , the flux integral has

to be computed with a quadrature formula from  $t_2^1$  to  $t_1^1$ . As before, the flux integral computed on this shared interface is not only added to  $\hat{V}_1^1$ , but also to the  $\hat{V}_2^2$ . The time interval, for which the flux contribution at the interface shared by an element  $Q_i$  and an adjacent element  $Q_j$  has to be computed when evolving  $\hat{V}_i^{n+1}$  to  $\hat{U}_i^{n+1}$ , is generally

$$[t_{ij}^*, t_i^{n+1}] = [\max(t_i^n, t_j^n), t_i^{n+1}] \quad (2.115)$$

In this manner, the algorithm continues by searching for elements satisfying the evolve condition (2.114). So all elements are evolved in a suitable order by evaluating the different terms of the right hand side of equation (2.112) for each element in an effective order. At each time, the interface fluxes are defined uniquely for both adjacent elements, making the scheme exactly conservative.

### 2.5.3 Discussion

The presented time integration yields an efficient solution method for unsteady 'advection dominated' problems and an interesting alternative to implicit or implicit/explicit time discretizations. The property 'advection dominated' depends heavily on the underlying problem. Focusing the discussion on the compressible Navier-Stokes equation, we have two effects which penalize our explicit time integration

- viscosity: if viscous physics dominate, the time step behaves like  $\Delta t \simeq \Delta x^2 / (2p + 1)^2$  causing a large number of time steps and thus high computational costs,
- compressibility: the maximum advection eigenvalue  $\lambda_{max}^A$  is composed of the flow speed magnitude and the speed of sound. If one is not interested in the propagation of the acoustic waves, the physical meaningful time step is determined by the flow speed. The ratio of flow speed and speed of sound is denoted by the Mach number  $Ma$ . The Mach number is used to characterize the compressibility of the flow, where low Mach numbers relate to low compressibility (a Mach number equal zero corresponds to the incompressible limit). For high Mach numbers the maximum advection eigenvalue is about the magnitude of the flow speed, yielding a explicit time step restriction in the range of the physical meaningful time step. For low Mach number flows, the eigenvalue is dominated by the speed of sound, yielding a time step which could be significant lower than the physical time step. Generally, the efficiency of an explicit time

discretization depends on the disparity of the problem inherent different time scales, namely the stiffness of the problem.

Thus, the presented time integration method and its application to the unsteady compressible Navier-Stokes equations is best suited for high Reynolds number and high Mach number flows. For problems with 'stiff' regions it is possible to combine our explicit discontinuous Galerkin method with an implicit space-time discontinuous Galerkin method [50], yielding an *IMEX* space-time discontinuous Galerkin scheme [54]. If we compare the efficiency of the CERKG method with the classical Cauchy-Kovalevskaya based approach, it is worth noting that for the construction of an approximative space-time solution for the three dimensional compressible Navier-Stokes equations the latter is more efficient, especially for high time orders. However, the main building blocks of the Galerkin based approach with the nodal type integration introduced above are matrix vector multiplications. It is possible to bring the efficiency of this new approach on the same level, if optimized matrix-vector multiplication routines are available. An advantage of the CERKG approach is that for non-linear problems the predictor corrector formulation presented in subsection 2.5.2.2 is as accurate as the standard formulation (2.108), whereas the predictor corrector formulation with a Cauchy-Kovalevskaya based predictor loses accuracy up to one order of magnitude in comparison. This loss in accuracy outweighs the efficiency gain of the predictor corrector formulation for the Cauchy-Kovalevskaya based predictor, but brings a performance increase of about 30% for the CERKG approach compared to the standard formulation (2.108). It is worth mentioning that another advantage of the CERKG approach is that it could be directly applied to every set of partial differential equations, as we only need evaluations of the fluxes at the nodal points, whereas a Cauchy-Kovalevskaya based approach requires the construction of a new Cauchy-Kovalevskaya procedure for every new set of partial differential equations.



## 3 Computations

### 3.1 Validation and Proof of Concept

#### 3.1.1 Accuracy and Applicability of the Polymorphic Nodal Elements

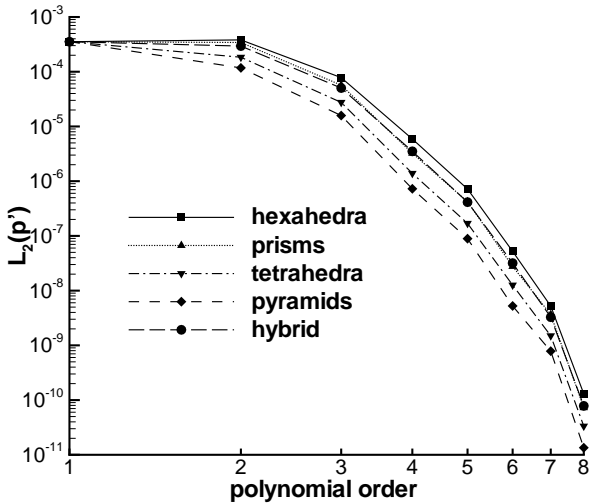
In a first step, the high order accuracy of the developed polynomial data representation is verified. We can either perform simple interpolation and  $L_2$  projection tests or directly use the polynomial representation framework in a discontinuous Galerkin based solver. We choose here the latter possibility. However, to neglect variational crimes we start with a simple linear problem - the linearized Euler equations, listed in Appendix D. For a linear problem the presented method with its approximation reduces to an exact discontinuous Galerkin scheme. To solve the characteristic wave propagation problem described in Appendix D, the hybrid grid detailed in Appendix C is chosen with periodic boundary conditions. In Table 3.1 the experimental order of convergence (EOC) for this test is plotted, where we choose as a representative setup an even and odd polynomial degree  $p \in \{3, 4\}$ . The results clearly indicate that

$n_h$	Nb cells	Nb DOF	$L_2(p')$	EOC	Nb DOF	$L_2(p')$	EOC
		$p = 3$			$p = 4$		
1	21	420	$5,03E - 5$	-	9.408	$3,51E - 6$	-
2	168	3360	$2,21E - 6$	4,5	75.264	$1,22E - 7$	4,8
3	567	11.340	$4,22E - 7$	4,1	19.845	$1,68E - 8$	4,9
4	1344	26.880	$1,22E - 7$	4,1	47.040	$4,06E - 9$	4,9

**Table 3.1:** Experimental order of convergence for  $p = 3$  and  $p = 4$ .

the optimal experimental order of convergence is achieved. As expected, for the linear problem the results do not change when we increase the interpolation order  $p_L$  for the approximation of the integrals or when we change the recursion parameters  $\ell_{2D}$  and  $\ell_{3D}$ , as in this special case the polynomial interpolation is exact. To further investigate the behavior of the discretization for higher polynomial approximations, five configurations were tested. In the first configuration the computational domain  $\Omega := [0; 2]^3$  is split into  $2^3$  hexahedral grid

cells. We plot in Figure 3.1 the  $L_2$  error norm of the pressure  $p'$  for polynomial order  $p = 1$  up to  $p = 8$  evaluated at  $t_{end} = 20$ . For the next configurations the hexahedral base grid was further split into tetrahedra, prisms or pyramids, according to Figure C.1, resulting in 48, 16 and 48 grid cells, respectively. In the last configuration the hybrid grid C with  $n = 1$  was used, resulting in 21 grid cells. We note that for the first four configurations the time steps do not



**Figure 3.1:** Double logarithmic plot of  $L_2$  error versus the polynomial order for different element types and grids.

differ over the computational domain, thus the local time stepping algorithm reduces to a global time stepping scheme. But for configuration five, due to the different grid cell types and their different in-spheres, the scheme runs in local time stepping modus. These results confirm the exponential convergence of the presented scheme with respect to  $p$ -refinement for smooth problems.

In a next step a convergence study for the compressible Navier-Stokes equations is performed, using the hybrid grid and the analytical solution described in Appendices C and F, respectively. For our test we choose the coefficients



$\kappa = 1.4$ ,  $Pr = 0.72$ ,  $\mu = 0.0001$ ,  $R = 287.14$  and  $\alpha = 0.5$ ,  $\omega = 10$ ,  $k = \pi$  with the dimension of the problem  $d = 3$ . The results of this numerical investigations are listed in Table 3.2 for  $p = 4$  and  $p = 5$  with  $\ell = (0, 0)$ , where we used  $t_{end} = 1$  and periodic boundary conditions. The results indicate that the optimal order of convergence  $EOC = p + 1$  for  $p$  odd and even is achieved, evaluating the presented numerical flux approximation for the viscous terms of the compressible Navier-Stokes equation. Up to now, the influence of the

$n_h$	Nb cells	$p = 4$			$p = 5$		
		Nb DOF	$L_2(\rho e)$	EOC	Nb DOF	$L_2(\rho e)$	EOC
2	168	5.880	$6,13E-3$	-	9.408	$3,80E-3$	-
4	1344	47.040	$1,91E-4$	5,0	75.264	$9,36E-5$	5,3
8	10752	376.320	$4,32E-6$	5,5	602.112	$1,54E-6$	5,9
16	86016	3.010.560	$1,22E-7$	5,1	4.816.896	$2,38E-8$	6,0

**Table 3.2:** Experimental order of convergence for  $p = 4$  and  $p = 5$  with  $\ell = (0, 0)$  and  $t_{end} = 1.0$ .

recursion parameter and thus the influence of non-linear flux functions on the accuracy of the nodal approximation has been neglected. To investigate the dependence of the solution quality on the choice of the recursion parameters  $\ell = (\ell_{2D}, \ell_{3D})$  and the interpolation order  $p_L$ , the homogeneous Euler equation with an analytical solution described in Appendix E are chosen. The results of numerical tests with a  $p = 6$  approximation for different parameters  $\ell$  and/or different interpolation orders  $p_L$  are listed in Tables 3.3-3.6. The general observation is that if we increase the number of interpolation points, the error norm decreases and the CPU time per element update (EU) increases. We also compared the nodal integration to the standard Gaussian integration, where we choose  $7^3 = 343$  tensor product Jacobi Gauss points for the volume integrals and  $7^2 = 49$  tensor product Jacobi Gauss points for each of the surface integrals. The results clearly show that it is possible to drastically reduce the evaluation points without losing accuracy.

At the end of this section preliminary results for a DG discretization with polygonal meshes are shown to demonstrate that the concept of polymorphic nodal elements can be used for non-standard elements as well. We propose to apply the recursion based algorithm introduced in section 2.1 to define efficient sets of interpolation points for polygonal grid cells. Numerical investigations indicate that for a general grid cell the shape dependent parameter  $p^*$ , which is the maximal possible interpolation order with surface points only, has the value

Int order $p_L$ and $\ell$	Nb Int points	$L_2(\rho)$	CPU/EU
$p_L = 6, \ell = (0, 0)$	117	$1,9654E - 05$	100%
$p_L = 6, \ell = (1, 0)$	124	$1,7455E - 05$	107%
$p_L = 6, \ell = (0, 1)$	147	$1,8112E - 05$	120%
$p_L = 6, \ell = (1, 1)$	154	$1,6055E - 05$	121%
$p_L = 6, \ell = (2, 0)$	136	$1,7399E - 05$	110%
$p_L = 6, \ell = (2, 1)$	166	$1,5832E - 05$	125%
$p_L = 7, \ell = (0, 0)$	160	$1,7586E - 05$	127%
$p_L = 8, \ell = (0, 0)$	214	$1,6336E - 05$	154%
$p_L = 7, \ell = (4, 2)$	512	$1,4770E - 05$	255%
Gauss Legendre points	637	$1,4665E - 05$	403%

**Table 3.3:** Results for different types of integration points for the hexahedra with  $p = 6$ . The domain  $\Omega$  is subdivided into 8 hexahedra.

Int order $p_L$ and $\ell$	Nb Int points	$L_2(\rho)$	CPUEU
$p_L = 6, \ell = (0, 0)$	98	$2,8744E - 04$	100%
$p_L = 6, \ell = (1, 0)$	106	$2,8256E - 04$	109%
$p_L = 6, \ell = (0, 1)$	103	$1,7078E - 04$	107%
$p_L = 6, \ell = (1, 1)$	111	$1,6332E - 04$	110%
$p_L = 7, \ell = (0, 0)$	138	$2,7298E - 04$	127%
$p_L = 8, \ell = (0, 0)$	187	$1,5537E - 04$	181%
$p_L = 7, \ell = (1, 1)$	159	$1,0978E - 04$	136%
Gauss Jacobi points	588	$9,8771E - 05$	425%

**Table 3.4:** Results for different types of integration points for  $p = 6$  pyramids. The domain  $\Omega$  is subdivided into 6 pyramids.

*number of sides minus one*, which we choose for all grid cell types discussed in this thesis. Starting from a mesh with triangles the corresponding dual mesh is constructed and used as the polygonal mesh, Figure 3.2a. The primal triangle mesh is no longer needed as it is only used to construct the dual mesh. The resulting polygonal mesh contains elements with four sides up to elements with seven sides. For the distribution of the interpolation points two different strategies are used for a polynomial approximation with degree  $p = 3$ . For the first strategy we directly use the recursion algorithm (2.16) with a fixed recursion parameter  $\ell_{2D}$  for all elements. If we choose  $\ell_{2D} = 0$ , test configuration A shown in Figure 3.2b, the resulting interpolation grid is

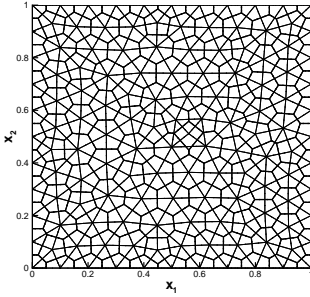
Int. order $p_L$ and $\ell$	Nb Int. points	$L_2(\rho)$	CPU/EU
$p_L = 6, \ell = (0, 0)$	101	$1,4853E - 05$	100%
$p_L = 6, \ell = (1, 0)$	110	$1,4235E - 05$	109%
$p_L = 6, \ell = (0, 1)$	116	$1,2260E - 05$	114%
$p_L = 6, \ell = (1, 1)$	125	$1,2250E - 05$	118%
$p_L = 7, \ell = (0, 0)$	141	$1,4210E - 05$	127%
$p_L = 8, \ell = (0, 0)$	188	$1,2925E - 05$	154%
$p_L = 7, \ell = (0, 1)$	165	$1,1562E - 05$	141%
Gauss Jacobi points	588	$1,1006E - 05$	424%

**Table 3.5:** Results for different types of integration points for  $p = 6$  prisms. The domain  $\Omega$  is subdivided into 8 hexahedra which are further subdivided into 2 prisms, yielding 16 grid cells.

