

Entropy Stable Summation-by-Parts Methods for Hyperbolic Conservation Laws on h/p Non-Conforming Meshes

INAUGURAL - DISSERTATION

zur

Erlangung des Doktorgrades

der Mathematisch-Naturwissenschaftlichen Fakultät

der Universität zu Köln

vorgelegt von

Lucas Friedrich

aus Bonn

Köln, 2019

Berichterstatter: Prof. Dr. Gregor Gassner
Prof. Dr. Thomas Sonar

Tag der mündlichen Prüfung: 18.01.2019

Abstract

In this work we present high-order primary conservative and entropy stable schemes for hyperbolic systems of conservation laws with geometric (h) and algebraic (p) non-conforming rectangular meshes. Throughout we rely on summation-by-parts (SBP) operators which discretely mimic the integration-by-parts rule to construct stable approximations. Thus, the discrete proofs of primary conservation and entropy stability can be done in a one-to-one fashion to the continuous analysis. Here, we consider different SBP operators based on finite difference as well as discontinuous Galerkin approaches.

We derive non-conforming schemes by extending ideas of high-order primary conservative and entropy stable SBP methods on conforming meshes. Here, special attention is given to the coupling between non-conforming elements. The coupling is instructed to entropy stable projection operators. However, these projection operators suffer from a suboptimal degree. Therefore, we develop degree preserving SBP operators where the norm matrix has a higher degree compared to classical SBP operators. With these operators it is possible to construct entropy stable projection operators which have the same degree as the SBP differentiation matrix.

Typically, high-order primary conservative and entropy stable schemes are semi-discrete methods with a discretized spatial domain and assuming continuity in time. Therefore, temporal errors are introduced when integrating the conservation laws in time with standard methods, e.g. Runge-Kutta schemes, for which the entropy can have an unpredictable temporal behaviour. Thus, we extend high-order primary conservative and entropy stable semi-discrete methods to fully-discrete schemes on conforming and non-conforming meshes. This results in an implicit space-time method.

We introduce a simple mesh generation strategy to obtain quadrilateral meshes surrounding two dimensional complex geometries. Finally, with the generated meshes we simulate a flow around a NACA0012 air foil to demonstrate the benefits of considering non-conforming elements for a practical simulation.

Kurzzusammenfassung

In dieser Arbeit präsentieren wir primär konservative und Entropie stabile Verfahren hoher Ordnung zum Lösen von Systemen von hyperbolischen Erhaltungsgleichungen auf geometrischen (h) und algebraischen (p) nicht-konformen rechteckigen Gittern. Dabei verwenden wir ausschließlich summation-by-parts (SBP) Operatoren, welche die partielle Integration diskret nachahmen um stabile Approximationen zu konstruieren. Dadurch können die Beweise von primär konservativen und Entropie stabilen Verfahren analog zur kontinuierlichen Analyse durchgeführt werden. In dieser Arbeit betrachten wir verschiedene SBP Operatoren basierend auf finite Differenzen sowie auf unstetigen Galerkin Ansätzen.

Wir leiten nicht-konforme Verfahren her indem wir die Ansätze von primär konservativen und Entropie stabilen SBP Verfahren auf konformen Gittern erweitern. Besondere Aufmerksamkeit ist der Kopplung von nicht-konformen Gittern gewidmet. Die Kopplung ist angewiesen auf Entropie stabile Projektionsoperatoren. Diese Projektionsoperatoren haben allerdings einen suboptimalen Grad. Dadurch entwickeln wir Grad erhaltene SBP Operatoren, wo die Norm Matrix eine höhere Genauigkeit als herkömmliche SBP Operatoren aufweist. Mit diesen Operatoren können wir Entropie stabile Projektionsoperatoren konstruieren welche denselben Grad haben wie die SBP Differtiationsmatrix.

Gewöhnlich sind primär konservative und Entropie stabile Verfahren semi-diskret mit einer räumlichen Diskretisierung. Die Zeit hingegen wird als kontinuierlich angenommen. Dadurch ergeben sich temporäre Fehler durch die Zeitintegration der Erhaltungsgleichungen mit Standardverfahren, wie z.B. Runge-Kutta Verfahren, wodurch die Entropie einen unberechenbaren Verlauf in der Zeit annehmen kann. Daher erweitern wir primär konservative und Entropie stabile semi-diskrete Methoden hoher Ordnung zu voll-diskreten Verfahren auf konformen und nicht-konformen Gittern. Die resultierenden Verfahren sind implizite Raum-Zeit Approximation. Wir führen eine simple Strategie zur Generierung von viereckigen Gitterelementen ein um zwei dimensionale komplexe Geometrien zu umgeben. Abschließend simulieren wir mit diesen Gittern die Strömung um einen NACA0012 Tragflügel. Mit dieser Simulationen demonstrieren wir die Vorteile von nicht-konformen Gittern für praktische Anwendungen.

Danksagung

Ich möchte mich bei meinem Betreuer Gregor Gassner für die Möglichkeit zum Promovieren und das entgegengebrachte Vertrauen bedanken. Neben der Forschung an interessanten Themen hatte ich das Vergnügen in einer Arbeitsgruppe zu arbeiten die arbeitstechnisch als auch privat toll harmonierte. Ich habe die letzten vier Jahre in der Arbeitsgruppe sehr genossen und werde die Arbeit als auch die Mittagspausen, Feierabendbiere und die intensiven sowie wahnsinnig stumpfsinnigen Gespräche sehr vermissen. Zusätzlich möchte ich mich bei David del Rey Fernández für die langjährige Kooperation und Freundschaft bedanken.

Großer Dank geht an meine Eltern, Jenny und Wolfgang, für Ihre Liebe, Rückhalt sowie finanzielle Unterstützung. Ohne euch hätte ich nicht die Gelegenheit zum Studieren gehabt und hätte daher diesen Berufsweg niemals einschlagen können. Zudem möchte ich mich bedanken bei meinen Brüdern, Andreas und Christian, mit denen ich immer viel zu lachen habe; meinen Brüser Jungs, Sven, Simon, Bengt, Manu und Max, die mich durch so viele Lebensabschnitte begleitet haben; meine Studienkollegen, Vincent, Johannes und Marvin, die mir eine super Studienzeit beschert haben; den TTC Duisdorf für den sportlichen Ausgleich als auch meine Schwiegermutter Gerda, die mehrfach auf Fiona aufgepasst hat und mir damit mehr Zeit einräumte diese Arbeit anzufertigen.

Der nächste Dank geht an einen Menschen der aktuell nicht mal einen Meter groß ist, aber dennoch mein Leben in so kurzer Zeit so wunderschön verändert hat; meine Tochter Fiona. Du hast mich erkennen lassen was im Leben wirklich wichtig ist. Es gibt nichts Schöneres als wenn ich nach Hause komme und ich dich sehe wie du mit einem Lächeln im Gesicht in meine Arme gelaufen kommst. Danke.

Zu guter Letzt gilt der größte Dank meiner besseren Hälfte und Ehefrau Franzi. Du hast beruflich als auch privat immer zu mir gehalten und uns beiden das größte Glück auf Erden geschenkt. Ich kann dir dafür nicht genug danken. Die letzten Jahre mit dir waren so wunderschön und ich freue mich wahnsinnig auf alle weiteren Jahre die unserer kleinen Familie noch bevorstehen. Ich liebe dich!

Contents

Abstract	iii
Kurzzusammenfassung	iv
Danksagung	v
List of Symbols and Abbreviations	x
1. Introduction	1
1.1. Goals of the Dissertation	3
1.2. Outline of the Dissertation	4
2. Energy Stable and Primary Conservative Summation-by-Parts Methods for the Linear Advection Equation on Conforming Meshes	5
2.1. Summation-by-Parts Methods for the One Dimensional Linear Advection Equation	7
2.1.1. Summation-by-Parts Operators	8
2.1.2. Summation-by-Parts Methods with Simultaneous-Approximation-Term	10
2.1.3. Discontinuous Galerkin Spectral Element Method	20
2.1.4. Hybrid Gauss-Trapezoidal-Lobatto Operators	28
2.2. Summation-by-Parts Methods for the Two Dimensional Linear Advection Equation	30
2.3. Numerical Verification of Summation-by-Parts Schemes for the Two Dimensional Linear Advection Equation	42
2.3.1. Comparison and Convergence of Summation-by-Parts Schemes	43
2.3.2. Numerical Verification of Primary Conservation and Energy Stability	47
2.4. Summary	50
3. Energy Stable and Primary Conservative Summation-by-Parts Methods for the Linear Advection Equation on Non-Conforming Meshes	51
3.1. History on Non-Conforming Summation-by-Parts Methods	54

3.2.	Construction of M-Compatible Projection Operators	59
3.2.1.	Projection Operators for Discontinuous Galerkin	59
3.2.2.	Projection Operators for Finite Difference and Hybrid Gauss-Trapezoidal-Lobatto	63
3.2.3.	Analysis of the Suboptimal Degree of the Projection Operators	65
3.3.	Degree Preserving Operators	66
3.4.	Non-Conforming Discrete Analysis for the Linear Advection Equation	68
3.4.1.	Non-Conforming Nodal Distributions	69
3.4.2.	Non-Conforming Interfaces with Hanging Corners	71
3.4.3.	Including Dissipation within the Numerical Surface Flux	75
3.5.	Numerical Verification of Non-Conforming Summation-by-Parts Schemes for the Two Dimensional Linear Advection Equation	78
3.5.1.	Comparison and Convergence of Summation-by-Parts Schemes	79
3.5.2.	Numerical Verification of Primary Conservation and Energy Stability	83
3.6.	Summary	85
4. Entropy Stable and Primary Conservative Summation-by-Parts Methods for Non-Linear Hyperbolic Systems on Conforming Meshes		87
4.1.	Entropy Stable Summation-by-Parts Methods for One Dimensional Hyperbolic Systems of Conservation Laws	90
4.1.1.	Skew-Symmetric Discretization	91
4.1.2.	Summation-by-Parts Flux Difference Scheme	93
4.2.	Entropy Stable Summation-by-Parts Methods for Two Dimensional Hyperbolic Systems of Conservation Laws	100
4.3.	Numerical Verification of Summation-by-Parts Schemes for Non-Linear Problems	111
4.3.1.	Comparison and Convergence of Summation-by-Parts Schemes	112
4.3.2.	Numerical Verification of Primary Conservation and Entropy Stability	114
4.4.	Summary	119
5. Entropy Stable and Primary Conservative Summation-by-Parts Methods for Non-Linear Hyperbolic Systems on Non-Conforming Meshes		120
5.1.	Non-Conforming Discrete Analysis for Non-Linear Problems	122
5.1.1.	Non-Conforming Nodal Distributions	122
5.1.2.	Non-Conforming Interfaces with Hanging Corners	126
5.1.3.	Including Dissipation within the Numerical Surface Flux	130

5.2. Numerical Verification of Non-Conforming Summation-by-Parts Schemes for Non-Linear Problems	134
5.2.1. Comparison and Convergence of Summation-by-Parts Schemes for Non-Linear Problems	135
5.2.2. Numerical Verification of Primary Conservation and Entropy Stability	138
5.3. Summary	139
6. Entropy Stable and Primary Conservative Summation-by-Parts Space-Time Methods for Non-Linear Hyperbolic Systems	142
6.1. Discrete Analysis for Summation-by-Parts Space-Time Methods . . .	143
6.1.1. Review of Finite Volume Methods	143
6.1.2. Summation-by-Parts Space-Time Methods	147
6.1.3. Extension to Non-Conforming Elements	157
6.1.4. Implicit Solver	158
6.2. Numerical Verification of Summation-by-Parts Space-Time Schemes	159
6.2.1. Comparison and Convergence of Summation-by-Parts Space-Time Schemes	159
6.2.2. Numerical Verification of Primary Conservation and Entropy Stability	164
6.3. Summary	166
7. Mesh Generation and Applications	168
7.1. Mesh Generator Implementation	172
7.1.1. Quadtree Data Structure	172
7.1.2. Final-Cut Domain Decomposition	174
7.1.3. Element Merging	176
7.1.4. Two-to-One Coupling	179
7.1.5. Non-Conforming Meshes for Complex Geometries	180
7.2. Application: Flow around a NACA0012 Air Foil	183
7.3. Summary	187
8. Global Summary	189
8.1. Conclusion	191
8.2. Outlook	192
A. Appendix	1
A.1. Primary Conservative and Energy Stable Discretization for the Linear Advection Equation on Conforming Curved Elements	1

Contents

A.2. Primary Conservative and Entropy Stable Discretization for Hyperbolic Systems of Conservation Laws on Conforming Curved Elements 9

Erklärung 30

Curriculum Vitae 31

Publikationsliste 32

List of Symbols and Abbreviations

In this work continuous functions are denoted by capital letters with a script type, e.g. \mathcal{A} . Discrete approximations of continuous functions are given by lower case letters, e.g. a . Vectors are denoted with bold letters, e.g. $\mathbf{a} = [a_0, \dots, a_N]^T$, while matrices are presented using capital letters with sans-serif font, e.g. \mathbf{A} .

Alphanumeric Symbols

$\mathbf{1}$	Vector of ones
a, b	Wave speed of the linear advection equation
\tilde{a}, \tilde{b}	Contravariant wave speed of the linear advection equation
\tilde{a}^p, \tilde{b}^p	Approximated contravariant wave speeds of degree p
bn	Number of boundary nodes
\mathcal{B}	Boundary condition
\mathbf{B}	Boundary matrix with $\mathbf{B} = \text{diag}(-1, 0, \dots, 0, +1)$
B_r	The r -th Bernoulli polynomial
$B_L E_t, B_R E_t$	Discrete energy growth on the boundaries
CFL	Courant-Friedrichs-Lewy number
\mathbf{D}	Differentiation matrix
$(DS), (DT)$	Space-time discretization is space and time, respectively
$(Dx), (Dy)$	Spatial discretization is x - and y -direction, respectively
\mathbf{e}_0	Standard basis vector with $\mathbf{e}_0 = (1, 0, \dots, 0)$
\mathbf{e}_N	Standard basis vector with $\mathbf{e}_N = (0, \dots, 0, 1)$
E	Element
\mathcal{E}	Total energy
EC	Denotes a flux is entropy conservative
ES	Denotes a flux is entropy stable
E_t^{tot}	Total energy growth
f, g	Discrete physical fluxes
\tilde{f}, \tilde{g}	Discrete contravariant physical fluxes
f^*, g^*	Discrete numerical surface fluxes
\tilde{f}^*, \tilde{g}^*	Discrete numerical contravariant surface fluxes
$f^\#, g^\#$	Discrete numerical volume fluxes
$\tilde{f}^\#, \tilde{g}^\#$	Discrete numerical contravariant volume fluxes

List of Symbols and Abbreviations

f^{Ent}	Discrete entropy flux function
\mathcal{F}, \mathcal{G}	Physical fluxes
$\tilde{\mathcal{F}}, \tilde{\mathcal{G}}$	Physical contravariant fluxes
\mathcal{F}^{ent}	Entropy flux function
h	Water height of the Shallow Water equations
\mathcal{H}	Implicit system for space-time discretization
\mathbf{I}	Identity matrix
IE_t	Discrete energy growth on an interface
IS_t	Discrete entropy growth on an interface
IU_t	Discrete growth of the primary quantities on an interface
J	Jacobian determinant
\mathbf{K}	Dissipation matrix
L	Denotes the left element
\mathbf{L}	Interpolation operator
\mathbf{M}	Diagonal symmetric positive definite norm matrix
Ma	Mach number
$\mathbf{n} = (n_1, n_2)$	Normal vector in two dimensional space
$N + 1$	Number of nodes on an element in one dimension
N_Q	Number of elements
N_t	Number of elements in t -direction
N_{Q_x}	Number of elements in x -direction
N_{Q_y}	Number of elements in y -direction
NE	Number of elements on an interface with hanging corners
$\mathcal{O}(\cdot)$	Error estimate
p	Polynomial degree
\mathcal{P}	Pressure
\mathbf{P}	Projection operator
\mathbf{Q}	Skew-symmetric SBP matrix with $\mathbf{Q} = \mathbf{M}\mathbf{D}$
R	Denotes the right element
Res	Residuum to validate primary conservation and entropy stability
S	Discrete entropy function
\mathcal{S}	Entropy function
\mathbf{S}	Surface matrix
SAT	Simultaneous approximation term
S_t^{tot}	Total entropy growth
t	Time in physical space
T	Final time
u	Numerical solution of a scaler conservation law
\mathbf{u}	Numerical solution of a system of conservation laws

List of Symbols and Abbreviations

u^*	Numerical surface state in time
$u^\#$	Numerical volume state in time
\mathcal{U}	Solution of a scalar conservation law
\mathbf{U}	Solution of a system of conservation laws of size N_{Eq}
\mathcal{U}_0	Initial condition
U_t^{tot}	Total growth of the primary quantities
\mathbf{v}	Discrete Entropy variables
\mathcal{V}	Entropy variables
VP	Volume part
x, y	Coordinates in physical space
Z_Q, Z_e, Z_{opt}	Optimized values for the construction of operators
γ	Adiabatic coefficient for the Euler equations
Γ	Interface curves describing the surface of a curved domain
Δ	Distances
Δ	Differencing matrix
ζ	Source term for a manufactured solution
ϑ	Random generated numbers
ν	Velocity
λ	Eigenvalue of the Jacobian flux
ξ, η	Coordinates in reference space
ξ, η	Nodal distribution in the reference space
Ξ	Denotes the Mortar element
ρ	Density
σ	Dissipation rate
τ	Time in reference space
v_B	Interface function defined on the boundary of Ω
φ	Basis function for variational formulation
ϕ	Test function for variational formulation
Φ	Function detecting badly shaped elements
Φ_{min}	Upper bound detecting badly shaped elements
χ	Two dimensional mapping function
Ψ^u	Entropy potential
Ψ^f, Ψ^g	Entropy flux potential
ω	Quadrature weights
Ω	Physical domain
$\hat{\Omega}$	Reference space

$(\mathcal{U}, \mathcal{W})_{\mathcal{L}_2(\Omega)}$	The \mathcal{L}_2 inner product
$\ \mathcal{U}\ _{\mathcal{L}_2(\Omega)}$	The \mathcal{L}_2 norm induced by $(\mathcal{U}, \mathcal{U})_{\mathcal{L}_2(\Omega)}$
$\langle \mathbf{u}, \mathbf{v} \rangle_{\mathbb{M}}$	Discrete inner product approximated with \mathcal{L}_2 inner product
$\ \mathbf{u}\ _{\mathbb{M}}$	The discrete norm induced by $\langle \mathbf{u}, \mathbf{u} \rangle_{\mathbb{M}}$
$\ \mathbf{u}\ _2$	Euclidean vector norm
$\{u\}$	Average of two states of u

Abbreviations

CFL	Courant-Friedrichs-Lewy
CPR	Correction procedure via reconstruction
DG	Discontinuous Galerkin
DGSEM	Discontinuous Galerkin Spectral Element Method
DOFS	Degrees of freedom
DP	Degree Preserving
EC	Entropy conservative
EOC	Experimental order of convergence
ES	Entropy stable
FD	Finite difference
FEM	Finite element method
FV	Finite volume
GMRES	Generalized minimal residual
HGTL	Hybrid Gauss-Trapezoidal-Lobatto
HOPR	High Order Preprocessor
IBP	Integration-by-parts
LAE	Linear advection equation
LG	Legendre-Gauss
LGL	Legendre-Gauss-Lobatto
MHD	Magnetohydrodynamics
ODE	Ordinary Differential Equation
PDE	Partial Differential Equation
SBP	Summation-by-parts
SAT	Simultaneous-Approximation-Term

1. Introduction

In large scale applications the effective use of computational resources is an important aspect of emerging numerical methods. In particular the distribution of the degrees of freedom are of special interest. This makes numerical methods a powerful tool for modelling experiments which are either too expensive or impossible to perform. As the power of computers has increased drastically over the last decades, we have come to rely on numerical schemes to simulate real world problems based on mathematical models.

Systems of *conservation laws* cover a broad range of numerical applications as they simulate the behaviour of numerous fluids. Conservation laws consists of partial differential equations tracking the fluid behaviour and evolution over time on a given spatial domain. Their solutions are quantities such as density or velocity. For conservation laws the conserved quantities can only change in time depending on the fluid flow into or out of the system. This is referred to as *primary conservative*.

For approximating the solution of partial differential equations there is a broad range of numerical methods based on finite differences, finite volumes or finite elements. Especially focusing on conservation laws high-order numerical methods have been constructed to solve the mathematical model and at the same time to discretely mimic primary conservation. Still, such numerical schemes do not necessarily represent the physical behaviour of a fluid correctly. A possible explanation for this phenomenon is that the scheme does not discretely obey the second law of thermodynamics. Therefore, over the past years numerical schemes have been developed which are primary conservative and at the same time discretely obey the second law of thermodynamics. Such schemes are denoted as *entropy stable*.

In 1993 Carpenter et al. [12] constructed high-order, primary conservative and energy (linear entropy) stable schemes for linear conservation laws. The method makes use of *summation-by-parts* operators based on finite differences. Such operators were first designed in 1974 by Kreiss and Scherer [72] and are constructed to discretely mimic the integration-by-parts rule. Therefore, the discrete analysis of primary conservative and energy stable schemes is done in a one-to-one fashion to the continuous analysis. Nowadays, numerous summation-by-parts operators

exist, e.g. operators based on discontinuous Galerkin approaches [42]. Therefore, the method of Carpenter et al. can be interpreted as a finite difference, discontinuous Galerkin or any other numerical approach by simply considering different summation-by-parts operators. Due to the variety, summation-by-parts operators had a great impact on numerically solving conservation laws in a stable manner. Schemes which rely on the use of summation-by-parts operators are denoted as *summation-by-parts methods*.

Designing primary conservative and entropy stable methods for non-linear problems is a complicated task. Back in 1984, Tadmor was the first to create such a scheme based on low-order finite volume methods [112]. Extension of this work to high-order entropy stable schemes with summation-by-parts operators have been developed relying on split forms, e.g. [42]. However, depending on the conservation law, finding such a split form can be cumbersome. A breakthrough on high-order entropy stable methods is the *Flux Difference* scheme developed by Fisher and Carpenter [33] in 2013. Here, the method of Tadmor for arbitrary conservation laws is extended to a general high-order scheme using summation-by-parts operators without considering any split forms.

All these introduced schemes rely on semi-discrete discretizations. Here, we focus on a spatial discretization and assume continuity in time. In order for the numerical solution to discretely obey the second law of thermodynamics we need to consider a small time step. A recent work by Friedrich et al. [38] expanded the high-order primary conservative and entropy stable Flux Difference method to a fully-discrete scheme without assuming exact quadrature. Thus, the physical law is not violated despite the time step.

When applying numerical schemes to real world problems the choice and design of the mesh is an important aspect. Here, we consider multi-block meshes with quadrilateral elements, where each element has its own nodal distribution. Entropy stable methods like the Flux Difference scheme are typically constructed on a conforming mesh. For such meshes the nodal distribution of each element and the element interfaces need to coincide. However, for certain problems one is interested in refining the mesh resolution in certain regions, e.g. at the surface of an air foil to observe the fluid behaviour. Obtaining such conforming quadrilateral meshes is cumbersome or even impossible. Further mesh flexibility is introduced by considering *non-conforming* elements. Here, neither the nodes nor the interfaces of the elements must coincide. For such meshes we can arbitrary refine the resolution of an element independent of the neighbor elements. In addition, as the

interfaces of elements do not necessarily have to coincide, it is easier to generate non-conforming meshes for simulating flows around complex geometries. The main question of this work is:

Can we maintain convergence order, primary conservation and entropy stability of a summation-by-parts scheme when focusing on non-conforming meshes?

Investigations on this topic have been done over the last years and will be presented in this work.

1.1. Goals of the Dissertation

The main goal of this dissertation is to propose high-order primary conservative and entropy stable summation-by-parts schemes on non-conforming meshes. Focusing on linear conservation laws Mattsson and Carpenter [83] were the first to derive such a scheme. Their method consists of projection operators connecting the non-conforming elements. In terms of energy stability the projection operators satisfy certain stability and accuracy conditions. However, due to these conditions the projection operators suffer from a suboptimal degree. Therefore, the method loses up to one order of accuracy compared with a conforming discretization. To overcome this order loss we present a new set of *degree preserving* summation-by-parts operators such that the resulting non-conforming scheme is of the same order as the conforming scheme.

Moving on to non-linear problems, the theory on entropy stable methods on non-conforming meshes is rare. In this dissertation we present the non-conforming high-order primary conservative and entropy stable method by Friedrich et al. [38]. The authors were the first to derive such a scheme on Cartesian meshes where neither the nodes nor the interfaces need to coincide. The key concept of this method is to combine the conforming Flux Difference method of Fisher and Carpenter for non-linear problems with the non-conforming summation-by-parts method by Mattsson and Carpenter for linear problems. Additionally, in comparison to linear conservation laws, a special modification of the term coupling the non-conforming elements needs to be considered.

Besides focusing on non-conforming elements in space for, e.g. simplifying the mesh generation, we apply non-conforming elements in time on fully-discrete schemes. Thus, we can consider different time steps for different elements gaining additionally flexibility and at the same time still ensuring primary conservation

and entropy stability.

Finally, in order to demonstrate the advantageous features of non-conforming meshes, we present a simple adaptive cut cell strategy to generate non-conforming meshes around complex geometries. Therefore, we simulate a flow around a NACA0012 air foil on a conforming and a non-conforming mesh and compare the results.

Parts of this dissertation have been published [38, 39] or are under review [40] in the Journal of Scientific Computing .

1.2. Outline of the Dissertation

In Chapter 2 we introduce energy stable and primary conservative summation-by-parts methods focusing on the one and two dimensional linear advection equation. Here, we demonstrate the features of summation-by-parts operators and introduce different operators derived from finite differences and discontinuous Galerkin approaches. In Chapter 3 we extend the energy stable and primary conservative method to a non-conforming mesh by introducing the method of Mattsson and Carpenter [83]. Here, special attention is paid to the construction of projection operators connecting non-conforming elements. As the operators have a suboptimal degree we introduce degree preserving summation-by-parts operators.

After introducing the linear theory we move on to non-linear conservation laws and introduce the concept of entropy in Chapter 4. We demonstrate the additional issues for non-linear problems and derive high-order entropy stable schemes. Here, we discuss split forms of conservation laws and introduce the Flux Difference scheme by Fisher and Carpenter [34]. In Chapter 5 we answer the question in the introduction and present the high-order primary conservative and entropy stable scheme on a non-conforming mesh by Friedrich et al. [38].

Furthermore we introduce a high-order primary conservative and entropy stable scheme based on a fully-discrete discretization in Chapter 6. This method can be applied to conforming and non-conforming elements. Finally, we discuss the generation of meshes in Chapter 7 and present numerical simulations.

Within each chapter, besides Chapter 7, proofs occur. Theoretical findings and proofs will be numerically verified at the end of each chapter.

2. Energy Stable and Primary Conservative Summation-by-Parts Methods for the Linear Advection Equation on Conforming Meshes

We are interested in solving hyperbolic conservation laws. To start off we focus on one dimensional scalar conservation laws

$$\frac{\partial \mathcal{U}}{\partial t} + \frac{\partial \mathcal{F}(\mathcal{U})}{\partial x} = 0, \quad (2.1)$$

with $t \in [0, T] \subset \mathbb{R}$, $x \in \Omega = [x_L, x_R] \subset \mathbb{R}$ and appropriate initial and boundary conditions. Throughout this thesis we denote \mathcal{U} as the *conservative variable* and \mathcal{F} as the *physical flux*. Conservation laws are *primary conservative*, in a sense that the conserved solution can only change in time depending on the fluid flow into or out of the system. Mathematically speaking, the time derivative of the integral of \mathcal{U} over Ω only depends on its boundary conditions and so is conserved over time within the domain Ω . This can be verified by applying Gauss' law on (2.1)

$$\int_{\Omega} \frac{\partial \mathcal{U}}{\partial t} d\Omega = v_B(\tau), \quad (2.2)$$

where $v_B(\tau) = -\mathcal{F}(\mathcal{U}(\tau, \xi))|_{\xi=x_L}^{x_R}$ is purely defined on the boundary. This holds for all hyperbolic conservation laws independent of the choice of the physical flux [31].

In this chapter we set the flux to be $\mathcal{F}(\mathcal{U}) = a\mathcal{U}$ which gives us the *linear advection equation (LAE)*

$$\frac{\partial \mathcal{U}}{\partial t} + a \frac{\partial \mathcal{U}}{\partial x} = 0, \quad (2.3)$$

where $a > 0$ describes the constant wave speed. The LAE is the simplest prototype of a hyperbolic conservation law. Later, in chapter 4, we focus on more general systems of hyperbolic equations.

Before attempting to solve (2.3) it is necessary to ensure that the model represents the physical behaviour of \mathcal{U} correctly or, in other words, that the model is *stable*. A mathematical model is said to be stable if its solution is bounded by its initial and boundary conditions. The choice of these conditions is an important component to ensure that the problem is well-posed [23]. For the LAE (2.3) these are defined by

$$\begin{aligned} \mathcal{U}(0, x) &= \mathcal{U}_0(x) & \text{for } x \in [x_L, x_R], \\ \mathcal{U}(t, x_L) &= \mathcal{B}(t) & \text{for } t \in [0, T]. \end{aligned} \quad (2.4)$$

Note, that the boundary conditions are one-sided as they are only defined on the left boundary x_L . Since the LAE transports information from left to right ($a > 0$) it is not instructed to a right boundary condition. Also, when considering a right boundary condition for the LAE, then the problem is not necessarily solvable [23].