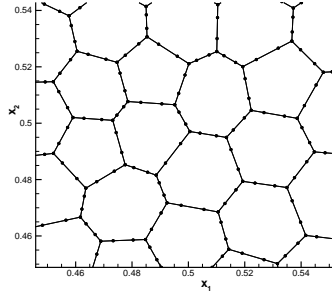
Int. order $p_L$ and $\ell$	Nb Int. points	$L_2(\rho)$	CPU/EU
$p_L = 6, \ell = (0, 0)$	84	$1,414E - 04$	100%
$p_L = 7, \ell = (0, 0)$	120	$1,4386E - 04$	113%
$p_L = 8, \ell = (0, 0)$	165	$1,3945E - 04$	135%
Gauss Jacobi points	539	$1,3790E - 04$	399%

**Table 3.6:** Results for different types of integration points for  $p = 6$  tetrahedra. The domain  $\Omega$  is subdivided into 6 tetrahedra.

only distributed at grid cell boundaries, as all grid cells have at least 4 sides. Similar to the discussion above it is favorable for non-linear problems to use more interpolation points, i.e. increasing the recursion parameter  $\ell_{2D}$ . The test configuration B shown in Figure 3.3a is constructed by choosing  $\ell_{2D} = 3$  for all grid cells. In this extreme case where quadrilaterals (4 sides, 3 recursive defined interior point layers) and heptagons (7 sides, 0 recursive defined interior point layers) arise, the resulting point distribution is non-uniform and seems to be not well suited. To avoid this uneven interpolation point distribution, our second strategy is to *fix* the number of recursions for every grid cell type, thus introducing the recursion parameter  $\ell_{2D}$  independently for every grid cell type. In Figure 3.3b the interpolation grid for a fixed recursion number  $r_{max} = 1$  (test configuration C) with a second order inner point distribution is shown, corresponding to the parameter  $\ell_{2D} = 2$  for quadrilaterals up to  $\ell_{2D} = 5$  for heptagons. To validate this discretizations the reduced two dimensional version

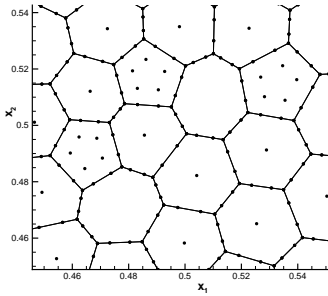


(a) primal and dual mesh

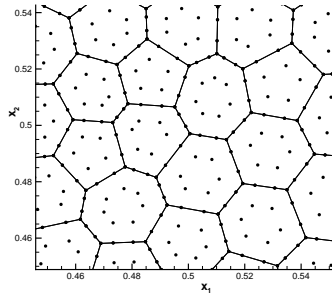


(b) configuration A

**Figure 3.2:** Primal and dual mesh ( $h = 0.1$ ) and detailed view of the interpolation grid ( $h = 0.025$ ) with  $p = 3$  ( $\pi_{2D} = 0$ ) interpolation.



(a) configuration B



(b) configuration C

**Figure 3.3:** Detailed view of the interpolation grid ( $h = 0.025$ ) for  $p = 3$  approximation with  $\ell_{2D} = 3$  or  $r_{max} = 1$ .

of the previous example for the Navier-Stokes equations is used, with the same parameters, except the parameter  $k$  which we change from  $\pi$  to  $2\pi$  and the dimension  $d$  from 3 to 2. For the grid refinement, four different regular triangle grids with typical mesh size  $h$  are constructed and then converted to polygonal meshes, similar to Figure 3.2a. The pre-computation of the surface and volume integral matrices is done on sub triangles with standard Gaussian integration. In Table 3.7 the results for configuration A and the results for the reference computation on the primal triangular grid with  $t_{end} = 0.5$  and exact boundary conditions are shown. We notice first that the expected order of convergence is achieved. Considering efficiency, the results on the primal mesh are more accurate, whereas the CPU time for configuration A is  $t_{CPU} = 378s$  and the CPU time for the primal configuration is  $t_{CPU} = 594s$ . The reasons for the CPU time advantage is that the resulting polygonal configuration has only about half the DOF and allows larger (explicit) time steps. To account for

$h$	Nb cells	$L_2(\rho_e)$	$EOC$	Nb cells	$L_2(\rho_e)$	$EOC$
	triangular configuration			configuration A		
0,2	62	$2,44E-3$	-	42	$1,28E-2$	-
0,1	226	$1,92E-4$	3,7	134	$1,31E-3$	3,3
0,05	896	$1,07E-5$	4,3	489	$7,16E-5$	4,2
0,025	3595	$6,42E-7$	4,1	1878	$4,77E-6$	3,9

**Table 3.7:** Experimental order of convergence for  $p = 3$  (10 DOF per grid cell) for reference test on primal triangular mesh and for test configuration A.

the non-linearity of the Navier-Stokes fluxes, computations with configuration B and C are performed and corresponding results are listed in Table 3.8. We notice that the accuracy of the solution is improved, approaching the quality of the primal configuration solution. As expected, the results of configuration C are more accurate compared to the results of configuration B. Considering the efficiency of the computations, the CPU time for test B is  $t_{CPU} = 380s$  and for test C  $t_{CPU} = 398s$ . The application to general polygonal meshes indicates a large potential compared to traditional triangular meshes with respect to computational efficiency. However, the nodal integration approach introduced in section 2.3 enhances efficiency already for standard shaped grid cells. The above presented evaluation of the scheme with different recursion parameters for the non-linear Euler equations reveal a four fold reduction in CPU time

h	Nb cells	$L_2(\rho e)$	$EOC$	$L_2(\rho e)$	$EOC$
		configuration B		configuration C	
0,2	42	$9,55E-3$	-	$5,17E-3$	-
0,1	134	$7,22E-4$	3,7	$4,25E-4$	3,6
0,05	489	$3,38E-5$	4,4	$2,64E-5$	4,0
0,025	1878	$1,84E-6$	4,2	$1,64E-6$	4,0

**Table 3.8:** Experimental order of convergence for  $p = 3$  (10 DOF per grid cell) for test configuration B and C.

without losing accuracy.

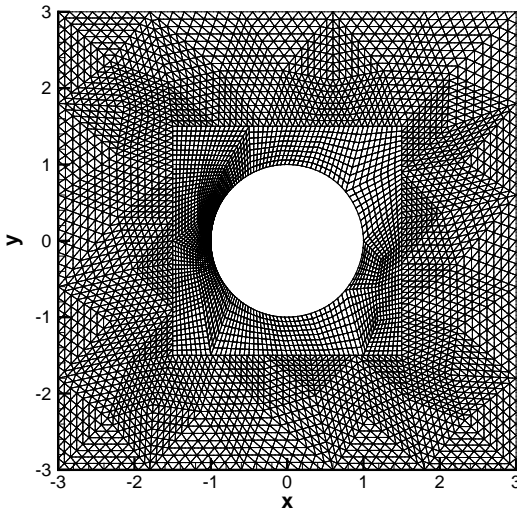
### 3.1.2 Efficiency of the Numerical Method

To classify the efficiency of the presented discretization for the three-dimensional compressible Navier-Stokes, the average CPU time per element update and per degree of freedom (CPU/EU/DOF) is evaluated. We list the results for  $p = 6$  trial functions (84 DOF/Element) and  $p_t = 2$  time polynomial in Table 3.9. Based on the above presented investigations, we choose for every grid cell type the most efficient combination (in terms of accuracy versus CPU time) of the parameters  $\ell$  and the interpolation order  $p_L$ . All CPU times were measured on one processor of an Intel Xeon Dual Core CPU with 2.66GHz. An equivalent measurement for a 6th order compact finite difference scheme with 4th order Runge-Kutta time integration, see Babucke et al. [6] for details, on the same CPU yields  $\sim 56\mu s$  and  $\sim 2\mu s$  on a SX8 vector processor.

Int order $p_L$ and $\ell$	Element type	CPU/EU/DOF [ $\mu s$ ]
$p_L = 6, \ell = (1, 1)$	hexahedron	39, 9
$p_L = 7, \ell = (1, 1)$	pyramid	43, 1
$p_L = 6, \ell = (0, 1)$	prism	31, 5
$p_L = 6, \ell = (0, 0)$	tetrahedron	27, 7

**Table 3.9:** CPU times for the three dimensional compressible Navier-Stokes equations with  $p = 6$ .

The evaluation of the local-time stepping influence is a difficult task, as the results strongly depend on the underlying problem. For a theoretical consideration of the potential we refer to Dumbser et al. [30]. To get an indication of the potential for a concrete example, a comparison with a global time stepping variant of the presented scheme is undertaken. We choose again the analytical solution from Appendix F. The parameters are  $\mu = 0.01$ ,  $\kappa = 1.4$ ,  $Pr = 0.72$  and  $k = 2\pi$ ,  $\omega = 0.5$ ,  $\gamma = 1.0$  and the dimension  $d = 2$ . The computational domain is a  $[-3; 3] \times [-3; 3]$  square, where a cylinder centered at the origin with radius equal to one is subtracted. We furthermore use exact boundary conditions and  $t_{end} = 0.5$ . In Figure 3.4 the grid with triangles, quadrilaterals, hanging nodes and curved boundary elements is shown. We choose polynomial



**Figure 3.4:** Computational grid with 8448 grid cells.

degrees from  $p = 1$  to  $p = 5$  for our approximation space and  $\ell_{2D} = 1$  for quadrilaterals and  $\ell_{2D} = 0$  for triangles. The results of this computations are

listed in Table 3.10. Although the results of the global time stepping scheme are slightly more accurate the comparison of the CPU times verifies that the local time stepping scheme is far more efficient.

$p$	Nb DOF	$L_2(U_4)$ (LTS)	$L_2(U_4)$ (GTS)	$\frac{CPU_{GTS}}{CPU_{LTS}}$
1	25344	$1,77E-02$	$1,70E-02$	<b>35,9</b>
2	50688	$4,82E-04$	$4,73E-04$	<b>37,7</b>
3	84480	$2,11E-05$	$2,01E-05$	<b>38,1</b>
4	126720	$4,97E-07$	$4,89E-07$	<b>38,9</b>
5	177408	$2,15E-08$	$2,07E-08$	<b>40,1</b>

**Table 3.10:** Comparison of local time stepping (LTS) and global time stepping (GTS).

In the last part of this subsection the high performance computing properties of the scheme are investigated. The presented discontinuous Galerkin discretization is fully MPI parallelized, where the crucial part for efficiency is the incorporation of dynamic load balancing, as the local time stepping algorithm naturally yields an inhomogeneous distribution of work load. For an extensive investigation of the parallel performance of the scheme combined with local  $hp$ -refinement,  $p$ -adaptation, dynamic load balancing and of course local time stepping, we refer to the PhD thesis of Frieder Lörcher [54]. To demonstrate the potential of this explicit discontinuous Galerkin scheme and give an indication of the parallelizability, a test case for the three dimensional Navier-Stokes equations with a source term and the exact solution given in Appendix F is chosen. We use the parameters  $\mu = 0.0001$ ,  $\kappa = 1.4$ ,  $Pr = 0.72$ ,  $R = 287.14$  and  $\gamma = 0.5$ ,  $\omega = 10.0$ ,  $k = \pi$ ,  $d = 3$ . This problem is periodic in a cubic domain with length 1. Thus when we choose for each processor an unit cube, the work load of the problem scales constant with the number of processors when periodic boundary conditions are supposed. All computations are performed on the Altix cluster of the Höchstleistungs-Rechenzentrum München (HLRB). In Table 3.11 the scale-up for a computation with  $14^3$   $p = 5$  hexahedra per unit cube is shown. We note that in the case  $n_P = 1$  no communication takes place. Defining the scale-up as  $\mathcal{S}(n_P) = \frac{t_{n_P=1}}{t_{n_P}}$  we notice that even for 2197 processors the efficiency is at an excellent rate of 94.5%.

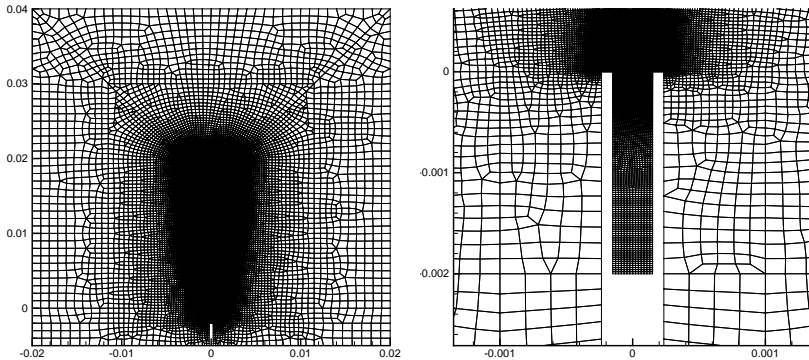
In the last example the efficiency of the scheme compared to an existing commercial CFD tool, namely Ansys CFX, is investigated. We simulate the flow of



$n_P$	1 ( $1^3$ )	8 ( $2^3$ )	64 ( $4^3$ )	512 ( $8^3$ )	2197 ( $13^3$ )
wallclock time[s]	803	816	818	834	851
efficiency $\mathcal{S}(n_P)$ [%]	100	98.5	98.2	96.4	94.5

**Table 3.11:** Results of the scale-up test with constant load per processor (153664 DOF)

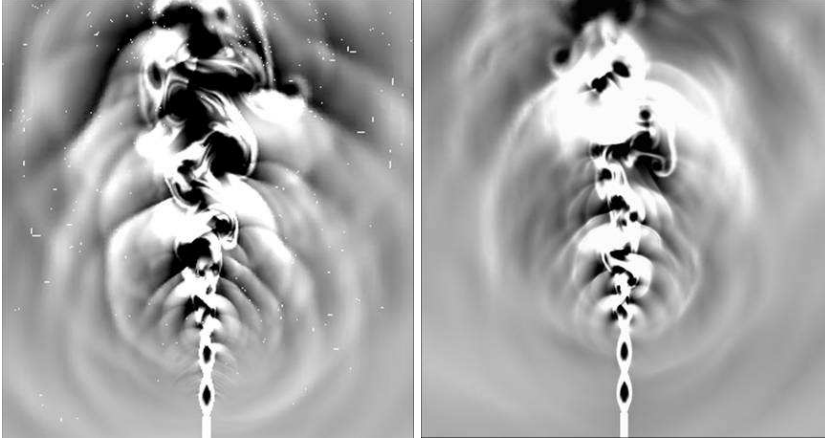
a single jet and are furthermore interested in the aeroacoustics. We plot in Figure 3.5 the geometrical setup and the unstructured mesh, consisting of 110,000 unstructured quadrilateral elements. We initialize the computation with air at one bar pressure with zero velocity. At the inlet the inflow has the velocity  $v = 500 \text{ m/s}$ , the temperature  $T = 220 \text{ K}$  and the pressure  $p = 2 \text{ bar}$ , which corresponds to the Mach number  $Ma = 1.4$  and Reynolds number  $Re_d \approx 40,000$  where  $d = 0.0003 \text{ m}$  is the inlet diameter. We plot the density contours from



**Figure 3.5:** Computational setup for the single jet example. All length scales are in meters.

$\rho \in [0.9; 1.3] \text{ kg/m}^3$  in Figure 3.6, which we obtained from a computations with  $p = 1$  polynomials (330,000 DOF). The corresponding CPU time using one processor of an Intel Dual-Core Xeon 2.6GHz is  $t_{CPU} = 18 \text{ h}$  for a simulation time of  $\Delta t = 1 \text{ ms}$ . We can clearly see the shock cells forming directly

after the inlet and a roll up in vortices of the jet. These two phenomena are sources for acoustic waves, which are propagated into the farfield. We plot in



**Figure 3.6:** Density distribution calculated with  $p = 1$  DG scheme (left) and CFX (right).

Figure 3.6 the corresponding results using Ansys CFX, which are kindly provided by Olaf Schönrock [70]. Ansys CFX uses a finite volume discretization of first up to second order in space and a first order implicit time discretization. The CPU time using one processor of an AMD Dual-Core Opteron 2218 2.6GHz is  $t_{CPU} = 252 h$  for a simulation time of  $\Delta t = 1 ms$ . Comparing the quality of the solution, especially the propagation and the amplitudes of the acoustic waves and the vortex structures, it is clear that the DG solution is superior.