The definition of *linear stability* is adapted from Gustaffson [50]:

Definition 1. Linear Stability

The problem (2.3) is linearly stable if the solution is bounded by its initial and boundary conditions

$$\|\mathcal{U}(t, \cdot)\|_{\mathcal{L}_2(\Omega)}^2 \leq \|\mathcal{U}_0\|_{\mathcal{L}_2(\Omega)}^2 + K(t) \int_0^t \mathcal{B}(\tau)^2 d\tau. \quad (2.5)$$

where $K(t) > 0$ is independent of the initial and boundary conditions and the norm $\|\cdot\|_{\mathcal{L}_2(\Omega)}$ is induced by the \mathcal{L}_2 inner product

$$(\mathcal{U}, \mathcal{W})_{\mathcal{L}_2(\Omega)} = \int_{\Omega} \mathcal{U}\mathcal{W} d\Omega, \quad \|\mathcal{U}\|_{\mathcal{L}_2(\Omega)}^2 = (\mathcal{U}, \mathcal{U})_{\mathcal{L}_2(\Omega)}. \quad (2.6)$$

Due to Definition 1 the solution is bounded and therefore it cannot "explode" in finite time. We will show that the LAE is stable by applying the energy method. Therefore, we multiply (2.3) with its solution \mathcal{U} and integrate over the domain Ω

$$\int_{\Omega} \mathcal{U} \frac{\partial \mathcal{U}}{\partial t} d\Omega + a \int_{\Omega} \mathcal{U} \frac{\partial \mathcal{U}}{\partial x} d\Omega = 0. \quad (2.7)$$

Applying the product rule and rearranging sides we get

$$\int_{\Omega} \frac{1}{2} \frac{\partial \mathcal{U}^2}{\partial t} d\Omega = -a \int_{\Omega} \frac{1}{2} \frac{\partial \mathcal{U}^2}{\partial x} d\Omega. \quad (2.8)$$

Now, we apply Leibniz rule on the left hand side and integrate the right hand side. By including the boundary condition we get

$$\frac{\partial}{\partial t} \|\mathcal{U}\|_{\mathcal{L}_2(\Omega)}^2 = -a \left(\mathcal{U}(t, x_R)^2 - \mathcal{B}(t)^2 \right). \quad (2.9)$$

As $\mathcal{U}(t, x_R)^2$ is non-negative, we get the following estimate

$$\frac{\partial}{\partial t} \|\mathcal{U}\|_{\mathcal{L}_2(\Omega)}^2 \leq a\mathcal{B}(t)^2. \quad (2.10)$$

Finally, by integrating in time over (2.10) and including the initial condition (2.4) we get

$$\|\mathcal{U}(t, \cdot)\|_{\mathcal{L}_2(\Omega)}^2 \leq \|\mathcal{U}_0\|_{\mathcal{L}_2(\Omega)}^2 + a \int_0^t \mathcal{B}(\tau)^2 d\tau, \quad (2.11)$$

which indicates that the LAE is stable. Equation (2.11) is referred as an *energy estimate*, as $\frac{1}{2}\mathcal{U}^2$ denotes the *energy* for the LAE.

Remark 1. The name *energy* can also be found in [24, 27, 111] and is used as in physical models the energy is typically described by a quadratic expression.

The total energy of the system is described by $\frac{1}{2} \|\mathcal{U}\|_{\mathcal{L}_2(\Omega)}^2$. In other words, the LAE is stable if its total energy is bounded by the initial and boundary conditions. In addition, the LAE is also declared as *energy stable*.

In this chapter we introduce numerical methods for hyperbolic conservation laws in one and two spatial dimensions. As for the continuous analysis above we desire schemes which are

- *Primary Conservative:* The numerical approximation must discretely recover the integral of its numerical solution over time (modulo the boundaries).
- *Energy Stable:* The numerical solution remains bounded by the initial and boundary conditions of the problem.
- *High-order:* The scheme can differentiate polynomials with high degree exactly; this implies that the order of the solution error is high-order.

Over the past years summation-by-parts (SBP) methods had a huge impact on solving hyperbolic equations. The main idea of these methods is to mimic the continuous analysis above with a set of discrete operators.

2.1. Summation-by-Parts Methods for the One Dimensional Linear Advection Equation

Back in 1974 Kreiss and Scherer laid the foundation for SBP methods [72]. They followed the idea of discretely applying the energy method to prove energy stability of their scheme. Therefore, they invented finite difference (FD) based *summation-by-parts Operators (SBP Operators)* which mimic the integration-by-parts (IBP)

rule. Thus, they were able to prove discrete stability for constant-coefficient and periodic problems without relying on differentiability or continuity of the numerical solution. The SBP operators are introduced in Section 2.1.1. Based on the work of Kreiss and Scherer, novel SBP methods including boundary conditions were presented by Strand [109], Olsson [93, 94] and Nordström and Carpenter [89]. As an extension of SBP methods, Carpenter, Gottlieb and Abarbanel [12] introduced the *Simultaneous-Approximation-Term (SAT)* to weakly impose boundary conditions and to consider energy stable multi-block SBP methods based on FD operators. This method is called the *SBP-SAT* scheme and is introduced in Section 2.1.2. Here, there are several extension of the SBP-SAT method to a large amount of two and three dimensional problems by considering special SATs [13, 15, 90]. In addition, the SBP-SAT method can be applied on unstructured meshes as in [25, 55, 56, 57].

Besides of the development of energy stable FD methods, discontinuous Galerkin (DG) methods had a major impact in solving hyperbolic systems, which was first introduced by Reed in 1973 [103]. This method can be interpreted as a combination of finite element and finite volume schemes as they are derived from the weak form of PDEs with basis functions and include numerical surface fluxes. The DG method is easily applicable to high-order applications by only changing the number of nodes within one element. Moreover, the DG method benefits from low dispersion and dissipation errors [2]. In Section 2.1.3 we introduce the derivations of the *Discontinuous Galerkin Spectral Element Method (DGSEM)* [54, 67].

Even though SBP-SAT FD methods and DGSEM follow a different motivation they have a lot in common. In 1996 Carpenter [11] proved that under given circumstances the differentiation operator of a collocation scheme is a SBP operator. In addition, Gassner [42] showed that the DGSEM can be interpreted as a SBP-SAT scheme when considering Legendre-Gauss-Lobatto nodes. This enables us to directly compare the FD and DG schemes with the SBP property within the same framework only by exchanging the SBP operator. In addition, we introduce a further SBP operator called *Hybrid Gauss-Trapezoidal-Lobatto (HGTL)* operator in Section 2.1.4 by Fernández [23]. With the SBP framework and operators we numerically verify the operator properties in Section 2.3.

2.1.1. Summation-by-Parts Operators

Next, we introduce SBP operators. In essence, the goal of SBP operators is to discretely mimic the IBP property [27, 71, 81, 82, 84]. Such operators are typically constructed in a reference space $\hat{\Omega} = [-1, +1]$, e.g. [39]. Therefore, we transform

the one dimensional conservation law (2.1) from $\Omega = [x_L, x_R]$ into $\hat{\Omega}$ with the affine map

$$\xi(x) = 2 \frac{x - x_L}{x_R - x_L} - 1. \quad (2.12)$$

The transformed PDE is represented by

$$\frac{\partial \mathcal{U}}{\partial t} + \frac{\partial \mathcal{F}(\mathcal{U})}{\partial x} = \frac{\partial \mathcal{U}}{\partial t} + \frac{\partial \mathcal{F}(\mathcal{U})}{\partial \xi} \frac{\partial \xi}{\partial x} = 0, \quad (2.13)$$

including $\frac{\partial \xi}{\partial x} = \frac{2}{\Delta x}$ with $\Delta x := x_R - x_L$ we get

$$\frac{\partial \mathcal{U}}{\partial t} + \frac{2}{\Delta x} \frac{\partial \mathcal{F}(\mathcal{U})}{\partial \xi} = 0. \quad (2.14)$$

For numerically solving (2.14) we consider discrete operators. Therefore, we introduce SBP operators.

Definition 2. Summation-by-Parts Operator

Given a set of $N + 1$ nodes $\boldsymbol{\xi} = [\xi_0, \dots, \xi_N]^T$ on the spatial domain $\hat{\Omega}$ with $\xi_0 = -1$ and $\xi_N = 1$. We define $\boldsymbol{\xi}^k := [\xi_0^k, \dots, \xi_N^k]^T$ with $k > 0$. A matrix operator $\mathbf{D} \in \mathbb{R}^{(N+1) \times (N+1)}$ is a SBP operator of degree p which approximates the first derivative $\frac{\partial}{\partial \xi}$ on the nodal distribution $\boldsymbol{\xi}$ in $\hat{\Omega}$, if

- $\mathbf{D} \boldsymbol{\xi}^k = \mathbf{M}^{-1} \mathbf{Q} \boldsymbol{\xi}^k = k \boldsymbol{\xi}^{k-1}$, $k = 0, 1, \dots, p$.
- \mathbf{M} denotes the norm matrix and is symmetric positive definite.
- $\mathbf{Q} + \mathbf{Q}^T = \mathbf{B} := \text{diag}(-1, 0, \dots, 0, 1)$.

We consider only SBP operators where the norm matrix \mathbf{M} is diagonal as this is indispensable later for proving stability for non-linear problems in Chapter 4. The norm matrix \mathbf{M} generates a discrete \mathcal{L}_2 inner product [23, 58] and norm such that

$$\langle \mathbf{u}, \mathbf{w} \rangle_{\mathbf{M}} = \mathbf{u}^T \mathbf{M} \mathbf{w} \approx \int_{-1}^{+1} \mathcal{U} \mathcal{W} d\xi = (\mathcal{U}, \mathcal{W})_{\mathcal{L}_2(\hat{\Omega})}, \quad \|\mathbf{u}\|_{\mathbf{M}}^2 = \langle \mathbf{u}, \mathbf{u} \rangle_{\mathbf{M}}, \quad (2.15)$$

where \mathcal{U}, \mathcal{W} are continuous functions on $\hat{\Omega}$ with their discrete projections onto the nodes $\boldsymbol{\xi}$

$$\begin{aligned} \mathbf{u} &:= [\mathcal{U}(\xi_0), \dots, \mathcal{U}(\xi_N)]^T, \\ \mathbf{w} &:= [\mathcal{W}(\xi_0), \dots, \mathcal{W}(\xi_N)]^T. \end{aligned} \quad (2.16)$$

Assuming that \mathbf{D} is of degree p , then the norm matrix \mathbf{M} is automatically of degree $2p - 1$, meaning that if $\mathcal{K} = \mathcal{U}\mathcal{W}$ is a polynomial of degree $2p - 1$, then

$$\langle \mathbf{u}, \mathbf{w} \rangle_{\mathbf{M}} = (\mathcal{U}, \mathcal{W})_{\mathcal{L}_2(\hat{\Omega})}, \quad (2.17)$$

see [23, 58]. Next, we examine the SBP property $\mathbf{M}\mathbf{D} + (\mathbf{M}\mathbf{D})^T = \mathbf{B}$ and point out its relation to IBP. Due to IBP we know

$$\left(\mathcal{U}, \frac{\partial \mathcal{W}}{\partial \xi} \right)_{\mathcal{L}_2(\hat{\Omega})} = \mathcal{U}\mathcal{W} \Big|_{-1}^{+1} - \left(\frac{\partial \mathcal{U}}{\partial \xi}, \mathcal{W} \right)_{\mathcal{L}_2(\hat{\Omega})}. \quad (2.18)$$

Now we approximate $\left(\mathcal{U}, \frac{\partial \mathcal{W}}{\partial \xi} \right)_{\mathcal{L}_2(\hat{\Omega})}$ using SBP operators

$$\begin{aligned} \langle \mathbf{u}, \mathbf{D}\mathbf{w} \rangle_{\mathbf{M}} &= \mathbf{u}^T \mathbf{M}\mathbf{D}\mathbf{w} = \mathbf{u}^T \mathbf{Q}\mathbf{w} = \mathbf{u}^T (\mathbf{B} - \mathbf{Q}^T) \mathbf{w}, \\ &= \mathbf{u}^T \mathbf{B}\mathbf{w} - (\mathbf{D}\mathbf{u})^T \mathbf{M}\mathbf{w}, \\ &= u_N w_N - u_0 w_0 - \langle \mathbf{D}\mathbf{u}, \mathbf{w} \rangle_{\mathbf{M}}. \end{aligned} \quad (2.19)$$

We see that (2.19) discretely mimics IBP (2.18). The terms $\mathbf{u}^T \mathbf{M}\mathbf{D}\mathbf{w}$ and $\mathbf{u}^T \mathbf{D}^T \mathbf{M}\mathbf{w}$ are approximations of $\left(\mathcal{U}, \frac{\partial \mathcal{W}}{\partial \xi} \right)_{\mathcal{L}_2(\hat{\Omega})}$ and $\left(\frac{\partial \mathcal{U}}{\partial \xi}, \mathcal{W} \right)_{\mathcal{L}_2(\hat{\Omega})}$, respectively. The boundary matrix \mathbf{B} refers to surface evaluations of the reference space, as we include the nodes at the surface of the reference space $\xi_0 = -1$ and $\xi_N = 1$.

Remark 2. Another class of such operators are *generalized SBP* operators [23, 99], where the surface nodes are not necessarily included. However, before the work of Chan in 2018 [17] it was not clear whether entropy stable schemes can be formulated using generalized SBP operators for problems like the compressible Euler equations. Therefore, in this work we focus on SBP operators where the surface nodes $\xi_0 = -1$ and $\xi_N = 1$ are included.

The mimic of IBP by SBP is the key concept in Section 2.1.2 when applying the energy method within the discrete analysis.

2.1.2. Summation-by-Parts Methods with Simultaneous-Approximation-Term

In this section we introduce the SBP-SAT method by Carpenter et al. [12]. This method is provably energy stable and primary conservative through the use of SBP operators. In addition, it is applicable to multi-block discretizations, which allows the scheme to perform on complex meshes. The SBP-SAT method is a semi-discrete method built from the method of lines. Here, we only discretize the

spatial domain and assume continuity in time. This leads to a discrete system or ordinary differential equations (ODEs) which is solved by standard Runge-Kutta time integration methods.

The SBP-SAT scheme was first introduced using SBP FD operators over the whole domain Ω . After describing the construction of these operators we introduce the SBP-SAT discretization and finally, we extend this method to a multi-block discretization.

Construction of SBP FD Operators

In this section, we explicitly construct classical SBP FD operators $(\mathbf{M}^{\text{FD}}, \mathbf{D}^{\text{FD}})$ as in [23]. Therefore we set $\boldsymbol{\xi} = [\xi_0, \dots, \xi_N]^T$ to be an uniformly distributed set of $N + 1$ nodes denoted on the spatial domain $\hat{\Omega}$ with

$$\xi_i = i \frac{2}{N} - 1, \quad (2.20)$$

for $i = 0, \dots, N$. The degree of \mathbf{D}^{FD} is denoted by p and the choice of the number of nodes $N + 1$ is arbitrary assuming that N is larger than a minimum number of nodes, see [24, 27, 111]. The construction of the SBP FD operators is adapted by [39].

The SBP FD operators consist of so called *interior and boundary* nodes. The interior nodes have at least p neighbor nodes to the left and right. The derivative on these nodes is approximated by an unique symmetric, central finite difference formula of degree $2p$. In comparison, the approximation of the derivative on the boundary nodes is constructed by a non-symmetric finite difference stencil of degree p . Here, we consider degrees of freedom at the boundary nodes of the operator \mathbf{M}^{FD} and \mathbf{Q}^{FD} . Focusing on $p = 2$ and setting the number of boundary nodes to $bn = 2p$, the operators have the following structure

$$\mathbf{M}^{\text{FD}} := \frac{2}{N} \text{diag}(h_0, \dots, h_3, 1, \dots, 1, h_3, \dots, h_0), \quad (2.21)$$

and $\mathbf{D}^{\text{FD}} := (\mathbf{M}^{\text{FD}})^{-1} \mathbf{Q}^{\text{FD}}$, where the structure of \mathbf{Q}^{FD} is given in Figure 2.1. For the classical SBP FD operators the free coefficients are calculated, such that the matrix $\mathbf{D}^{\text{FD}} = (\mathbf{M}^{\text{FD}})^{-1} \mathbf{Q}^{\text{FD}}$ is of degree p

$$\mathbf{D}^{\text{FD}} \boldsymbol{\xi}^k = k \boldsymbol{\xi}^{k-1} \quad k = 0, \dots, p, \quad (2.22)$$

$$\Leftrightarrow \mathbf{Q}^{\text{FD}} \boldsymbol{\xi}^k = k \mathbf{M}^{\text{FD}} \boldsymbol{\xi}^{k-1} \quad k = 0, \dots, p, \quad (2.23)$$

$$\mathbf{Q}^{\text{FD}} = \begin{bmatrix}
 \begin{array}{cccc|cccccccc}
 -\frac{1}{2} & q_{12} & q_{13} & q_{14} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 -q_{12} & 0 & q_{23} & q_{24} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 -q_{13} & -q_{23} & 0 & q_{34} & -\frac{1}{12} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 -q_{14} & -q_{24} & -q_{34} & 0 & \frac{2}{3} & -\frac{1}{12} & 0 & 0 & 0 & 0 & 0 & 0 \\
 \hline
 0 & 0 & \frac{1}{12} & -\frac{2}{3} & 0 & \frac{2}{3} & -\frac{1}{12} & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & \frac{1}{12} & -\frac{2}{3} & 0 & \frac{2}{3} & -\frac{1}{12} & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & \frac{1}{12} & -\frac{2}{3} & 0 & \frac{2}{3} & -\frac{1}{12} & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & \frac{1}{12} & -\frac{2}{3} & 0 & \frac{2}{3} & -\frac{1}{12} & 0 & 0 \\
 \hline
 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{12} & -\frac{2}{3} & 0 & q_{34} & q_{24} & q_{14} \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{12} & -q_{34} & 0 & q_{23} & q_{13} \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -q_{24} & -q_{23} & 0 & q_{12} \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -q_{14} & -q_{13} & -q_{12} & \frac{1}{2}
 \end{array}
 \end{bmatrix}$$

Figure 2.1.: Structure of the SBP FD operator of degree $p = 2$ with free boundary parameters. The figure is reproduced from [39].

where it is convenient to solve (2.23) as all parameters depend linearly on each other. For degrees of $p = 1, 2$, the operators \mathbf{M}^{FD} and \mathbf{Q}^{FD} are fully specified, for $p = 3, 4$ however this is not the case [23]. For $p \geq 5$ we need to consider more boundary nodes than $2p$ to ensure a diagonal positive definite norm matrix, where the minimum number of boundary nodes is found in [3]. Here, we choose the free parameters so that the truncation error is minimized. To do so we define

$$\boldsymbol{\epsilon}_{p+1} := \mathbf{D}^{\text{FD}} \boldsymbol{\xi}^{p+1} - (p+1) \boldsymbol{\xi}^p, \quad (2.24)$$

and then minimize

$$Z_e := \boldsymbol{\epsilon}_{p+1}^T \mathbf{M}^{\text{FD}} \boldsymbol{\epsilon}_{p+1}, \quad (2.25)$$

under the assumption that all entries of \mathbf{M}^{FD} are positive. If parameters are still not specified, we desire to have small coefficients within \mathbf{Q}^{FD} to avoid round-off errors. Therefore, we calculate

$$Z_Q := \mathbf{1}^T (\mathbf{Q}^{\text{FD}} \circ \mathbf{Q}^{\text{FD}}) \mathbf{1}, \quad (2.26)$$

where \circ is the Hadamard product and $\mathbf{1}$ is a vector of ones of size $N + 1$. Again

we minimize (2.26) ensuring that all entries of \mathbf{M}^{FD} are positive. A pseudocode description of the construction of $\mathbf{M}^{\text{FD}}, \mathbf{Q}^{\text{FD}}$ is provided in Algorithm 1.

Input: N and p
Output: \mathbf{M}^{FD} and \mathbf{Q}^{FD}
Set $bn := 2p$;
Solve (2.23);
while one or more entries of \mathbf{M}^{FD} are negative **do**
 | Set $bn := bn + 1$;
 | Solve (2.23) again;
end
Minimize \mathbf{Z}_e in (2.25) under the constraint $h_i > 0$ for $i = 0, \dots, bn - 1$;
Minimize \mathbf{Z}_Q in (2.26) under the constraint $h_i > 0$ for $i = 0, \dots, bn - 1$;

Algorithm 1: Construction of classical SBP FD operators.

From now on we denote SBP finite difference operators as *FD operators*.

SBP-SAT Discretization

Given a SBP operator \mathbf{D}^{FD} with its corresponding norm matrix \mathbf{M}^{FD} , we present the SBP-SAT discretization by Carpenter et al. [12]. For a general scalar hyperbolic conservation law the SBP-SAT discretization is defined by

$$\frac{\Delta x}{2} \mathbf{u}_t = -\mathbf{D}^{\text{FD}} \mathbf{f} + \mathbf{SAT}, \quad (2.27)$$

where

$$\mathbf{u}_t := \left[\frac{\partial \mathcal{U}(t, \xi_0)}{\partial t}, \dots, \frac{\partial \mathcal{U}(t, \xi_N)}{\partial t} \right]^T. \quad (2.28)$$

The term $\mathbf{D}^{\text{FD}} \mathbf{f}$ approximates the spatial derivative of the physical flux of $\frac{\partial \mathcal{F}(\mathbf{u})}{\partial x}$. The SAT term is presented as a penalty term to discretely incorporate the boundary condition \mathcal{B} . The choice of the SAT depends on the corresponding conservation law and its boundary conditions. Focusing on the LAE (2.3) we present the SBP-SAT discretization

$$\frac{\Delta x}{2} \mathbf{u}_t = -a \mathbf{D}^{\text{FD}} \mathbf{u} + a (\mathbf{M}^{\text{FD}})^{-1} \mathbf{e}_0 (u_0 - \mathcal{B}(t)), \quad (2.29)$$

where $\mathbf{e}_0 := (1, 0, \dots, 0)^T$ and \mathcal{B} describes the boundary condition as in (2.4). As the boundary conditions only depend on the left boundary, the SAT also only refers to the left boundary. Such a SAT is known as an *upwind SAT*.

Remark 3. In case $a < 0$, the SAT would only depend on the right boundary and would be denoted as *downwind SAT*.

We demonstrate how the continuous energy analysis is mimicked at the semi-discrete level with SBP operators to prove stability of semi-discrete approximations. Here, as we assume continuity in time, we discretely model (2.10). The growth in the total primary quantity $\frac{\partial}{\partial t} \int_{x_L}^{x_R} \mathcal{U} dx$ is approximated by

$$\begin{aligned} \frac{\partial}{\partial t} \int_{x_L}^{x_R} \mathcal{U} dx &= \frac{\Delta x}{2} \int_{-1}^{+1} \frac{\partial \mathcal{U}}{\partial t} d\xi = \frac{\Delta x}{2} \left(\mathbf{1}, \frac{\partial \mathcal{U}}{\partial t} \right)_{\mathcal{L}_2(\hat{\Omega})}, \\ &\approx \frac{\Delta x}{2} \langle \mathbf{1}, \mathbf{u}_t \rangle_{\text{MFD}} = \frac{\Delta x}{2} \mathbf{1}^T \mathbf{M}^{\text{FD}} \mathbf{u}_t. \end{aligned} \quad (2.30)$$

In addition, the growth in total energy, described by $\frac{1}{2} \frac{\partial}{\partial t} \|\mathcal{U}\|_{\mathcal{L}_2(\Omega)}^2$ is discretely modeled by

$$\begin{aligned} \frac{1}{2} \frac{\partial}{\partial t} \|\mathcal{U}\|_{\mathcal{L}_2(\Omega)}^2 &= \int_{x_L}^{x_R} \mathcal{U} \frac{\partial \mathcal{U}}{\partial t} dx = \frac{\Delta x}{2} \int_{-1}^{+1} \mathcal{U} \frac{\partial \mathcal{U}}{\partial t} d\xi = \frac{\Delta x}{2} \left(\mathcal{U}, \frac{\partial \mathcal{U}}{\partial t} \right)_{\mathcal{L}_2(\hat{\Omega})}, \\ &\approx \frac{\Delta x}{2} \langle \mathbf{u}, \mathbf{u}_t \rangle_{\text{MFD}} = \frac{\Delta x}{2} \mathbf{u}^T \mathbf{M}^{\text{FD}} \mathbf{u}_t. \end{aligned} \quad (2.31)$$

With (2.30) and (2.31) we define the fundamental definition of *discrete primary conservation* and *discrete energy stability*.

Definition 3. Primary Conservative Scheme for Scalar Conservation Laws

A semi-discrete scheme for a scalar conservation law is said to be primary conservative if the temporal derivative of the total discrete primary quantity is

$$\frac{\Delta x}{2} \mathbf{1}^T \mathbf{M}^{\text{FD}} \mathbf{u}_t = v_B(t), \quad (2.32)$$

where $v_B(t)$ purely depends on the evaluations on the boundary of the domain.

Definition 4. Energy Stable Scheme for the LAE

A semi-discrete scheme for the one dimensional LAE is said to be energy stable if the temporal derivative of the total discrete energy is bounded by its boundary condition

$$\frac{\Delta x}{2} \mathbf{u}^T \mathbf{M}^{\text{FD}} \mathbf{u}_t \leq \frac{a}{2} \mathcal{B}(t)^2. \quad (2.33)$$

Note, that Definition 4 is a discrete mimic of Definition 1 as integration in time gives us the semi-discrete energy estimate

$$\Delta x \|\mathbf{u}(t, \cdot)\|_{\text{MFD}}^2 \leq \Delta x \|\mathbf{u}(0, \cdot)\|_{\text{MFD}}^2 + \int_0^t a \mathcal{B}(\tau)^2 d\tau. \quad (2.34)$$

In addition, Definition 3 and 4 do not rely on exact integration of the scheme. We will prove that the SBP-SAT discretization (2.29) for the LAE is primary conservative and energy stable.

Theorem 1. *Considering SBP operators $(\mathbf{M}^{\text{FD}}, \mathbf{D}^{\text{FD}})$ satisfying the SBP property $\mathbf{M}^{\text{FD}}\mathbf{D}^{\text{FD}} + (\mathbf{M}^{\text{FD}}\mathbf{D}^{\text{FD}})^T = \mathbf{B}$, then discretization (2.29) is primary conservative and energy stable for the LAE.*

Proof. To prove primary conservation, we multiply (2.29) by $\mathbf{1}^T\mathbf{M}^{\text{FD}}$

$$\frac{\Delta x}{2}\mathbf{1}^T\mathbf{M}^{\text{FD}}\mathbf{u}_t = -a\mathbf{1}^T\mathbf{M}^{\text{FD}}\mathbf{D}^{\text{FD}}\mathbf{u} - a\mathbf{1}^T\mathbf{e}_0(u_0 - \mathcal{B}(t)). \quad (2.35)$$

Due to the SBP property $\mathbf{M}^{\text{FD}}\mathbf{D}^{\text{FD}} = \mathbf{Q}^{\text{FD}} = \mathbf{B} - \mathbf{Q}^{\text{FD},T}$ we get

$$\frac{\Delta x}{2}\mathbf{1}^T\mathbf{M}^{\text{FD}}\mathbf{u}_t = -a\mathbf{1}^T\mathbf{B}\mathbf{u} + a\mathbf{u}^T\mathbf{M}^{\text{FD}}\mathbf{D}^{\text{FD}}\mathbf{1} - a(u_0 - \mathcal{B}(t)). \quad (2.36)$$

Assuming that $\mathbf{D}^{\text{FD}}\mathbf{1} = \mathbf{0}$ (satisfied for all SBP operators of degree $p > 0$) we arrive at

$$\frac{\Delta x}{2}\mathbf{1}^T\mathbf{M}^{\text{FD}}\mathbf{u}_t = -au_N + a\mathcal{B}(t), \quad (2.37)$$

which indicates that the scheme is primary conservative as the right hand side purely depends on evaluations on the boundary [24].

Besides primary conservation, we claim that the SBP-SAT scheme is energy stable. Applying the energy method, we first multiply (2.29) with $\mathbf{u}^T\mathbf{M}^{\text{FD}}$ and get

$$\frac{\Delta x}{2}\mathbf{u}^T\mathbf{M}^{\text{FD}}\mathbf{u}_t = -a\mathbf{u}^T\mathbf{M}^{\text{FD}}\mathbf{D}^{\text{FD}}\mathbf{u} - a\mathbf{u}^T\mathbf{e}_0(u_0 - \mathcal{B}(t)). \quad (2.38)$$

Due to the SBP property (2.19) we know

$$\begin{aligned} \mathbf{u}^T\mathbf{M}^{\text{FD}}\mathbf{D}^{\text{FD}}\mathbf{u} &= u_N^2 - u_0^2 - \mathbf{u}^T\mathbf{D}^{\text{FD},T}\mathbf{M}^{\text{FD}}\mathbf{u}, \\ \Leftrightarrow \mathbf{u}^T\mathbf{M}^{\text{FD}}\mathbf{D}^{\text{FD}}\mathbf{u} &= \frac{1}{2}u_N^2 - \frac{1}{2}u_0^2. \end{aligned} \quad (2.39)$$

Including (2.39) in (2.38) gives us

$$\frac{\Delta x}{2}\mathbf{u}^T\mathbf{M}^{\text{FD}}\mathbf{u}_t = -\frac{a}{2}u_N^2 - \frac{a}{2}u_0^2 + au_0\mathcal{B}(t). \quad (2.40)$$

As for the continuous analysis we know $u_N^2 \geq 0$,

$$\frac{\Delta x}{2}\mathbf{u}^T\mathbf{M}^{\text{FD}}\mathbf{u}_t \leq -\frac{a}{2}u_0^2 + au_0\mathcal{B}(t). \quad (2.41)$$

By completing the square we arrive at

$$\frac{\Delta x}{2} \mathbf{u}^T \mathbf{M}^{\text{FD}} \mathbf{u}_t \leq -\frac{a}{2} (u_0 - \mathcal{B}(t))^2 + \frac{a}{2} \mathcal{B}(t)^2 \quad (2.42)$$

Due to the fact that a square of real numbers is positive we get the following discrete estimate for (2.10)

$$\frac{\Delta x}{2} \mathbf{u}^T \mathbf{M}^{\text{FD}} \mathbf{u}_t \leq \frac{a}{2} \mathcal{B}(t)^2. \quad (2.43)$$

□

Remark 4. The presented SBP-SAT method (2.29) is instructed to SBP operators where the surface nodes $\xi_0 = -1$ and $\xi_0 = +1$ are included. For nodal distributions without the surface nodes an alternative energy stable discretization is given by the SBP CPR (correction procedure via reconstruction) method [100].

For a more flexible scheme we next consider the SBP-SAT discretization on a multi-block mesh.

SBP-SAT Discretization on Multi-Block Meshes

When moving on to real world applications for two or three dimensional problems, we typically consider complex meshes with numerous elements, e.g. when simulating a flow surrounding an air foil. An advantageous feature of the SBP-SAT method is that it can also be applied as a *multi-block method*. Multi-block methods decompose the domain Ω into N_Q non-overlapping elements

$$\Omega = \bigcup_{q=1}^{N_Q} E_q. \quad (2.44)$$

On multi-block meshes we construct local nodal numerical solutions on each element denoted by \mathbf{u}^q . The global numerical solution \mathbf{u} is given by uniting the local solutions

$$\mathbf{u} = \bigcup_{q=1}^{N_Q} \mathbf{u}^q. \quad (2.45)$$

For simplicity we assume all elements have the width Δx . A one dimensional multi-block mesh with two elements (left L and right R) is visualized in Figure 2.2.