#### 3.1.3 Unresolved Problems

In this subsection two main problems of high order finite element (discontinuous Galerkin) approximations are discussed. One is shock capturing and the other is high order representation of geometry boundaries. We note that there are several other unresolved issues, but this two problems are specific and

unique problems which should be solved to allow a wider field of applications of unstructured high order finite element (discontinuous Galerkin) schemes.

### 3.1.3.1 Shock Capturing

When solving compressible fluid flow it is possible that for advection dominated problems (large Mach numbers,  $Ma \geq 1$ ) discontinuities, namely shocks, develop. Polynomial approximations of discontinuities lead to the well known Gibbs phenomenon. A consequence is the appearance of oscillations around the discontinuity. These oscillations can cause unphysical solutions, such as negative density and pressure, and consequently non-hyperbolic Jacobi advection matrices, which causes the code to abort. Another problem is that these oscillations and the associated errors propagate through the computational domain during the computation and pollute the whole solution accuracy. To control the oscillations, several shock capturing techniques, borrowing from the finite volume framework are developed. Essentially, shock capturing for discontinuous Galerkin is split into two parts. First part is to find the so called troubled cells, cells which contain (or are supposed to contain) a discontinuity. After flagging these cells, the discontinuous Galerkin polynomial is replaced in these cells with a weighted essentially non oscillatory (WENO) reconstructed polynomial [85] or a hermite WENO reconstructed polynomial [7]. There are several problems related to this task. First of all, two different schemes have to be implemented. Especially the implementation and development of unstructured high order (H)WENO reconstructions is a non-trivial task. Recently, Dumbser et al. [27, 29] developed a new (Hermite type) reconstruction on unstructured grids, where they incorporate ideas from the modal discontinuous Galerkin framework and thus developed a unified framework for DG and FV schemes. Considering the locality of the (H)WENO limited DG scheme, a high order accurate reconstruction needs a large stencil, which reduces the efficiency of the local time stepping feature and the high parallelizability of the method. A main disadvantage of finite volume type limiters for discontinuous Galerkin schemes is the extreme loss of accuracy when replacing a DG polynomial with a reconstructed polynomial. Even in the case where the reconstruction has the same theoretical order of accuracy, the quality of a finite element type solution, especially its subcell resolution capabilities, and a finite volume type solution on same grids are not comparable [27]. This effect is furthermore increased with increasing approximation order and is *independent* of the solution regularity. We will demonstrate this claim below using the famous double Mach reflection example of Woodward and Collela [82]. This accuracy difference increases the

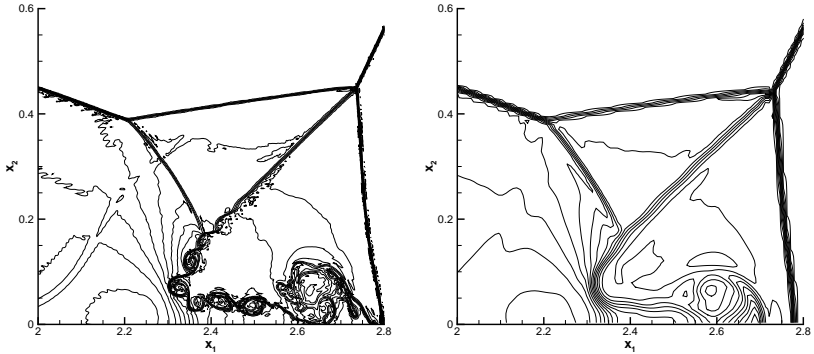
burden on the trouble cell indicator, as the accuracy of a wrong detected cell drastically decreases. To take the high order approximation quality and the corresponding subcell resolution into account, Persson and Peraire [64] introduced a new way of adding artificial viscosity to the problem. They constructed a local resolution based indicator and used it to determine a cell-wise constant viscosity for a Laplacian additional diffusion term. In contrast to the finite element inspired artificial approach of Jaffre et al. [44], where the diffusion is only added in the volume integral, Persson and Peraire propose to use a consistent treatment of the artificial viscosity terms. Numerical tests indicate that a consistent treatment of artificial shock capturing viscosity yields more accurate and more stable results. The main disadvantage of this approach is that the amount of needed viscosity is a priori not known. Too much viscosity decreases the solution accuracy, while insufficient artificial viscosity results in unstable computations. Furthermore the higher the artificial viscosity, the smaller the explicit time step. As long as the distribution of viscosity is concentrated locally in the vicinity of discontinuities, the loss in efficiency is limited due to the local time stepping feature. Persson and Peraire propose to use artificial viscosity which is proportional to the local spatial resolution

$$\nu_{art} \simeq \frac{h}{p+1} \simeq \delta_{shock}, \quad (3.1)$$

where  $\nu_{art}$  is the artificial viscosity,  $\delta_{shock}$  the artificial shock width,  $h$  the diameter and  $p$  the polynomial degree of the considered grid cell. This implies that for high order discontinuous Galerkin polynomials less artificial viscosity is needed for stabilization. However up to now, no parameter free implementations of this approach are known.

To highlight the potential of this new artificial viscosity approach, the famous double Mach reflection example of Woodward and Collela [82] is considered. Here, we choose the cartesian setup similar to [68]. The computational domain  $\Omega = [0, 4] \times [0, 1]$  is split into a regular cartesian grid with a mesh size of  $h$ . The initial solution is a right-moving Mach 10 shock starting at position  $x_1 = 1/6$  and  $x_2 = 0$  spanning a  $60^\circ$  angle with the  $x_1$ -axis. A reflecting wall boundary condition starts at  $x_1 = 1/6$  for  $x_2 = 0$ . The other boundary is prescribed with the exact moving (post) shock conditions. The characteristic part of this solution is the double Mach stem region and the shear layer instability. We plot in the right part of Figure 3.7 this region obtained with a classic WENO FV scheme of fifth order accuracy and  $h = 1/120$ , which was kindly provided by Jens Uitzmann [77]. In the left part of Figure 3.7, a DG solution using  $p = 4$  polynomials and  $h = 1/120$  with the Persson and Peraire artificial

viscosity approach is shown. We note that when using a WENO-FV based limiter for the DG scheme, the best solution in the flagged troubled cells is comparable to the pure WENO-FV solution. Thus comparing the numerical shock width and especially the shear layer, it is obvious that the quality of the DG solution is superior even for discontinuities. To get an impression

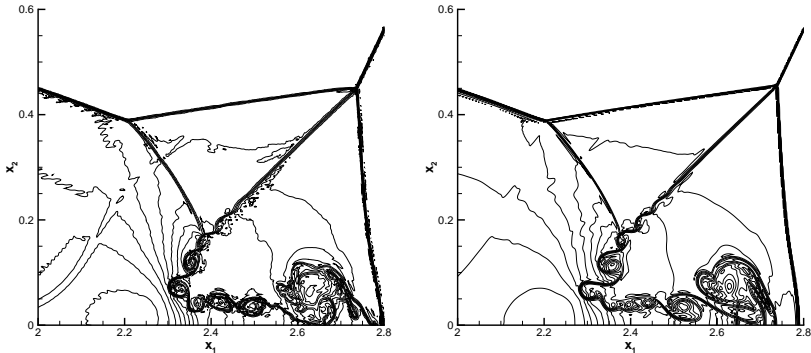


**Figure 3.7:** 30 equally spaced density contour lines from  $\rho = 1.5 \dots 21.5$ . left: DG scheme with  $h = 1/120$  and  $p = 4$ . right: 5th order WENO-FV scheme with  $h = 1/120$ .

of the capabilities of the different approaches another calculation using the WENO-FV scheme on a refined mesh ( $h = 1/480$ ) is done. In Figure 3.8 the resulting double Mach stem region is plotted and compared with the former DG result. We can clearly see now that both solutions have approximately the same shock thickness and approximately the same details in the shear layer roll up. Summarizing, the problem is that no accurate and problem independent (parameter free) shock capturing techniques exist, which makes the simulation of flow problems containing a discontinuity a time consuming and difficult task.

### 3.1.3.2 High Order Geometric Boundary Representation

Bassi and Rebay [9] were the first who investigated the importance of high order accurate boundary geometry representation within the discontinuous Galerkin framework. They found out that for the simulation of inviscid Euler equations



**Figure 3.8:** 30 equally spaced density contour lines from  $\rho = 1.5 \dots 21.5$ . left: DG scheme with  $h = 1/120$  and  $p = 4$ . right: 5th order WENO-FV scheme with  $h = 1/480$ .

the approximation of the boundary geometry should be at least *parabolic* (to get a uniform high order approximation of the problem, it is clear that the boundary order should at least be equal to the order of the discontinuous Galerkin discretization). Bassi and Rebay found an example where their discontinuous Galerkin discretization with  $p \geq 1$  and straight sided boundary grid cells converges to the (physically) wrong solution. They also proposed to use just a high order accurate distribution of the normal vectors, without constructing grid cells with curved boundaries. This gives an accurate treatment of the (wall) boundary condition, which seems to be the main effect of higher order boundary representations [51]. Although, using only high order normal vectors without considering curved elements results in a non-conservative scheme. Krivodonova and Berger [51] investigated the non-conservativity for some two dimensional test cases and found no overall negative effect on the accuracy of the solution. It seems that this investigations depend at least on the underlying problem and cannot be generalized. The main problem for both approaches is how to get the normal vectors at the surface evaluation points. Bassi and Rebay proposed to use a spline reconstruction of the boundary using only data from the (boundary) grid. While this is trivial in two dimensions, the extension to three dimension is nearly impossible for complex geometries. Especially

intersecting surfaces, which cause sharp intersection lines (e.g. trailing edge for wings), are extremely difficult to handle as well as geometric singularities. Furthermore there is only few literature which deals with spline reconstruction on arbitrary unstructured grids. Another approach proposed recently by Luo et al. [58] is to incorporate and store a representation of the geometry and use this analytical boundary description to compute the normal vectors. While this seems to be the most accurate way to determine the normal vectors and thus the most promising approach for the future, incorporating and managing the geometry is a non-trivial task with respect to coding and drastically increases the complexity of the boundary approximation. However, if we are interested in solving problems with complex geometries, where unstructured meshing is self-evident, using high order polynomial discontinuous Galerkin approximations and consequently large grid cells to maintain the computational efficiency of our method, a high order accurate boundary treatment is mandatory.

## 3.2 Application

### 3.2.1 Direct Numerical Simulation 2D

#### 3.2.1.1 Boundary Layer Instability

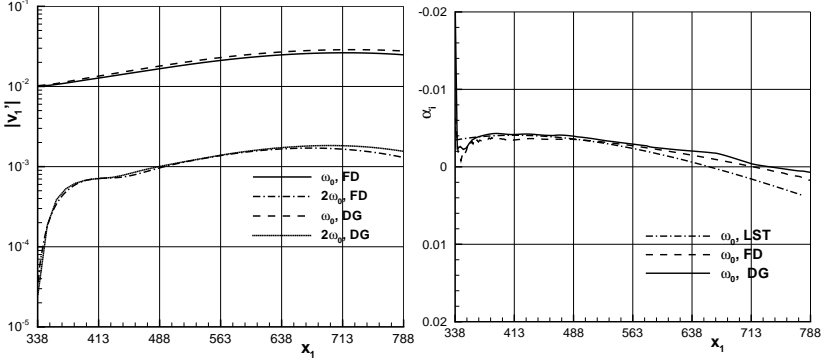
We consider in this example the evolution of a Tollmien-Schlichting wave in a subsonic compressible boundary layer. The computational domain  $\Omega$  extends from  $x_1 = 337$  to  $x_1 = 890$  and  $x_2 = 0$  to  $x_2 = 23$ . We choose subsonic inflow and outflow boundary conditions and at  $x_2 = 0$  isothermal wall conditions with  $T_w = 296[K]$ . The initial solution of the computation is obtained from a similarity solution with Mach number  $Ma_\infty = 0.8$  and  $T_\infty = 280[K]$ . The Reynolds number  $Re := \frac{\rho_\infty v_1 \delta_1}{\mu(T_\infty)} = 1000$ , based on the displacement thickness at the inflow  $\delta_1$ . Using  $\delta_1$  as the reference length, we get  $\delta_1 = 1$  at the inflow and the boundary layer thickness  $\delta_{99} = 2.95$  and  $\delta_{99} = 4.8$  at the inflow and outflow, respectively. The temperature dependence of viscosity  $\mu$  is modeled using Sutherland's law

$$\mu(T) = \mu(T_\infty) T^{3/2} \frac{1 + T_s}{T + T_s}, \quad (3.2)$$

with  $\mu(T_\infty) = 1.735 \cdot 10^{-5} \left[ \frac{kg}{m.s} \right]$  and  $T_s = 110.4[K]$ .

The inflow at  $x_1 = 337$  is superimposed with a forcing term, composed of the eigenfunction of the Tollmien-Schlichting wave with the fundamental frequency  $\omega_0 = 0.0688$ . For a detailed description of the similarity solution and the eigenfunction, we refer to Babucke et al. [5]. The computational domain was subdivided in  $48 \times 22$  regular quadrilaterals and discretized with  $p = 6$  and  $\ell_{2D} = 1$ , resulting in 29568 DOF. The end time of the simulation was set to  $\frac{t_{end}}{T_0} = 37$ , where  $T_0 = \frac{2\pi}{\omega_0} \approx 92$ , to ensure a periodic solution. To analyze our results we apply a discrete Fourier analysis using one period of the forcing frequency  $T_0$  from  $\frac{t}{T_0} = 36$  to  $\frac{t}{T_0} = 37$ . We plot the maximal amplitude of  $v_1$  with respect to  $x_2$  as a function of  $x_1$  in Figure 3.9. For comparison, corresponding results obtained with a 6th order compact finite difference code with  $330 \times 150$  grid points and 4th order Runge Kutta time integration [5] are included, showing good agreement. We furthermore plot the amplification rate  $\alpha_i$  of the velocity  $v_1$  based on the maximal amplitude in Figure 3.9. Again, the result is in good accordance to the reference result [5] and the predictions of the linear stability theory.





**Figure 3.9:** Maximum amplitudes of  $v_1$  (left). Amplification rate  $\alpha_i$  of  $u_1$  based on maximum amplitude (right).