The corresponding SBP-SAT discretization for a multi-block mesh with an upwind SAT is

$$\frac{\Delta x}{2} \mathbf{u}_t^q = -a \mathbf{D}^{\text{FD}} \mathbf{u}^q - a (\mathbf{M}^{\text{FD}})^{-1} \mathbf{e}_0 (u_0^q - u_N^{q-1}), \quad (2.46)$$

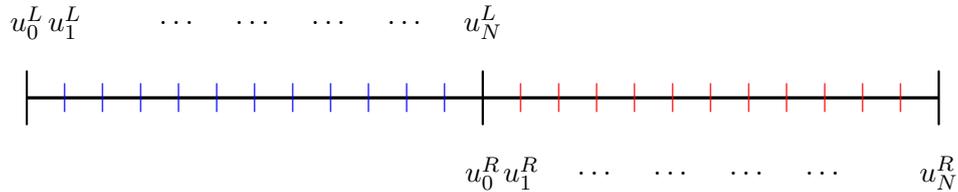


Figure 2.2.: One dimensional multi-block mesh with two elements. Here u_N^L and u_0^R are both defined on the interface. However, their values differ in general due to discontinuities or approximation errors of the solution.

with $u_N^0 = \mathcal{B}(t)$. Here we can see the convenience of constructing SBP operators in a reference space as we do not have to construct an operator for each element individually. Note, that even though u_0^q and u_N^{q-1} are defined on the same spatial coordinate (see Figure 2.2 for element L and R), they do not necessarily have the same value due to approximation errors or a discontinuous solution.

We approximate the growth in the total primary quantity $\frac{\partial}{\partial t} \int_{\Omega} \mathcal{U} d\Omega$ by

$$\frac{\partial}{\partial t} \int_{\Omega} \mathcal{U} d\Omega = \sum_{q=1}^{N_Q} \int_{E_q} \frac{\partial \mathcal{U}}{\partial t} dE_q \approx \frac{\Delta x}{2} \sum_{q=1}^{N_Q} \mathbf{1}^T \mathbf{M}^{\text{FPD}} \mathbf{u}_t^q. \quad (2.47)$$

Accordingly, for a multi-block discretization the growth in total energy $\frac{1}{2} \frac{\partial}{\partial t} \|\mathcal{U}\|_{\mathcal{L}_2(\Omega)}^2$ is approximated by

$$\frac{1}{2} \frac{\partial}{\partial t} \|\mathcal{U}\|_{\mathcal{L}_2(\Omega)}^2 \approx \frac{\Delta x}{2} \sum_{q=1}^{N_Q} \mathbf{u}^{q,T} \mathbf{M}^{\text{FPD}} \mathbf{u}_t^q. \quad (2.48)$$

Similar as in Section (2.1.2), we define primary conservation and discrete energy stability for SBP methods on a multi-block mesh.

Definition 5. Primary Conservative Scheme for Scalar Conservation Laws on Multi-Block Meshes

A semi-discrete scheme on a multi-block mesh for a scalar conservation law is said to be primary conservative if the temporal derivative of the total discrete primary quantity is

$$\frac{\Delta x}{2} \sum_{q=1}^{N_Q} \mathbf{1}^T \mathbf{M}^{\text{FD}} \mathbf{u}_t^q = v_B(t), \quad (2.49)$$

where $v_B(t)$ purely depends on the evaluations on the boundary of the domain.

Definition 6. Energy Stable Scheme for the LAE on Multi-Block Meshes

A semi-discrete scheme on a multi-block mesh for the one dimensional LAE is said to be energy stable if the temporal derivative of the total discrete energy is bounded by its boundary conditions

$$\frac{\Delta x}{2} \sum_{q=1}^{N_Q} \mathbf{u}^{q,T} \mathbf{M}^{\text{FD}} \mathbf{u}_t^q \leq \frac{a}{2} \mathcal{B}(t)^2. \quad (2.50)$$

As for a single element in Section (2.1.2), we will proof that the SBP-SAT method (2.46) is primary conservative and energy stable on a multi-block mesh.

Theorem 2. *The multi-block discretization (2.46) is primary conservative and energy stable for the LAE.*

Proof. In order to prove primary conservation, we multiply (2.46) with $\mathbf{1}^T \mathbf{M}^{\text{FD}}$

$$\frac{\Delta x}{2} \mathbf{1}^T \mathbf{M}^{\text{FD}} \mathbf{u}_t^q = -a \mathbf{1}^T \mathbf{M}^{\text{FD}} \mathbf{D}^{\text{FD}} \mathbf{u}^q - a \mathbf{1}^T \mathbf{e}_0 (u_0^q - u_N^{q-1}). \quad (2.51)$$

Considering the analogous steps as in Section 2.1.2 we obtain

$$\frac{\Delta x}{2} \mathbf{1}^T \mathbf{M}^{\text{FD}} u_t^R = -a u_N^q + a u_N^{q-1}. \quad (2.52)$$

Summing over all elements the terms on the interior interfaces vanish and thus gives us

$$\frac{\Delta x}{2} \sum_{q=1}^{N_Q} \mathbf{1}^T \mathbf{M}^{\text{FD}} \mathbf{u}_t^q = -a u_N^{N_Q} + a \mathcal{B}(t), \quad (2.53)$$

which proves that the discretization is primary conservative.

To verify that the SBP-SAT multi-block method is energy stable we apply the energy method as in Section 2.1.2.

$$\frac{\Delta x}{2} \mathbf{u}^q \mathbf{M}^{\text{FD}} \mathbf{u}_t^q = -a \mathbf{u}^{q,T} \mathbf{M}^{\text{FD}} \mathbf{D}^{\text{FD}} \mathbf{u}^q - a \mathbf{u}^{q,T} \mathbf{e}_0 (u_0^q - u_N^{q-1}). \quad (2.54)$$

By considering the analogous derivations as in Section 2.1.2 we get

$$\frac{\Delta x}{2} \mathbf{u}^{q,T} \mathbf{M}^{\text{FD}} \mathbf{u}_t^q = -\frac{a}{2} (u_N^q)^2 - \frac{a}{2} (u_0^q)^2 + a u_0^q u_N^{q-1}. \quad (2.55)$$

By including this in (2.48) we arrive at

$$\begin{aligned} & \frac{\Delta x}{2} \sum_{q=1}^{N_Q} \mathbf{u}^{q,T} \mathbf{M}^{\text{FD}} \mathbf{u}_t^q \\ &= \underbrace{-\frac{a}{2} u_N^{N_Q} - \frac{a}{2} (u_0^1)^2 + \frac{a}{2} u_0^1 \mathcal{B}(t)}_{=:BT} + \underbrace{\sum_{q=2}^{N_Q} \left(-\frac{a}{2} (u_N^{q-1})^2 - \frac{a}{2} (u_0^q)^2 + a u_0^q u_N^{q-1} \right)}_{=:IT}. \end{aligned} \quad (2.56)$$

Next, we analyze the term BT and IT individually, where BT refers to the values at the boundary (boundary terms) of Ω and IT refers to the interior interfaces (interior terms). Here, BT is exactly the right hand side of (2.40) (ignoring the element index) and thus we can recover (2.43) which gives us the estimate

$$BT \leq \frac{a}{2} \mathcal{B}(t)^2. \quad (2.57)$$

In comparison to the SBP-SAT method on a single element, the multi-block scheme introduces the coupling term IT . To ensure that (2.10) is discretely satisfied, we need to guarantee that IT is non-positive. As IT contains a binomial we get

$$IT = -\frac{a}{2} \underbrace{\sum_{q=2}^{N_Q} (u_N^{q-1} - u_0^q)^2}_{\geq 0} \leq 0. \quad (2.58)$$

Including (2.57) and (2.58) in (2.56) we arrive at

$$\Delta x \sum_{q=1}^{N_Q} \mathbf{u}^{q,T} \mathbf{M}^{\text{FD}} \mathbf{u}_t^q \leq a \mathcal{B}(t)^2, \quad (2.59)$$

which indicates energy stability of the multi-block SBP-SAT method. \square

Throughout this section we considered the upwind SAT as in (2.46). However, such SATs are not uniquely defined. An alternative SAT is the *central SAT* denoted by

$$\mathbf{SAT} = -\frac{a}{2} (\mathbf{M}^{\text{FD}})^{-1} \left(\mathbf{e}_0 (u_0^q - u_N^{q-1}) - \mathbf{e}_N (u_N^q - u_0^{q+1}) \right). \quad (2.60)$$

Introducing a right boundary condition $\mathcal{B}_N(t) = \mathcal{U}(t, x_R)$ we get the discrete energy equation.

$$\Delta x \sum_{q=1}^{N_Q} \mathbf{u}^{q,T} \mathbf{M}^{\text{FD}} \mathbf{u}_t^q = a \left(u_0^1 \mathcal{B}(t) - u_N^{N_Q} \mathcal{B}_N(t) \right). \quad (2.61)$$

As the energy is no longer described by an inequality, such a SBP-SAT scheme with a central SAT is denoted as *energy conservative*. Note, that when we consider periodic boundary conditions $\mathcal{B}(t) = u_N^{N_Q}$ and $\mathcal{B}_N(t) = u_0^1$ we get

$$\Delta x \sum_{q=1}^{N_Q} \mathbf{u}^{q,T} \mathbf{M}^{\text{FD}} \mathbf{u}_t^q = 0, \quad (2.62)$$

which is an useful result when verifying primary conservation and energy stability/conservation in Section 2.3. Note that when considering a central SAT with periodic boundary conditions we obtain from (2.53)

$$\frac{\Delta x}{2} \sum_{q=1}^{N_Q} \mathbf{1}^T \mathbf{M}^{\text{FD}} \mathbf{u}_t^q = 0. \quad (2.63)$$

To summarize, we introduced the SBP-SAT method on a single element and on a multi-block mesh. The key concept of this method lies within the construction of the norm matrix \mathbf{M}^{FD} and differentiation matrix \mathbf{D}^{FD} . If these operators satisfy the SBP property

$$\mathbf{M}^{\text{FD}} \mathbf{D}^{\text{FD}} + (\mathbf{M}^{\text{FD}} \mathbf{D}^{\text{FD}})^T = \mathbf{B}, \quad (2.64)$$

then it is possible to mimic the continuous stability and conservation analysis in a one-to-one fashion. This indicates stability and conservation of the scheme. Another scheme which satisfies these properties is the *Discontinuous Galerkin Spectral Element Method (DGSEM)*.

2.1.3. Discontinuous Galerkin Spectral Element Method

In this section we introduce the DGSEM. The DGSEM is a special class of discontinuous Galerkin methods (DG methods). We first introduce the general DG method and then describe the derivation of the DGSEM. This method is of special interest, as it can be interpreted as a SBP-SAT method provided a non uniform nodal distribution.

The Discontinuous Galerkin Method

In this section we introduce the general DG method. As for the SBP-SAT method, the DG method can be applied as a multi-block method. Thus, we subdivide the spatial domain Ω into non-overlapping elements as in (2.44). In comparison to the SBP-SAT method, the DG approach is derived from the weak formulation similar to the finite element method (FEM). Both methods, DG and FEM, assume continuity of the numerical solution on an element. However, the standard FEM assumes continuity on the interfaces connecting two elements, which indicates global continuity on the spatial domain Ω . This is insufficient for the solutions of hyperbolic conservation laws as these are typically discontinuous. In comparison to the FEM, the DG method does not assume a continuous solution at the interfaces. To handle discontinuities a numerical surface flux is introduced as in the finite volume method. To derive the weak formulation we multiply (2.1) with an arbitrary discontinuous test function $\phi := \phi(x)$ defined on $\Omega = [x_L, x_R]$ and integrate over the spatial domain

$$\int_{\Omega} \left(\frac{\partial \mathcal{U}}{\partial t} + \frac{\partial \mathcal{F}}{\partial x} \right) \phi \, d\Omega = 0, \quad (2.65)$$

with $\mathcal{F} := \mathcal{F}(\mathcal{U})$. As the test function is arbitrary, (2.65) needs to be satisfied on every element

$$\int_{E_q} \frac{\partial \mathcal{U}}{\partial t} \phi \, dE_q + \int_{E_q} \frac{\partial \mathcal{F}}{\partial x} \phi \, dE_q = 0, \quad (2.66)$$

for $q = 1, \dots, N_Q$. For further extension it is helpful to transform (2.66) into the reference element $\hat{\Omega} = [-1, +1]$ as in (2.12).

$$\frac{\Delta x}{2} \int_{-1}^{+1} \frac{\partial \mathcal{U}}{\partial t} \phi \, d\xi + \int_{-1}^{+1} \frac{\partial \mathcal{F}}{\partial \xi} \phi \, d\xi = 0, \quad (2.67)$$

Within (2.66) or (2.67) we assume that \mathcal{U} is differentiable. To weaken this assumption for the DG approach, we derive the weak formulation. Assuming that $\phi \in \mathcal{C}^\infty(E_q)$ we apply integration by parts on (2.67) and get the weak formulation

$$\frac{\Delta x}{2} \int_{-1}^{+1} \frac{\partial \mathcal{U}}{\partial t} \phi \, d\xi + \mathcal{F} \phi \Big|_{-1}^{+1} - \int_{-1}^{+1} \mathcal{F} \frac{\partial \phi}{\partial \xi} \, d\xi = 0. \quad (2.68)$$

When deriving the DG method we focus on (2.68) and so \mathcal{U} is not assumed to be differentiable on Ω .

For the DGSEM we approximate \mathcal{U} on E_q by the local numerical solution \mathcal{U}^q . As for the multi-block SBP-SAT method (2.45) the global numerical solution \mathcal{U}^{num} is

$$\mathcal{U}(t, x) \approx \mathcal{U}^{num}(t, x) = \bigcup_{q=1}^{N_Q} \mathcal{U}^q(t, x). \quad (2.69)$$

The local solution \mathcal{U}^q is defined by a polynomial basis within $\hat{\Omega}$

$$\mathcal{U}^q = \sum_{j=0}^N u_j^q(t) \varphi_j(\xi), \quad (2.70)$$

where $\{u_j^q\}_{j=0}^N$ are time dependent coefficients. Here, $\{\varphi_j\}_{j=0}^N$ is a set of basis functions in space. The choice of these functions depend on the considered DG implementation. We set these functions to be continuous on $[-1, +1]$, which indicates a continuous numerical solution within each element E_q . However, this is not assumed at the element interfaces as demonstrated in Figure 2.3.

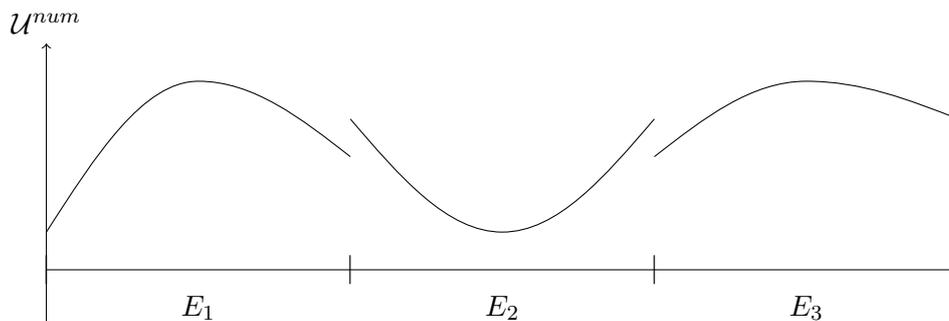


Figure 2.3.: Discontinuous Galerkin solutions are assumed to be continuous on an element but not on an interface.

This is the main difference between the discontinuous Galerkin and the continuous Galerkin method or FEM. As discontinuities might occur at the interface, the surface term $\mathcal{F}\phi|_{-1}^{+1}$ in (2.68) is not uniquely defined. This issue is not only seen for DG methods but also for finite volume methods (FV methods). To resolve the surface term we introduce the *numerical surface flux* denoted by f^* . This flux depends on the interface value of the "left" and "right" element and is defined by (approximated) Riemann solvers [116]. A well known numerical flux is the local Lax-Friedrichs flux

$$f^*(\mathcal{U}^R, \mathcal{U}^L) = \frac{1}{2} (\mathcal{F}(\mathcal{U}^R) + \mathcal{F}(\mathcal{U}^L)) - \frac{1}{2} \max_{z \in \{\mathcal{U}^L, \mathcal{U}^R\}} \left| \frac{\partial \mathcal{F}}{\partial \mathcal{U}}(z) \right| (\mathcal{U}^R - \mathcal{U}^L). \quad (2.71)$$

Besides introducing the numerical solution and the numerical flux, we need a numerical representation \mathcal{F}^q of the physical flux \mathcal{F} within E_q . Simply evaluating $\mathcal{F}^q = \mathcal{F}(\mathcal{U}^q)$ can cause non-linearity within the basis functions (e.g. terms like $\frac{1}{1+\phi^2}$) for which the numerical integration can become cumbersome and expensive. To avoid these non-linearities we set

$$\mathcal{F}^q(t, \xi) = \sum_{j=0}^N f_j^q(t) \varphi_j(\xi), \quad (2.72)$$

with time dependent coefficients $\{f_j^q\}_{j=0}^N$. Note, that $\mathcal{F}^q \neq \mathcal{F}(\mathcal{U}^q)$ for non-linear fluxes which introduces aliasing errors [53]. Assuming the aliasing error to be sufficiently small and inserting the numerical solution \mathcal{U}^q , the numerical flux representation \mathcal{F}^q and the numerical surface flux f^* in (2.68) and assuming the test function ϕ to be an arbitrary linear combination of the basis functions φ_i we get a DG formulation in the weak form

$$\frac{\Delta x}{2} \sum_{j=0}^N \frac{\partial u_j^q}{\partial t} \int_{-1}^{+1} \varphi_j \varphi_i d\xi + a f^* \varphi_i \Big|_{-1}^{+1} - a \sum_{j=0}^N f_j^q \int_{-1}^{+1} \varphi_j \frac{\partial \varphi_i}{\partial \xi} d\xi = 0, \quad (2.73)$$

or alternatively in the inner product notation

$$\frac{\Delta x}{2} \sum_{j=0}^N \frac{\partial u_j^q}{\partial t} (\varphi_j, \varphi_i)_{\mathcal{L}_2(\hat{\Omega})} + a f^* \varphi_i \Big|_{-1}^{+1} - a \sum_{j=0}^N f_j^q \left(\varphi_j, \frac{\partial \varphi_i}{\partial \xi} \right)_{\mathcal{L}_2(\hat{\Omega})} = 0, \quad (2.74)$$

for $i = 0, \dots, N$. The derivation of a DG method depends strongly on the choice of the basis functions and the choice of the quadrature rule to approximate the integrals. To evaluate these integrals at low cost we focus on the DGSEM implementation.

DGSEM Discretization

In this section we introduce the discretization of the DGSEM. In order to obtain a high-order scheme, the integrals of the weak formulation are approximated with Legendre-Gauss (LG) or Legendre-Gauss-Lobatto (LGL) quadrature [53] on $\hat{\Omega}$. Let $\omega_0, \dots, \omega_N$ and ξ_0, \dots, ξ_N denote the quadrature weights and nodes in $[-1, +1]$. We set the basis functions to be Lagrange basis polynomials of degree p . In contrast to FD schemes, the number of points within the DGSEM is linked to the polynomial degree by $p = N$. The Lagrange polynomials need to be defined on a set of interpolation nodes. The crux of the nodal DGSEM is to collocate the interpolation nodes with the nodes of the corresponding quadrature. The

considered Lagrange polynomials ℓ_j for $j = 0, \dots, N$ are defined by

$$\ell_j(\xi) = \prod_{k=0, k \neq j}^N \frac{\xi - \xi_k}{\xi_j - \xi_k}, \quad (2.75)$$

which satisfy the Kronecker delta property, i.e. $\ell_j(\xi_k) = \delta_{jk}$. With these basis functions the time dependent coefficients of \mathcal{U}^q and \mathcal{F}^q are directly evaluated

$$\begin{aligned} \mathcal{U}^q(t, \xi_i) &= \sum_{j=0}^N u_j^q(t) \ell_j(\xi_i) = \sum_{j=0}^N u_j^q(t) \delta_{ij} = u_i(t), \\ \mathcal{F}^q(t, \xi_i) &= \sum_{j=0}^N f_j^q(t) \ell_j(\xi_i) = \sum_{j=0}^N f_j^q(t) \delta_{ij} = f_i(t), \end{aligned} \quad (2.76)$$

which indicates that the DGSEM is a *nodal* scheme as the coefficients are directly evaluated at the nodes. The weak formulation of the DGSEM is defined by

$$\frac{\Delta x}{2} \sum_{j=0}^N \frac{\partial u_j^q}{\partial t} (\ell_j, \ell_i)_{\mathcal{L}_2(\hat{\Omega})} + f^* \ell_i \Big|_{-1}^{+1} - \sum_{j=0}^N f_j^q (\ell_j, \ell_i)_{\mathcal{L}_2(\hat{\Omega})} = 0, \quad (2.77)$$

for $i = 0, \dots, N$. Next, we use quadrature to evaluate the integrals within (2.77). First we consider

$$(\ell_j, \ell_i)_{\mathcal{L}_2(\hat{\Omega})} \approx \sum_{k=0}^N \omega_k \ell_j(\xi_k) \ell_i(\xi_k) = \sum_{k=0}^N \omega_k \delta_{jk} \delta_{ik} = \omega_i \delta_{ij}. \quad (2.78)$$

Due to collocation the integral is approximated by $\omega_k \delta_{ij}$. Considering the LG quadrature the evaluation of (2.78) is an equality as the LG quadrature can integrate polynomials of degree $2N + 1$ exactly. Thus, $\{\ell_j\}_{j=0}^N$ is an orthogonal basis with respect to $(\cdot, \cdot)_{\mathcal{L}_2(\hat{\Omega})}$. For the LGL quadrature however (2.78) is indeed an approximation as this quadrature is only exact for polynomials of degree $2N - 1$. In [44] it is shown that an exact integration of the Lagrange basis polynomials on LGL nodes leads to a non-orthogonal basis within $(\cdot, \cdot)_{\mathcal{L}_2(\hat{\Omega})}$ and thus increases the cost of the integral approximation. Therefore, we accept the integration error within (2.78). This technique is also referred to as *mass lumping*.

For the second integral in (2.77) we get

$$\left(\ell_j, \frac{\partial \ell_i}{\partial \xi} \right)_{\mathcal{L}_2(\hat{\Omega})} = \sum_{k=0}^N \omega_k \ell_j(\xi_k) \frac{\partial \ell_i}{\partial \xi}(\xi_k) = \omega_j \frac{\partial \ell_i}{\partial \xi}(\xi_j). \quad (2.79)$$

Note that (2.79) is exact for either LG or LGL quadrature. Including these evaluations in the weak formulation we obtain

$$\frac{\Delta x}{2} \omega_i \frac{\partial u_i^q}{\partial t} + f^* \ell_i \Big|_{-1}^{+1} - \sum_{j=0}^N \omega_j \frac{\partial \ell}{\partial \xi}(\xi_j) f_j^q = 0, \quad (2.80)$$

for $i = 0, \dots, N$. For a more compact notation we rewrite (2.80) in terms of norm and differentiation matrices and define

$$\mathbf{u}^q := [u_0^q, \dots, u_N^q]^T, \quad \mathbf{f}^q := [f_0^q, \dots, f_N^q]^T, \quad \mathbf{u}_t^q := \frac{\partial \mathbf{u}^q}{\partial t}. \quad (2.81)$$

To distinguish the norm matrices of a FD and DG scheme, we denote the norm and differentiation matrix for the DGSEM by \mathbf{M}^{DG} and \mathbf{D}^{DG} . As for FD operators the norm matrix of the DGSEM is a diagonal matrix

$$\mathbf{M}^{\text{DG}} = \text{diag}(\omega_0, \dots, \omega_N), \quad (2.82)$$

due to exact integration or mass lumping in (2.78). The spatial derivative of u_i^q is defined by

$$\frac{\partial}{\partial x} u_i^q = \frac{\partial}{\partial x} u^q(\xi_i, t) = \sum_{j=0}^N \frac{\partial \ell_j}{\partial \xi}(\xi_i) u_j^q(t), \quad (2.83)$$

for $i = 0, \dots, N$. In a compact notation (2.83) is equivalent to

$$\frac{\partial}{\partial x} \mathbf{u}^q = \mathbf{D}^{\text{DG}} \mathbf{u}^q, \quad (2.84)$$

with

$$[\mathbf{D}^{\text{DG}}]_{ij} = \frac{\partial \ell_j}{\partial \xi}(\xi_i), \quad (2.85)$$

for $i, j = 0, \dots, N$. To handle the numerical flux term we introduce the surface matrix \mathbf{S}^{DG} and the numerical surface flux vector \mathbf{f}^*

$$\begin{aligned} [\mathbf{S}^{\text{DG}}]_{ij} &= -\delta_{0j} \ell_i(-1) + \delta_{Nj} \ell_i(+1), \\ \mathbf{f}^* &= [f^*(\mathcal{U}^q(t, \xi = -1)), 0, \dots, 0, f^*(\mathcal{U}^q(t, \xi = 1))]^T, \end{aligned} \quad (2.86)$$

which gives us $\mathbf{S}^{\text{DG}} \mathbf{f}^* = f^* \ell_i \Big|_{-1}^{+1}$. Finally, we get the DGSEM discretization in a compact matrix-vector notation

$$\frac{\Delta x}{2} \mathbf{M}^{\text{DG}} \mathbf{u}_t^q + \mathbf{S}^{\text{DG}} \mathbf{f}^* - \mathbf{D}^{\text{DG}, T} \mathbf{M}^{\text{DG}} \mathbf{f}^q = 0. \quad (2.87)$$

Our main interest is to obtain an energy stable discretization. The SBP-SAT method was proven to be energy stable in Section 2.1.2 by considering FD operators satisfying the SBP property (2.64). For the DGSEM, the coefficients of the operators $\mathbf{M}^{\text{DG}}, \mathbf{D}^{\text{DG}}, \mathbf{S}^{\text{DG}}$ depend on the corresponding quadrature. As mentioned before, the LG quadrature is of higher degree than the LGL quadrature. However, when considering the LGL nodes we obtain a SBP operator as proven in Lemma 1. The key concept of this proof is that the LGL nodes include the surface nodes ($\xi_0 = -1$ and $\xi = +1$) whereas the LG nodes do not.

Lemma 1. *Considering the LGL nodes $\{\xi_i\}_{i=0}^N$ and weights $\{\omega_i\}_{i=0}^N$. Then the norm matrix \mathbf{M}^{DG} and the differentiation matrix \mathbf{D}^{DG} satisfy the SBP property (2.64)*

$$\mathbf{Q}^{\text{DG}} + \mathbf{Q}^{\text{DG},T} = \mathbf{B}, \quad (2.88)$$

with $\mathbf{Q}^{\text{DG}} = \mathbf{M}^{\text{DG}}\mathbf{D}^{\text{DG}}$.

Proof. First we look at a single entry of \mathbf{Q}^{DG}

$$[\mathbf{Q}^{\text{DG}}]_{ij} = [\mathbf{M}^{\text{DG}}\mathbf{D}^{\text{DG}}]_{ij} = \omega_i \frac{\partial \ell_j}{\partial \xi}(\xi_i), \quad (2.89)$$

with $i, j = 0, \dots, N$. An entry of \mathbf{Q}^{DG} is equal to $\left(\ell_i, \frac{\partial \ell_j}{\partial \xi}\right)_{\mathcal{L}_2(\hat{\Omega})}$ as

$$\left(\ell_i, \frac{\partial \ell_j}{\partial \xi}\right)_{\mathcal{L}_2(\hat{\Omega})} = \sum_{k=0}^N \omega_k \ell_i(\xi_k) \frac{\partial \ell_j}{\partial \xi}(\xi_k) = \omega_i \frac{\partial \ell_j}{\partial \xi}(\xi_i) = [\mathbf{Q}^{\text{DG}}]_{ij}, \quad (2.90)$$

as $\ell_i \frac{\partial \ell_j}{\partial \xi}$ is a polynomial of degree $2N - 1$. Looking at an arbitrary coefficient of $\mathbf{Q}^{\text{DG}} + \mathbf{Q}^{\text{DG},T}$ we get

$$[\mathbf{Q}^{\text{DG}} + \mathbf{Q}^{\text{DG},T}]_{ij} = [\mathbf{Q}^{\text{DG}}]_{ij} + [\mathbf{Q}^{\text{DG}}]_{ji} = \left(\ell_i, \frac{\partial \ell_j}{\partial \xi}\right)_{\mathcal{L}_2(\hat{\Omega})} + \left(\frac{\partial \ell_i}{\partial \xi}, \ell_j\right)_{\mathcal{L}_2(\hat{\Omega})}. \quad (2.91)$$

Due to IBP we arrive at

$$[\mathbf{Q}^{\text{DG}} + \mathbf{Q}^{\text{DG},T}]_{ij} = \ell_i \ell_j \Big|_{-1}^{+1} = \ell_i(+1)\ell_j(+1) - \ell_i(-1)\ell_j(-1). \quad (2.92)$$

As we consider LGL nodes we know $\xi_0 = -1$ and $\xi_N = +1$ which gives us

$$[\mathbf{Q}^{\text{DG}} + \mathbf{Q}^{\text{DG},T}]_{ij} = \ell_i(\xi_N)\ell_j(\xi_N) - \ell_i(\xi_0)\ell_j(\xi_0) = \delta_{iN}\delta_{Nj} - \delta_{i0}\delta_{0j}. \quad (2.93)$$

So for $\mathbf{Q}^{\text{DG}} + \mathbf{Q}^{\text{DG},T}$ holds

$$\left[\mathbf{Q}^{\text{DG}} + \mathbf{Q}^{\text{DG},T}\right]_{ij} = \begin{cases} -1 & \text{if } i = j = 0, \\ +1 & \text{if } i = j = N, \\ 0 & \text{else} \end{cases} \quad (2.94)$$

which is precisely the definition of \mathbf{B} and so we get the SBP property (2.64)

$$\mathbf{Q}^{\text{DG}} + \mathbf{Q}^{\text{DG},T} = \mathbf{B}. \quad (2.95)$$

□

From now on when referring to DG operators we focus on differentiation matrices \mathbf{D}^{DG} with their corresponding norm matrices \mathbf{M}^{DG} constructed on LGL nodes due to the SBP property (2.64). For the LGL nodes the surface matrix \mathbf{S}^{DG} reduces to \mathbf{B} since

$$[\mathbf{S}^{\text{DG}}]_{ij} = -\delta_{0j}\ell_i(-1) + \delta_{Nj}\ell_i(+1) = \delta_{iN}\delta_{Nj} - \delta_{i0}\delta_{0j} = [\mathbf{B}]_{ij}. \quad (2.96)$$

Focusing on LGL nodes (2.87) is equivalent to

$$\frac{\Delta x}{2} \mathbf{M}^{\text{DG}} \mathbf{u}_t^q + \mathbf{B} \mathbf{f}^* - \mathbf{D}^{\text{DG},T} \mathbf{M}^{\text{DG}} \mathbf{f}^q = 0. \quad (2.97)$$

In order to compare the DGSEM discretization (2.97) with the SBP-SAT discretization (2.46) we rearrange the terms within the weak formulation. Due to the SBP property (2.64) we get the so called *strong formulation*.

$$\frac{\Delta x}{2} \mathbf{u}_t^q = -\mathbf{D}^{\text{DG}} \mathbf{f}^q - (\mathbf{M}^{\text{DG}})^{-1} \mathbf{B} (\mathbf{f}^* - \mathbf{f}^q). \quad (2.98)$$

Remark 5. Rearranging (2.97) to (2.98) is a discrete mimic of IBP.

To fully specify the DGSEM we need to select the numerical flux. As for SATs, the numerical flux depends on the considered conservation law. Focusing on the LAE we get

$$\frac{\Delta x}{2} \mathbf{u}_t^q = -a \mathbf{D}^{\text{DG}} \mathbf{u}^q - (\mathbf{M}^{\text{DG}})^{-1} \mathbf{B} (\mathbf{f}^* - a \mathbf{u}^q). \quad (2.99)$$

As the LAE transports information from left to right ($a > 0$) we consider the upwind flux

$$f^* (\mathcal{U}^R, \mathcal{U}^L) = a \mathcal{U}^L, \quad (2.100)$$

which is equivalent to the local Lax Friedrichs flux (2.71). For the LAE with $a > 0$ this is the exact solution of the Riemann problem [116]. Including (2.100) in the

numerical flux vector, leading to $\mathbf{f}^* = a [u_N^{q-1}, 0, \dots, 0, u_N^q]^T$ with $u_N^0 = \mathcal{B}(t)$, we get the strong formulation for the LAE

$$\frac{\Delta x}{2} \mathbf{u}_t^q = -a \mathbf{D}^{\text{DG}} \mathbf{u}^q - a (\mathbf{M}^{\text{DG}})^{-1} \mathbf{e}_0 (u_0^q - u_N^{q-1}). \quad (2.101)$$

Remarkably, when comparing the discretization of the DGSEM (2.101) and the SBP-SAT method (2.46) we have the same discretization besides different SBP operators with their corresponding nodes. Even when considering a central numerical flux $f^*(\mathcal{U}^R, \mathcal{U}^L) = a \frac{\mathcal{U}^L + \mathcal{U}^R}{2}$ the DGSEM discretization is equivalent to the energy conservative SBP-SAT scheme with a symmetric SAT (2.60). Therefore, the proofs of primary conservation and energy stability in Section 2.1.2 hold for the DGSEM. This is an outstanding result as we can apply different approaches within the same discretization only by replacing the SBP operator. Even when considering completely new SBP operators, regardless of the fact if they are derived from a weak formulation or based on finite differences, we can insert these operators in (2.101) or (2.46) to obtain a primary conservative and energy stable scheme. From now on, when referring to a method based on SBP operators (\mathbf{M}, \mathbf{D}) , we name such a scheme a *SBP method*. Here, (\mathbf{M}, \mathbf{D}) can be finite difference or discontinuous Galerkin operators. Different SBP operators can be found in [23]. Due to good properties as shown in the numerical results sections we describe the *Hybrid Gauss-Trapezoidal-Lobatto (HGTL)* operators in Section 2.1.4 which satisfies the SBP property (2.64).

2.1.4. Hybrid Gauss-Trapezoidal-Lobatto Operators

Finite Difference operators of degree p which satisfy the SBP property (2.64) have the flexibility to consider an arbitrary number of nodes by increasing the number of interior points. The structure of the operator as in Figure 2.1 does not change considering uniform distributed nodes. However, the approximation error for high-order ($p > 2$) FD schemes on uniform distributed nodes is higher than on e.g. LGL or Chebyshev nodes [27]. In comparison to the FD operators, increasing the number of nodes for the DG operators increases the order of the approximation which can lead to unnecessary high cost of the scheme.

In order to maintain the flexibility of a FD operator and obtain a reduced error with no effect on the order of accuracy, Fernández [26] developed HGTL operators. Here, we consider equally spaced interior nodes, but non uniform distributed nodes at the boundary. The construction of these operators is similar as for the FD operators as in Section 2.1.2. In order to construct a SBP operator on such a

set of nodes with a diagonal positive definite norm matrix it is essential that a quadrature rule with positive weights exists. Deriving the optimal position of the non uniform boundary nodes while at the same time ensuring that a quadrature rule with positive weights exists is a difficult task [27]. Therefore, we first consider a quadrature rule with positive weights. Quadrature rules for nodal distributions with a non uniform distribution of nodes at the boundary and equally spaced nodes in the interior were purposed by Alpert [4] and are denoted as Hybrid Gauss-Trapezoidal quadrature rules. When considering this quadrature rule in the reference space $\hat{\Omega} = [-1, +1]$, the boundary points -1 and $+1$ are not included, which is essential for the SBP property in this work (2.64). Therefore, Fernández modified these nodes by including the boundaries and denoted the new quadrature rule as *Hybrid Gauss-Trapezoidal Lobatto* rule or HGTL rule. Let $N + 1$ denote the number of nodes and bn the number of boundary nodes. The position of the boundary nodes are derived by solving (2.102) for $\{(\tilde{\omega}_i, \tilde{x}_i)\}_{i=0}^{bn-p-1}$

$$\sum_{i=0}^{bn-p-1} \tilde{\omega}_i \tilde{x}_i = \frac{B_{r+1}(bn-p-1)}{r+1}, \quad \text{for } r = 0, \dots, 2(bn-p-1), \quad (2.102)$$

$$\tilde{x}_0 = 0,$$

where $B_r(\cdot)$ is the r -th Bernoulli polynomial with $B_0 = 1$. The first equation within (2.102) is adapted by [4] and the second equation is to ensure that the nodes at the boundary are included. The nodal distribution of ξ in $[-1, +1]$ is derived by transformation

$$\xi_i = \frac{2}{N} \tilde{x}_i - 1, \quad \xi_{N-i} = \frac{2}{N} (N - \tilde{x}_i) - 1, \quad (2.103)$$

for $i = 0, \dots, bn-p-1$. All other nodes within ξ are assumed to be uniformly distributed as in (2.20).

Remark 6. Note that only $bn-p$ boundary nodes can be non uniformly distributed as all interior nodes need p equally spaced neighbor nodes to the left and right as its derivative is approximated via a central finite difference formula. For example, let the boundary nodes be ξ_0, \dots, ξ_{bn-1} . To approximate the derivative on the interior node ξ_{bn} we need p equally spaced neighbors $\xi_{bn-p}, \dots, \xi_{bn+p}$. Therefore, the only boundary nodes which can be distributed non uniformly are $\xi_0, \dots, \xi_{bn-p-1}$.

Rather than considering the quadrature rule of Alpert which has positive weights up to degree 20 [4], we assume the norm matrix to have bn free coefficients (as in (2.21) for $p = 2$). This gives us more degrees of freedom (*DOFS*) to solve the degree equation (2.23).

With this nodal distribution we can construct the HGTL operators in a one-to-one fashion to FD operators in Section 2.1.2 as presented in Algorithm 2.

Input: N and p
Output: M and Q
Set $bn := 2p$;
Solve (2.102) to obtain ξ ;
Solve (2.23);
while one or more entries of M are negative or (2.23) not solvable **do**
 | Set $bn := bn + 1$;
 | Solve (2.102) and (2.23) again;
end
Minimize Z_e in (2.25) under the constraint $h_i > 0$ for $i = 0, \dots, bn$;
Minimize Z_Q in (2.26) under the constraint $h_i > 0$ for $i = 0, \dots, bn$;

Algorithm 2: Construction of HGTL operators.

Up to this point we introduced SBP operators such as FD, DG and HGTL operators. Including these operators in discretization (2.46) gives the one dimensional SBP method. In real world applications one typically considers two or three dimensional non-linear systems of conservation laws on complex geometries and meshes. Therefore, we next introduce the SBP method for the two dimensional linear advection equation.

2.2. Summation-by-Parts Methods for the Two Dimensional Linear Advection Equation

In general two dimensional scalar conservation laws are defined by

$$\frac{\partial \mathcal{U}}{\partial t} + \frac{\partial \mathcal{F}(\mathcal{U})}{\partial x} + \frac{\partial \mathcal{G}(\mathcal{U})}{\partial y} = 0, \quad (2.104)$$

with $t \in [0, T] \subset \mathbb{R}$ and $(x, y) \in \Omega \subset \mathbb{R}^2$. In this section we focus on the two dimensional linear advection equation such that $\mathcal{F}(\mathcal{U}) = a\mathcal{U}$ and $\mathcal{G}(\mathcal{U}) = b\mathcal{U}$, where $a, b > 0$ denote the wave speeds

$$\frac{\partial \mathcal{U}}{\partial t} + a \frac{\partial \mathcal{U}}{\partial x} + b \frac{\partial \mathcal{U}}{\partial y} = 0. \quad (2.105)$$

To derive an energy estimate we consider Dirichlet boundary and initial conditions

$$\begin{aligned} \mathcal{U}(0, x, y) &= \mathcal{U}_0(x, y) && \text{for } (x, y) \in \Omega, \\ \mathcal{U}(t, x, y) &= \mathcal{B}(t, x, y) && \text{for } t \in [0, T] \text{ and } (x, y) \in \Omega^-, \end{aligned} \quad (2.106)$$

where $\Omega^- := \{(x, y) \in \partial\Omega \mid (an_1 + bn_2 < 0)\}$ and $\mathbf{n} = (n_1, n_2)^T$. Here, n_1, n_2 are the x and y components of the outward pointing normal to $\partial\Omega$. As for the one dimensional LAE we obtain the energy estimate

$$\frac{\partial}{\partial t} \|\mathcal{U}(t, \cdot, \cdot)\|_{\mathcal{L}_2(\Omega)}^2 \leq \int_{\partial\Omega^-} \mathcal{B}(t, x, y)^2 |an_1 + bn_2| d\partial\Omega^-, \quad (2.107)$$

or alternatively by integrating in time

$$\|\mathcal{U}(t, \cdot, \cdot)\|_{\mathcal{L}_2(\Omega)}^2 \leq \|\mathcal{U}_0\|_{\mathcal{L}_2(\Omega)}^2 + \int_0^t \int_{\partial\Omega^-} \mathcal{B}(\tau, x, y)^2 |an_1 + bn_2| d\tau d\partial\Omega^-. \quad (2.108)$$

As the solution is bounded by the initial and boundary conditions the two dimensional LAE is energy stable (2.5). Here, we introduce a two dimensional SBP discretization that remains energy stable and primary conservative. Therefore, we subdivide Ω into N_Q non-overlapping elements as in (2.45). Here, each element is assumed to be a quadrilateral where in general the interfaces between elements can be curved. As for the one dimensional case we transform the element E into the reference space $\hat{\Omega} := [-1, +1] \times [-1, +1]$. Let (x, y) denote the coordinates within the physical space E and (ξ, η) to be the coordinates within the computational space $\hat{\Omega}$. As such, we consider a transformation $\chi : \hat{\Omega} \rightarrow E$ such that

$$(x, y)^T = \chi(\xi, \eta). \quad (2.109)$$

Let $\chi_1, \dots, \chi_4 \in \mathbb{R}^2$ describe the corners of an element E . The element is described by curves $\Gamma_1, \dots, \Gamma_4$ representing the interface as in Figure 2.4. With the interface curves $\Gamma_1, \dots, \Gamma_4$ we construct an interpolation between the "bottom" and "top" curve denoted by χ^{bt} and an interpolation between the "left" and "right" curve denoted by χ^{lr} with

$$\begin{aligned} \chi^{bt}(\xi, \eta) &:= \frac{1-\eta}{2} \Gamma_1(\xi) + \frac{1+\eta}{2} \Gamma_3(\xi), \\ \chi^{lr}(\xi, \eta) &:= \frac{1-\xi}{2} \Gamma_4(\eta) + \frac{1+\xi}{2} \Gamma_2(\eta). \end{aligned} \quad (2.110)$$

Simply adding χ^{bt} and χ^{lr} to obtain χ is insufficient, as e.g. corners do not match

$$\begin{aligned} \chi^{bt}(\xi, -1) + \chi^{lr}(\xi, -1) &= \Gamma_1(\xi) + \frac{1-\xi}{2} \Gamma_4(-1) + \frac{1+\xi}{2} \Gamma_2(-1), \\ &\neq \Gamma_1(\xi) = \chi(\xi, -1). \end{aligned} \quad (2.111)$$

Therefore, we introduce a correction term χ^{cor} defined by

$$\begin{aligned} \chi^{cor}(\xi, \eta) &:= \frac{1}{4} (\chi_1(1-\xi)(1-\eta) + \chi_2(1+\xi)(1-\eta)) \\ &\quad + \frac{1}{4} (\chi_3(1+\xi)(1+\eta) + \chi_4(1-\xi)(1+\eta)). \end{aligned} \quad (2.112)$$

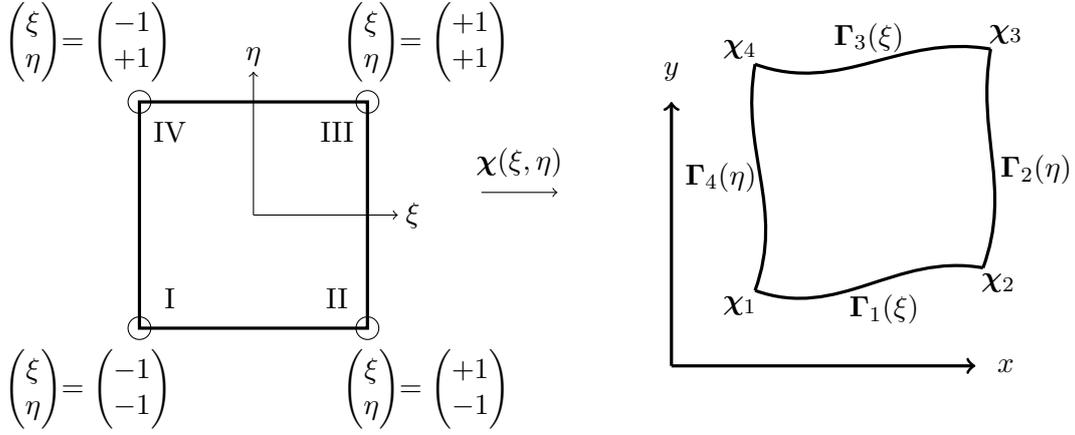


Figure 2.4.: Transforming the reference space $\hat{\Omega} = [-1, +1] \times [-1, +1]$ into an element with curved interfaces.

With the correction term we define the transformation

$$\boldsymbol{\chi} := \boldsymbol{\chi}^{bt} + \boldsymbol{\chi}^{lr} - \boldsymbol{\chi}^{cor}. \quad (2.113)$$

Thus, we have the mapping of the physical coordinates (x, y) in terms of the computational coordinates (ξ, η)

$$(x, y)^T = \boldsymbol{\chi}(\xi, \eta). \quad (2.114)$$

With $\boldsymbol{\chi}$ we transform the two dimensional conservation law (2.104) into the reference space. Assuming $\xi = \xi(x, y)$ and $\eta = \eta(x, y)$ we can rewrite the derivatives of the physical fluxes by applying the chain rule

$$\begin{aligned} \frac{\partial \mathcal{F}}{\partial x} &= \frac{\partial \mathcal{F}}{\partial \xi} \frac{\partial \xi}{\partial x} + \frac{\partial \mathcal{F}}{\partial \eta} \frac{\partial \eta}{\partial x}, \\ \frac{\partial \mathcal{G}}{\partial y} &= \frac{\partial \mathcal{G}}{\partial \xi} \frac{\partial \xi}{\partial y} + \frac{\partial \mathcal{G}}{\partial \eta} \frac{\partial \eta}{\partial y}. \end{aligned} \quad (2.115)$$

Note, that for reasons of simplicity we suppress the dependence of \mathcal{U} on the physical fluxes. As we constructed the mapping function $\boldsymbol{\chi}$ such that $x = x(\xi, \eta)$ and $y = y(\xi, \eta)$ we do not directly obtain $\frac{\partial \xi}{\partial x}, \frac{\partial \eta}{\partial x}, \frac{\partial \xi}{\partial y}, \frac{\partial \eta}{\partial y}$. However, we can rewrite these unknown terms by the inverse of the Jacobian of $\boldsymbol{\chi}$ as in [67]

$$\begin{pmatrix} \frac{\partial \xi}{\partial x} & \frac{\partial \eta}{\partial x} \\ \frac{\partial \xi}{\partial y} & \frac{\partial \eta}{\partial y} \end{pmatrix} = \begin{pmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{pmatrix}^{-1}. \quad (2.116)$$

Calculating the inverse matrix leads to

$$\begin{pmatrix} \frac{\partial \xi}{\partial x} & \frac{\partial \eta}{\partial x} \\ \frac{\partial \xi}{\partial y} & \frac{\partial \eta}{\partial y} \end{pmatrix} = \frac{1}{J} \begin{pmatrix} \frac{\partial y}{\partial \eta} & -\frac{\partial y}{\partial \xi} \\ -\frac{\partial x}{\partial \eta} & \frac{\partial x}{\partial \xi} \end{pmatrix}, \quad (2.117)$$

with the determinant $J = \frac{\partial x}{\partial \xi} \frac{\partial y}{\partial \eta} - \frac{\partial x}{\partial \eta} \frac{\partial y}{\partial \xi}$. Here, we obtain the so called *metric terms* $\frac{\partial x}{\partial \xi}$, $\frac{\partial x}{\partial \eta}$, $\frac{\partial y}{\partial \xi}$, $\frac{\partial y}{\partial \eta}$. With the metric terms we rewrite the physical fluxes (2.115) in reference coordinates as

$$\begin{aligned} \frac{\partial \mathcal{F}}{\partial x} &= \frac{1}{J} \left(\frac{\partial \mathcal{F}}{\partial \xi} \frac{\partial y}{\partial \eta} - \frac{\partial \mathcal{F}}{\partial \eta} \frac{\partial y}{\partial \xi} \right), \\ \frac{\partial \mathcal{G}}{\partial y} &= \frac{1}{J} \left(\frac{\partial \mathcal{G}}{\partial \eta} \frac{\partial x}{\partial \xi} - \frac{\partial \mathcal{G}}{\partial \xi} \frac{\partial x}{\partial \eta} \right). \end{aligned} \quad (2.118)$$

Including (2.118) in (2.115) we obtain the two dimensional conservation law on E transformed in the reference space $\hat{\Omega}$

$$J \frac{\partial \mathcal{U}}{\partial t} + \frac{\partial \mathcal{F}}{\partial \xi} \frac{\partial y}{\partial \eta} - \frac{\partial \mathcal{F}}{\partial \eta} \frac{\partial y}{\partial \xi} + \frac{\partial \mathcal{G}}{\partial \eta} \frac{\partial x}{\partial \xi} - \frac{\partial \mathcal{G}}{\partial \xi} \frac{\partial x}{\partial \eta} = 0. \quad (2.119)$$

For a more compact notation we introduce the *contravariant fluxes* $\tilde{\mathcal{F}}$ and $\tilde{\mathcal{G}}$ defined as

$$\begin{aligned} \tilde{\mathcal{F}} &:= \mathcal{F} \frac{\partial y}{\partial \eta} - \mathcal{G} \frac{\partial x}{\partial \eta}, \\ \tilde{\mathcal{G}} &:= \mathcal{G} \frac{\partial x}{\partial \xi} - \mathcal{F} \frac{\partial y}{\partial \xi}, \end{aligned} \quad (2.120)$$

for which we can rewrite (2.119) by

$$J \frac{\partial \mathcal{U}}{\partial t} + \frac{\partial \tilde{\mathcal{F}}}{\partial \xi} + \frac{\partial \tilde{\mathcal{G}}}{\partial \eta} = 0. \quad (2.121)$$

Note, that for (2.121) we assume that χ is smooth such that $\frac{\partial^2 \chi}{\partial \xi \partial \eta} = \frac{\partial^2 \chi}{\partial \eta \partial \xi}$.

Based on (2.121) we present a SBP discretization. It can be derived from a finite difference perspective using SATs or from the weak formulation of the PDE. Here, we choose the weak formulation. Thus, we approximate the conserved variable \mathcal{U} and the contravariant fluxes $\tilde{\mathcal{F}}$, $\tilde{\mathcal{G}}$ with a set of basis functions. As such, we consider Lagrange basis functions of degree N in each spatial direction, e.g.,

$$\mathcal{U}(t, \xi, \eta) \Big|_{E_q} \approx \mathcal{U}^q(t, \xi, \eta) = \sum_{i,j=0}^N u_{ij}^q(t) \ell_i(\xi) \ell_j(\eta), \quad (2.122)$$

and similar for $\tilde{\mathcal{F}}$ and $\tilde{\mathcal{G}}$ with \tilde{f}_{ij}^q and \tilde{g}_{ij}^q . Note, that selecting the test function to be a product of two Lagrange functions both defined in one dimension we obtain a tensor product basis. With this basis we can decouple the derivatives and the numerical fluxes in each spatial direction. Therefore, we can still use the one dimensional SBP operators as in Section 2.1. For ease of presentation we assume that the nodal distribution within an element is the same in each spatial direction. However, due to the tensor product decoupling we could allow the nodal distribution to differ in ξ - and η -direction. Using (2.122) to approximate the solution of the conservation law, we multiply (2.121) by a test function ϕ and integrate over the reference space

$$\int_{-1}^{+1} \int_{-1}^{+1} \left(J \frac{\partial \mathcal{U}^q}{\partial t} + \frac{\partial \tilde{\mathcal{F}}^q}{\partial \xi} + \frac{\partial \tilde{\mathcal{G}}^q}{\partial \eta} \right) \phi \, d\xi \, d\eta = 0, \quad (2.123)$$

As for the one dimensional case we represent the test function by the basis functions $\phi = \sum_{i,j=0}^N \phi_{ij} \ell_i(\xi) \ell_j(\eta)$ to obtain a degree N approximation \mathcal{U}^q . Note, that $\ell_i(\xi) \ell_j(\eta)$ for $i, j = 0, \dots, N$ is a sufficient basis for degree N polynomials. As the test function is an arbitrary polynomial of degree N , equation (2.123) is equivalent to

$$\int_{-1}^{+1} \int_{-1}^{+1} \left(J \frac{\partial \mathcal{U}^q}{\partial t} + \frac{\partial \tilde{\mathcal{F}}^q}{\partial \xi} + \frac{\partial \tilde{\mathcal{G}}^q}{\partial \eta} \right) \ell_i(\xi) \ell_j(\eta) \, d\xi \, d\eta = 0, \quad (2.124)$$

All integrals present in (2.124) are approximated with $N + 1$ LGL nodes and weights, e.g.,

$$\begin{aligned} \int_{\hat{\Omega}} J \frac{\partial \mathcal{U}^q}{\partial t} \ell_i(\xi) \ell_j(\eta) \, d\xi \, d\eta &\approx \sum_{n,m=0}^N \omega_n \omega_m \left(\sum_{k,l=0}^N J_{kl} \frac{\partial u_{kl}^q}{\partial t} \ell_k(\xi_n) \ell_l(\eta_m) \right) \ell_i(\xi_n) \ell_j(\eta_m), \\ &= \omega_i \omega_j J_{ij} \frac{\partial u_{ij}^q}{\partial t}. \end{aligned} \quad (2.125)$$

Note, that for simplicity we suppress the time variable t . When approximating the integrals concerning the contravariant fluxes we first apply IBP to obtain a weak formulation. To resolve discontinuities across element interfaces we introduce

numerical surface fluxes denoted by \tilde{f}^*, \tilde{g}^* . The approximation is given by

$$\begin{aligned} \int_{\tilde{\Omega}} \frac{\partial \tilde{\mathcal{F}}^q}{\partial \xi} \ell_i(\xi) \ell_j(\eta) d\xi d\eta &= \int_{-1}^{+1} \ell_j(\eta) \left(\int_{-1}^{+1} \frac{\partial \tilde{\mathcal{F}}^q}{\partial \xi} \ell_i(\xi) d\xi \right) d\eta, \\ &= \int_{-1}^{+1} \ell_j(\eta) \left(\tilde{\mathcal{F}}^q \ell_i \Big|_{-1}^{+1} - \int_{-1}^{+1} \tilde{\mathcal{F}}^q \frac{\partial \ell_i}{\partial \xi}(\xi) d\xi \right) d\eta, \\ &\approx \omega_j \left(\tilde{f}^{*,q}(\xi_N, \eta_j) \delta_{iN} - \tilde{f}^{*,q}(\xi_0, \eta_j) \delta_{i0} - \sum_{k=0}^N \mathbf{Q}_{ki} \tilde{f}_{kj}^q \right) \end{aligned} \quad (2.126)$$

with $\mathbf{Q} = \mathbf{MD}$ and

$$\begin{aligned} \int_{\tilde{\Omega}} \frac{\partial \tilde{\mathcal{G}}^q}{\partial \eta} \ell_i(\xi) \ell_j(\eta) d\xi d\eta &= \int_{-1}^{+1} \ell_i(\xi) \left(\int_{-1}^{+1} \frac{\partial \tilde{\mathcal{G}}^q}{\partial \eta} \ell_j(\eta) d\eta \right) d\xi, \\ &= \int_{-1}^{+1} \ell_i(\xi) \left(\tilde{\mathcal{G}}^q \ell_j \Big|_{-1}^{+1} - \int_{-1}^{+1} \tilde{\mathcal{G}}^q \frac{\partial \ell_j}{\partial \eta}(\eta) d\eta \right) d\xi, \\ &\approx \omega_i \left(\tilde{g}^{*,q}(\xi_i, \eta_N) \delta_{Nj} - \tilde{g}^{*,q}(\xi_i, \eta_0) \delta_{0j} - \sum_{l=0}^N \mathbf{Q}_{lj} \tilde{g}_{il}^q \right). \end{aligned} \quad (2.127)$$

In total this gives us the weak form

$$\begin{aligned} \omega_i \omega_j J_{ij} \frac{\partial u_{ij}^q}{\partial t} + \omega_j \left(\tilde{f}^{*,q}(\xi_N, \eta_j) \delta_{iN} - \tilde{f}^{*,q}(\xi_0, \eta_j) \delta_{i0} - \sum_{k=0}^N \mathbf{Q}_{ki} \tilde{f}_{kj}^q \right) \\ + \omega_i \left(\tilde{g}^{*,q}(\xi_i, \eta_N) \delta_{Nj} - \tilde{g}^{*,q}(\xi_i, \eta_0) \delta_{0j} - \sum_{l=0}^N \mathbf{Q}_{lj} \tilde{g}_{il}^q \right) = 0, \end{aligned} \quad (2.128)$$

and since we consider SBP operators with $\mathbf{Q}_{ki} + \mathbf{Q}_{ik} = \mathbf{B}_{ik}$ we get the equivalent two dimensional strong form

$$\begin{aligned} \omega_i \omega_j J_{ij} \frac{\partial u_{ij}^q}{\partial t} + \omega_j \left([\tilde{f}_{Nj}^{*,q} - \tilde{f}_{Nj}^q] \delta_{iN} - [\tilde{f}_{0j}^{*,q} - \tilde{f}_{0j}^q] \delta_{i0} + \sum_{k=0}^N \mathbf{Q}_{ik} \tilde{f}_{kj}^q \right) \\ + \omega_i \left([\tilde{g}_{iN}^{*,q} - \tilde{g}_{iN}^q] \delta_{Nj} - [\tilde{g}_{i0}^{*,q} - \tilde{g}_{i0}^q] \delta_{0j} + \sum_{l=0}^N \mathbf{Q}_{jl} \tilde{g}_{il}^q \right) = 0, \end{aligned} \quad (2.129)$$

with $\tilde{f}_{ij}^{*,q} = \tilde{f}^{*,q}(\xi_i, \eta_j)$ and $\tilde{g}_{ij}^{*,q} = \tilde{g}^{*,q}(\xi_i, \eta_j)$.

Note, that even though we derived (2.129) using a DG formulation we can apply the strong form discretization using an arbitrary SBP operator. Therefore, we do not include the operator superscript DG within (2.129).

As for the one dimensional case we consider the physical fluxes for the two dimensional LAE defined in (2.105). We prove that discretization (2.129) is primary conservative and energy stable for the LAE on a Cartesian mesh. For curved elements the discretization (2.129) needs to be modified to obtain a discrete primary conservative and energy stable scheme as presented in Appendix A.1. Focusing on a Cartesian mesh the metric terms are constant

$$\begin{aligned} \frac{\partial y}{\partial \eta} &= \frac{\Delta y}{2}, \quad \frac{\partial x}{\partial \eta} = 0, \\ \frac{\partial x}{\partial \xi} &= \frac{\Delta x}{2}, \quad \frac{\partial y}{\partial \xi} = 0, \end{aligned} \tag{2.130}$$

and $J = \frac{\Delta x \Delta y}{4}$. We consider a mesh on the domain $\Omega = [x_L, x_R] \times [y_U, y_O]$ with N_{Q_x} and N_{Q_y} elements in x - and y -direction. In total we have $N_Q = N_{Q_x} N_{Q_y}$ elements. Next, we define primary conservation and energy stability for the two dimensional LAE.

Definition 7. Primary Conservative Scheme for Two Dimensional Scalar Conservation Laws

A semi-discrete method approximating a two dimensional scalar conservation law is said to be primary conservative if the time derivative of the total discrete primary quantity is

$$\sum_{q=1}^{N_Q} \sum_{i,j=0}^N J_{ij} \omega_i \omega_j \frac{\partial u_{ij}^q}{\partial t} = v_B(t), \tag{2.131}$$

where $v_B(t)$ purely depends on evaluations on the boundary of the domain.