### 3.2.1.2 Plane Mixing Layer

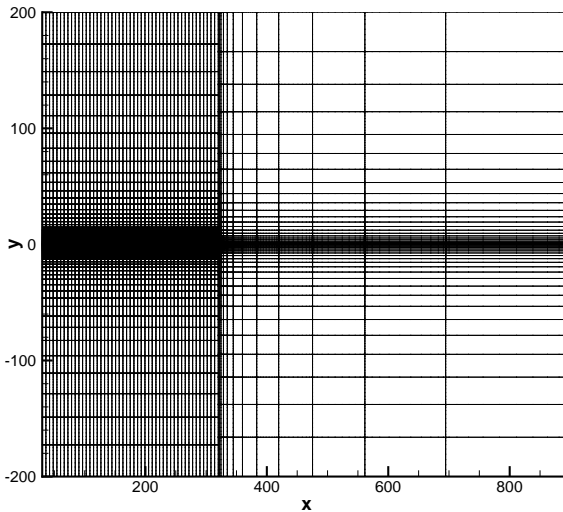
In this subsection, numerical results of a two-dimensional direct numerical simulation of the instability of mixing layer are presented. We adopted the situation simulated by Colonius, Lele, and Moin [24] and later by Babucke, Kloker, and Rist [6]. The initial condition for the mixing layer is the laminar solution to the steady compressible two-dimensional boundary layer equations. The Mach number of the upper and the lower stream are  $Ma_1 = 0.5$  and  $Ma_2 = 0.25$ , respectively. The Reynolds number  $Re = \rho_1 u_1 \delta / \mu = 500$  is based on the vorticity thickness on the inflow

$$\delta(x_0) := \left( \frac{\Delta u}{|\partial u / \partial y|_{max}} \right)_{x_0}, \quad (3.3)$$

which is also used to normalize the length scales. The computational domain starts at  $x_0 = 30$ , which yields  $\delta(x_0) = 1$ , and extends to  $x = 900$  and  $y = \pm 200$ . Velocities are normalized with  $u_1$  and all other quantities by their corresponding values in the upper stream. The flow is forced at the inflow with eigenfunctions found from the spatial viscous linear stability theory, see [6] and references therein for more details. The disturbances are composed of the fundamental frequency  $\omega_0 = 0.6293$  and the three subharmonics. The amplitudes of the

eigenfunctions are normalized by their maximum value of  $u_1$  and then scaled by the amplitude factor 0.002. In accordance to the simulation of Colonius et al. [24], the phase shift is  $\Delta\Theta = -0.028$  for the first,  $\Delta\Theta = 0.141$  for the second and  $\Delta\Theta = 0.391$  radians for the third disturbance.

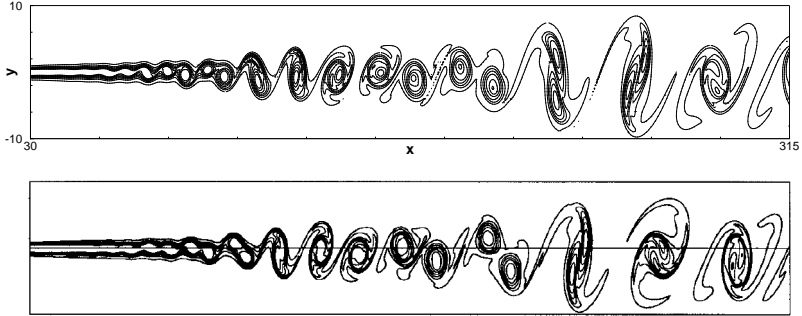
In Figure 3.10 the computational grid is shown. We used trial functions with  $p = 6$  in all grid zones. The grid resolution in the region  $[30; 320] \times [-12; 12]$  is 27 grid cells in  $y$ -direction corresponding to  $\Delta y \in [0.65; 2.75]$  and 230 grid cells in  $x$ -direction corresponding to  $\Delta x \in [0.75; 4]$  resulting in  $\approx 200000$  DOF.



**Figure 3.10:** Computational grid for mixing layer simulation.

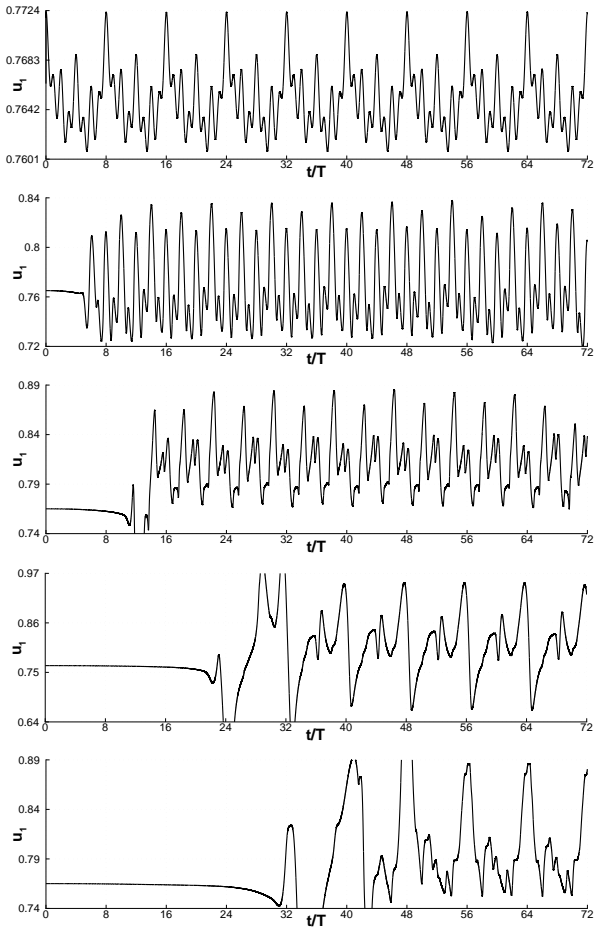
We plot in Figure 3.11 the vorticity contour levels at  $\frac{t}{T_0} = 68$  and included the corresponding results of Colonius et al. [24] for comparison. The numerical results show good agreement. The basic physical mechanism as the roll-up of vortices and the successive pairings of these vortices due to the subharmonic disturbances coincide very well. The streamwise spatial development of the mixing layer is nearly the same. There are some small differences locally, which may be due to different approximation errors or due to small differences in the forcing at the inflow. We refer to [6] for a comparison of the eigenfunctions

from the viscous theory and the ones found from the inviscid theory, which were used in [24]. To prepare for a discrete Fourier analysis in time, we first

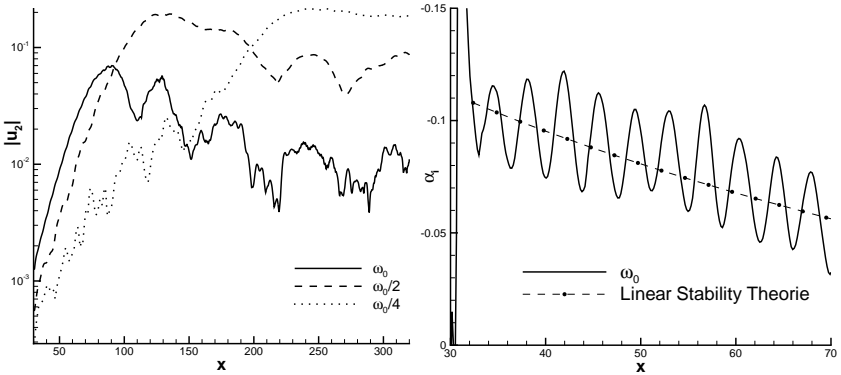


**Figure 3.11:** Mixing Layer. Vorticity contours in near-field mixing region at  $\frac{t}{T} = 68$  from contour level  $-0.26$  to  $0.02$  with increment  $0.04$  are shown. The result from Colonius et al. [24] is shown below.

look at the temporal behavior of the near-field flow region. We plot the time history of the streamwise velocity  $u_1$  at  $y = 0$  and different streamwise positions  $x \in \{30, 75, 130, 230, 315\}$  in Figure 3.12. These results clearly indicate that after  $\frac{t}{T} \approx 48$  the solution is almost periodic. To compare our results with the linear stability theory we apply a discrete Fourier analysis using eight periods of the lowest forced frequency  $\omega_0$  from  $\frac{t}{T} = 64$  to  $\frac{t}{T} = 72$ . In the left plot of Figure 3.13, the maximal amplitude of  $u_2$  with respect to  $y$  as a function of  $x$  is shown. In the first part of the domain, where the magnitude of the amplitudes are small, the growth is nearly exponentially, showing good agreement with linear stability theory. We furthermore plotted the amplification rate  $\alpha_i$  of the velocity  $u_2$  for the  $\omega_0$  mode, based on the maximal amplitude shown in the left picture. Again, the result is in good accordance with linear stability theory.



**Figure 3.12:** Streamwise velocity  $u_1$  as a function of time at the axes  $y = 0$  and at the various streamwise locations  $x = 30, \dots, 315$  (top down).



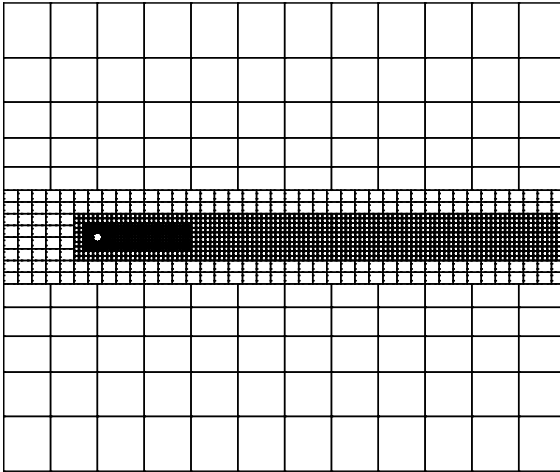
**Figure 3.13:** Maximum amplitudes of  $u_2$  (left). Amplification rate  $\alpha_i$  of  $u_2$  based on maximum amplitude (right).

## 3.2.2 Direct Numerical Simulation 3D

### 3.2.2.1 Laminar Time Periodic Flow Past a Sphere

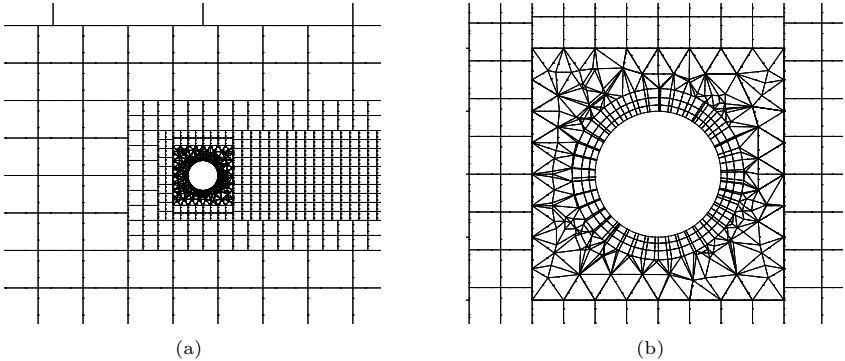
We consider in this example the flow past a sphere with radius  $r = 1$  centered at  $\vec{x}_0 = (0, 0, 0)^T$  and solve the three dimensional unsteady compressible Navier-Stokes equations with free stream Mach number  $Ma_\infty = 0.3$  and Reynolds number  $Re = 300$  based on the diameter of the sphere. We choose the free stream sound speed  $c_\infty = 1$  and set the free stream density to  $\rho_\infty = 1$ . The computational domain is given by  $-20 \leq x_1 \leq 100$  and  $-50 \leq x_2, x_3 \leq 50$ . To fully exploit our discontinuous Galerkin scheme a initial discretization with  $hp$ -refinement is chosen. Based on the previous investigations it is favorable to use a hexahedral mesh. However it shows that it needs an exaggerated effort to build such a mesh, using actual commercial grid generators. In contrast to this, tetrahedra meshes are easy to generate, but may not yield the best performance. To overcome this issue, we exploit the flexibility property of a discontinuous Galerkin based method and introduce a 'block unstructured' approach. We sub-divide the computational domain in several blocks, which we mesh independently, i.e. we do not force that the grids in adjacent blocks are conforming. This allows us to define blocks with pure hexahedral meshing and blocks where we use tetrahedral grids. To simplify the meshing, we defined

a block centered around the sphere with edge length of 4, which we discretized with curved prisms in the boundary layer of the sphere and tetrahedra elsewhere with a typical mesh length of  $h \approx 0.1$ . To capture the von Kármán vortex street a hexahedral block with typical mesh length of  $h = 0.5$  is build in the wake domain of the sphere. The nearfield is discretized with hexahedral blocks with a typical mesh length  $h \approx 3$  and the farfield with hexahedral blocks with typical mesh length  $h \approx 10$ . The whole computational mesh contains 30,250 grid cells. We plot in Figure 3.14 a projection of the grid onto the  $x_1 - x_3$ -plane at  $x_2 = 0$ , which clearly shows the non-conforming block unstructured setting. In Figure 3.15 two different zoomed-in views of this projected grid



**Figure 3.14:** Projection of the computational grid on the  $x_1 - x_3$ -plane at  $x_2 = 0$ .

are plotted. We note that an intersection of arbitrary unstructured blocks generates polygonal faces with number of sides up to eight. For the construction of the point sets for grid cells with such faces we can still use the construction guideline introduced in subsection 2.1.2.2. For the definition of the surface point sets we do not only use the point sets defined for triangles and quadrilaterals,



**Figure 3.15:** Zoomed views of the projected sphere example grid.

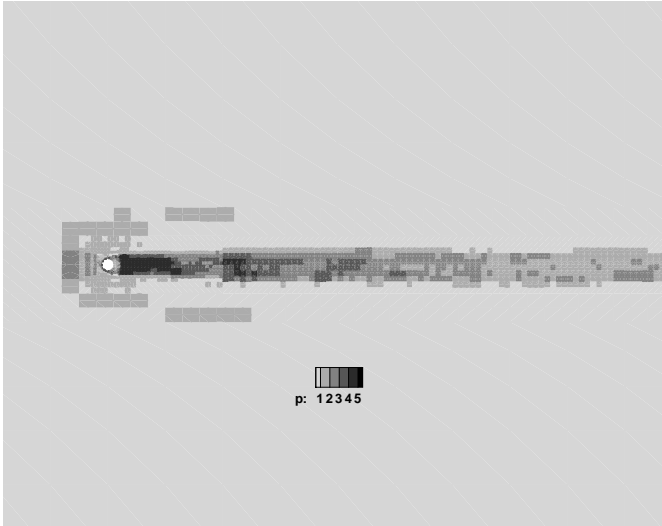
but also the point sets defined for polygonals used in subsection 3.1.1. To complete our initial  $hp$ -refined discretization an approximation with  $p = 3$  is chosen everywhere, except in the farfield, where we choose linear polynomials. This sums up to 570,460 DOF per physical variable. To further enhance the efficiency of the computation,  $p$ -adaption with  $1 \leq p \leq 5$  is applied. Inspired by the spectral decaying indicator of Persson and Peraire [64] we choose the following adaption indicator

$$\text{Ind} = \log_{10} \left( \frac{\sum_{j=n_M(p-1)+1}^{n_m(p)} (\hat{u}_j)^2}{1 + \sum_{j=2}^{n_m(p)} (\hat{u}_j)^2} + 10^{-16} \right), \quad (3.4)$$

where  $\hat{\mathbf{u}}$  are the modal coefficients of the momentum in  $x_1$  direction:  $\rho v_1$ . The range of this indicator is  $0 < \text{Ind} \leq -16$ , where lower indicators corresponds to better resolutions. With the proposed initial discretization, the solution is reasonable well resolved, yielding a range of occurring indicators of  $-5.5 \leq \text{Ind} \leq -16$ . We choose the following adaption controls for this computation

$$p_{\text{new}} = \begin{cases} \min(5, p_{\text{old}} + 1), & \text{if } \text{Ind} \leq -7, \\ \max(1, p_{\text{old}} - 1), & \text{if } \text{Ind} \geq -8.5. \end{cases} \quad (3.5)$$