Definition 8. Energy Stable Scheme for the Two Dimensional LAE

A semi-discrete method approximating the two dimensional LAE is said to be energy stable if

$$\begin{aligned} \sum_{q=1}^{N_Q} \sum_{i,j=0}^N J_{ij} \omega_i \omega_j \frac{1}{2} \frac{\partial (u_{ij}^q)^2}{\partial t} &\leq \sum_{n_y=1}^{N_{Q_y}} \frac{\tilde{a}}{2} \sum_{j=0}^N \omega_j \mathcal{B}(t, x_L, y_j^{n_y})^2 \\ &+ \sum_{n_x=1}^{N_{Q_x}} \frac{\tilde{b}}{2} \sum_{i=0}^N \omega_i \mathcal{B}(t, x_i^{n_x}, y_U)^2, \end{aligned} \tag{2.132}$$

as (2.132) approximates (2.107). Here, we define $\tilde{a} = \frac{a\Delta y}{2}$ and $\tilde{b} = \frac{b\Delta x}{2}$. The physical coordinats $x_i^{n_x}$ refers to the x -component in the n_x element on the i -th node. The same holds for the y -component $y_j^{n_y}$.

To prove that (2.129) conserves the primary quantity and is energy stable, we first examine the discrete growth in the primary quantity and energy on a single element. Therefore, we proof Lemma 2.

Lemma 2. *The discrete growth of the primary quantity and energy on a single element of discretization (2.129) are*

$$\sum_{i,j=0}^N J_{ij}\omega_i\omega_j \frac{\partial u_{ij}}{\partial t} = - \sum_{j=0}^N \omega_j \left(\tilde{f}_{Nj}^* - \tilde{f}_{0j}^* \right) - \sum_{i=0}^N \omega_i \left(\tilde{g}_{iN}^* - \tilde{g}_{i0}^* \right), \quad (2.133)$$

and

$$\begin{aligned} \sum_{i,j=0}^N J_{ij}\omega_i\omega_j \frac{1}{2} \frac{\partial u_{ij}^2}{\partial t} &= - \sum_{j=0}^N \omega_j \left(\left[u\tilde{f}^* - \frac{\tilde{a}}{2}u^2 \right]_{Nj} - \left[u\tilde{f}^* - \frac{\tilde{a}}{2}u^2 \right]_{0j} \right) \\ &\quad - \sum_{i=0}^N \omega_i \left(\left[u\tilde{g}^* - \frac{\tilde{b}}{2}u^2 \right]_{iN} - \left[u\tilde{g}^* - \frac{\tilde{b}}{2}u^2 \right]_{i0} \right). \end{aligned} \quad (2.134)$$

Note, that for clarity we suppress the element index q as we only focus on a single element.

Proof. First, we prove (2.133). Therefore, we sum over (2.128) for $i, j = 0, \dots, N$ to apply the discrete integration over the element.

$$\begin{aligned} \sum_{i,j=0}^N J_{ij}\omega_i\omega_j \frac{\partial u_{ij}}{\partial t} &= - \sum_{i,j=0}^N \omega_j \left(\tilde{f}_{Nj}^* \delta_{iN} - \tilde{f}_{0j}^* \delta_{i0} + \sum_{k=0}^N \mathbf{Q}_{ki} \tilde{f}_{kj} \right) \\ &\quad - \sum_{i,j=0}^N \omega_i \left(\tilde{g}_{iN}^* \delta_{Nj} - \tilde{g}_{i0}^* \delta_{0j} + \sum_{l=0}^N \mathbf{Q}_{lj} \tilde{g}_{il} \right). \end{aligned} \quad (2.135)$$

Rearranging the sums over i, j, l we obtain

$$\begin{aligned} \sum_{i,j=0}^N J_{ij}\omega_i\omega_j \frac{\partial u_{ij}}{\partial t} &= - \sum_{j=0}^N \omega_j \left(\tilde{f}_{Nj}^* - \tilde{f}_{0j}^* + \sum_{k=0}^N \tilde{f}_{kj} \sum_{i=0}^N \mathbf{Q}_{ki} \right) \\ &\quad - \sum_{i=0}^N \omega_i \left(\tilde{g}_{iN}^* - \tilde{g}_{i0}^* + \sum_{l=0}^N \tilde{g}_{il} \sum_{j=0}^N \mathbf{Q}_{lj} \right). \end{aligned} \quad (2.136)$$

Assuming that the SBP operator is of degree $p > 0$, we know $\sum_{i=0}^N \mathbf{Q}_{ki} = 0$ as $\sum_{i=0}^N \mathbf{D}_{ki} = 0$ and thus

$$\sum_{i,j=0}^N J_{ij} \omega_i \omega_j \frac{\partial u_{ij}}{\partial t} = - \sum_{j=0}^N \omega_j \left(\tilde{f}_{Nj}^* - \tilde{f}_{0j}^* \right) - \sum_{i=0}^N \omega_i \left(\tilde{g}_{iN}^* - \tilde{g}_{i0}^* \right). \quad (2.137)$$

Note, that (2.137) holds for an arbitrary scalar conservation law as we did not specify the physical flux. To prove (2.134) we first write the physical flux as

$$\begin{aligned} \tilde{f}_{ij} &= \frac{\Delta y}{2} f_{ij} = \tilde{a} u_{ij}, \\ \tilde{g}_{ij} &= \frac{\Delta x}{2} g_{ij} = \tilde{b} u_{ij}. \end{aligned} \quad (2.138)$$

We then multiply (2.128) by u_{ij} and sum over all nodes $i, j = 0, \dots, N$ to find

$$\begin{aligned} \sum_{i,j=0}^N J_{ij} \omega_i \omega_j \frac{1}{2} \frac{\partial u_{ij}^2}{\partial t} &= - \sum_{i,j=0}^N \omega_j \left(u_{Nj} \tilde{f}_{Nj}^* \delta_{iN} - u_{0j} \tilde{f}_{0j}^* \delta_{i0} - \sum_{k=0}^N \mathbf{Q}_{ki} u_{ij} \tilde{f}_{kj} \right) \\ &\quad - \sum_{i,j=0}^N \omega_i \left(u_{iN} \tilde{g}_{iN}^* \delta_{Nj} - u_{i0} \tilde{g}_{i0}^* \delta_{0j} - \sum_{l=0}^N \mathbf{Q}_{lj} u_{ij} \tilde{g}_{il} \right). \end{aligned} \quad (2.139)$$

Rearranging sums gives us

$$\begin{aligned} \sum_{i,j=0}^N J_{ij} \omega_i \omega_j \frac{1}{2} \frac{\partial u_{ij}^2}{\partial t} &= - \sum_{j=0}^N \omega_j \left(u_{Nj} \tilde{f}_{Nj}^* - u_{0j} \tilde{f}_{0j}^* - \tilde{a} \sum_{i=0}^N \sum_{k=0}^N \mathbf{Q}_{ki} u_{ij} u_{kj} \right) \\ &\quad - \sum_{i=0}^N \omega_i \left(u_{iN} \tilde{g}_{iN}^* - u_{i0} \tilde{g}_{i0}^* - \tilde{b} \sum_{j=0}^N \sum_{l=0}^N \mathbf{Q}_{lj} u_{ij} u_{il} \right). \end{aligned} \quad (2.140)$$

Due to the SBP property (2.64) we know $2\mathbf{Q}_{ki} = \mathbf{Q}_{ki} - \mathbf{Q}_{ik} + \mathbf{B}_{ki}$ and thus

$$2 \sum_{i=0}^N \sum_{k=0}^N \mathbf{Q}_{ki} u_{ij} u_{kj} = \sum_{i=0}^N \sum_{k=0}^N \mathbf{Q}_{ki} u_{ij} u_{kj} - \sum_{i=0}^N \sum_{k=0}^N \mathbf{Q}_{ik} u_{ij} u_{kj} + \sum_{i=0}^N \sum_{k=0}^N \mathbf{B}_{ki} u_{ij} u_{kj}. \quad (2.141)$$

By rearranging indices we obtain

$$\sum_{i=0}^N \sum_{k=0}^N \mathbf{Q}_{ki} u_{ij} u_{kj} = \frac{1}{2} \sum_{i=0}^N \sum_{k=0}^N \mathbf{B}_{ki} u_{ij} u_{kj} = \frac{1}{2} u_{Nj}^2 - \frac{1}{2} u_{0j}^2. \quad (2.142)$$

Therefore, we can rewrite (2.140) by

$$\begin{aligned} \sum_{i,j=0}^N J_{ij} \omega_i \omega_j \frac{1}{2} \frac{\partial u_{ij}^2}{\partial t} &= - \sum_{j=0}^N \omega_j \left(\left[u \tilde{f}^* - \frac{\tilde{a}}{2} u^2 \right]_{Nj} - \left[u \tilde{f}^* - \frac{\tilde{a}}{2} u^2 \right]_{0j} \right) \\ &\quad - \sum_{i=0}^N \omega_i \left(\left[u \tilde{g}^* - \frac{\tilde{b}}{2} u^2 \right]_{iN} - \left[u \tilde{g}^* - \frac{\tilde{b}}{2} u^2 \right]_{i0} \right), \end{aligned} \quad (2.143)$$

and thus gives us (2.134). \square

Within the proof of Lemma 2 we were able to rewrite the volume contributions in term surface contribution. This has been done by considering SBP operators as they mimic the IBP rule. Note, that the proof of Lemma 2 relied on a Cartesian mesh. For curved meshes however, discretization (2.128) does not reduce to surface contributions when discretely applying the energy method. This is due to the fact that \tilde{a} and \tilde{b} are non-constant and thus the discrete fluxes $\tilde{f} = \tilde{a}u$ and $\tilde{g} = \tilde{b}u$ introduce non-linear terms. The problems of discretizing hyperbolic conservation laws with non-linear fluxes is discussed in Section 4. As mentioned before, the discretization of (2.105) needs a slight modification. The modification of the discretization and the proof of primary conservation and energy stability for curved meshes is demonstrated in Appendix A.1. With Lemma 2 we finally can prove primary conservation and energy stability for (2.129).

Theorem 3. *Considering a Cartesian mesh and an upwind numerical surface flux, then the discretization (2.129) is primary conservative and energy stable.*

Proof. To obtain a primary conservative and energy stable scheme on the entire domain we must sum all discrete growths of the primary quantity and energy on all elements. Due to the use of SBP operators and Lemma 2, the total discrete growths depend on the interfaces of the elements. Here, we look at the vertical interfaces. The contribution of the horizontal interfaces is done in an analogous fashion. We first focus on the interior interfaces, meaning that the interface connects two elements as in Figure 2.5. Due to Lemma 2 the terms referring to the interior interfaces are

$$IU_t := \sum_{j=0}^N \omega_j \tilde{f}_{0j}^{*,R} - \sum_{j=0}^N \omega_j \tilde{f}_{Nj}^{*,L}, \quad (2.144)$$

$$IE_t := \sum_{j=0}^N \omega_j \left(u_{0j}^R \tilde{f}_{0j}^{*,R} - \frac{\tilde{a}}{2} (u_{0j}^R)^2 \right) - \sum_{j=0}^N \omega_j \left(u_{Nj}^L \tilde{f}_{Nj}^{*,L} - \frac{\tilde{a}}{2} (u_{Nj}^L)^2 \right), \quad (2.145)$$

where the superscript L and R refer to the left and right element, respectively. Here, IU_t and IE_t approximate the surface integral of $\frac{\partial \mathcal{U}}{\partial t}$ and $\frac{1}{2} \frac{\partial \mathcal{U}^2}{\partial t}$ on a single



Figure 2.5.: A single interface connecting two neighboring elements. This figure is reproduced from [38].

interface. As we focus on a one dimensional interface (first component of u and f^* is fixed), we set

$$u_j^L := u_{N_L j}^L, \quad u_j^R := u_{0j}^R, \quad (2.146)$$

and the same for \tilde{f}^* . For a conforming mesh both elements share the same numerical flux, meaning $\tilde{f}^* := \tilde{f}^{*,L} = \tilde{f}^{*,R}$. So, we see directly that

$$IU_t = \sum_{j=0}^N \omega_j \tilde{f}_j^* - \sum_{j=0}^N \omega_j \tilde{f}_j^* = 0. \quad (2.147)$$

As IU_t is zero, the total growth in the primary quantity (2.131) only depends on the interfaces at the boundary of Ω , which indicates primary conservation. Note, that the derivation holds independent of the choice of the numerical surface flux.

To obtain an energy stable scheme we must specify the numerical flux function. As for the one dimensional LAE, we consider the upwind flux

$$\tilde{f}^*(u^R, u^L) = \tilde{f}(u^L) = \tilde{a}u^L. \quad (2.148)$$

Note, that the upwind flux is an exact solution to the Riemann problem as we consider positive wave speeds. Focusing on IE_t we get

$$\begin{aligned} IE_t &= \tilde{a} \sum_{j=0}^N \omega_j \left(u_j^R u_j^L - \frac{1}{2} (u_j^R)^2 \right) - \tilde{a} \sum_{j=0}^N \omega_j \left(u_j^L u_j^L - \frac{1}{2} (u_j^L)^2 \right), \\ &= -\frac{\tilde{a}}{2} \sum_{j=0}^N \omega_j (u_j^R - u_j^L)^2 \leq 0, \end{aligned} \quad (2.149)$$

as $\tilde{a} = a \frac{\Delta y}{2} > 0$ (positive wave speed) and due to positive quadrature weights.

Secondly, we pay attention to the interfaces at the physical boundary. On the right boundary of the domain we are not instructed to a boundary condition, see (2.4). Including the upwind flux we get

$$\begin{aligned}
 B_R E_t &:= -\tilde{a} \sum_{j=0}^N \omega_j \left(u_{Nj}^{B_R} u_{Nj}^{B_R} - \frac{1}{2} \left(u_{Nj}^{B_R} \right)^2 \right), \\
 &= -\frac{\tilde{a}}{2} \sum_{j=0}^N \omega_j \left(u_{Nj}^{B_R} \right)^2 \leq 0.
 \end{aligned} \tag{2.150}$$

At last, we look at the interfaces on the left boundary of the domain with the upwind flux including the boundary condition

$$B_L E_t := \tilde{a} \sum_{j=0}^N \omega_j \left(u_{0j}^{B_L} \mathcal{B}(t, x_L, y_j^{n_y}) - \frac{1}{2} \left(u_{0j}^{B_L} \right)^2 \right), \tag{2.151}$$

where we assume that the interface is the n_y -th interface on the left boundary. As for the one dimensional LAE we complete the square, use the fact that a square of real numbers is always positive and arrive at

$$B_L E_t \leq \frac{\tilde{a}}{2} \sum_{j=0}^N \omega_j \mathcal{B}(t, x_L, y_j^{n_y})^2. \tag{2.152}$$

By focusing on the interior, right and left interfaces we considered all vertical interfaces. Note, that the derivations for all horizontal interfaces is analogous. By summing over all interfaces we obtain the discrete energy estimate

$$\begin{aligned}
 \sum_{q=1}^{N_Q} \sum_{i,j=0}^N J_{ij} \omega_i \omega_j \frac{1}{2} \frac{\partial \left(u_{ij}^q \right)^2}{\partial t} &\leq \underbrace{\sum_{n_y=1}^{N_{Q_y}} \frac{\tilde{a}}{2} \sum_{j=0}^N \omega_j \mathcal{B}(t, x_L, y_j^{n_y})^2}_{\text{Impact vertical interfaces}} \\
 &+ \underbrace{\sum_{n_x=1}^{N_{Q_x}} \frac{\tilde{b}}{2} \sum_{i=0}^N \omega_i \mathcal{B}(t, x_i^{n_x}, y_U)^2}_{\text{Impact horizontal interfaces}},
 \end{aligned} \tag{2.153}$$

which is precisely the definition of energy stability. \square

Throughout this proof we considered an upwind flux for the numerical surface flux due to positive wave speeds, see (2.100).

Remark 7. When considering a central flux with periodic boundary conditions and

re-deriving the proofs in Theorem 3, we obtain

$$\begin{aligned} \sum_{q=1}^{N_Q} \sum_{i,j=0}^N J_{ij} \omega_i \omega_j \frac{\partial u_{ij}^q}{\partial t} &= 0, \\ \sum_{q=1}^{N_Q} \sum_{i,j=0}^N J_{ij} \omega_i \omega_j \frac{1}{2} \frac{\partial (u_{ij}^q)^2}{\partial t} &= 0, \end{aligned} \tag{2.154}$$

which indicates primary and energy conservation.

To summarize, we extended the SBP method to the two dimensional LAE. This method is provable primary conservative and energy stable considering arbitrary SBP operators (FD, DG, HGTL). Due to the SBP operators we were able to reduce volume contributions to surface contributions when discretely applying the energy method, which is a discrete mimic of the IBP rule. In the next section we verify primary conservation and energy stability/conservation as well as high-order convergence and compare the error properties of the different SBP operators.

2.3. Numerical Verification of Summation-by-Parts Schemes for the Two Dimensional Linear Advection Equation

In this section we demonstrate the convergence properties as well as primary conservation and energy stability of the two dimensional SBP scheme. To do so, we focus on the LAE (2.105) on the domain $\Omega = [0, 1] \times [0, 1]$ and set the wave speeds to $a = b = 1$. We enforce periodic boundary conditions to ensure that $\int_{\Omega} \mathcal{U} d\Omega$ is constant over time. Unless stated otherwise, the upwind SAT/upwind flux is chosen. The time step is given by the *CFL* (Courant-Friedrichs-Lewy) condition [39]

$$\Delta t := \frac{CFL}{\max\{a, b\}} \frac{\min\{\frac{\Delta x}{2}, \frac{\Delta y}{2}\}}{N + 1}, \tag{2.155}$$

where Δx and Δy denote the width and height of an element and $N + 1$ denotes the number of nodes of an element in one spatial direction. In total, an element consists of $(N + 1)^2$ nodes. For simplicity we assume that all elements have the the same width, height and nodal distribution. To integrate the approximation in time we use the five-stage, fourth-order low-storage Runge-Kutta method of Carpenter and Kennedy [14] and set the final time to $T = 0.1$.

The numerical results are divided into two components. First, we compare different SBP operators with respect to their error and time step properties. Secondly, we verify primary conservation and energy stability of the SBP scheme.

2.3.1. Comparison and Convergence of Summation-by-Parts Schemes

In this section we compare the error properties of the two dimensional SBP scheme with FD, DG and HGTL operators via convergence analysis. For the convergence study, we consider the exact solution

$$\mathcal{U}(0, x, y) = 2 + \sin(2\pi(x - at)) + \cos(2\pi(y - bt)). \quad (2.156)$$

The experimental order of convergence (*EOC*) is determined by

$$EOC_d = \frac{\log\left(\frac{\mathcal{L}_{2,d}}{\mathcal{L}_{2,d-1}}\right)}{\log\left(\sqrt{\frac{DOFS_{d-1}}{DOFS_d}}\right)}, \quad (2.157)$$

where $\mathcal{L}_{2,d}$ denotes the \mathcal{L}_2 error and $DOFS_d$ denotes the total number of degrees of freedom (*DOFS*) in the domain at the d -th mesh level. Here, the \mathcal{L}_2 error is calculated using the quadrature weights of the corresponding norm matrix

$$\mathcal{L}_2 \text{ error} := \sqrt{\sum_{q=1}^{N_Q} \sum_{i,j=0}^N J_{ij} \omega_i \omega_j \left(\mathcal{U}(\cdot, x_i^q, y_j^q) - u_{ij}^q\right)^2}, \quad (2.158)$$

where x_i^q and y_j^q denote the components in physical space on the i - and j -th node within the q -th element. For the convergence study, we increase the number of $DOFS_d$ by increasing the number of elements, but not the number of nodes within an element. We note that this is an untypical approach when considering FD and HGTL operators. As in [70], the *DOFS* are increased by setting the number of elements fix and increasing the number of nodes within the element. However, when increasing the number of nodes for DG operators, the degree of the operator would change in each mesh refinement step. By setting the number of nodes within the element to be constant and increasing the number of elements, the order of the DG scheme remains constant when refining the mesh for the convergence study. In this section we chose the number of nodes and the elements such that the $DOFS_d$ remain constant at the d -th mesh level for all SBP operators. When considering FD or HGTL operators the d -th mesh level consists of 4^d elements within the spatial domain (2^d in the x - and y -direction) and for DG operators the d -th mesh level has 4^{d+2} elements (2^{d+2} in the x - and y -direction). The DG operators includes $(p+1)^2$ nodes per element ($p+1$ in the x - and y -direction). By setting the nodes for FD and HGTL in one spatial direction to $4(p+1)$, the $DOFS_d$ remain constant at the d -th mesh level.

$$DOFS_d = \underbrace{4^d}_{\#FD/HGTL \text{ elements}} \overbrace{(4(p+1))^2}^{\#FD/HGTL \text{ nodes}} = \underbrace{4^{d+2}}_{\#DG \text{ elements}} \overbrace{(p+1)^2}^{\#DG \text{ nodes}}. \quad (2.159)$$

Besides calculating the error and the *EOC* of the scheme, we are interested in the maximum step. For linear problems with periodic boundary conditions it is possible to rewrite the semi-discrete formulation in the following global form

$$\mathbf{u}_t = \mathbf{A}\mathbf{u}. \quad (2.160)$$

Let $\{\lambda_i\}_{i=1}^{N_Q(N+1)^2}$ denote the eigenvalues of \mathbf{A} . The low storage five-stage, fourth-order Runge-Kutta scheme is said to be stable, in a sense that the simulation does not crash, if the time step satisfies

$$\underbrace{\left(1 + (\lambda_i \Delta t) + \frac{1}{2}(\lambda_i \Delta t)^2 + \frac{1}{6}(\lambda_i \Delta t)^3 + \frac{1}{24}(\lambda_i \Delta t)^4 + \frac{1}{200}(\lambda_i \Delta t)^5\right)}_{=:R_i(\Delta t)} \leq 1, \quad (2.161)$$

for all $i = 1, \dots, N_Q(N+1)^2$ [14]. To obtain the maximum time step we calculate Δt_i^* , such that

$$R_i(\Delta t_i^*) = 1. \quad (2.162)$$

The maximum time step is determined by

$$\Delta t^* := \min_{1 \leq i \leq N_Q(N+1)^2} \{\Delta t_i^*\}. \quad (2.163)$$

Since the time step is dependent on the number of points of the spatial resolution and wave speed, we normalize the maximum time step which gives us the maximum *CFL*

$$CFL^* := \Delta t^* \frac{(N+1) \max\{a, b\}}{\min\{\frac{\Delta x}{2}, \frac{\Delta y}{2}\}}. \quad (2.164)$$

Finally, we solve the LAE with different sets of SBP operators with degrees of $p = 2, 3, 4$. For all SBP operators the *EOC* and the maximum *CFL* number is shown in the Tables 2.1-2.3.

Focusing on $p = 2$ in Table 2.1 we obtain an expected *EOC* of $p + 1$ [39]. Here, in each mesh level the HGTL operator has the best error properties. The FD operator is compatible as its maximum *CFL* number is larger ($2.80 > 2.77$) but also its \mathcal{L}_2 error is slightly higher. More significant is the difference towards the DG operator, where the \mathcal{L}_2 error is about two times as large as for the error of the HGTL operator and its maximum *CFL* number of 2.05 is noticeably smaller.

For $p = 3$ in Table 2.2 we also obtain an *EOC* of $p + 1$. The maximum *CFL* number decays for all SBP operators which is excepted when increasing the degree p . Again the FD operator has the best maximum *CFL* number of 2.39 whereas

EOC and maximum CFL number for solving the LAE with DG, FD and HGTL operators for $p = 2, 3, 4$.

<i>DOFS</i>	\mathcal{L}_2DG	<i>EOCDG</i>	\mathcal{L}_2FD	<i>EOCFD</i>	\mathcal{L}_2HGTL	<i>EOCHGTL</i>
576	9.63E-03		3.37E-03		2.25E-03	
2304	1.13E-03	3.1	5.54E-04	2.6	3.80E-04	2.6
9216	1.44E-04	3.0	9.03E-05	2.6	5.85E-05	2.7
36864	1.80E-05	3.0	1.06E-05	3.1	6.90E-06	3.1
147456	2.25E-06	3.0	1.32E-06	3.0	8.56E-07	3.0
589824	2.87E-07	3.0	1.64E-07	3.0	1.07E-07	3.0
	Max. <i>CFL</i>	2.06	Max. <i>CFL</i>	2.80	Max. <i>CFL</i>	2.77

Table 2.1.: Setting $p = 2$, $CFL = 1$, $T = 0.1$ and the number of nodes per element in one dimension for FD and HGTL operators to $N + 1 = 12$.

<i>DOFS</i>	\mathcal{L}_2DG	<i>EOCDG</i>	\mathcal{L}_2FD	<i>EOCFD</i>	\mathcal{L}_2HGTL	<i>EOCHGTL</i>
1024	4.19E-04		4.73E-04		1.19E-04	
4096	2.62E-05	4.0	3.07E-05	3.9	6.70E-06	4.2
16384	1.49E-06	4.1	2.25E-06	3.8	3.58E-07	4.2
65536	9.42E-08	4.0	1.37E-07	4.0	2.14E-08	4.1
262144	5.89E-09	4.0	8.47E-09	4.0	1.22E-09	4.1
1048576	3.70E-10	4.0	5.24E-10	4.0	7.55E-11	4.0
	Max. <i>CFL</i>	1.63	Max. <i>CFL</i>	2.39	Max. <i>CFL</i>	2.35

Table 2.2.: Setting $p = 3$, $CFL = 1$, $T = 0.1$ and the number of nodes per element in one dimension for FD and HGTL operators to $N + 1 = 16$.

<i>DOFS</i>	\mathcal{L}_2DG	<i>EOCDG</i>	\mathcal{L}_2FD	<i>EOCFD</i>	\mathcal{L}_2HGTL	<i>EOCHGTL</i>
1600	1.48E-05		5.54E-05		4.43E-06	
6400	4.96E-07	4.9	2.13E-06	4.7	1.43E-07	5.0
25600	1.23E-08	5.3	9.56E-08	4.5	5.94E-09	4.6
102400	4.29E-10	4.8	5.10E-09	4.2	3.21E-10	4.2
409600	1.34E-11	5.0	2.79E-10	4.2	8.74E-11	1.9
1638400	8.56E-13	4.0	1.36E-11	4.4	3.91E-11	1.2
	Max. <i>CFL</i>	1.38	Max. <i>CFL</i>	2.16	Max. <i>CFL</i>	2.13

Table 2.3.: Setting $p = 4$, $CFL = 1$, $T = 0.1$ and the number of nodes per element in one dimension for FD and HGTL operators to $N + 1 = 20$.

the DG operator has the worst ($CFL^* = 1.63$). In comparison to Table 2.1 the DG operator has a better error than the FD operator. Still, the error is not better than that obtained by the HGTL operator. Comparing the HGTL and the FD operator we notice a similar behaviour of the error properties and maximum CFL number as in Table 2.1.

Next, we focus on Table 2.3 for $p = 4$. As in Table 2.1-2.2 the maximum CFL number decreases, but the relationship between these numbers for the SBP operators remain the same (FD best, DG worst). Here, the DG and HGTL operator have similar error properties. When considering less $DOFS$ than 409600, the HGTL operator is slightly better, but for more $DOFS$ the DG operator has the best error properties. Both of these operators have a better error than the FD operator (ignoring the last mesh level), but again the FD operator has the highest maximum CFL number of 2.16. We note that for $p = 4$ we do not obtain an EOC of $p + 1$. A possible explanation for this behaviour is the fourth order time integration method. Presumably the error of time integration method dominates the spatial error of the SBP discretization. Therefore, we set up an extra run with $CFL = 0.01$ and present the results in Table 2.4

EOC for solving the LAE with with a reduced CFL number.

$DOFS$	$\mathcal{L}_2 DG$	$EOCDG$	$\mathcal{L}_2 FD$	$EOCFD$	$\mathcal{L}_2 HGTL$	$EOCHGTL$
1600	1.48E-05		5.54E-05		4.43E-06	
6400	4.96E-07	4.9	2.13E-06	4.7	1.43E-07	5.0
25600	1.24E-08	5.3	7.99E-08	4.7	5.04E-09	4.8
102400	4.17E-10	4.9	2.66E-09	4.9	1.97E-10	4.7

Table 2.4.: Setting $p = 4$, $CFL = 0.01$, $T = 0.1$ and the number of nodes per element in one dimension for FD and HGTL operators to $N + 1 = 20$. We only considered four mesh levels due to $CFL = 0.01$.

Due to computational effort in Table 2.4 we only considered four mesh levels. Here, we verify a convergence order of $EOC \approx p + 1$. Note, that in Table 2.3 and 2.4 the \mathcal{L}_2 errors for the first two mesh levels remain the same. This indicates that the time step only effects the error for a fine mesh resolution.

To summarize, we observe that when considering FD operator the maximum CFL number is higher than for operators on nodes which are not uniformly distributed. However, for high-order simulations ($p > 2$) the DG and HGTL operators have better error properties. For $p = 3$ we observe the error of the HGTL operator

is smaller than for DG, but considering a fine mesh at $p = 4$, than the DG operator has the best error. Due to the results presented in Table 2.1-2.3 we cannot claim which operator is "the best", but we can see that all operators can be taken into consideration when solving hyperbolic conservation laws.