With this adaptation settings the overall number of DOF was further decreased from 570,460 to  $\approx 350,000$  during the computation, resulting in an increased resolution in the boundary layer and in the region of the von Karman vortex street. The distribution of the local polynomial degrees  $p$  is plotted in Figure 3.16, which shows that all DOF are concentrated in the vicinity and in the wake of the sphere. The process of  $p$ -adaptation can be fully integrated into the

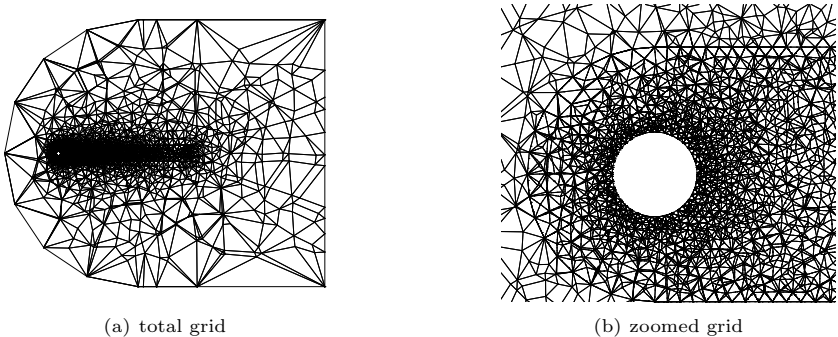


**Figure 3.16:** Distribution of local polynomial degree  $p$  at  $t_{end} = 1000$ .

local time stepping algorithm without synchronizing the grid cells to a common time level, as increasing or decreasing the order of the polynomial is purely local. Because of the rather large time scale of this problem compared to the maximal explicit time step, we allow the grid cell to adapt every 500 time steps. The minimum time step in this calculation is approximately  $\Delta t_{min} \approx 6 \cdot 10^{-4}$ . To get a fully developed periodic solution, we choose the simulation time for this example as  $t_{end} = 1000$ . The local time stepping feature of our scheme increases the average time step of the calculation to  $\Delta t_{avg} \approx 0.0115$ , yielding a speed up factor of  $\frac{\Delta t_{avg}}{\Delta t_{min}} \approx 18.6$  due to the local time stepping. To com-



pare the effectiveness of the hybrid approach with  $p$ -adaptation a second more 'classic' configuration with a pure conforming tetrahedral mesh is considered. The computational domain in this configuration extends from  $x_1 = -20$  to  $x_1 = 100$  and  $x_2, x_3 = \pm 30$ . The grid consists of  $\approx 160,000$  tetrahedra, where the wake of the sphere is resolved with  $h \approx 0.5$  and the surface of the sphere is discretized using triangles with  $h \approx 0.1$ . We plot the projection of the grid on the  $x_1 - x_3$ -plane at  $x_2 = 0$  in Figure 3.17. For the calculation polynomials



**Figure 3.17:** Visualization of the tetrahedra grid for the sphere example.

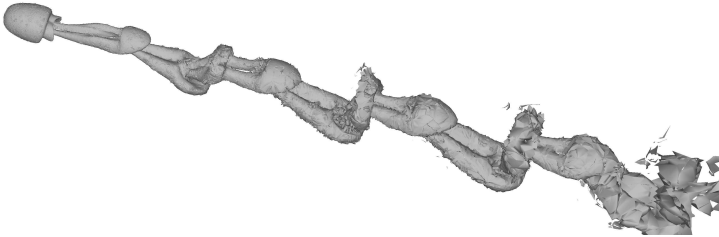
with degree  $p = 3$  are chosen, resulting in  $\approx 3,000,000$  DOF. Although the number of DOF are about one magnitude higher compared to the hybrid configuration the solution accuracy is slightly worse. Especially the discretization of the boundary layer with prisms yields a better approximation. If we compare the overall wallclock CPU time of the simulations with 100 processors of the HLRB Altix cluster, the hybrid configuration with  $p$ -adaptation needs  $\simeq 20$  hours and the tetrahedral configuration needs  $\simeq 120$  hours. Due to the local time stepping and the  $p$ -adaptation, balancing of the work load per processor is a non-trivial task. For the hybrid configuration with  $p$ -adaptation the parallel speed up with 100 processors is  $\approx 80\%$ , although the average number of DOF per processor is only  $\approx 3,500$ . We refer to Lörcher [54] for a detailed description of the dynamic load balancing algorithm and for an thorough investigation of the parallel speed up and scale up. The outcome of this investigations clearly confirms the good parallelizability of the proposed method.

The resulting mean drag and lifting coefficient  $C_{d/l}$ , their amplitudes  $\Delta C_{d/l}$  and the Strouhal number of the simulation with the hybrid configuration are plotted in Table 3.12 and are showing good agreement compared to results from the literature and to results obtained with the tetrahedral configuration [33].

	$C_d$	$\Delta C_d$	$C_l$	$\Delta C_l$	$Str$
hybrid	0.669	0.0033	-0.067	0.015	0.138
tet	0.673	0.0031	-0.065	0.015	0.135
Tomboulides [73]	0.671	0.0028	—	—	0.136
Johnson&Patel [45]	0.656	0.0035	-0.069	0.016	0.137

**Table 3.12:** Force coefficients and Strouhal number.

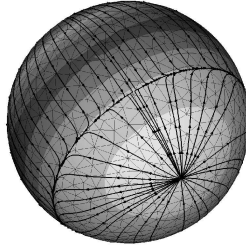
Figure 3.18 reveals the hairpin structure of the vortices [45] by using the  $\lambda_2$  vortex detection criterium. In the last Figure 3.19 the instantaneous distribution of the pressure coefficient and the limiting streamlines are plotted, where we can clearly identify the separation line.



**Figure 3.18:** Isometric view of instantaneous  $\lambda_2$  isosurface for the  $Re = 300$  case.

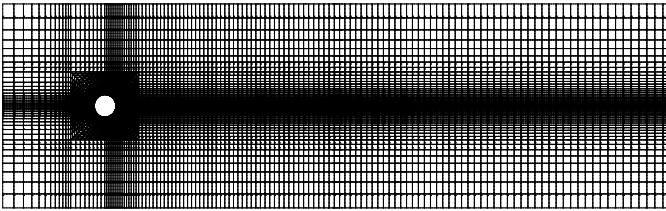
### 3.2.2.2 Weak Turbulent Flow Past a Sphere

In this last example we increase the Reynolds number based on the diameter to  $Re = 1000$ . To compare our simulation to the incompressible results obtained by Toumboulides and Orszag [74] we choose a low Mach number of  $Ma_\infty = 0.3$  and use a similar setup. The sphere is originated at  $\vec{x} = (0, 0, 0)^T$  with a radius



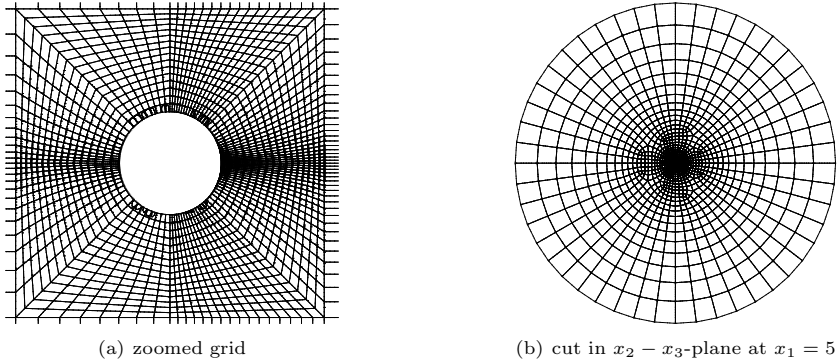
**Figure 3.19:** Distribution of the pressure coefficient and limiting streamlines on the sphere surface.

of  $r = 0.5$ . The cylindrical computational domain extends from  $x_1 = -4.5$  to  $x_1 = 25$  with a diameter of nine. We choose for the discretization of the domain a grid similar to the one used by Toumboulides and Orszag [74]. A cut of the grid in the  $x_1 - x_3$ -plane at  $x_2 = 0$  is shown in Figure 3.20. This two dimensional grid consists of about 8,000 quadrilateral grid cells. A de-



**Figure 3.20:** Visualization of the grid in  $x_1 - x_3$ -plane at  $x_2 = 0$ .

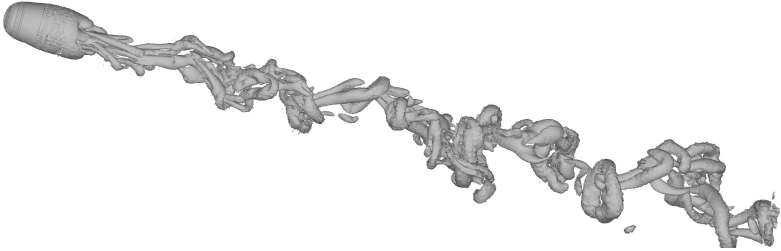
tailed view of the region in the vicinity of the sphere is plotted in Figure 3.21a. In Figure 3.21b a cut of the grid in the  $x_2 - x_3$ -plane at  $x_1 = 5$  is shown. The overall grid consists of 154,016 hexahedral grid cells. We choose for the calculation an approximation with a  $p = 5$  polynomial resulting in 8,624,896 DOF per conservative variable. We choose the recursion parameter  $\ell = (1,1)$  which yields 105 interpolation points per hexahedra and thus 16,171,680 interpolation points per conservative variable. The calculation was performed



**Figure 3.21:** Detailed visualization of the grid.

on 500 processors of the HLRB Altix cluster, resulting in a wallclock time  $t_{CPU} \approx 7000[s]$  per characteristic time interval  $\Delta t = \frac{2r}{v_\infty}$ . The minimum time step in this calculation is  $\Delta t_{min} \approx 8.6 \cdot 10^{-5}$  and the average time step is about  $\Delta t_{avg} \approx 2.1 \cdot 10^{-3}$  resulting in a speed up of  $\frac{\Delta t_{avg}}{\Delta t_{min}} \approx 24.5$  due to the local time stepping feature.

Increasing the Reynolds number to 1000 causes the appearance of smaller scales and a more chaotic flow due to a Kelvin-Helmholtz like instability of the shear layer. To illustrate this behaviour, isosurfaces of  $\lambda_2$  are plotted in Figure 3.22. Furthermore the contour plot of the second component of the vorticity  $\vec{\omega} = \vec{\nabla} \times \vec{v}$  on a  $x_1 - x_3$ -plane at  $x_2 = 0$  is shown in Figure 3.23 with 20 regular contour levels from  $-4 \leq \omega_2 \leq 4$ . The Strouhal number in this calculation results to  $Str = 0.191$  showing good agreement to the values reported by Toumboulides and Orszag [74] and the references therein.



**Figure 3.22:** Isometric view of instantaneous  $\lambda_2$  isosurface for the  $Re = 1000$  case.



**Figure 3.23:** Distribution of instantaneous  $\omega_2$  on a  $x_1 - x_3$ -plane at  $x_2 = 0$ .



## 4 Prospects

The aim of this work is to enhance the computational efficiency of a discontinuous Galerkin based solver for the *time dependent* nonlinear *compressible* Navier-Stokes equations. Although we concentrate on solving the Navier-Stokes equations the developed framework can directly be applied to linear or nonlinear first or second order hyperbolic/parabolic conservation laws. In fact the structure of the developed code allows to easily incorporate new conservation laws, e.g. [56] and [1], as essentially only the new physical fluxes  $\vec{F}^A(U)$  and  $\vec{F}^D(U, \vec{\nabla}U)$  have to be implemented.

The main building blocks of the new methodology are a special treatment of second order terms, a hybrid usage of modal and nodal polynomial representations, and a novel time discretization method. Compared to the state of the art we observed in numerical experiments a speed-up of about one order of magnitude because of the local time stepping feature, a reduction of the computational effort of about 30% due to the predictor corrector formulation, a factor of four due to the nodal type integration and an additional saving of about 30% due to the Riemann solver based numerical viscous fluxes.

In addition to the unresolved problems discussed in section 3.1.3 an unstructured adaptive high order accurate method has its own unique challenges. If we are interested in large scale computations on massively parallel clusters with  $\mathcal{O}(10^3)$  processors, the management and post-processing of the resulting data can cause severe difficulties. Due to the piecewise high order polynomial representation of the solution even the 'simple' visualization of specific quantities is non-trivial if one is interested in an accurate representation of the subcell resolution. Considering the simulation of turbulent flows, evaluation of the numerical results needs computations of global mean values and other statistical quantities, which is another non-trivial task due to the unstructured meshes. Thus an important task for the future is the development of accurate and efficient post-processing tools suitable for unstructured, non-conforming, locally refined high order discontinuous Galerkin data.

An interesting challenge for the future is to combine the developed discontinuous Galerkin method with a turbulence modeling approach. Motivated by the good results of Collis [21], who combined a discontinuous Galerkin scheme with the variational multiscale (VMS) approach [43], it seems to be promising to extend our method with the VMS approach. A successful incorporation of turbulence modeling within this efficient high order accurate methodology would open a huge field of physically relevant flow applications.



## A Lebesgue Constants

$p_L$	$n_L$	$\Lambda$	$n_L$	$\Lambda$	$n_L$	$\Lambda$
	<b>tri</b> ( $\ell_{2D} = 0$ )		<b>quad</b> ( $\ell_{2D} = 0$ )		<b>quad</b> ( $\ell_{2D} = 1$ )	
1	3	1.0	4	1.5	4	1.5
2	6	1.7	8	3.0	8	3.0
3	10	2.1	12	4.0	13	3.2
4	15	3.8	17	4.2	20	5.3
5	21	3.2	24	5.8	28	4.6
6	28	4.6	32	7.5	37	4.5
7	36	6.8	40	15.3	48	5.1
8	45	7.5	49	14.5	60	7.5
9	55	8.6	60	21.0	73	8.0
10	66	11.2	72	28.6	88	10.8
11	78	18.8	84	61.8	104	14.8
12	91	20.2	97	62.7	121	15.4

**Table A.1:** Lebesgue constants  $\Lambda$  and number of interpolation points  $n_L$  for the two dimensional interpolation points.