2.3.2. Numerical Verification of Primary Conservation and Energy Stability

As was shown in Section 2.1.2, the SBP scheme is primary conservative and energy stable. The SBP discretization is defined by

$$\omega_i \omega_j J_{ij} \frac{\partial u_{ij}^q}{\partial t} = Res(u_{ij}^q), \quad (2.165)$$

with

$$\begin{aligned} Res(u_{ij}^q) = & -\omega_j \left([\tilde{f}_{Nj}^{*,q} - \tilde{f}_{Nj}^q] \delta_{iN} - [\tilde{f}_{0j}^{*,q} - \tilde{f}_{0j}^q] \delta_{i0} + \sum_{k=0}^N \mathbf{Q}_{ik} \tilde{f}_{kj}^q \right) \\ & - \omega_i \left([\tilde{g}_{iN}^{*,q} - \tilde{g}_{iN}^q] \delta_{Nj} - [\tilde{g}_{i0}^{*,q} - \tilde{g}_{i0}^q] \delta_{0j} + \sum_{l=0}^N \mathbf{Q}_{jl} \tilde{g}_{il}^q \right), \end{aligned} \quad (2.166)$$

with $i, j = 0, \dots, N$ and $q = 1, \dots, N_Q$. We choose the numerical flux to be

$$f^*(\mathcal{U}^R, \mathcal{U}^L) = a \frac{\mathcal{U}^R + \mathcal{U}^L}{2} - \sigma \frac{|a|}{2} (\mathcal{U}^R - \mathcal{U}^L), \quad (2.167)$$

with $\sigma \in [0, 1]$ and similar for g^* . Setting $\sigma = 0$, we get the central flux for which the SBP discretization remains energy conservative and for $\sigma = 1$ we obtain the upwind flux/upwind SAT for which the scheme is energy stable. Here, we note that energy conservative schemes do not necessarily represent the correct physical behaviour of a wave as in real world application the energy must dissipate due to discontinuities. However, just for verification that the numerical scheme is implemented correctly, we set $\sigma = 0$ to verify energy conservation. Therefore, we multiply (2.165) with u_{ij}^q and sum over all elements and nodes

$$\sum_{q=1}^{N_Q} \sum_{i,j=0}^N \omega_i \omega_j J_{ij} u_{ij}^q \frac{\partial u_{ij}^q}{\partial t} = \sum_{q=1}^{N_Q} \sum_{i,j=0}^N u_{ij}^q Res(u_{ij}^q). \quad (2.168)$$

Similarly, primary conservation is described by

$$\sum_{q=1}^{N_Q} \sum_{i,j=0}^N \omega_i \omega_j J_{ij} \frac{\partial u_{ij}^q}{\partial t} = \sum_{q=1}^{N_Q} \sum_{i,j=0}^N Res(u_{ij}^q). \quad (2.169)$$

We verify conservation of the primary quantity and energy by calculating

$$\begin{aligned}
 E_t^{tot} &:= \sum_{q=1}^{N_Q} \sum_{i,j=0}^N u_{ij}^q Res(u_{ij}^q), \\
 U_t^{tot} &:= \sum_{q=1}^{N_Q} \sum_{i,j=0}^N Res(u_{ij}^q).
 \end{aligned}
 \tag{2.170}$$

Here, E_t^{tot} and U_t^{tot} describe the discrete growth of the total energy and the discrete growth of the total primary quantity. We consider the discontinuous initial condition

$$\mathcal{U}(t, x) = \begin{cases} \vartheta_1 & \text{if } x \leq y, \\ \vartheta_2 & \text{else,} \end{cases}
 \tag{2.171}$$

with periodic boundary conditions. Here, ϑ_1 and ϑ_2 are uniformly generated random numbers in $[0, 1]$. The random initial condition is chosen to demonstrate conservation independent of the initial condition. We calculate $(E_t^{tot})_k$ and $(U_t^{tot})_k$ for $k = 1, \dots, 1000$ independently by considering a FD operator of degree $p = 3$ and $N_Q = 100$ elements with $(N + 1) = 16^2$ nodes (16 nodes in x - and y -direction).

Calculating the discrete growth in the total energy and the total primary quantity

$\ U_t^{tot}\ _2$	5.51E-13
$\ E_t^{tot}\ _2$	7.19E-13

Table 2.5.: Calculating the growth in the total discrete primary quantity U_t^{tot} and total discrete energy E_t^{tot} for 1000 different random initial conditions. The growth is presented within the euclidean norm. Here, we verify conservation of the primary quantity and energy.

In Table 2.5 we verify primary and energy conservation. Besides only considering random initial data, we are interested of how the total primary quantity and total energy behave in time. Therefore we consider the initial condition

$$\mathcal{U}(t, x) = \begin{cases} 2 & \text{if } 0.3 \leq x \leq 0.7 \text{ and } 0.3 \leq y \leq 0.7, \\ 0 & \text{else.} \end{cases}
 \tag{2.172}$$

We set $T = 10$, $CFL = 1$ and focus again on a FD operator of degree $p = 3$ with $N_Q = 100$ and $(N + 1)^2 = 16^2$. We run the simulation twice considering either

the central or upwind flux. The total discrete primary quantity and energy are defined by

$$\begin{aligned}
 E^{tot} &= \sum_{q=1}^{N_Q} \sum_{i,j=0}^N \frac{1}{2} \left(u_{ij}^q \right)^2, \\
 U^{tot} &= \sum_{q=1}^{N_Q} \sum_{i,j=0}^N u_{ij}^q,
 \end{aligned}
 \tag{2.173}$$

and are plotted over time in Figure 2.6 and 2.7.

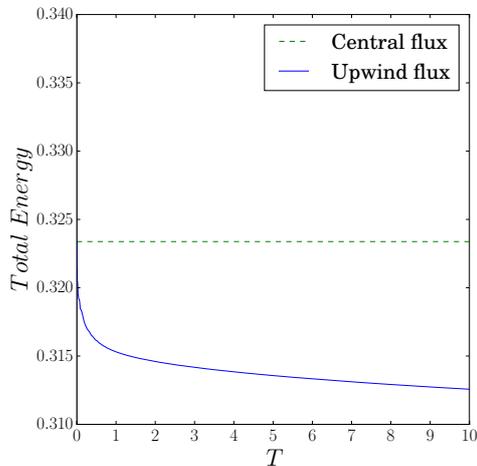


Figure 2.6.: Evolution of the total discrete energy of the solution with and without dissipation. We see that the total discrete energy is conserved when considering the central flux ($\sigma = 0$) and that it decays using the upwind flux ($\sigma = 1$).

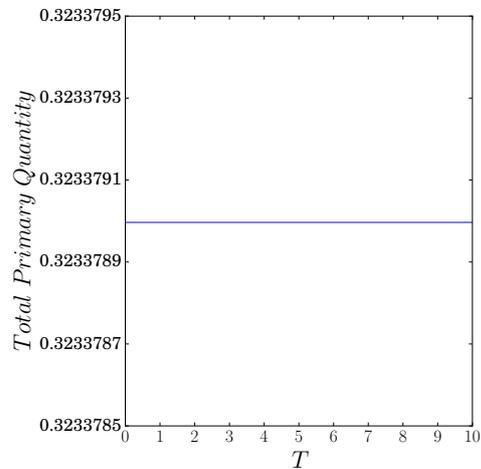


Figure 2.7.: A plot that demonstrates the conservation of the integrated conserved variables. The plot does not depend on the choice of the numerical surface flux.

Here, we verify that the total discrete energy is conserved when considering a central flux. By introducing a dissipation term ($\sigma > 0$) within the numerical flux, the total discrete energy is dissipated over time. In comparison, the total discrete primary quantity does not depend on the choice of the numerical surface flux.

Remark 8. In this section we only focused on FD operators. However, we obtain the same results when considering any other SBP operator.

2.4. Summary

In Section 2.1 we introduced the SBP method for the one dimensional linear advection equation. The linear advection equation is naturally primary conservative and energy stable when considering appropriate boundary conditions, which we demonstrated using integration-by-parts. To maintain this property within the numerical scheme, we introduced summation-by-parts operators. These operators are defined to discretely mimic the integration-by-parts rule. Given a summation-by-parts operator the SBP scheme is provably primary conservative and energy stable. Such operators can be derived from finite difference formulas or from a weak formulation of the conservation law. In extension we introduced the two dimensional SBP method focusing on quadrilateral elements and using the tensor product. We implemented the SBP method and compared the numerical errors and the maximum time step for different summation-by-parts operators. Based on the degree of the differentiation matrix, we verified that the SBP method is high-order. Also, we analyzed the behaviour of the total discrete primary quantity and energy with discontinuous initial data. Since the total discrete primary quantity remained constant and the total discrete energy did not increase, we demonstrated that our implementation of the SBP scheme is primary conservative and energy stable.

3. Energy Stable and Primary Conservative Summation-by-Parts Methods for the Linear Advection Equation on Non-Conforming Meshes

For two (or even three) dimensional problems the distribution of elements or, in other words, the mesh is substantial. Many real world problems contain a wide range of length scales. The efficient approximation of such problems necessitates the ability to judiciously distribute degrees of freedom. Therefore, we focus on *non-conforming numerical approximations*. Non-conforming numerical approximations offer increased flexibility for applications that require high resolution in a localized area of the computational domain or near complex geometries. For example, consider a flow around a cylinder as in Figure 3.1.

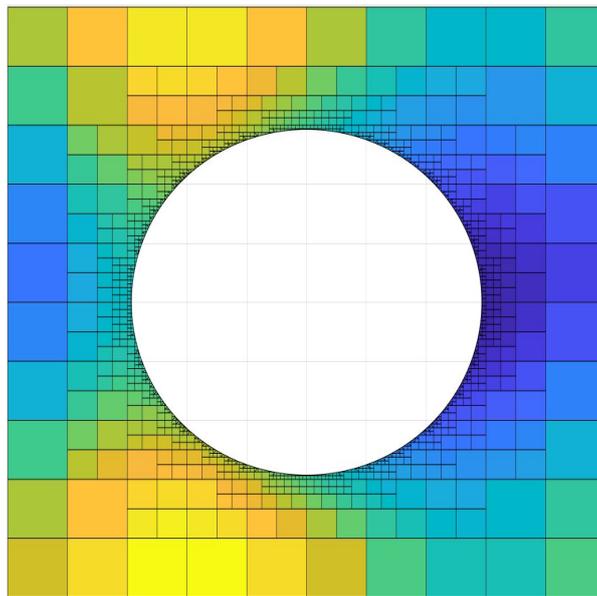


Figure 3.1.: A constructed non-conforming mesh to simulate a flow around a cylinder.

We are interested in how fluids behave near object, e.g. near the cylinder in Figure 3.1. In terms of computationally efficiency we desire a mesh with fine elements at the cylinder surface and a coarse mesh at the boundary of the domain. For more details on mesh configuration see Chapter 7. In general, designing a conforming mesh around a geometry is cumbersome. The main difficulties occur due to the interfaces and the nodal distributions of the elements which need to coincide with the neighbouring element, see Figure 3.2.

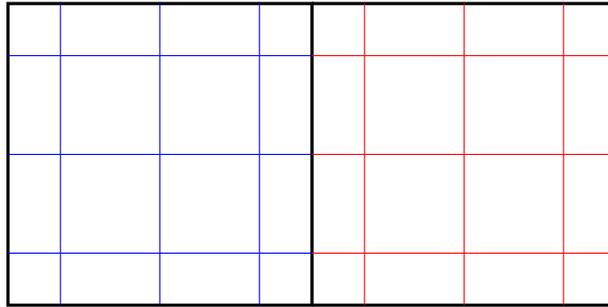


Figure 3.2.: Conforming elements where the interfaces and the nodal distribution coincide. The figure is reproduced from [38].

In order to overcome these problems, we consider meshes where neither the nodal distributions of the element nor the interfaces need to coincide. Focusing on a quadrilateral mesh, we allow the nodal distribution to differ between elements (Figure 3.3), sometimes called p refinement or algebraic non-conforming. In addition the mesh can contain hanging corners (Figure 3.4), sometimes called h refinement or geometric non-conforming. Combining p and h refinement we obtain a fully h/p non-conforming approach (Figure 3.5).

In this chapter we present a primary conservative and energy stable non-conforming SBP method. The key to the non-conforming approximation is how the numerical fluxes or SATs between neighbor interfaces are treated. In comparison, for conforming approximations as in the previous chapter the interface points between two neighboring elements coincide. This allowed for a straightforward definition of unique numerical surface fluxes to account for how information is transferred between neighbors. It is then possible to determine numerical surface fluxes that guaranteed energy stability/conservation of the conforming approximation.

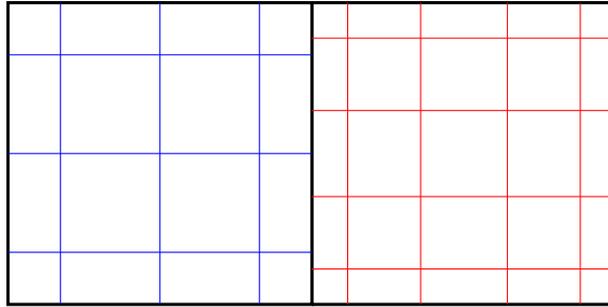


Figure 3.3.: Neighboring elements with different nodal distributions. The figure is reproduced from [38].

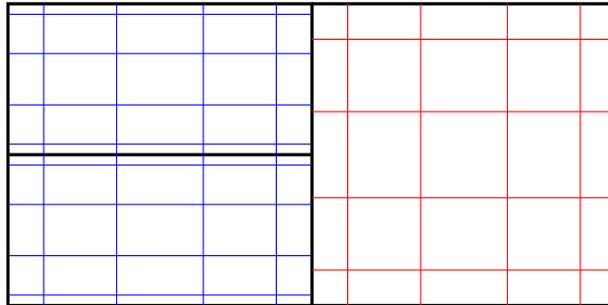


Figure 3.4.: Neighboring elements with a hanging corners. The figure is reproduced from [38].

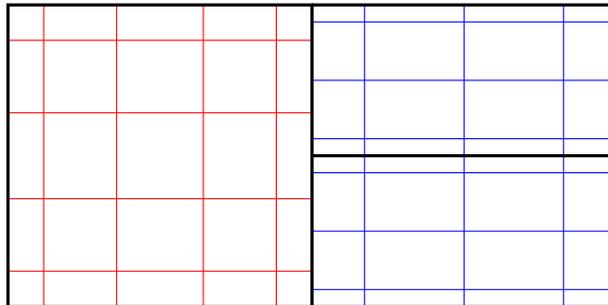


Figure 3.5.: Neighboring elements with a hanging corner and differing nodal distribution. The figure is reproduced from [38].

This chapter is structured as followed: First we will gather insight into the development of non-conforming SBP methods in the past. All these methods rely on interpolation/projection operators to connect different types of non-conforming elements. Based on these methods we will introduce a fully conservative (primary and energy conservative) discretization for the LAE on algebraic or p non-conforming elements. Next, we extend the method to geometric or h non-conforming elements. As fully conservative schemes do not reflect the actual physics of a conservation law, we include numerical interface dissipation and obtain an energy stable scheme. Before, doing numerical tests, we investigate in the properties of the interpolation/projection operators. It is worth noting, that non-conforming approximations (in comparison to conforming approximations) are less accurate due to the reduction of the convergence rate [39]. Therefore, we introduce a new set of SBP operators and interpolation/projection operators, for which the optimal convergence rate is retained. Finally, we verify the properties as primary conservation, energy stability/conservation and high-order accuracy for the introduced SBP operators.

3.1. History on Non-Conforming Summation-by-Parts Methods

In this section we introduce existing non-Conforming SBP methods. Therefore, we focus on a simple prototype of non-conforming elements as in Figure 3.3. Here, we have two elements where the interfaces coincide, but not the nodal distributions (p refinement). We differ between a left element L and a right element R . In each element we consider a different number and set of nodes. Let $\boldsymbol{\eta}^L, \boldsymbol{\eta}^R$ denote the set of vertical nodes of size $N_L + 1$ and $N_R + 1$, respectively. For each set of nodes we consider a different SBP operator on each element $(\mathbf{M}_L, \mathbf{D}_L)$ and $(\mathbf{M}_R, \mathbf{D}_R)$. As nodal distributions do not coincide, it is not possible to do a point-to-point transfer between the elements to evaluate the numerical flux/SAT. For the SBP-SAT method, Mattsson and Carpenter [83] developed interpolation operators that result in a primary conservative and energy stable scheme for linear hyperbolic systems on Cartesian meshes. Here, the constructed operators

$$\mathbf{P}_{L2R} \in \mathbb{R}^{(N_R+1) \times (N_L+1)}, \quad \mathbf{P}_{R2L} \in \mathbb{R}^{(N_L+1) \times (N_R+1)}, \quad (3.1)$$

interpolate the nodes from one element to the neighbouring non-conforming element, see Figure 3.6. The subscript " $L2R$ " denotes the interpolation from the left to the right element. Mattsson and Carpenter were able to re-derive the proof for primary conservation and energy stability for the SBP-SAT method for linear systems of conservation laws provided that the interpolation operators satisfy the

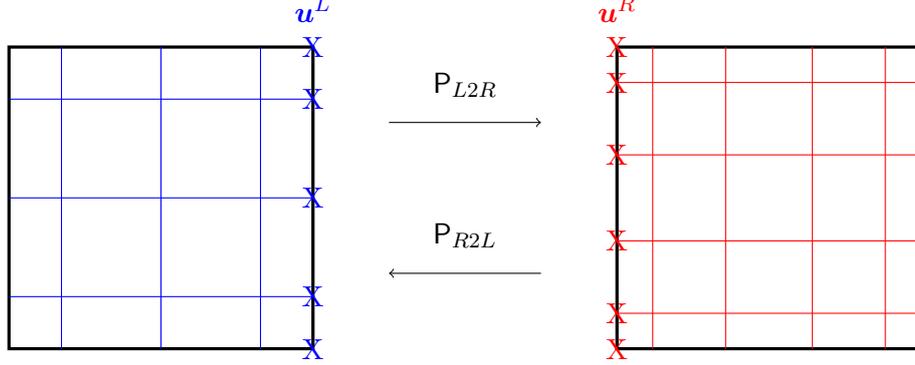


Figure 3.6.: Interpolating the nodes on the interface from one element to another.

M-compatibility condition

$$\mathbf{P}_{L2R}^T \mathbf{M}_R = \mathbf{M}_L \mathbf{P}_{R2L}. \quad (3.2)$$

The M-compatibility condition can also be found in [16, 38, 39, 70, 80, 83, 96, 97] and is necessary to obtain an energy stable non-conforming SBP method. To apply the two dimensional discretization for scalar conservation laws (2.129) on a non-conforming mesh, we need to specify the SAT/numerical flux. Let \mathbf{u}^L denote the solution on L over the nodes along its right interface and let \mathbf{u}^R denote the solution on R over the nodes along its left interface (see Figure 3.6). Here, the one dimensional vectors \mathbf{u}^L and \mathbf{u}^R are of different sizes, as we consider a non-conforming mesh. Thus, the projection operators \mathbf{P}_{L2R} and \mathbf{P}_{R2L} are non-square matrices. We discretize R with (2.129) and focus on the left surface. For the LAE, Mattsson and Carpenter choose the following SAT to obtain an energy conservative scheme

$$\mathbf{SAT}_{EC} := [\tilde{\mathbf{f}}_{EC}^{*,R} - \tilde{\mathbf{f}}^R] = \frac{\tilde{a}}{2} \mathbf{P}_{L2R} \mathbf{u}^L - \frac{\tilde{a}}{2} \mathbf{u}^R, \quad (3.3)$$

where the subscript EC denotes that the SAT is energy conservative. Also, we included the index R within the numerical flux as the numerical flux differs on the left and right element. By rearranging (3.3) and inserting $\tilde{\mathbf{f}}^R = \tilde{a} \mathbf{u}^R$ we obtain the following energy conservative numerical surface flux

$$\tilde{\mathbf{f}}_{EC}^{*,R} = \tilde{a} \frac{\mathbf{u}^R + \mathbf{P}_{L2R} \mathbf{u}^L}{2}, \quad (3.4)$$

which can be interpreted as a non-conforming central flux. To get an energy stable scheme we include interface dissipation by

$$\tilde{\mathbf{f}}_{ES}^{*,R} = \tilde{\mathbf{f}}_{EC}^{*,R} - \frac{|\tilde{a}|}{2} \left(\frac{1}{2} (\mathbf{P}_{L2R} \mathbf{P}_{R2L} + \mathbf{I}_R) \mathbf{u}^R - \mathbf{P}_{L2R} \mathbf{u}^L \right), \quad (3.5)$$

where \mathbf{I}_R denotes the identity matrix of size N_R+1 . Here, the subscript *ES* denotes that the numerical surface flux is energy stable. The flux (3.5) can be interpreted as a non-conforming Lax-Friedrichs flux. Note, that when considering a conforming mesh, meaning that $\mathbf{P}_{L2R} = \mathbf{P}_{R2L} = \mathbf{I}_R$, we get the classical upwind flux (2.148).

Due to (3.2), the interpolation operators depend on the norm matrices. In [83] a certain set of such interpolation operators is given for SBP FD operator. These operators are sufficient for certain test cases, but not in general, e.g. the interpolation operator connecting FD operators of degree $p_L = 2$ and $p_R = 3$ is not provided. Constructing the projection operators in general is cumbersome as the projection operators depend on at least two elements (even more for h refinement).

For DG schemes, Kopriva [66, 69] invented the so called *DG Mortar method* for arbitrary conservation laws. The method is also applicable to curvilinear elements. However, for simplicity of the presentation we focus on Cartesian meshes (\tilde{a} remains constant for the left and right element). Here, the operators \mathbf{P}_{L2R} and \mathbf{P}_{R2L} are derived by a discrete \mathcal{L}_2 projection and are denoted as *projection operators*. In comparison to the method of Mattsson and Carpenter, the mortar method constructs the projection operators by first projecting the solution on the interface nodes of the elements onto an intermediate grid and then projecting back the numerical flux to the surface of the corresponding element as in Figure 3.7.

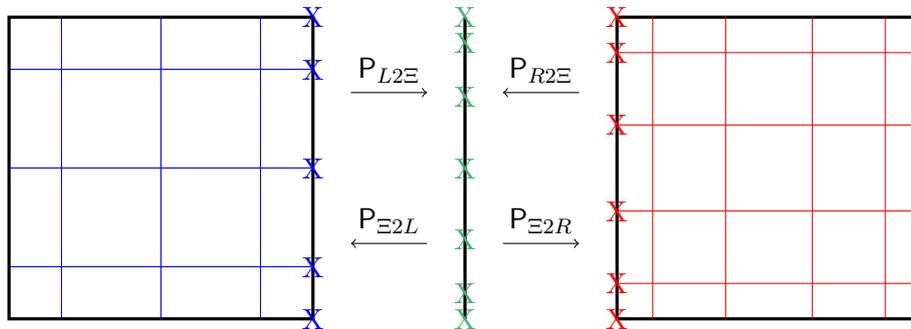


Figure 3.7.: Two conforming elements with non-conforming nodal distributions and an intermediate grid.

Here, the intermediate grid is referred to as *mortar* (the "cement") as it connects adjacent elements (the "bricks"). The operators \mathbf{P}_{L2E} and \mathbf{P}_{R2E} project the solution on the left and right element to the mortar, whereas \mathbf{P}_{E2L} and \mathbf{P}_{E2R} denote the projection back to the element. The main advantage of this approach is that

one can construct interpolation/projection operators independently of the neighbouring elements as all element interfaces are projected to the same set of nodes. This overcomes the difficulty of constructing projection operators as in the paper of Mattsson and Carpenter. In addition, the mortar method has excellent parallel computing properties [9, 60].

The mortar has its own set of nodes $\boldsymbol{\eta}^\Xi$ and a norm matrix \mathbf{M}_Ξ . In [69] the nodes are typically chosen to be LG or LGL nodes. For Koprivas' method, a unique numerical surface flux is evaluated on the mortar. Considering the LAE and a Lax-Friedrichs numerical flux on the mortar, then the flux is calculated by

$$\tilde{\mathbf{f}}^{*,\Xi} = \tilde{a} \frac{\mathbf{P}_{L2\Xi} \mathbf{u}^L + \mathbf{P}_{R2\Xi} \mathbf{u}^R}{2} - \frac{|\tilde{a}|}{2} (\mathbf{P}_{R2\Xi} \mathbf{u}^R - \mathbf{P}_{L2\Xi} \mathbf{u}^L). \quad (3.6)$$

Using the back projection we get the numerical fluxes for each element

$$\tilde{\mathbf{f}}^{*,L} = \mathbf{P}_{\Xi 2L} \tilde{\mathbf{f}}^{*,\Xi}, \quad \tilde{\mathbf{f}}^{*,R} = \mathbf{P}_{\Xi 2R} \tilde{\mathbf{f}}^{*,\Xi}. \quad (3.7)$$

The mortar method is primary conservative for all system of hyperbolic conservation laws on Cartesian or curvilinear meshes. However, the method is not necessarily entropy stable/conservative (energy stable/conservative for the LAE). One reason is that the projection operators do not necessarily satisfy the M-compatibility condition (3.2) when applying a discrete projection.

A more recent non-conforming method was developed by Kozdon and Wilcox [70]. Here, projection operators are constructed by also introducing an intermediate grid called the *glue grid*, as it "glues" the elements together. In addition, the operators are constructed such that they satisfy the M-compatibility condition

$$\begin{aligned} \mathbf{P}_{L2\Xi}^T \mathbf{M}_\Xi &= \mathbf{M}_L \mathbf{P}_{\Xi 2L}, \\ \mathbf{P}_{R2\Xi}^T \mathbf{M}_\Xi &= \mathbf{M}_R \mathbf{P}_{\Xi 2R}. \end{aligned} \quad (3.8)$$

Note, that (3.8) can be satisfied independent of the neighboring element. After deriving operators satisfying (3.8), we obtain the projection operators connecting the left and right element by

$$\mathbf{P}_{L2R} = \mathbf{P}_{\Xi 2R} \mathbf{P}_{L2\Xi}, \quad \mathbf{P}_{R2L} = \mathbf{P}_{\Xi 2L} \mathbf{P}_{R2\Xi}. \quad (3.9)$$

This construction maintains the M-compatibility condition (3.2) as

$$\begin{aligned} \mathbf{P}_{L2R}^T \mathbf{M}_R &= \mathbf{P}_{L2\Xi}^T \underbrace{\mathbf{P}_{\Xi 2R}^T \mathbf{M}_R}_{=\mathbf{M}_\Xi \mathbf{P}_{R2\Xi}} = \underbrace{\mathbf{P}_{L2\Xi}^T \mathbf{M}_\Xi}_{=\mathbf{M}_L \mathbf{P}_{\Xi 2L}} \mathbf{P}_{R2\Xi}, \\ &= \mathbf{M}_L \mathbf{P}_{\Xi 2L} \mathbf{P}_{R2\Xi} = \mathbf{M}_L \mathbf{P}_{R2\Xi}. \end{aligned} \quad (3.10)$$

Note, that the norm matrices are symmetric and thus $M^T = M$. The method is primary conservative and energy stable on Cartesian meshes for linear systems of hyperbolic conservation laws. Here, the numerical fluxes are uniquely determined on the glue grid as for the mortar method (3.7). To finally obtain an energy stable/conservative scheme the SAT needs to be modified by a correction term **COR**

$$\begin{aligned} \mathbf{SAT} &:= [\tilde{\mathbf{f}}^{*,R} - \tilde{\mathbf{f}}^R] + \mathbf{COR}, \\ &= [\mathbf{P}_{\Xi 2R} \tilde{\mathbf{f}}^{*,\Xi} - \tilde{\mathbf{f}}^R] + \frac{\tilde{a}}{2} (\mathbf{u}^R - \mathbf{P}_{\Xi 2R} \mathbf{P}_{R 2\Xi} \mathbf{u}^R). \end{aligned} \quad (3.11)$$

Note, that after the evaluation of the numerical flux on the glue grid, the SAT purely depends on the right element. Thus, the storage of the projection operators can be done locally on each element without considering their neighbors. We verify energy conservation directly by inseting the central numerical flux.

$$\begin{aligned} \mathbf{SAT}_{EC} &= \left[\frac{\tilde{a}}{2} \mathbf{P}_{\Xi 2R} (\mathbf{P}_{R 2\Xi} \mathbf{u}^R + \mathbf{P}_{L 2\Xi} \mathbf{u}^L) - \tilde{a} \mathbf{u}^R \right] + \frac{\tilde{a}}{2} (\mathbf{u}^R - \mathbf{P}_{\Xi 2R} \mathbf{P}_{R 2\Xi} \mathbf{u}^R), \\ &= \frac{\tilde{a}}{2} \underbrace{\mathbf{P}_{\Xi 2R} \mathbf{P}_{L 2\Xi}}_{=\mathbf{P}_{L 2R}} \mathbf{u}^L - \frac{\tilde{a}}{2} \mathbf{u}^R = \frac{\tilde{a}}{2} \mathbf{P}_{L 2R} \mathbf{u}^L - \frac{\tilde{a}}{2} \mathbf{u}^R, \end{aligned} \quad (3.12)$$

which is precisely the same numerical SAT as in the approach of Mattsson and Carpenter (3.3). To include numerical dissipation we consider the Lax-Friedrichs flux on the intermediate/glue grid as in (3.6).

In addition, Kozdon and Wilcox [70] extended their scheme to curvilinear meshes, but due to this extension the scheme is not primary conservative. Deriving primary conservative and at the same time energy stable schemes on a non-conforming curvilinear mesh with h/p refinement is one of today's research topics. Focusing on curved meshes Carpenter et al. [16] developed an energy stable and primary conservative scheme considering p refinement by modifying the SAT. Up to now there is no energy stable SBP method for general conservation laws considering h refinement on a curved mesh.

In this work we aim to simulate non-linear problems on complex geometries as in Figure 3.1. A generation of such meshes is cumbersome, especially for conforming meshes, as here all interfaces need to coincide. However, this is not required for non-conforming meshes with h refinement. As we want to ensure an energy/entropy stable simulation, we consider Cartesian meshes for element refinement.

3.2. Construction of M-Compatible Projection Operators

In this section we construct M-compatible projection operators for all introduced SBP operators in Section 2.1. As in the last section, we purely focus on p refinement as the extension for h refinement is straightforward (see Section 3.4.2). Here, we focus on an element L and an intermediate grid Ξ as in Figure 3.8.

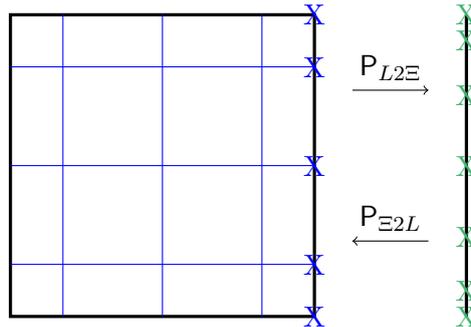


Figure 3.8.: Single element and an intermediate grid.

Let $\boldsymbol{\eta}^L$ and $\boldsymbol{\eta}^\Xi$ denote the one dimensional set of nodes of size $N_L + 1$ and $N_\Xi + 1$ in η -direction and \mathbf{M}_L and \mathbf{M}_Ξ denote the corresponding diagonal norm matrices. Besides satisfying the M-compatibility condition (3.2), we also discuss the accuracy of the projection operators $\mathbf{P}_{L2\Xi}$ and $\mathbf{P}_{\Xi2L}$. Therefore, we define the degree of a projection operator.

Definition 9. Degree of a Projection Operator

The projection operators $\mathbf{P}_{L2\Xi}$ is of degree p , if

$$\mathbf{P}_{L2\Xi}(\boldsymbol{\eta}^L)^k = (\boldsymbol{\eta}^\Xi)^k, \quad (3.13)$$

for $k = 0, \dots, p$.

How to construct such projection operators of high degree depends on the corresponding SBP operators defined on the element L . First, we construct projection operators for elements with DG operators.