<b>tetrahedron/<math>p_L</math></b>	1	2	3	4	5	6	7	8	9	10	11
$n_L$	4	10	20	35	56	84	120	165	220	286	364
$\ell = (0,0)/\Lambda$	1.0	2.0	2.9	4.0	6.4	7.9	10.8	17.6	22.0	34.8	36.5
<b>hexahedron/<math>p_L</math></b>	1	2	3	4	5	6	7	8	9	10	11
$n_L$	8	20	32	50	80	117	160	214	280	358	448
$\ell = (0,0)/\Lambda$	1.5	5.0	6.4	8.8	17.0	20.3	41.5	47.6	103.6	201.3	454.2
$n_L$	8	20	32	50	81	124	172	226	298	389	492
$\ell = (1,0)/\Lambda$	1.5	5.0	6.4	8.8	11.6	35.6	37.1	46.6	103.2	113.5	148.2
$n_L$	8	20	38	68	104	147	208	280	364	472	592
$\ell = (0,1)/\Lambda$	1.5	5.0	4.8	15.6	11.2	13.0	30.4	32.7	52.0	111.6	323.5
$n_L$	8	20	38	68	105	154	220	298	394	509	642
$\ell = (1,1)/\Lambda$	1.5	5.0	4.8	15.6	8.9	18.1	13.3	31.0	49.4	58.0	78.0
$n_L$	8	20	32	51	88	136	184	245	336	444	552
$\ell = (2,0)/\Lambda$	1.5	5.0	6.4	7.8	9.1	14.0	30.2	28.3	40.2	56.9	124.2
$n_L$	8	20	38	69	112	166	238	329	438	570	726
$\ell = (2,1)/\Lambda$	1.5	5.0	4.8	5.9	8.9	11.1	12.5	20.3	21.2	31.5	61.2

**Table A.2:** Lebesgue constants  $\Lambda$  and number of interpolation points  $n_L$  for the three dimensional interpolation sets with different parameters  $\ell = (\ell_{3D}, \ell_{2D})$ .

<b>pentahedron/<math>p_L</math></b>	1	2	3	4	5	6	7	8	9	10	11
$n_L$	6	15	26	42	67	101	141	188	248	322	407
$\ell = (0, 0)/\Lambda$	1.7	3.7	4.4	6.0	8.1	21.4	22.7	42.3	96.7	112.1	175.2
$n_L$	6	15	26	43	72	110	152	205	278	365	458
$\ell = (1, 0)/\Lambda$	1.7	3.7	4.4	5.9	10.0	11.2	23.4	24.2	61.6	74.2	167.8
$n_L$	6	15	29	51	79	116	165	224	296	382	482
$\ell = (0, 1)/\Lambda$	1.7	3.7	4.1	9.4	7.2	15.6	15.0	34.2	70.8	86.8	117.6
$n_L$	6	15	29	52	84	125	179	247	329	428	545
$\ell = (1, 1)/\Lambda$	1.7	3.7	4.1	5.7	10.0	8.7	13.0	17.2	33.3	34.5	60.2
<b>pyramid/<math>p_L</math></b>	1	2	3	4	5	6	7	8	9	10	11
$n_L$	5	13	25	42	66	98	138	187	247	319	403
$\ell = (0, 0)/\Lambda$	1.5	3.0	4.2	6.8	9.7	15.6	24.5	39.7	71.4	146.9	366.2
$n_L$	5	13	25	43	70	106	150	205	275	359	455
$\ell = (1, 0)/\Lambda$	1.5	3.0	4.2	5.3	7.2	11.4	20.0	20.8	54.8	38.6	83.6
$n_L$	5	13	26	45	70	103	146	199	263	339	428
$\ell = (0, 1)/\Lambda$	1.5	3.0	3.8	8.4	9.0	13.1	20.2	32.8	65.5	137.7	360.6
$n_L$	5	13	26	46	74	111	159	219	292	380	484
$\ell = (1, 1)/\Lambda$	1.5	3.0	3.8	6.0	7.0	9.5	12.9	18.0	27.4	27.4	42.1

**Table A.3:** Lebesgue constants  $\Lambda$  and number of interpolation points  $n_L$  for the three dimensional interpolation sets with different parameters  $\ell = (\ell_{3D}, \ell_{2D})$ .



## B Coefficients for CERK schemes

exponent $o$	$b_{o1}^t$
0	1

**Table B.1:** Coefficients  $b_{oj}^t$  with  $p_t = 0$  with  $n_{stages} = 1$ .

Stage	$a_{j1}$
2	1

**Table B.2:** Coefficients  $a_{jl}$  with  $p_t = 1$  with  $n_{stages} = 2$ .

exponent $o$	$b_{o1}^t$	$b_{o2}^t$
0	1	0
1	$\frac{-1}{1}$	$\frac{1}{2}$

**Table B.3:** Coefficients  $b_{oj}^t$  with  $p_t = 1$  with  $n_{stages} = 2$ .

Stage	$a_{j1}$	$a_{j2}$	$a_{j3}$
2	$\frac{12}{23}$		
3	$\frac{-68}{375}$	$\frac{368}{375}$	
4	$\frac{31}{144}$	$\frac{529}{1152}$	$\frac{125}{384}$

Table B.4: Coefficients  $a_{j1}$  with  $p_t = 2$  with  $n_{stages} = 4$ .

exponent $o$	$b_{o1}^t$	$b_{o2}^t$	$b_{o3}^t$	$b_{o4}^t$
0	1	0	0	0
1	$\frac{-65}{48}$	$\frac{529}{384}$	$\frac{125}{128}$	-1
2	$\frac{41}{72}$	$\frac{-529}{576}$	$\frac{-125}{192}$	1

Table B.5: Coefficients  $b_{oj}^t$  with  $p_t = 2$  with  $n_{stages} = 4$ .

Stage	$a_{j1}$	$a_{j2}$	$a_{j3}$	$a_{j4}$	$a_{j5}$
2	$\frac{1}{6}$				
3	$\frac{44}{1369}$	$\frac{363}{1369}$			
4	$\frac{3388}{4913}$	$\frac{-8349}{4913}$	$\frac{8140}{4913}$		
5	$\frac{-36764}{408375}$	$\frac{767}{1125}$	$\frac{-32708}{136125}$	$\frac{210392}{408375}$	
6	$\frac{1697}{18876}$	0	$\frac{50653}{116160}$	$\frac{299693}{1626240}$	$\frac{3375}{11648}$

Table B.6: Coefficients  $a_{j1}$  with  $p_t = 3$  with  $n_{stages} = 6$ .

exponent $o$	$b_{o1}^t$	$b_{o2}^t$	$b_{o3}^t$	$b_{o4}^t$	$b_{o5}^t$	$b_{o6}^t$
0	1	0	0	0	0	0
1	$\frac{-104217}{37466}$	0	$\frac{1861101}{230560}$	$\frac{-63869}{293440}$	$\frac{-1522125}{762944}$	$\frac{165}{131}$
2	$\frac{1806901}{618189}$	0	$\frac{-2178079}{380424}$	$\frac{6244423}{5325936}$	$\frac{982125}{190736}$	$\frac{-461}{131}$
3	$\frac{-866577}{824252}$	0	$\frac{12308679}{5072320}$	$\frac{-7816583}{10144640}$	$\frac{-624375}{217984}$	$\frac{196}{131}$

Table B.7: Coefficients  $b_{oj}^t$  with  $p_t = 3$  with  $n_{stages} = 6$ .

---

Stage	$a_{j1}$	$a_{j2}$	$a_{j3}$	$a_{j4}$	$a_{j5}$	$a_{j6}$	$a_{j7}$
2	$\frac{1}{6}$						
3	$\frac{1}{16}$	$\frac{3}{16}$					
4	$\frac{1}{4}$	$-\frac{3}{4}$	1				
5	$-\frac{3}{4}$	$\frac{15}{4}$	-3	$\frac{1}{2}$			
6	$\frac{369}{1372}$	$-\frac{243}{343}$	$\frac{297}{343}$	$\frac{1485}{9604}$	$\frac{297}{4802}$		
7	$-\frac{133}{4512}$	$\frac{1113}{6016}$	$\frac{7945}{16544}$	$-\frac{12845}{24064}$	$-\frac{315}{24064}$	$\frac{156065}{198528}$	
8	$\frac{83}{945}$	0	$\frac{248}{825}$	$\frac{41}{180}$	$\frac{1}{36}$	$\frac{2401}{38610}$	$\frac{6016}{20475}$

Table B.8: Coefficients  $a_{jl}$  with  $p_t = 4$  with  $n_{stages} = 8$ .

exponent $o$	$b_{o1}^t$	$b_{o2}^t$	$b_{o3}^t$	$b_{o4}^t$	$b_{o5}^t$	$b_{o6}^t$	$b_{o7}^t$	$b_{o8}^t$
0	1	0	0	0	0	0	0	0
1	$-\frac{3292}{819}$	0	$\frac{5112}{715}$	$-\frac{123}{52}$	-63	$-\frac{40817}{33462}$	$\frac{18048}{5915}$	-18
2	$\frac{17893}{2457}$	0	$-\frac{43568}{2145}$	$\frac{3161}{224}$	$\frac{1061}{224}$	$\frac{60025}{50193}$	$-\frac{637696}{53235}$	$\frac{75}{13}$
3	$-\frac{4369}{819}$	0	$\frac{1344}{68}$	$-\frac{1495}{78}$	-413	$\frac{2401}{1321}$	$\frac{96356}{3315}$	-139
4	$\frac{596}{315}$	0	$-\frac{1984}{275}$	$\frac{118}{15}$	2	$-\frac{0604}{6435}$	$-\frac{48128}{6825}$	4

Table B.9: Coefficients  $b_{oj}^t$  with  $p_t = 4$  with  $n_{stages} = 8$ .



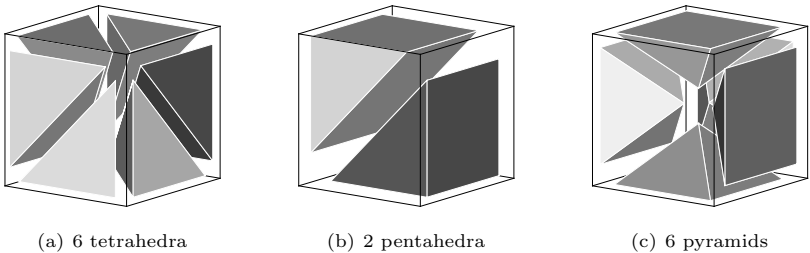


## C A Hybrid Grid for h-Refinement Studies

The computational domain  $\Omega := [0; 2]^3$  is split into 8 regular cubic subdomains  $\Omega_i = \vec{x}_i + [0; 1]^3$   $i = 1, \dots, 8$  with

$$\begin{aligned} \vec{x}_1 &:= (0, 0, 0)^T, \quad \vec{x}_2 := (1, 0, 0)^T, \quad \vec{x}_3 := (0, 1, 0)^T, \\ \vec{x}_4 &:= (0, 0, 1)^T, \quad \vec{x}_5 := (1, 1, 0)^T, \quad \vec{x}_6 := (0, 1, 1)^T, \\ \vec{x}_7 &:= (1, 0, 1)^T, \quad \vec{x}_8 := (1, 1, 1)^T. \end{aligned} \quad (\text{C.1})$$

For our  $h$ -refinement tests we introduce the parameter  $n_h \geq 1$ . For a given  $n_h$ , we first split every subdomain  $\Omega_i$  into  $n_h^3$  regular hexahedral elements. To generate the hybrid mesh, we furthermore split the hexahedra in the first domain  $i = 1$  into tetrahedra, in the domains  $i = 2, 3, 4$  into prisms and in the domain  $i = 8$  into pyramids. We illustrate the different hexahedra splittings in Figure C.1 (please note that the front pyramid is blanked for better visualization purpose). For  $n_h = 1$  the hybrid prototype mesh consists of 21 grid cells. With increasing refinement parameter  $n_h$  it is now possible



**Figure C.1:** Visualization of the different hybrid meshes.

to define a sequence of regularly refined hybrid meshes, which is needed for convergence studies.



## D The Linearized Euler Equations

The linearized Euler equations (LEE) are a linear model problem used in the simulation of acoustic wave propagation. The structure of the LEE equations fits well in the considered class of problems when we neglect the diffusion part, i.e.  $\vec{F}^D = 0$ . The conservative variables are given as  $U = (\rho', u', v', w', p')^T$  and the LEE fluxes as  $\vec{F}^A := (F_1^A, F_2^A, F_3^A)^T := (\underline{A}_1 U, \underline{A}_2 U, \underline{A}_3 U)^T$  with the Jacobian matrices

$$\underline{A}_1 = \begin{pmatrix} u_0 & \rho_0 & 0 & 0 & 0 \\ 0 & u_0 & 0 & 0 & \frac{1}{\rho_0} \\ 0 & 0 & u_0 & 0 & 0 \\ 0 & 0 & 0 & u_0 & 0 \\ 0 & \kappa p_0 & 0 & 0 & u_0 \end{pmatrix}, \quad \underline{A}_2 = \begin{pmatrix} v_0 & 0 & \rho_0 & 0 & 0 \\ 0 & v_0 & 0 & 0 & 0 \\ 0 & 0 & v_0 & 0 & \frac{1}{\rho_0} \\ 0 & 0 & 0 & v_0 & 0 \\ 0 & 0 & \kappa p_0 & 0 & v_0 \end{pmatrix}, \quad (\text{D.1})$$

$$\underline{A}_3 = \begin{pmatrix} w_0 & 0 & 0 & \rho_0 & 0 \\ 0 & w_0 & 0 & 0 & 0 \\ 0 & 0 & w_0 & 0 & 0 \\ 0 & 0 & 0 & w_0 & \frac{1}{\rho_0} \\ 0 & 0 & 0 & \kappa p_0 & w_0 \end{pmatrix},$$

where  $U_0 := (\rho_0, u_0, v_0, w_0, p_0)^T$  is the background flow. As an example for an analytical solution, a planar wave is initialized such that it contains only fluctuations in the right moving characteristic wave with the Eigenvalue  $u_0 + c_0$

$$U = \underline{R} W, \quad (\text{D.2})$$

with  $W = \hat{W} \sin(\vec{k} \cdot \vec{x})$  and the Eigenvector matrix

$$\underline{R} = \begin{pmatrix} n_1 & n_2 & n_3 & \frac{\rho_0}{2c_0} & \frac{\rho_0}{2c_0} \\ 0 & -n_3 & n_2 & \frac{n_1}{2} & -\frac{n_1}{2} \\ n_3 & 0 & -n_1 & \frac{n_2}{2} & -\frac{n_2}{2} \\ -n_2 & n_1 & 0 & \frac{n_3}{2} & -\frac{n_3}{2} \\ 0 & 0 & 0 & \frac{\rho_0}{2c_0} & \frac{\rho_0}{2c_0} \end{pmatrix}, \quad (\text{D.3})$$

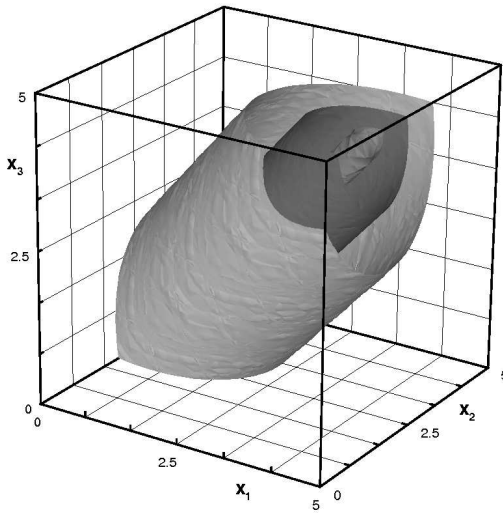
with  $c_0 = \sqrt{\kappa \frac{p_0}{\rho_0}}$ . We choose the perturbation of the characteristic variable vector  $\hat{W} = (0, 0, 0, 1e-3, 0)^T$ , the normal vector of the wave  $\vec{n} = (1, 0, 0)^T$ , the wave number vector  $\vec{k} = (\pi, 0, 0)^T$  and the background flow  $U_0 = (1, 0, 0, 0, \frac{1}{\kappa})^T$  with  $\kappa = 1.4$ , resulting in  $c_0 = 1$ .