3.2.1. Projection Operators for Discontinuous Galerkin

In this section we derive projection operators for elements with DG operators. Therefore, we adapt the Mortar element approach of Kopriva [69]. Here, we assume that the nodes on the element and on the intermediate grid/mortar are LGL

nodes with corresponding Lagrange basis function ℓ_i^L, ℓ_j^{Ξ} with $i = 0, \dots, N_L$ and $j = 0, \dots, N_{\Xi}$. For the mortar method we set $N_{\Xi} > N_L$.

First, we discuss the projection from the element to the mortar ($\mathbf{P}_{L2\Xi}$). We assume that the solution is evaluated at $\mathbf{u} = (u_0, \dots, u_{N_L})^T$ with the polynomial ansatz $\mathcal{U}(\eta) = \sum_{i=0}^{N_L} \ell_i^L(\eta) u_i$. We want to project \mathbf{u} onto the mortar to obtain $\mathbf{u}^{\Xi} = (u_0^{\Xi}, \dots, u_{N_{\Xi}}^{\Xi})^T$ with

$$\mathcal{U}^{\Xi}(\eta) = \sum_{j=0}^{N_{\Xi}} \ell_j^{\Xi}(\eta) u_j^{\Xi}. \quad (3.14)$$

Note, that $\mathcal{U}(\eta) \neq \mathcal{U}^{\Xi}(\eta)$ for a polynomial of higher degree as $N_L \neq N_{\Xi}$. The operator $\mathbf{P}_{L2\Xi}$ is created by a \mathcal{L}_2 projection onto the mortar

$$\begin{aligned} (\mathcal{U}, \ell_j^{\Xi})_{\mathcal{L}_2(\hat{\Omega})} &= (\mathcal{U}^{\Xi}, \ell_j^{\Xi})_{\mathcal{L}_2(\hat{\Omega})}, \\ \Leftrightarrow \sum_{i=0}^{N_L} (\ell_i^L, \ell_j^{\Xi})_{\mathcal{L}_2(\hat{\Omega})} u_i &= \sum_{i=0}^{N_{\Xi}} (\ell_i^{\Xi}, \ell_j^{\Xi})_{\mathcal{L}_2(\hat{\Omega})} u_i^{\Xi}, \end{aligned} \quad (3.15)$$

for $j = 0, \dots, N_{\Xi}$. In [69] the evaluation of the discrete inner products can be done using exact integration or mass lumping. However, to obtain projection matrices satisfying the M-compatibility condition (3.2), we consider LGL quadrature rules. The \mathcal{L}_2 inner product on the left hand side of (3.15) is evaluated exactly on $N_{\Xi} + 1$ LGL nodes

$$(\ell_i^L, \ell_j^{\Xi})_{\mathcal{L}_2(\hat{\Omega})} = \sum_{k=0}^{N_{\Xi}} \omega_k^{\Xi} \ell_i^L(\eta_k^{\Xi}) \ell_j^{\Xi}(\eta_k^{\Xi}) = \sum_{k=0}^{N_{\Xi}} \omega_k^{\Xi} \ell_i^L(\eta_k^{\Xi}) \delta_{jk} = \omega_j^{\Xi} \ell_i^L(\eta_j^{\Xi}), \quad (3.16)$$

for $i = 0, \dots, N_L, j = 0, \dots, N_{\Xi}$. Note, that this is an exact evaluation of the \mathcal{L}_2 product due to the accuracy of the LGL quadrature (exact for polynomials of degree $2N_{\Xi} - 1$ and $N_{\Xi} > N_L$). The \mathcal{L}_2 product on the right hand side of (3.15) is also evaluated on $N_{\Xi} + 1$ LGL nodes. Therefore, the inner product is approximated by an integration rule with mass lumping, e.g. [9],

$$(\ell_i^{\Xi}, \ell_j^{\Xi})_{\mathcal{L}_2(\hat{\Omega})} \approx \sum_{k=0}^{N_{\Xi}} \omega_k^{\Xi} \ell_i^{\Xi}(\eta_k^{\Xi}) \ell_j^{\Xi}(\eta_k^{\Xi}) = \sum_{k=0}^{N_{\Xi}} \omega_k^{\Xi} \delta_{ik} \delta_{jk} = \delta_{ij} \omega_j^{\Xi}, \quad (3.17)$$

for $i, j = 0, \dots, N_{\Xi}$. Next, we define the rectangular *interpolation* operator

$$[\mathbf{L}_{L2\Xi}]_{ij} := \ell_j^L(\eta_i^{\Xi}), \quad (3.18)$$

with $i = 0, \dots, N_L$ and $j = 0, \dots, N_{\Xi}$. We rewrite (3.15) in a compact matrix-vector notation

$$\mathbf{M}_{\Xi} \mathbf{L}_{L2\Xi} \mathbf{u} = \mathbf{M}_{\Xi} \mathbf{u}^{\Xi} \Leftrightarrow \underbrace{\mathbf{L}_{L2\Xi}}_{:=\mathbf{P}_{L2\Xi}} \mathbf{u} = \mathbf{u}^{\Xi}. \quad (3.19)$$

So the projection operator to move the solution from the element onto the mortar is equivalent to an interpolation operator. However, this does not hold for projecting the solution from the mortar back to the element.

To construct the operator $\mathbf{P}_{\Xi2L}$ we again consider a \mathcal{L}_2 projection. Here, we assume to have a discrete evaluation of the solution on the mortar $\mathbf{u}^{\Xi} = (u_0^{\Xi}, \dots, u_{N_{\Xi}}^{\Xi})^T$ with $\mathcal{U}^{\Xi}(\eta) = \sum_{i=0}^{N_{\Xi}} \ell_i^{\Xi}(\eta) u_i^{\Xi}$ and seek the solution on the element $\mathbf{u} = (u_0, \dots, u_{N_L})^T$ with

$$\mathcal{U}(\eta) = \sum_{i=0}^{N_L} \ell_i^L(\eta) u_i. \quad (3.20)$$

The \mathcal{L}_2 projection back to the element is

$$\begin{aligned} (\mathcal{U}^{\Xi}, \ell_j^L)_{\mathcal{L}_2(\hat{\Omega})} &= (\mathcal{U}, \ell_j^L)_{\mathcal{L}_2(\hat{\Omega})}, \\ \Leftrightarrow \sum_{i=0}^{N_{\Xi}} (\ell_i^{\Xi}, \ell_j^L)_{\mathcal{L}_2(\hat{\Omega})} u_i^{\Xi} &= \sum_{i=0}^{N_L} (\ell_i^L, \ell_j^L)_{\mathcal{L}_2(\hat{\Omega})} u_i, \end{aligned} \quad (3.21)$$

for $j = 0, \dots, N_L$. The \mathcal{L}_2 inner product on the left hand side of (3.21) is computed exactly using $N_{\Xi} + 1$ LGL points. The \mathcal{L}_2 inner product on the right in (3.21) is approximated with mass lumping at $N_L + 1$ LGL nodes. Thus, we obtain

$$(\ell_i^{\Xi}, \ell_j^L)_{\mathcal{L}_2(\hat{\Omega})} = \sum_{k=0}^M \omega_k^{\Xi} \ell_i^{\Xi}(\eta_k^{\Xi}) \ell_j^L(\eta_k^{\Xi}) = \omega_i^{\Xi} \ell_j^L(\eta_i^{\Xi}), \quad (3.22)$$

where $i = 0, \dots, N_{\Xi}$, $j = 0, \dots, N_L$ and

$$(\ell_i^L, \ell_j^L)_{\mathcal{L}_2(\hat{\Omega})} \approx \sum_{k=0}^N \omega_k^L \ell_i^L(\eta_k^L) \ell_j^L(\eta_k^L) = \delta_{ij} \omega_j^L, \quad (3.23)$$

for $i, j = 0, \dots, N_L$. Again, we write (3.21) in a compact matrix-vector notation which gives us

$$\mathbf{L}_{L2\Xi}^T \mathbf{M}_{\Xi} \mathbf{u}^{\Xi} = \mathbf{M}_L \mathbf{u}. \quad (3.24)$$

As $\mathbf{L}_{L2\Xi} = \mathbf{P}_{L2\Xi}$ (3.19) we obtain

$$\underbrace{\mathbf{M}_L^{-1} \mathbf{P}_{L2\Xi}^T \mathbf{M}_{\Xi}}_{:=\mathbf{P}_{\Xi2L}} \mathbf{u}^{\Xi} = \mathbf{u}, \quad (3.25)$$

where we introduce the projection operator (not an interpolation operator) from the mortar back to the element with $N_L + 1$ nodes. With this approach we construct projection operators satisfying the M-compatibility condition (3.2), i.e.,

$$\mathbf{P}_{\Xi 2L} = \mathbf{M}_L^{-1} \mathbf{P}_{L2\Xi}^T \mathbf{M}_\Xi \Leftrightarrow \mathbf{M}_L \mathbf{P}_{\Xi 2L} = \mathbf{P}_{L2\Xi}^T \mathbf{M}_\Xi. \quad (3.26)$$

To summarize, the projection operators for DG elements is constructed as followed

- The projection from the element to the mortar is given by an interpolation using the DG ansatz $\mathbf{P}_{L2\Xi} = \mathbf{L}_{L2\Xi}$.
- The projection back to the element is constructed by $\mathbf{P}_{\Xi 2L} = \mathbf{M}_L^{-1} \mathbf{P}_{L2\Xi}^T \mathbf{M}_\Xi$.

Here, as we adapt the method of Kopriva and set the nodes on the intermediate grid to be LGL nodes. However, in general the choice of the nodes and weights is nearly arbitrary, provided that the degree of \mathbf{M}_Ξ is greater or equal to the degree of \mathbf{M}_L . For example, the nodes on the mortar/intermediate grid can be chosen to be FD or HGTL nodes.

Besides satisfying the M-compatibility condition, we are interested in the accuracy of the projection operators. As we consider $N_L + 1$ LGL nodes, the corresponding differentiation matrix \mathbf{D}_L is of degree $p = N_L$. Focusing on a conforming mesh, the discretization (2.129) for the LAE can solve for polynomials of degree N_L exactly. To extend this result to a non-conforming discretization, we desire projection operators which are also of degree N_L . As the operator $\mathbf{P}_{L2\Xi}$ is constructed by a direct interpolation from LGL nodes to the nodes on the intermediate grid, it is naturally an operator of degree N_L . However, this does not hold for the back projection $\mathbf{P}_{\Xi 2L}$ as shown in Lemma 3.

Lemma 3. *Assume that $\boldsymbol{\eta}^L$ are LGL nodes of size $N_L + 1$. Let $\boldsymbol{\eta}^\Xi$ be an arbitrary set of nodes of size $N_\Xi + 1$ with a norm matrix \mathbf{M}_Ξ of degree $\geq 2N_L - 1$. Then the projection operator $\mathbf{P}_{\Xi 2L}$ is only of degree $N_L - 1$.*

Proof. As $\boldsymbol{\eta}^L$ are LGL nodes of size $N_L + 1$, the corresponding norm matrix \mathbf{M}_L is of degree $2N_L - 1$ [24]. To obtain the degree of the projection operator we first examine $(\ell_i^L, \eta^k)_{\mathcal{L}_2(\hat{\Omega})}$, where $i = 0, \dots, N_L$ and $k = 0, \dots, N_L - 1$. As $\ell_i^L \eta^k$ is a polynomial of degree $\leq 2N_L - 1$, the approximation using $\langle \cdot, \cdot \rangle_{\mathbf{M}_L}$ or $\langle \cdot, \cdot \rangle_{\mathbf{M}_\Xi}$ are

both exact. Therefore,

$$\begin{aligned}
 & \langle \ell_i^L, \eta^k \rangle_{\mathbf{M}_\Xi} = \langle \ell_i^L, \eta^k \rangle_{\mathbf{M}_L}, \\
 \Leftrightarrow & \sum_{j=0}^{N_\Xi} \omega_j^\Xi \ell_i^L(\eta_j^\Xi) (\eta_j^\Xi)^k = \sum_{j=0}^{N_L} \omega_j^L \ell_i^L(\eta_j^L) (\eta_j^L)^k, \\
 \Leftrightarrow & \sum_{j=0}^{N_\Xi} \omega_j^\Xi \ell_i^L(\eta_j^\Xi) (\eta_j^\Xi)^k = \omega_i^L (\eta_i^L)^k.
 \end{aligned} \tag{3.27}$$

Using the definition of the projection operator $\mathbf{P}_{L2\Xi}$ (3.19) we can rewrite (3.27) in a compact matrix vector notation

$$\begin{aligned}
 & \mathbf{P}_{L2\Xi}^T \mathbf{M}_\Xi (\boldsymbol{\eta}^\Xi)^k = \mathbf{M}_L (\boldsymbol{\eta}^L)^k, \\
 \Leftrightarrow & \mathbf{M}_L^{-1} \mathbf{P}_{L2\Xi}^T \mathbf{M}_\Xi (\boldsymbol{\eta}^\Xi)^k = (\boldsymbol{\eta}^L)^k, \\
 \Leftrightarrow & \mathbf{P}_{\Xi2L} (\boldsymbol{\eta}^\Xi)^k = (\boldsymbol{\eta}^L)^k,
 \end{aligned} \tag{3.28}$$

for $k = 0, \dots, N_L - 1$. So the projection operator can only project polynomials of degree $N_L - 1$ exactly. \square

As we see in Lemma 3, the operator can only project polynomials up to a degree $N_L - 1$ exactly. This result is not only found for projection operators for DG operators, but also for projection operators for FD operators as in [70, 83]. Here, all introduced projection operators are of one degree lower than the degree of the differentiation matrix \mathbf{D} . A discussion on this topic can be found in Section 3.2.3

3.2.2. Projection Operators for Finite Difference and Hybrid Gauss-Trapezoidal-Lobatto

In this section we discuss the construction of projection operators for FD and HGTL operators. Again, we focus on an element L and an intermediate grid Ξ and assume that the differentiation matrix \mathbf{D}_L is of degree p . As for the projection operators for DG, we want the projection operators to satisfy

$$\begin{aligned}
 \mathbf{P}_{L2\Xi} (\boldsymbol{\eta}^L)^k &= (\boldsymbol{\eta}^\Xi)^k, \\
 \mathbf{P}_{\Xi2L} (\boldsymbol{\eta}^\Xi)^l &= (\boldsymbol{\eta}^L)^l,
 \end{aligned} \tag{3.29}$$

for $k = 0, \dots, p$ and $l = 0, \dots, p - 1$. As we need to ensure that the projection operators satisfy the \mathbf{M} -compatibility condition (3.2) we seek a projection operator

that satisfies

$$\begin{aligned} \mathbf{P}_{L2\Xi} \left(\boldsymbol{\eta}^L \right)^k &= \left(\boldsymbol{\eta}^\Xi \right)^k, \\ \mathbf{M}_L^{-1} \mathbf{P}_{L2\Xi}^T \mathbf{M}_\Xi \left(\boldsymbol{\eta}^\Xi \right)^l &= \left(\boldsymbol{\eta}^L \right)^l, \end{aligned} \quad (3.30)$$

or alternatively

$$\begin{aligned} \mathbf{P}_{L2\Xi} \left(\boldsymbol{\eta}^L \right)^k &= \left(\boldsymbol{\eta}^\Xi \right)^k, \\ \mathbf{P}_{L2\Xi}^T \mathbf{M}_\Xi \left(\boldsymbol{\eta}^\Xi \right)^l &= \mathbf{M}_L \left(\boldsymbol{\eta}^L \right)^l, \end{aligned} \quad (3.31)$$

for $k = 0, \dots, p$ and $l = 0, \dots, p - 1$.

Remark 9. When solving the system (3.31) including the condition $\mathbf{P}_{L2\Xi}^T \mathbf{M}_\Xi \left(\boldsymbol{\eta}^\Xi \right)^p = \mathbf{M}_L \left(\boldsymbol{\eta}^L \right)^p$ (the back projection should be of degree p), then the system is not solvable for FD and HGTL operators presented in Section 2.1.2 and 2.1.4. This result holds for an arbitrary choice of the intermediate grid. For more details see Section 3.2.3.

Note, that after choosing an intermediate grid with nodes and quadrature weights, the conditions (3.31) only include the unknown projection operator $\mathbf{P}_{L2\Xi}$. We assume that the construction of this operator is a dense matrix, so it introduces $(N_L + 1)(N_\Xi + 1)$ degrees of freedom. In general, the conditions (3.31) are insufficient to fully specify $\mathbf{P}_{L2\Xi}$. Therefore, we need to consider further conditions.

We reiterate, that for FD and HGTL operators the differentiation matrix \mathbf{D}_L is of degree p only at the boundary nodes. On the interior nodes however, the operator has a degree of $2p$ due to a symmetric, central finite difference formula. To establish further conditions, we set the quadrature rule on the intermediate grid to be derived from a HGTL or FD operator. Therefore, the nodes $\boldsymbol{\eta}^\Xi$ also consist of boundary and interior nodes. In order to mimic the accuracy properties of the differentiation matrix, we set the following conditions for the projection operator:

$$\begin{aligned} \mathbf{P}_{L2\Xi} \left(\boldsymbol{\eta}^L \right)^k &= \left(\boldsymbol{\eta}^\Xi \right)^k, & \text{for } k = 0, \dots, p \\ \mathbf{P}_{L2\Xi}^T \mathbf{M}_\Xi \left(\boldsymbol{\eta}^\Xi \right)^l &= \mathbf{M}_L \left(\boldsymbol{\eta}^L \right)^l, & \text{for } l = 0, \dots, p - 1, \\ \left(\mathbf{P}_{L2\Xi} \left(\boldsymbol{\eta}^L \right)^{\tilde{k}} \right)_i &= \left(\left(\boldsymbol{\eta}^\Xi \right)^{\tilde{k}} \right)_i, & \text{for } \tilde{k} = 0, \dots, 2p, \\ \left(\mathbf{P}_{L2\Xi}^T \mathbf{M}_\Xi \left(\boldsymbol{\eta}^\Xi \right)^{\tilde{l}} \right)_i &= \left(\mathbf{M}_L \left(\boldsymbol{\eta}^L \right)^{\tilde{l}} \right)_i, & \text{for } \tilde{l} = 0, \dots, 2p, \end{aligned} \quad (3.32)$$

and $i = bn, \dots, N + 2 - bn$. Still, (3.32) does not necessarily fully specify the projection matrix $\mathbf{P}_{L2\Xi}$, for which we next use optimization constraints. Therefore, we use any remaining degrees of freedom to ensure that the modulus of each eigenvalue of the matrix

$$\mathbf{Z}_{opt} := \mathbf{M}_L - \mathbf{P}_{L2\Xi}^T \mathbf{M}_\Xi \mathbf{P}_{L2\Xi} \quad (3.33)$$

is as close to zero as possible [39]. We chose the optimization of minimizing the eigenvalues of \mathbf{Z}_{opt} as

$$\begin{aligned} \mathbf{Z}_{opt} &= \mathbf{M}_L - \mathbf{P}_{L2\Xi}^T \mathbf{M}_\Xi \mathbf{P}_{L2\Xi} = \mathbf{M}_L \left(\mathbf{I}_L - \mathbf{M}_L^{-1} \mathbf{P}_{L2\Xi}^T \mathbf{M}_\Xi \mathbf{P}_{L2\Xi} \right), \\ &= \mathbf{M}_L \left(\mathbf{I}_L - \mathbf{P}_{\Xi 2L} \mathbf{P}_{L2\Xi} \right). \end{aligned} \quad (3.34)$$

So, when $\mathbf{Z}_{opt} \approx 0$, then $\mathbf{I}_L \approx \mathbf{P}_{\Xi 2L} \mathbf{P}_{L2\Xi}$ which indicates a minimization of the projection error. Note, that optimizing the projection operators such that $\mathbf{I}_L \approx \mathbf{P}_{\Xi 2L} \mathbf{P}_{L2\Xi}$ is the same optimization criteria as in the work of Kozdon and Wilcox [70].

3.2.3. Analysis of the Suboptimal Degree of the Projection Operators

In Section 3.2.1 and 3.2.2 it was not possible to construct projection operators of degree p which at the same time satisfy the M-compatibility condition (3.2). Up to now, all constructed SBP operators had norm matrices of degree $2p - 1$ [23, 58]. A drawback of such operators is that it is not possible to construct $\mathbf{P}_{L2\Xi}$ and $\mathbf{P}_{\Xi 2L}$ of degree p , as proven in the following theorem, where we adapt the ideas of Lundquist and Nordström [80].

Theorem 4. *Given a degree p SBP operator \mathbf{D}_L with norm matrix \mathbf{M}_L of degree $2p - 1$. Assuming an intermediate grid with a norm matrix \mathbf{M}_Ξ of degree $\geq 2p - 1$ such that the matrices \mathbf{M}_L and \mathbf{M}_Ξ are different in terms of the norm matrices having different errors, then it is not possible to construct M-compatible projection operators $\mathbf{P}_{L2\Xi}$ and $\mathbf{P}_{\Xi 2L}$ that are both of degree p .*

Proof. We assume that it is possible to construct M-compatible operators of degree p and seek a contradiction. Therefore, we have the degree p projection operator $\mathbf{P}_{L2\Xi}$ and $\mathbf{P}_{\Xi 2L}$ which satisfy

$$\begin{aligned} \mathbf{P}_{L2\Xi} \left(\boldsymbol{\eta}^L \right)^k &= \left(\boldsymbol{\eta}^\Xi \right)^k, \\ \mathbf{P}_{\Xi 2L} \left(\boldsymbol{\eta}^\Xi \right)^k &= \left(\boldsymbol{\eta}^L \right)^k, \\ \mathbf{P}_{\Xi 2L}^T \mathbf{M}_L &= \mathbf{M}_\Xi \mathbf{P}_{L2\Xi}, \end{aligned} \quad (3.35)$$

for $k = 0, \dots, p$. Focusing on the third equation and multiplying it from left by $\left((\boldsymbol{\eta}^\Xi)^p\right)^T$ and from the right by $\left(\boldsymbol{\eta}^L\right)^p$ we get

$$\begin{aligned} \left((\boldsymbol{\eta}^\Xi)^p\right)^T \mathbf{P}_{\Xi 2L}^T \mathbf{M}_L \left(\boldsymbol{\eta}^L\right)^p &= \left((\boldsymbol{\eta}^\Xi)^p\right)^T \mathbf{M}_\Xi \mathbf{P}_{L 2\Xi} \left(\boldsymbol{\eta}^L\right)^p, \\ \left(\mathbf{P}_{\Xi 2L} \left(\boldsymbol{\eta}^\Xi\right)^p\right)^T \mathbf{M}_L \left(\boldsymbol{\eta}^L\right)^p &= \left((\boldsymbol{\eta}^\Xi)^p\right)^T \mathbf{M}_\Xi \mathbf{P}_{L 2\Xi} \left(\boldsymbol{\eta}^L\right)^p. \end{aligned} \quad (3.36)$$

Due to the assumptions (3.35) the projection operators are of degree p and thus

$$\left(\left(\boldsymbol{\eta}^L\right)^p\right)^T \mathbf{M}_L \left(\boldsymbol{\eta}^L\right)^p = \left((\boldsymbol{\eta}^\Xi)^p\right)^T \mathbf{M}_\Xi \left(\boldsymbol{\eta}^\Xi\right)^p. \quad (3.37)$$

However, the norm matrix \mathbf{M}_L cannot integrate η^{2p} exactly. Also, if \mathbf{M}_Ξ is of degree $2p - 1$, then the right-hand side of (3.37) also does not integrate exactly. The error terms for each quadrature rule are different, so the equality (3.37) cannot hold in general. For the case where the degree of \mathbf{M}_Ξ is larger than $2p - 1$, then the equality (3.37) cannot hold since the right-hand side performs exact integration of η^{2p} , whereas the right hand side produces a quadrature error. \square

The contradiction within Theorem 4 holds due to the accuracy of the norm matrix \mathbf{M}_L . This contradiction would not occur, if \mathbf{M}_L would be of degree $2p$ or higher. Therefore, we construct SBP operators with norm matrices of degree $\geq 2p$ for which projection operators of degree p satisfying the \mathbf{M} -compatibility condition can be built. We denote such SBP operators as *degree preserving* as in [39]. The construction of a SBP degree preserving, element based finite difference operator is given in Section 3.3

Remark 10. The existence of a quadrature rule with positive weights of degree $\geq 2p$ is sufficient for the existence of a degree preserving SBP operator of degree p [23].

3.3. Degree Preserving Operators

As shown in Section 3.2.3 it is not possible to construct \mathbf{M} -compatible projection operators of degree p if the norm matrix of the SBP operator is only of degree $2p - 1$. Here, p refers to the degree of the differentiation matrix of the SBP operator. In this section we explicitly construct degree preserving SBP operators (\mathbf{M}, \mathbf{D}) , where the degree of \mathbf{D} is p and the degree of the norm matrix \mathbf{M} is $\geq 2p$. As for all other SBP operators in this work, we construct the SBP degree preserving operator (DP operator) in a reference space $\hat{\Omega} := [-1, 1]$. We construct the DP operator in a similar fashion to the FD operators and consider an interior stencil of

degree $2p$ and boundary nodes with free parameters. Since the norm matrix needs to be at least one degree higher than for the FD operators in Section 2.1.2, we naturally need more free coefficients. These coefficients are obtained by increasing the number of boundary nodes at least by one. For example, for $p = 2$ the norm matrix \mathbf{M} is

$$\mathbf{M} := \frac{2}{N} \text{diag}(h_0, \dots, h_4, 1, \dots, 1, h_4, \dots, h_0), \quad (3.38)$$

where $N + 1$ denotes the number of nodes. The upper left corner of \mathbf{Q} has the following structure:

$$(\mathbf{Q})_{(1:5,1:7)} = \begin{bmatrix} -\frac{1}{2} & q_{12} & q_{13} & q_{14} & q_{15} & 0 & 0 \\ -q_{12} & 0 & q_{23} & q_{24} & q_{25} & 0 & 0 \\ -q_{13} & -q_{23} & 0 & q_{34} & q_{35} & 0 & 0 \\ -q_{14} & -q_{24} & -q_{34} & 0 & q_{45} & -\frac{1}{12} & 0 \\ -q_{15} & -q_{25} & -q_{35} & -q_{45} & 0 & \frac{2}{3} & -\frac{1}{12} \end{bmatrix}.$$

With these configurations we will focus on the construction of the DP operators. As for FD operators the degrees of freedom of \mathbf{Q} referring to the interior nodes are the same. Here, we again consider a symmetric central difference formula. Note, that the DP operator is element based as in [28]. By changing the number of nodes, we must re-calculate the degrees of freedom at the boundary blocks. This is similar to a DG operator, as the operator coefficients change by increasing or decreasing the number of nodes.

Let $\boldsymbol{\xi}$ be an uniformly distributed set of nodes within the reference space $[-1, +1]$ and $\boldsymbol{\xi}_0 = -1$, $\boldsymbol{\xi}_N = +1$. The free coefficients are determined by solving

$$\begin{aligned} \mathbf{Q}\boldsymbol{\xi}^k &= k\mathbf{M}\boldsymbol{\xi}^{k-1} & k = 0, \dots, p, \\ \mathbf{1}^T \mathbf{M}\boldsymbol{\xi}^k &= \frac{\zeta_N^{k+1} - \zeta_0^{k+1}}{k+1} & k = 0, \dots, 2p+1. \end{aligned} \quad (3.39)$$

In case of negative weights for a fixed value N , we increase the number of boundary nodes bn to obtain more free coefficients for satisfying the property of a diagonal norm matrix with positive weights.

By solving (3.39), the DP operator is not necessarily fully specified. As in Section 2.1.2 we construct the DP operators using the same constraints and optimization criteria. A description of the construction is provided in Algorithm 3.

Input: N and p
Set $bn := 2p + 1$;
Solve (3.39);
while *one or more entries of M are negative* **do**
 Set $bn := bn + 1$;
 if $2bn > N + 1$ **then**
 | Error: Chose a different N ;
 else
 | Solve (3.39) again;
 end
end
Minimize Z_e in (2.25) under the constraint $h_i > 0$ for $i = 0, \dots, bn - 1$;
Minimize Z_Q in (2.26) under the constraint $h_i > 0$ for $i = 0, \dots, bn - 1$;

Algorithm 3: Construction of degree preserving operators.

By following these steps we create SBP operators (M, D) where D is of degree p and M is of degree $2p + 1$. Note, that as for existing SBP operators like those in Section 2.1.2, a minimum number of nodes needs to be considered to solve for (3.39). Creating the corresponding projection operators for the DP operators is done in the same way as for FD and HGTL in Section 3.2.2.

3.4. Non-Conforming Discrete Analysis for the Linear Advection Equation

In this section we discuss the construction of the non-conforming SBP scheme and prove primary conservation for energy stability for the LAE. Again, we consider discretization (2.129). As the elements are no longer conforming, we need to choose an appropriate numerical surface flux/SAT. Here, we need to differ between elements with conforming (p refinement) and non-conforming interfaces (h refinement). We will first describe how to determine a fully conservative scheme for conforming interfaces, but differing nodal distribution. Then, we extend these results to consider meshes with non-conforming interfaces (hanging corners). Finally, we determine an energy stable scheme by including interface dissipation.

As discussed in the Section 3.1, it would be efficient to store the projection operators within the elements and to choose the SAT as in (3.11). For this ansatz, we need to calculate the numerical surface flux on the intermediate grid. However, when moving onto general non-linear problems in later sections, it is (up to now) not possible to evaluate the numerical surface flux on an intermediate grid in

a stable manner. The reason for this is that when determining the numerical flux of the projected solution $\mathbf{P}_{L2\Xi}u^L$ and $\mathbf{P}_{R2\Xi}u^R$, then non-linear projections get introduced, e.g. $(\mathbf{P}_{L2\Xi}u^L)^2$. A discussion on this issue is provided in Section 5. Therefore, within the mathematical derivations we neglect the intermediate grid and store the projection operators $\mathbf{P}_{L2R}, \mathbf{P}_{R2L}$ and the numerical surface fluxes $\tilde{f}^{*,L}, \tilde{f}^{*,R}$ for the left and right element on the interfaces. The derivations will be consistent with the stability and conservation proofs for non-linear problems in Section 5. In addition, the derivations in this section are more compact as we do not consider the projection on an intermediate grid. Still, an intermediate grid is an useful analytical tool and has been used throughout this work to construct projection operators.

3.4.1. Non-Conforming Nodal Distributions

In this section we show how to create a fully conservative scheme on a mesh with conforming interfaces, but non-conforming nodal distributions (p refinement). As shown in Lemma 2 the primary quantity and energy growth is dependent on the numerical surface fluxes at an interface. Note, that Lemma 2 refers to a single element and thus holds for conforming and non-conforming meshes. As such, we elect to consider all terms related to a single shared interface of a left and right element.