## E An Analytical Solution for the Euler Equations

The considered analytical solution is a three dimensional variation of the isentropic vortex convection problem of Hu and Shu [42]

$$\begin{aligned}
 \vec{r}(\vec{x}, t) &= \vec{r}_{vortex} \times (\vec{x} - \vec{x}_0 - \vec{v}_0 \cdot t), \\
 \delta v &= \frac{v_{max}}{2\pi} \exp\left(\frac{1 - \left(\frac{|\vec{r}|}{r_0}\right)^2}{2}\right), \\
 \vec{v}(\vec{x}, t) &= \vec{v}_0 + \delta v \cdot \vec{r}, \\
 \frac{T}{T_0} &= 1 - \frac{\kappa - 1}{2} \left(\frac{\delta v}{c_0}\right)^2, \\
 \rho(\vec{x}, t) &= \rho_0 \left(\frac{T}{T_0}\right)^{\frac{1}{\kappa-1}}, \\
 p(\vec{x}, t) &= p_0 \left(\frac{T}{T_0}\right)^{\frac{\kappa}{\kappa-1}}.
 \end{aligned} \tag{E.1}$$

If we choose the rotational axis of the vortex  $\vec{r}_{vortex} = (0, 0, 1)^T$  and furthermore  $\rho_0 = p_0 = R = 1$ , then the standard two dimensional problem is recovered. For our test problem we choose the following constant background flow  $(\rho_0, \vec{v}_0^T, p_0) = (1, 1, 1, 1, \frac{1}{\kappa})$ ,  $\kappa = 1.4$ , the rotational axis of the vortex  $\vec{r}_{vortex} = (1, -0.5, 1)^T$ , the initial center of the vortex  $\vec{x}_0 = (0.5, 0.5, 0.5)^T$ , the amplitude of the vortex  $v_{max} = 0.1$ , the halfwidth of the vortex  $r_0 = 1$  and the end time of the simulation  $t_{end} = 4$ . The computational domain  $\Omega := [0; 5]^3$  with exact boundary conditions prescribed. The solution to this problem at time  $t = 2$  is shown in Figure E.1.



**Figure E.1:** 3D isentropic vortex. Isosurfaces of density.

## F An Analytical Solution for the Navier-Stokes Equations

If we are interested in a convergence study for the compressible Navier-Stokes equations (NSE) an exact solution is needed. However, up to now no two or three dimensional exact solutions for the unsteady compressible NSE are known. We thus construct an exact solution, but for the inhomogeneous NSE. We choose an arbitrary smooth solution, insert it into the NSE and calculate the right hand side, which is in the general case not zero. This right hand side is used as a source term for the inhomogeneous problem, which has now a known exact solution. If we choose

$$U = (\sin(\beta)\gamma + 2, \sin(\beta)\gamma + 2, \sin(\beta)\gamma + 2, \sin(\beta)\gamma + 2, (\sin(\beta)\gamma + 2)^2)^T, \quad (\text{F.1})$$

with  $\beta := k \sum_{j=1}^d x_j - \omega t$ , we get the following corresponding source term

$$S = \gamma \begin{pmatrix} \cos(\beta) (dk - \omega) \\ \cos(\beta) A + \sin(2\beta)\gamma k (\kappa - 1) \\ \cos(\beta) A + \sin(2\beta)\gamma k (\kappa - 1) \\ \cos(\beta) A + \sin(2\beta)\gamma k (\kappa - 1) \\ \cos(\beta) B + \sin(2\beta)\gamma (dk\kappa - \omega) + \sin(\beta) \left( \frac{dk^2\mu\kappa}{Pr} \right) \end{pmatrix}, \quad (\text{F.2})$$

with

$$\begin{aligned} A &= -\omega + \frac{k}{d-1} \left( (-1)^{d-1} + \kappa(2d-1) \right) \\ B &= \frac{1}{2} \left( (d^2 + \kappa(6+3d))k - 8\omega \right). \end{aligned} \quad (\text{F.3})$$

We made for the derivation of this analytical solution the assumption that the fluid dependent coefficients  $\mu$  and  $Pr$  are constant. If we choose  $d = 2$  for the two dimensional case, the 4th row of the solution and of the source term have to be canceled. The parameter  $\gamma$ ,  $\omega$  and  $k$  can be chosen arbitrarily and are related to spatial and temporal scale of the analytical solution.





## Bibliography

- [1] C. Altmann, G. Gassner, F. Lörcher, and C.-D. Munz. An explicit discontinuous Galerkin scheme for ideal Magnetohydrodynamics with an inherent hyperbolic divergence cleaning sub-cycling mechanism. *in preparation*.
- [2] D. Arnold. *An Interior Penalty Finite Element Method with Discontinuous Elements*. PhD thesis, The University of Chicago, 1979.
- [3] D. N. Arnold, F. Brezzi, B. Cockburn, and L. D. Marini. Discontinuous Galerkin methods for elliptic problems. In B. Cockburn, G. Karniadakis, and Chi-Wan Shu, editors, *Discontinuous Galerkin Methods. Lecture Notes in Computational Science and Engineering*, pages 89–101. Springer, 2000.
- [4] D. N. Arnold, F. Brezzi, B. Cockburn, and L. D. Marini. Unified analysis of discontinuous Galerkin methods for elliptic problems. *SIAM J. Numer. Anal.*, 39(5):1749–1779, 2002.
- [5] A. Babucke. *Direct Numerical Simulation of Noise-Generation Mechanisms in the Mixing Layer of a Jet*. Dissertation, Universität Stuttgart, 2009.
- [6] A. Babucke, M. J. Kloker, and U. Rist. DNS of a plane mixing layer for the investigation of sound generation mechanisms. *Comput. and Fluids*, 37:360–368, 2008.
- [7] D.S. Balsara, C. Altmann, C.-D. Munz, and M. Dumbser. A sub-cell based indicator for troubled zones in RKDG schemes and a novel class of hybrid RKDG+HWENO schemes. *J. Comput. Phys.*, 226:586–620, 2008.
- [8] 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.
- [9] F. Bassi and S. Rebay. A high-order discontinuous Galerkin finite element method solution of the 2D Euler equations. *J. Comput. Phys.*, 138:251–285, 1997.

- [10] F. Bassi and S. Rebay. Numerical evaluation of two discontinuous Galerkin methods for the compressible Navier-Stokes equations. *Int. J. Numer. Meth. Fluids*, 40:197–207, 2002.
- [11] M. Ben-Artzi and J. Falcovitz. A second-order Godunov-type scheme for compressible fluid dynamics. *J. Comput. Phys.*, 55:1–32, 1984.
- [12] R. Biswas, K.D. Devine, and J. Flaherty. Parallel, adaptive finite element methods for conservation laws. *Appl. Numer. Math.*, 14:255–283, 1994.
- [13] Y. Cheng and C.-W. Shu. A discontinuous Galerkin finite element method for time dependent partial differential equations with higher order derivatives. *Math. Comput.*, 77:699–730, 2008.
- [14] B. Cockburn, S. Hou, and C.-W. Shu. The Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws IV: The multidimensional case. *Math. Comput.*, 54:545–581, 1990.
- [15] B. Cockburn, S. Y. Lin, and C.-W. Shu. TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws III: One dimensional systems. *J. Comput. Phys.*, 84:90–113, 1989.
- [16] B. Cockburn and C.-W. Shu. TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws II: General framework. *Math. Comput.*, 52:411–435, 1989.
- [17] B. Cockburn and C.-W. Shu. The Runge-Kutta local projection  $p^1$ -discontinuous Galerkin method for scalar conservation laws. *M<sup>2</sup>AN*, 25:337–361, 1991.
- [18] B. Cockburn and C.-W. Shu. The local discontinuous Galerkin method for time-dependent convection diffusion systems. *SIAM J. Numer. Anal.*, 35:2440–2463, 1998.
- [19] B. Cockburn and C.-W. Shu. The Runge-Kutta discontinuous Galerkin method for conservation laws V: Multidimensional systems. *J. Comput. Phys.*, 141:199–224, 1998.
- [20] S.S. Collis. Monitoring unresolved scales in multiscale turbulence modeling. *Phys. Fluids*, 13:1800–1806, 2001.

- [21] S.S. Collis. The DG/VMS method for unified turbulence simulation. In *Proc. of the 32nd AIAA Fluid Dynamics Conference and Exhibit*, 2002.
- [22] S.S. Collis. Discontinuous Galerkin methods for turbulence simulation. In *Proc. of the Summer Programm*, 2002.
- [23] S.S. Collis and Kaveh Ghayour. Discontinuous Galerkin methods for compressible DNS. In *Proceedings of FEDSM, ASME/JSME Joint Fluids Engineering Conference*, 2003.
- [24] T. Colonius, S. K. Lele, and P. Moin. Sound generation in a mixing layer. *J. Fluid Mech.*, 330:375–409, 1997.
- [25] K.D. Devine, J.E. Flaherty, S.R. Wheat, and A.B. Maccabe. Parallel partitioning strategies for the adaptive solution of conservation laws. Technical Report 94-1, Rensselaer Polytechnic Institute.
- [26] K.D. Devine, J.E. Flaherty, S.R. Wheat, and A.B. Maccabe. A massively parallel adaptive finite element method with dynamic load balancing. Technical Report 93-0936C, SANDIA, 1993.
- [27] M. Dumbser, D. Balsara, E. F. Toro, and C.-D. Munz. A unified framework for the construction of one-step finite-volume and discontinuous Galerkin schemes on unstructured meshes. *J. Comput. Phys*, 227:8209–8253, 2008.
- [28] M. Dumbser, C. Enaux, and E. F. Toro. Finite volume schemes of very high order of accuracy for stiff hyperbolic balance laws. *J. Comput. Phys.*, 227:3971–4001, 2008.
- [29] M. Dumbser, M. Käser, V. A. Titarev, and E. F. Toro. Quadrature-free non-oscillatory finite volume schemes on unstructured meshes for nonlinear hyperbolic systems. *J. Comput. Phys*, 226:204–243, 2007.
- [30] M. Dumbser, M. Käser, and E. F. Toro. An arbitrary high–order discontinuous Galerkin method for elastic waves on unstructured meshes – V. Local time stepping and p–adaptivity. *Geophys. J. Int.*, 171:695–717, 2007.
- [31] M. Dumbser and C.-D. Munz. Building blocks for arbitrary high order discontinuous Galerkin schemes. *J. Sci. Comput.*, 27:215–230, 2006.

- [32] J. E. Flaherty, R. M. Loy, M. S. Shephard, B. K. Szymanski, J. D. Teresco, and L. H. Ziantz. Adaptive local refinement with octree load balancing for the parallel solution of three-dimensional conservation laws. *J. Para. Dist. Comput.*, 47(2):139–152, 1997.
- [33] G. Gassner, F. Lörcher, C.-D. Munz, and J. S. Hesthaven. Polymorphic nodal elements and their application in discontinuous Galerkin methods. *J. Comput. Phys.*, (228):1573–1590, 2009.
- [34] G. Gassner, F. Lörcher, and C.-D. Munz. A contribution to the construction of diffusion fluxes for finite volume and discontinuous Galerkin schemes. *J. Comput. Phys.*, 224(2):1049–1063, 2007.
- [35] G. Gassner, F. Lörcher, and C.-D. Munz. A discontinuous Galerkin scheme based on a space-time expansion. II. Viscous flow equations in multi dimensions. *J. Sci. Comp.*, 34(3):260–286, 2008.
- [36] S. Gottlieb and C.-W. Shu. Total variation diminishing Runge-Kutta schemes. *Math. Comput.*, 67:73–85, 1998.
- [37] R. Hartmann and P. Houston. Symmetric interior penalty DG methods for the compressible Navier-Stokes equations I: Method formulation. *Int. J. Numer. Anal. Model.*, 3(1):1–20, 2006.
- [38] J.S. Hesthaven. From electrostatics to almost optimal nodal sets for polynomial interpolation in simplex. *SIAM J. Numer. Anal.*, 35(2):655–676, 1998.
- [39] J.S. Hesthaven, S. Gottlieb, and D. Gottlieb. *Spectral Methods for Time-Dependent Problems*. Cambridge University Press, Cambridge, 2006.
- [40] J.S. Hesthaven and T. Warburton. Nodal high-order methods on unstructured grids I: Time-domain solution of Maxwell’s equations. *J. Comput. Phys.*, 181(1):186–221, 2002.
- [41] J.S. Hesthaven and T. Warburton. *Nodal Discontinuous Galerkin Methods: Algorithms, Analysis, and Applications*. Springer Verlag, New York, 2008.
- [42] C. Hu and C.-W. Shu. Weighted essentially non-oscillatory schemes on triangular meshes. *J. Comput. Phys.*, 150:97–127, 1999.

- [43] T.J.R. Hughes, L. Mazzei, and K.E. Jansen. Large eddy simulation and the variational multiscale method. *Comp. Vis. Sci.*, 3:47–59, 2000.
- [44] J. Jaffre, C. Johnson, and A. Szepessy. Convergence of the discontinuous Galerkin finite element method for hyperbolic conservation laws. *Math. Mod. Meth. Appl. Sci.*, 182:546–585, 1995.
- [45] T.A. Johnson and V.C. Patel. Flow past a sphere up to a Reynolds number of 300. *J. Fluid. Mech.*, 378:19–70, 1999.
- [46] A. Kanevsky, M.H. Carpenter, D. Gottlieb, and J.S. Hesthaven. Application of implicit-explicit high-order Runge-Kutta methods to discontinuous Galerkin schemes. *J. Comput. Phys.*, 225:1753–1781, 2007.
- [47] G. E. Karniadakis and S. Sherwin. *Spectral/hp Element Methods for Computational Fluid Dynamics*. Numerical Mathematics and Scientific Computation. Oxford University Press, USA, 2005.
- [48] M. Käser and M. Dumbser. An arbitrary high order discontinuous Galerkin method for elastic waves on unstructured meshes I: The two-dimensional isotropic case with external source terms. *Geophys. J. Int.*, 166:855–877, 2006.
- [49] J. Kevorkian. *Partial Differential Equations. Analytical Solution Techniques. Second Edition*. Springer, Berlin, 2000.
- [50] C.M. Klaij, J. J. W. van der Vegt, and H. van der Ven. Spacetime discontinuous Galerkin method for the compressible Navier-Stokes equations. *J. Comput. Phys.*, 217(2):589–611, 2006.
- [51] L. Krivodonova and M. Berger. High-order accurate implementation of solid wall boundary conditions in curved geometries. *J. Comput. Phys.*, 211:492–512, 2006.
- [52] Y. Liu, M. Vinokur, and Z. J. Wang. Spectral difference method for unstructured grids I: Basic formulation. *J. Comput. Phys.*, 216:780–801, 2006.
- [53] I. Lomtev and G. E. Karniadakis. A discontinuous Galerkin method for the Navier-Stokes equations. *Int. J. Numer. Meth. Fluids*, 29:587–603, 1999.

- [54] F. Lörcher. *Predictor-Corrector DG schemes for the numerical solution of the compressible Navier-Stokes equations in complex domains*. Dissertation, Universität Stuttgart, 2009.
- [55] F. Lörcher, G. Gassner, and C.-D. Munz. A discontinuous Galerkin scheme based on a space-time expansion. I. Inviscid compressible flow in one space dimension. *J. Sci. Comp.*, 32(2):175–199, 2007.
- [56] F. Lörcher, G. Gassner, and C.-D. Munz. An explicit discontinuous Galerkin scheme with local time-stepping for general unsteady diffusion equations. *J. Comput. Phys.*, 227(11):5649–5670, 2008.
- [57] F. Lörcher and C.-D. Munz. Lax-Wendroff-type schemes of arbitrary order in several space dimensions. *IMA J. Num. Anal.*, 27:593–615, 2007.
- [58] H. Luo, J.D. Baum, and R. Löhner. On the computation of steady-state compressible flows using a discontinuous Galerkin method. *Int. J. Numer. Meth. Engrg.*, 73:597–623, 2008.
- [59] J. A. Nitsche. Über ein Variationsprinzip zur Lösung von Dirichlet-Problemen bei Verwendung von Teilräumen, die keinen Randbedingungen unterworfen sind. *Abh. Math. Sem. Univ. Hamburg*, 36:9–15, 1971.
- [60] B. Owren and M. Zennaro. Order barriers for continuous explicit Runge-Kutta methods. *Math. Comp.*, 56:645–661, 1991.
- [61] B. Owren and M. Zennaro. Derivation of efficient continuous explicit Runge-Kutta methods. *SIAM J. Sci. Stat. Comput.*, 13:1488–1501, 1992.
- [62] J. Peraire and P.-O. Persson. The compact discontinuous Galerkin (CDG) method for elliptic problems. *SIAM J. Sci. Comput.*, 30(4):1806–1824, 2008.
- [63] P.-O. Persson and J. Peraire. An efficient low memory implicit DG algorithm for time dependent problems. In *Proc. of the 44th AIAA Aerospace Sciences Meeting and Exhibit*, 2006.
- [64] P.-O. Persson and J. Peraire. Sub-cell shock capturing for discontinuous Galerkin methods. In *Proc. of the 44th AIAA Aerospace Sciences Meeting and Exhibit*, 2006.

- [65] P.-O. Persson, J. Peraire, and J. Bonet. Discontinuous Galerkin solution of the Navier-Stokes equations on deformable domains. In *Proc. of the 45th AIAA Aerospace Sciences Meeting and Exhibit*, 2007.
- [66] J. Qiu. A numerical comparison of the Lax-Wendroff discontinuous Galerkin method based on different numerical fluxes. *J. Sci. Comp.*, 30:345–367.
- [67] J. Qiu, B. C. Khoo, and C.-W. Shu. A numerical study for the performance of the Runge-Kutta discontinuous Galerkin method based on different numerical fluxes. *J. Comput. Phys.*, 212:540 – 565, 2006.
- [68] J. Qiu and C.-W. Shu. Hermite WENO schemes and their application as limiters for Runge-Kutta discontinuous Galerkin method II: Two dimensional case. *Comput. Fluids*, 34:642–663, 2005.
- [69] W.H. Reed and T.R. Hill. Triangular mesh methods for the neutron transport equation. Technical Report LA-UR-73-479, Los Alamos Scientific Laboratory, 1973.
- [70] Olaf Schönrock. *Numerical prediction of flow induced noise in free jets of high Mach numbers*. Dissertation, Universität Stuttgart, 2009.
- [71] Y. Sun, Z. J. Wang, and Y. Liu. Spectral (finite) volume method for conservation laws on unstructured grids VI: Extension to viscous flow. *J. Comput. Phys.*, 215:41–58, 2006.
- [72] W. Sutherland. The viscosity of gases and molecular force. *Phil. Mag.*, 36:507–531, 1893.
- [73] A.G. Tomboulides. *Direct and large-eddy simulation of wake flows: Flow past a sphere*. PhD thesis, Princeton University, 1993.
- [74] A.G. Tomboulides and S.A. Orszag. Numerical investigation of transitional and weak turbulent flow past a sphere. *J. Fluid Mech.*, 416:45–73, 2000.
- [75] E.F. Toro. *Riemann Solvers and Numerical Methods for Fluid Dynamics*. Springer, June 1999.
- [76] E.F. Toro, M. Spruce, and W. Speares. Restoration of the contact surface in the Harten–Lax–van Leer Riemann solver. *J. Shock Wave*, 4:25–34, 1994.

- [77] Jens Utzmann. *A Domain Decomposition Method for the Efficient Direct Simulation of Aeroacoustic Problems*. Dissertation, Universität Stuttgart, 2008.
- [78] 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(2):546–585, 2002.
- [79] L. Wang and D. J. Mavriplis. Implicit solution of the unsteady Euler equations for high-order accurate discontinuous Galerkin discretizations. *J. Comput. Phys.*, 225:1994–2015, 2007.
- [80] T. Warburton. An explicit construction of interpolation nodes on the simplex. *J. Engrg. Math.*, 56:247–262, 2006.
- [81] T. C. Warburton and G. E. Karniadakis. A discontinuous Galerkin method for the viscous MHD equations. *J. Comput. Phys.*, 152(2):608–641, 1999.
- [82] P. Woodward and P. Colella. The numerical simulation of two-dimensional fluid flow with strong shocks. *J. Comput. Phys.*, 54:115–173, 1984.
- [83] J. Yan and C.-W. Shu. A local discontinuous Galerkin method for KdV type equations. *SIAM J. Numer. Anal.*, 40(2):769–791, 2002.
- [84] J. Yan and C.-W. Shu. Local discontinuous Galerkin methods for partial differential equations with higher order derivatives. *J. Sci. Comput.*, 17(1-4), 2002.
- [85] J. Zhu, J. Qiu, C.-W. Shu, and M. Dumbser. Runge-Kutta discontinuous Galerkin method using WENO limiters II: Unstructured meshes. *J. Comput. Phys.*, 227:4330–4353, 2008.



## List of Tables

2.1	Stability numbers for the diffusion flux approximation . . . . .	36
2.2	Stability numbers $\alpha$ for the pure one dimensional scalar and linear advection case. . . . .	44
2.3	Stability numbers $\beta$ for the pure one dimensional scalar and linear diffusion case. . . . .	44
3.1	Experimental order of convergence for $p = 3$ and $p = 4$ . . . . .	51
3.2	Experimental order of convergence for $p = 4$ and $p = 5$ with $\ell = (0, 0)$ and $t_{end} = 1.0$ . . . . .	53
3.3	Results for different types of integration points for the hexahedra with $p = 6$ . The domain $\Omega$ is subdivided into 8 hexahedra. . . . .	54
3.4	Results for different types of integration points for $p = 6$ pyramids. The domain $\Omega$ is subdivided into 6 pyramids. . . . .	54
3.5	Results for different types of integration points for $p = 6$ prisms. The domain $\Omega$ is subdivided into 8 hexahedra which are further subdivided into 2 prisms, yielding 16 grid cells. . . . .	55
3.6	Results for different types of integration points for $p = 6$ tetrahedra. The domain $\Omega$ is subdivided into 6 tetrahedra. . . . .	55
3.7	Experimental order of convergence for $p = 3$ (10 DOF per grid cell) for reference test on primal triangular mesh and for test configuration A. . . . .	57
3.8	Experimental order of convergence for $p = 3$ (10 DOF per grid cell) for test configuration B and C. . . . .	58
3.9	CPU times for the three dimensional compressible Navier-Stokes equations with $p = 6$ . . . . .	58
3.10	Comparison of local time stepping (LTS) and global time stepping (GTS). . . . .	60
3.11	Results of the scale-up test with constant load per processor (153664 DOF). . . . .	61
3.12	Force coefficients and Strouhal number. . . . .	78

A.1	Lebesgue constants $\Lambda$ and number of interpolation points $n_L$ for the two dimensional interpolation points. . . . .	85
A.2	Lebesgue constants $\Lambda$ and number of interpolation points $n_L$ for the three dimensional interpolation sets with different parameters $\ell = (\ell_{3D}, \ell_{2D})$ . . . . .	86
A.3	Lebesgue constants $\Lambda$ and number of interpolation points $n_L$ for the three dimensional interpolation sets with different parameters $\ell = (\ell_{3D}, \ell_{2D})$ . . . . .	87
B.1	Coefficients $b_{oj}^t$ with $p_t = 0$ with $n_{stages} = 1$ . . . . .	89
B.2	Coefficients $a_{jl}$ with $p_t = 1$ with $n_{stages} = 2$ . . . . .	89
B.3	Coefficients $b_{oj}^t$ with $p_t = 1$ with $n_{stages} = 2$ . . . . .	89
B.4	Coefficients $a_{jl}$ with $p_t = 2$ with $n_{stages} = 4$ . . . . .	90
B.5	Coefficients $b_{oj}^t$ with $p_t = 2$ with $n_{stages} = 4$ . . . . .	90
B.6	Coefficients $a_{jl}$ with $p_t = 3$ with $n_{stages} = 6$ . . . . .	90
B.7	Coefficients $b_{oj}^t$ with $p_t = 3$ with $n_{stages} = 6$ . . . . .	90
B.8	Coefficients $a_{jl}$ with $p_t = 4$ with $n_{stages} = 8$ . . . . .	91
B.9	Coefficients $b_{oj}^t$ with $p_t = 4$ with $n_{stages} = 8$ . . . . .	91

## List of Figures

2.1	Quadrilateral with $p_L = 9$ ( $\ell_{2D} = 0$ ). From left to right: pure equidistant distribution, LGL points with equidistant nesting, LGL points with LGL-type nesting and optimized points. . . .	15
2.2	Sequence of steps 1-4 of a computation with 3 different elements and local time-stepping . . . . .	47
3.1	Double logarithmic plot of $L_2$ error versus the polynomial order for different element types and grids. . . . .	52
3.2	Primal and dual mesh ( $h = 0.1$ ) and detailed view of the interpolation grid ( $h = 0.025$ ) with $p = 3$ ( $\pi_{2D} = 0$ ) interpolation. .	56
3.3	Detailed view of the interpolation grid ( $h = 0.025$ ) for $p = 3$ approximation with $\ell_{2D} = 3$ or $r_{max} = 1$ . . . . .	56
3.4	Computational grid with 8448 grid cells. . . . .	59
3.5	Computational setup for the single jet example. All length scales are in meters. . . . .	61
3.6	Density distribution calculated with $p = 1$ DG scheme (left) and CFX (right). . . . .	62
3.7	30 equally spaced density contour lines from $\rho = 1.5 \dots 21.5$ . left: DG scheme with $h = 1/120$ and $p = 4$ . right: 5th order WENO-FV scheme with $h = 1/120$ . . . . .	65
3.8	30 equally spaced density contour lines from $\rho = 1.5 \dots 21.5$ . left: DG scheme with $h = 1/120$ and $p = 4$ . right: 5th order WENO-FV scheme with $h = 1/480$ . . . . .	66
3.9	Maximum amplitudes of $v_1$ (left). Amplification rate $\alpha_i$ of $u_1$ based on maximum amplitude (right). . . . .	69
3.10	Computational grid for mixing layer simulation. . . . .	70
3.11	Mixing Layer. Vorticity contours in near-field mixing region at $\frac{t}{T} = 68$ from contour level $-0.26$ to $0.02$ with increment $0.04$ are shown. The result from Colonius et al. [24] is shown below. . .	71
3.12	Streamwise velocity $u_1$ as a function of time at the axes $y = 0$ and at the various streamwise locations $x = 30, \dots, 315$ (top down). .	72

3.13	Maximum amplitudes of $u_2$ (left). Amplification rate $\alpha_i$ of $u_2$ based on maximum amplitude (right). . . . .	73
3.14	Projection of the computational grid on the $x_1 - x_3$ -plane at $x_2 = 0$ . . . . .	74
3.15	Zoomed views of the projected sphere example grid. . . . .	75
3.16	Distribution of local polynomial degree $p$ at $t_{end} = 1000$ . . . . .	76
3.17	Visualization of the tetrahedra grid for the sphere example. . . . .	77
3.18	Isometric view of instantaneous $\lambda_2$ isosurface for the $Re = 300$ case. . . . .	78
3.19	Distribution of the pressure coefficient and limiting streamlines on the sphere surface. . . . .	79
3.20	Visualization of the grid in $x_1 - x_3$ -plane at $x_2 = 0$ . . . . .	79
3.21	Detailed visualization of the grid. . . . .	80
3.22	Isometric view of instantaneous $\lambda_2$ isosurface for the $Re = 1000$ case. . . . .	81
3.23	Distribution of instantaneous $\omega_2$ on a $x_1 - x_3$ -plane at $x_2 = 0$ . . . . .	81
C.1	Visualization of the different hybrid meshes. . . . .	93
E.1	3D isentropic vortex. Isosurfaces of density. . . . .	98

## Lebenslauf

06.03.1980	Geboren in Lustenau (A)
1986-1990	Volksschule Fussach (A)
1990-1998	Bundesgymnasium Blumenstraße, Bregenz (A)
1998	Allgemeine Hochschulreife (Matura)
1998-1999	Studium Astrophysik, Universität Innsbruck (A)
1999-2002	Studium Luft -und Raumfahrttechnik, Universität Stuttgart (D)
2002	Zulassung zum Parallelstudium Luft-und Raumfahrttechnik/Mathematik
2002-2005	Studium Luft-und Raumfahrttechnik Studium Mathematik
2005	Diplom Luft-und Raumfahrttechnik Diplom Mathematik
2005	Wissenschaftlicher Mitarbeiter am Mathematisches Institut D der Technische Universität Graz (A)
2005 - 2008	Wissenschaftlicher Mitarbeiter am Institut für Aero-und Gasdynamik der Universität Stuttgart (D)
2008	Dissertation: <i>Discontinuous Galerkin Methods for the Un- steady Compressible Navier-Stokes Equations</i>

Stuttgart, September 2008

Gregor Gassner