The numerical approximation is primary and energy conservative provided

$$IU_t := \langle \mathbf{1}^R, \tilde{\mathbf{f}}^{*,R} \rangle_{\mathbf{M}_R} - \langle \mathbf{1}^L, \tilde{\mathbf{f}}^{*,L} \rangle_{\mathbf{M}_L}, \quad (3.40)$$

$$\begin{aligned} IE_t := & \langle \mathbf{u}^R, \tilde{\mathbf{f}}^{*,R} \rangle_{\mathbf{M}_R} - \frac{\tilde{a}}{2} \langle \mathbf{1}^R, (\mathbf{u}^R)^2 \rangle_{\mathbf{M}_R} \\ & - \langle \mathbf{u}^L, \tilde{\mathbf{f}}^{*,L} \rangle_{\mathbf{M}_L} + \frac{\tilde{a}}{2} \langle \mathbf{1}^L, (\mathbf{u}^L)^2 \rangle_{\mathbf{M}_L}, \end{aligned} \quad (3.41)$$

are both zero. Here, (3.40) and (3.41) are adapted from Section 2.2 (Equation (2.144) and (2.145)) and rewritten in a compact matrix vector form where

$$\begin{aligned} \tilde{\mathbf{f}}^{*,L} &= [\tilde{f}_{N_L 0}^{*,L}, \dots, \tilde{f}_{N_L N_L}^{*,L}]^T, \\ \tilde{\mathbf{f}}^{*,R} &= [\tilde{f}_{0 0}^{*,R}, \dots, \tilde{f}_{0 N_R}^{*,R}]^T, \end{aligned} \quad (3.42)$$

and $\mathbf{1}^L$ and $\mathbf{1}^R$ describe vectors of size $N_L + 1$ and $N_R + 1$. Within (3.40) and (3.41), the subtractions involve two discrete inner products with differing nodal distributions between the left and right elements. Therefore, we require projection

operators that move information from the left node distribution to the right and vice versa. The construction of these operators for FD, DG, HGTL and operators can be found in Section 3.2 and 3.3.

Now, we address the issue of p refinement where the mesh may contain differing nodal distributions. To do so, we consider the energy conservative fluxes in a modified way. Therefore, we introduce the projection operators within the numerical flux

$$\begin{aligned}\tilde{\mathbf{f}}_{EC}^{*,L}(\mathbf{u}^R, \mathbf{u}^L) &= \frac{1}{2} \left(\tilde{\mathbf{f}}(\mathbf{P}_{R2L}\mathbf{u}^R) + \tilde{\mathbf{f}}(\mathbf{u}^L) \right), \\ \tilde{\mathbf{f}}_{EC}^{*,R}(\mathbf{u}^R, \mathbf{u}^L) &= \frac{1}{2} \left(\tilde{\mathbf{f}}(\mathbf{u}^R) + \tilde{\mathbf{f}}(\mathbf{P}_{L2R}\mathbf{u}^L) \right).\end{aligned}\tag{3.43}$$

We will prove that the scheme is primary and energy conservative, if the projection operators satisfy the M-compatibility condition (3.2).

Theorem 5. *Considering the fluxes within (3.43). The scheme is primary and energy conservative, meaning that (3.40) and (3.41) are zero, if*

$$\mathbf{P}_{R2L}^T \mathbf{M}_L = \mathbf{M}_R \mathbf{P}_{L2R}.\tag{3.44}$$

Proof. First, we focus on (3.40) and include the fluxes (3.43). We get

$$\begin{aligned}IU_t &= \frac{\tilde{a}}{2} \langle \mathbf{1}^R, \mathbf{u}^R \rangle_{\mathbf{M}_R} + \frac{\tilde{a}}{2} \langle \mathbf{1}^R, \mathbf{P}_{L2R}\mathbf{u}^L \rangle_{\mathbf{M}_R} \\ &\quad - \frac{\tilde{a}}{2} \langle \mathbf{1}^L, \mathbf{P}_{R2L}\mathbf{u}^R \rangle_{\mathbf{M}_L} - \frac{\tilde{a}}{2} \langle \mathbf{1}^L, \mathbf{u}^L \rangle_{\mathbf{M}_L}.\end{aligned}\tag{3.45}$$

Writing out the inner product we have

$$\begin{aligned}IU_t &= \frac{\tilde{a}}{2} \mathbf{1}^{R,T} \mathbf{M}_R \mathbf{u}^R + \frac{\tilde{a}}{2} \mathbf{1}^{R,T} \mathbf{M}_R \mathbf{P}_{L2R} \mathbf{u}^L \\ &\quad - \frac{\tilde{a}}{2} \mathbf{1}^{L,T} \mathbf{M}_L \mathbf{P}_{R2L} \mathbf{u}^R - \frac{\tilde{a}}{2} \mathbf{1}^{L,T} \mathbf{M}_L \mathbf{u}^L.\end{aligned}\tag{3.46}$$

Assuming that the projection operators can project a constant exactly, meaning $\mathbf{P}_{L2R}\mathbf{1}^L = \mathbf{1}^R$ and $\mathbf{P}_{R2L}\mathbf{1}^R = \mathbf{1}^L$ we get

$$\begin{aligned}IU_t &= \frac{\tilde{a}}{2} \left(\mathbf{1}^{L,T} \mathbf{P}_{L2R}^T \mathbf{M}_R \mathbf{u}^R + \mathbf{1}^{R,T} \mathbf{M}_R \mathbf{P}_{L2R} \mathbf{u}^L - \mathbf{1}^{L,T} \mathbf{M}_L \mathbf{P}_{R2L} \mathbf{u}^R - \mathbf{1}^{R,T} \mathbf{P}_{R2L}^T \mathbf{M}_L \mathbf{u}^L \right), \\ &= \frac{\tilde{a}}{2} \left(\mathbf{1}^{L,T} \underbrace{\left(\mathbf{P}_{L2R}^T \mathbf{M}_R - \mathbf{M}_L \mathbf{P}_{R2L} \right)}_{=0} \mathbf{u}^R + \mathbf{1}^{R,T} \underbrace{\left(\mathbf{M}_R \mathbf{P}_{L2R} - \mathbf{P}_{R2L}^T \mathbf{M}_L \right)}_{=0} \mathbf{u}^L \right), \\ &= 0,\end{aligned}\tag{3.47}$$

due to the M-compatibility equation (3.2).

Secondly, we focus on (3.41) to prove energy conservation. Including the fluxes (3.43) we get

$$\begin{aligned} IE_t &= \frac{\tilde{a}}{2} \langle \mathbf{u}^R, \mathbf{P}_{L2R} \mathbf{u}^L \rangle_{M_R} - \frac{\tilde{a}}{2} \langle \mathbf{u}^L, \mathbf{P}_{R2L} \mathbf{u}^R \rangle_{M_L}, \\ &= \frac{\tilde{a}}{2} (\mathbf{u}^R)^T M_R \mathbf{P}_{L2R} \mathbf{u}^L - \frac{\tilde{a}}{2} (\mathbf{u}^L)^T M_L \mathbf{P}_{R2L} \mathbf{u}^R. \end{aligned} \quad (3.48)$$

Rearranging terms gives us

$$\begin{aligned} IE_t &= \frac{\tilde{a}}{2} (\mathbf{u}^R)^T \underbrace{(M_R \mathbf{P}_{L2R} - \mathbf{P}_{R2L}^T M_L)}_{=0} \mathbf{u}^L, \\ &= 0, \end{aligned} \quad (3.49)$$

again due to the M-compatibility equation (3.2). This indicates that the scheme is primary and energy conservative when considering the energy conservative fluxes (3.43). \square

Based on this proof, we can construct energy conservative schemes on non-conforming meshes with conforming interfaces (p refinement). To introduce additional flexibility, we next consider elements where the interfaces may not coincide (h refinement).

3.4.2. Non-Conforming Interfaces with Hanging Corners

In Section 3.4.1 we discussed the coupling of elements with a coinciding interface but differing nodal distributions. As such, each numerical interface flux only depends on one neighboring element. For example the numerical surface flux $\tilde{\mathbf{f}}_{EC}^{*,L}$ in (3.43) only contained the projection operator \mathbf{P}_{R2L} , so it only depends on one neighboring element R . This is acceptable if the interfaces have no hanging corners, however for the more general case of h refinement as in Figure 3.9 the interface coupling requires addressing contributions from many elements. Throughout this section we focus on discrete meshes as in Figure 3.9. For the h refinement discretization we adapt the results derived in the previous section for conforming interfaces. Therefore, we consider all left elements as if they are *one* large element $L = \bigcup_{i=1}^{NE} L_i$, where NE denotes the number of left neighbors of the right element R . Thus, we see that L has a conforming interface with element R (red line). To obtain the nodes of L in the reference space we first map the nodes of L_i from

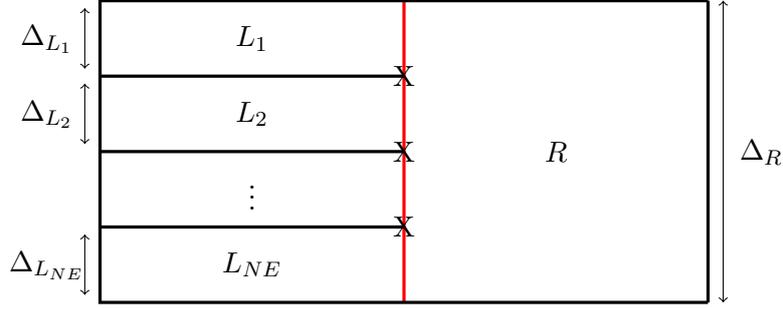


Figure 3.9.: h refinement with hanging corners X . The figure is reproduced from [38].

$[-1, +1]$ to $\left[2 \sum_{j=0}^{i-1} \frac{\Delta L_j}{\Delta R} - 1, 2 \sum_{j=0}^i \frac{\Delta L_j}{\Delta R} - 1\right]$ by

$$\tilde{\eta}_k^{L_i} = \frac{\Delta L_i}{\Delta R} (\eta_k^{L_i} + 1) + 2 \sum_{j=0}^{i-1} \frac{\Delta L_j}{\Delta R} - 1, \quad (3.50)$$

for $k = 0, \dots, N_{L_i}$ and $i = 1, \dots, N_{L_{NE}}$. Here, η^{L_i} denotes the vertical nodes of the element L_i and Δ denotes its height. The nodes of L on the red line in Figure 3.9 are defined by

$$\boldsymbol{\eta}^L := (\tilde{\eta}_0^{L_1}, \dots, \tilde{\eta}_{N_{L_1}}^{L_1}, \dots, \tilde{\eta}_0^{L_{NE}}, \dots, \tilde{\eta}_{N_{L_{NE}}}^{L_{NE}})^T. \quad (3.51)$$

Due to (3.50) all coefficients of $\boldsymbol{\eta}^L$ are in the reference space $[-1, +1]$. The corresponding norm matrix of L is given by

$$\mathbf{M}_L := \frac{1}{\Delta R} \begin{pmatrix} \Delta_{L_1} \mathbf{M}_{L_1} & & 0 \\ & \ddots & \\ 0 & & \Delta_{L_{NE}} \mathbf{M}_{L_{NE}} \end{pmatrix}. \quad (3.52)$$

With the nodes $\boldsymbol{\eta}^L$ and the norm matrix \mathbf{M}_L we can construct projection operators $\mathbf{P}_{L2R}, \mathbf{P}_{R2L}$ which satisfy the \mathbf{M} -compatibility condition (3.2):

$$\mathbf{P}_{L2R}^T \mathbf{M}_R = \mathbf{M}_L \mathbf{P}_{R2L}. \quad (3.53)$$

The "large" projection operators can be interpreted into parts that contribute from/to each of the left elements with the following block structure

$$\mathbf{P}_{L2R} = \begin{bmatrix} \mathbf{P}_{L_1 2R} & \dots & \mathbf{P}_{L_{NE} 2R} \end{bmatrix}, \quad (3.54)$$

and

$$\mathbf{P}_{R2L} = \begin{bmatrix} \mathbf{P}_{R2L_1} \\ \vdots \\ \mathbf{P}_{R2L_{NE}} \end{bmatrix}. \quad (3.55)$$

With this new notation we get a modified M-compatibility condition

$$\Delta_{L_i} \mathbf{P}_{R2L_i}^T \mathbf{M}_{L_i} = \Delta_R \mathbf{M}_R \mathbf{P}_{L_i 2R}, \quad i = 1, \dots, NE. \quad (3.56)$$

When constructing \mathbf{P}_{R2L} and \mathbf{P}_{L2R} both operators are of a certain degree, say p_{R2L} and p_{L2R} (both ≥ 0), meaning

$$\begin{aligned} \mathbf{P}_{R2L} (\boldsymbol{\eta}^R)^k &= (\boldsymbol{\eta}^L)^k, & \text{for } k = 0, \dots, p_{R2L}, \\ \mathbf{P}_{L2R} (\boldsymbol{\eta}^L)^k &= (\boldsymbol{\eta}^R)^k, & \text{for } k = 0, \dots, p_{L2R}. \end{aligned} \quad (3.57)$$

Due to (3.54) and (3.55) the subdomain operators have the following properties

$$\begin{aligned} \mathbf{P}_{R2L_i} (\boldsymbol{\eta}^R)^k &= (\tilde{\boldsymbol{\eta}}^{L_i})^k, & \text{for } i = 1, \dots, NE \text{ and } k = 0, \dots, p_{R2L}, \\ \sum_{i=1}^{N_L} \mathbf{P}_{L_i 2R} (\tilde{\boldsymbol{\eta}}^{L_i})^k &= (\boldsymbol{\eta}^R)^k, & \text{for } k = 0, \dots, p_{L2R}. \end{aligned} \quad (3.58)$$

Considering the special case $k = 0$ we get

$$\begin{aligned} \mathbf{P}_{R2L_i} \mathbf{1}^R &= \mathbf{1}^{L_i}, & \text{for } i = 1, \dots, NE, \\ \sum_{i=1}^{N_L} \mathbf{P}_{L_i 2R} \mathbf{1}^{L_i} &= \mathbf{1}^R, \end{aligned} \quad (3.59)$$

which describes the projection of a constant from one element into another. So the operators \mathbf{P}_{R2L_i} with $i = 1, \dots, NE$ are of degree p_{R2L} and therefore can project a constant exactly. In comparison, the operators $\mathbf{P}_{L_i 2R}$ for $i = 1, \dots, NE$ cannot necessarily project a constant exactly.

As in Section 3.4.1 we need to choose a numerical surface fluxes so that the scheme is primary and energy conservative. We note that for the h non-conforming case (just like p non-conforming) the result of Lemma 2 is still valid. Therefore, only a non-conforming numerical surface flux is needed to construct a fully conservative SBP method. Therefore, we analyze all terms which are related to the interface connecting L_1, \dots, L_{NE} and R . Similar to (3.40) and (3.41), we arrive at the following terms

$$IU_t = \langle \mathbf{1}^R, \tilde{\mathbf{f}}^{*,R} \rangle_{\mathbf{M}_R} - \sum_{i=1}^{NE} \langle \mathbf{1}^{L_i}, \tilde{\mathbf{f}}^{*,L_i} \rangle_{\mathbf{M}_{L_i}}, \quad (3.60)$$

$$IE_t = \langle \mathbf{u}^R, \tilde{\mathbf{f}}^{*,R} \rangle_{\mathbf{M}_R} - \frac{\tilde{a}_R}{2} \langle \mathbf{1}^R, (\mathbf{u}^R)^2 \rangle_{\mathbf{M}_R} - \sum_{i=1}^{NE} \left(\langle \mathbf{u}^{L_i}, \tilde{\mathbf{f}}^{*,L_i} \rangle_{\mathbf{M}_{L_i}} - \frac{\tilde{a}_{L_i}}{2} \langle \mathbf{1}^{L_i}, (\mathbf{u}^{L_i})^2 \rangle_{\mathbf{M}_{L_i}} \right), \quad (3.61)$$

which need to be zero to obtain a discretely fully conservative scheme. Note, that due to the different heights of the subdomains L_i we get different metric terms, such that $\tilde{a}_{L_i} \neq \tilde{a}_R$.

Theorem 6. *Given a set of projection operators that satisfy (3.56), then the numerical surface fluxes*

$$\tilde{\mathbf{f}}_{EC}^{*,R} = \frac{\tilde{a}_R}{2} \left(\mathbf{u}^R + \sum_{i=1}^{NE} \mathbf{P}_{L_i 2R} \mathbf{u}^{L_i} \right), \quad (3.62)$$

and

$$\tilde{\mathbf{f}}_{EC}^{*,L_i} = \frac{\tilde{a}_{L_i}}{2} \left(\mathbf{P}_{R 2L_i} \mathbf{u}^R + \mathbf{u}^{L_i} \right), \quad (3.63)$$

for $i = 1, \dots, NE$ lead to a fully conservative scheme.

Proof. The proof is done analogously to Theorem 5. First, we prove primary conservation. Inserting the numerical surface flux gives us

$$\begin{aligned} IU_t &= \left\langle \mathbf{1}^R, \frac{\tilde{a}_R}{2} \left(\mathbf{u}^R + \sum_{i=1}^{NE} \mathbf{P}_{L_i 2R} \mathbf{u}^{L_i} \right) \right\rangle_{\mathbf{M}_R} \\ &\quad - \sum_{i=1}^{NE} \left\langle \mathbf{1}^{L_i}, \frac{\tilde{a}_{L_i}}{2} \left(\mathbf{P}_{R 2L_i} \mathbf{u}^R + \mathbf{u}^{L_i} \right) \right\rangle_{\mathbf{M}_{L_i}}, \\ &= \frac{\tilde{a}_R}{2} \mathbf{1}^{R,T} \mathbf{M}_R \mathbf{u}^R - \sum_{i=1}^{NE} \frac{\tilde{a}_{L_i}}{2} \mathbf{1}^{L_i,T} \mathbf{M}_{L_i} \mathbf{P}_{R 2L_i} \mathbf{u}^R \\ &\quad + \sum_{i=1}^{NE} \frac{\tilde{a}_R}{2} \mathbf{1}^{R,T} \mathbf{M}_R \mathbf{P}_{L_i 2R} \mathbf{u}^{L_i} - \sum_{i=1}^{NE} \frac{\tilde{a}_{L_i}}{2} \mathbf{1}^{L_i,T} \mathbf{M}_{L_i} \mathbf{u}^{L_i}. \end{aligned} \quad (3.64)$$

Note, that on a Cartesian mesh $\tilde{a}_R = a \frac{\Delta_R}{2}$ and $\tilde{a}_{L_i} = a \frac{\Delta_{L_i}}{2}$ for all $i = 1, \dots, NE$. Therefore, using the modified M-compatibility condition (3.56) on the second and

third summand we get

$$\begin{aligned}
 IU_t &= \frac{\tilde{a}_R}{2} \mathbf{1}^{R,T} \mathbf{M}_R \mathbf{u}^R - \frac{\tilde{a}_R}{2} \left(\sum_{i=1}^{NE} \mathbf{P}_{L_i 2R} \mathbf{1}^{L_i} \right)^T \mathbf{M}_R \mathbf{u}^R \\
 &\quad + \sum_{i=1}^{NE} \frac{\tilde{a}_{L_i}}{2} \left(\mathbf{P}_{R 2L_i} \mathbf{1}^R \right)^T \mathbf{M}_{L_i} \mathbf{u}^{L_i} - \sum_{i=1}^{NE} \frac{\tilde{a}_{L_i}}{2} \mathbf{1}^{L_i,T} \mathbf{M}_{L_i} \mathbf{u}^{L_i}.
 \end{aligned} \tag{3.65}$$

Due to the accuracy conditions of the projection operators (3.59) we arrive at

$$\begin{aligned}
 IU_t &= \frac{\tilde{a}_R}{2} \mathbf{1}^{R,T} \mathbf{M}_R \mathbf{u}^R - \frac{\tilde{a}_R}{2} \mathbf{1}^{R,T} \mathbf{M}_R \mathbf{u}^R \\
 &\quad + \sum_{i=1}^{NE} \frac{\tilde{a}_{L_i}}{2} \mathbf{1}^{L_i,T} \mathbf{M}_{L_i} \mathbf{u}^{L_i} - \sum_{i=1}^{NE} \frac{\tilde{a}_{L_i}}{2} \mathbf{1}^{L_i,T} \mathbf{M}_{L_i} \mathbf{u}^{L_i}, \\
 &= 0,
 \end{aligned} \tag{3.66}$$

which proves primary conservation. To obtain energy conservation we look at (3.61) and insert the numerical surface fluxes

$$\begin{aligned}
 IE_t &= \frac{\tilde{a}_R}{2} \left\langle \mathbf{u}^R, \mathbf{u}^R + \sum_{i=1}^{NE} \mathbf{P}_{L_i 2R} \mathbf{u}^{L_i} \right\rangle_{\mathbf{M}_R} - \frac{\tilde{a}_R}{2} \left\langle \mathbf{1}^R, (\mathbf{u}^R)^2 \right\rangle_{\mathbf{M}_R} \\
 &\quad - \sum_{i=1}^{NE} \frac{\tilde{a}_{L_i}}{2} \left(\left\langle \mathbf{u}^{L_i}, \mathbf{P}_{R 2L_i} \mathbf{u}^R + \mathbf{u}^{L_i} \right\rangle_{\mathbf{M}_{L_i}} - \left\langle \mathbf{1}^{L_i}, (\mathbf{u}^{L_i})^2 \right\rangle_{\mathbf{M}_{L_i}} \right), \\
 &= \sum_{i=1}^{NE} \frac{\tilde{a}_R}{2} (\mathbf{u}^R)^T \mathbf{M}_R \mathbf{P}_{L_i 2R} \mathbf{u}^{L_i} - \sum_{i=1}^{NE} \frac{\tilde{a}_{L_i}}{2} (\mathbf{u}^{L_i})^T \mathbf{M}_{L_i} \mathbf{P}_{R 2L_i} \mathbf{u}^R.
 \end{aligned} \tag{3.67}$$

As the norm matrix is diagonal we know that $\langle \mathbf{u}^R, \mathbf{u}^R \rangle_{\mathbf{M}_R} = \langle \mathbf{1}^R, (\mathbf{u}^R)^2 \rangle_{\mathbf{M}_R}$. Finally, due to the modified M-compatibility condition (3.56) we get

$$IE_t = 0, \tag{3.68}$$

which indicates entropy conservation. \square

Based on this proof, we can construct fully conservative schemes for meshes with non-conforming interfaces.

3.4.3. Including Dissipation within the Numerical Surface Flux

In Sections 3.4.1 and 3.4.2 we derived fully conservative schemes for the LAE on non-conforming meshes with h/p refinement. From these results, we can include

interface dissipation to obtain an energy stable discretization for arbitrary non-conforming rectangular meshes. Now we focus on the general case, where we have differing nodal distributions as well as hanging corners as in Figure 3.9. As in Section 3.4.2 we assume that the projection operators satisfy the modified M-compatibility condition (3.56).

Theorem 7. *The non-conforming SBP scheme is primary conservative and energy stable, for the following numerical surface fluxes*

$$\tilde{\mathbf{f}}_{ES}^{*,L_i} = \tilde{\mathbf{f}}_{EC}^{*,L_i} - \frac{|\tilde{a}_{L_i}|}{2} \left(\mathbf{P}_{R2L_i} \mathbf{u}^R - \mathbf{u}^{L_i} \right), \quad (3.69)$$

$$\tilde{\mathbf{f}}_{ES}^{*,R} = \tilde{\mathbf{f}}_{EC}^{*,R} - \frac{|\tilde{a}_R|}{2} \sum_{i=1}^{NE} \mathbf{P}_{L_i2R} \left(\mathbf{P}_{R2L_i} \mathbf{u}^R - \mathbf{u}^{L_i} \right). \quad (3.70)$$

Proof. By including dissipation we can prove primary conservation by substituting the new fluxes (3.69) and (3.70) into (3.60)

$$IU_t = \left\langle \mathbf{1}^R, \tilde{\mathbf{f}}_{ES}^{*,R} \right\rangle_{\mathbf{M}_R} - \sum_{i=1}^{NE} \left\langle \mathbf{1}^{L_i}, \tilde{\mathbf{f}}_{ES}^{*,L_i} \right\rangle_{\mathbf{M}_{L_i}} \quad (3.71)$$

Due to Theorem 1 we know that

$$\left\langle \mathbf{1}^R, \tilde{\mathbf{f}}_{EC}^{*,R} \right\rangle_{\mathbf{M}_R} - \left\langle \mathbf{1}^{L_i}, \tilde{\mathbf{f}}_{EC}^{*,L_i} \right\rangle_{\mathbf{M}_{L_i}} = 0. \quad (3.72)$$

Therefore, IU_t reduces to

$$\begin{aligned} IU_t = & - \frac{|\tilde{a}_R|}{2} \sum_{i=1}^{NE} \mathbf{1}^{R,T} \mathbf{M}_R \mathbf{P}_{L_i2R} \left(\mathbf{P}_{R2L_i} \mathbf{u}^R - \mathbf{u}^{L_i} \right) \\ & + \sum_{i=1}^{NE} \frac{|\tilde{a}_{L_i}|}{2} \mathbf{1}^{L_i,T} \mathbf{M}_{L_i} \left(\mathbf{P}_{R2L_i} \mathbf{u}^R - \mathbf{u}^{L_i} \right). \end{aligned} \quad (3.73)$$

Due to the modified M-compatibility condition (3.56) we arrive at

$$\begin{aligned} IU_t = & - \sum_{i=1}^{NE} \frac{|\tilde{a}_{L_i}|}{2} \left(\mathbf{P}_{R2L_i} \mathbf{1}^R \right)^T \mathbf{M}_{L_i} \left(\mathbf{P}_{R2L_i} \mathbf{u}^R - \mathbf{u}^{L_i} \right) \\ & + \sum_{i=1}^{NE} \frac{|\tilde{a}_{L_i}|}{2} \mathbf{1}^{L_i,T} \mathbf{M}_{L_i} \left(\mathbf{P}_{R2L_i} \mathbf{u}^R - \mathbf{u}^{L_i} \right). \end{aligned} \quad (3.74)$$

As \mathbf{P}_{R2L_i} can project a constant exactly (3.59), we find that

$$\begin{aligned} IU_t &= - \sum_{i=1}^{NE} \frac{|\tilde{a}_{L_i}|}{2} \mathbf{1}^{L_i,T} \mathbf{M}_{L_i} \left(\mathbf{P}_{R2L_i} \mathbf{u}^R - \mathbf{u}^{L_i} \right) \\ &\quad + \sum_{i=1}^{NE} \frac{|\tilde{a}_{L_i}|}{2} \mathbf{1}^{L_i,T} \mathbf{M}_{L_i} \left(\mathbf{P}_{R2L_i} \mathbf{u}^R - \mathbf{u}^{L_i} \right), \\ &= 0, \end{aligned} \tag{3.75}$$

which leads to a primary conservative scheme.

To prove energy stability we include (3.69) and (3.70) in (3.61) and adapt the results from Theorem 1 to find that

$$\begin{aligned} IE_t &= - \frac{|\tilde{a}_R|}{2} \sum_{i=1}^{NE} \mathbf{u}^{R,T} \mathbf{M}_R \mathbf{P}_{L_i2R} \left(\mathbf{P}_{R2L_i} \mathbf{u}^R - \mathbf{u}^{L_i} \right) \\ &\quad + \sum_{i=1}^{NE} \frac{|\tilde{a}_{L_i}|}{2} \mathbf{u}^{L_i,T} \mathbf{M}_{L_i} \left(\mathbf{P}_{R2L_i} \mathbf{u}^R - \mathbf{u}^{L_i} \right). \end{aligned} \tag{3.76}$$

Again, we apply the modified M-compatibility condition (3.56) and obtain

$$\begin{aligned} IE_t &= - \sum_{i=1}^{NE} \frac{|\tilde{a}_{L_i}|}{2} \left(\mathbf{P}_{R2L_i} \mathbf{u}^R \right)^T \mathbf{M}_{L_i} \left(\mathbf{P}_{R2L_i} \mathbf{u}^R - \mathbf{u}^{L_i} \right) \\ &\quad + \sum_{i=1}^{NE} \frac{|\tilde{a}_{L_i}|}{2} \mathbf{u}^{L_i,T} \mathbf{M}_{L_i} \left(\mathbf{P}_{R2L_i} \mathbf{u}^R - \mathbf{u}^{L_i} \right), \\ &= - \sum_{i=1}^{NE} \frac{|\tilde{a}_{L_i}|}{2} \left(\mathbf{P}_{R2L_i} \mathbf{u}^R - \mathbf{u}^{L_i} \right)^T \mathbf{M}_{L_i} \left(\mathbf{P}_{R2L_i} \mathbf{u}^R - \mathbf{u}^{L_i} \right) \leq 0, \end{aligned} \tag{3.77}$$

since each \mathbf{M}_{L_i} is a symmetric positive definite matrix, so the non-conforming scheme is energy stable. \square

Note, that the dissipation terms (3.69) and (3.70) are non-symmetric. An alternative choice would be

$$\tilde{\mathbf{f}}_{ES,alt}^{*,L_i} = \tilde{\mathbf{f}}_{EC}^{*,L_i} - \frac{|\tilde{a}_{L_i}|}{2} \mathbf{P}_{R2L_i} \left(\mathbf{u}^R - \mathbf{P}_{L_i2R} \mathbf{u}^{L_i} \right), \tag{3.78}$$

$$\tilde{\mathbf{f}}_{ES,alt}^{*,R} = \tilde{\mathbf{f}}_{EC}^{*,R} - \frac{|\tilde{a}_R|}{2} \left(\mathbf{u}^R - \sum_{i=1}^{NE} \mathbf{P}_{L_i2R} \mathbf{u}^{L_i} \right). \tag{3.79}$$

However, for (3.78) and (3.79) the proof in Theorem 7 would not hold. This is due to the degree of the projection operators (3.59), as $\mathbf{P}_{L_i 2R}$ cannot necessarily project a constant exactly. Also, when considering a constant initial solution the dissipation term for $\tilde{\mathbf{f}}_{ES,alt}^{*,L_i}$ would not vanish and therefore the behaviour of the solution would become unphysical. Note, that the proof of Theorem 7 also holds for deriving an energy stable scheme for geometric conforming interfaces but differing nodal distributions (p refinement) by setting $NE = 1$. Here, the choice of the numerical surface flux (3.69),(3.70) or (3.78),(3.79) do not affect primary conservation and energy stability nor does the solution behave unphysical.

To summarize, we derived a high-order, primary conservative and energy stable SBP method for the LAE on general h/p non-conforming meshes. In the next section we will numerically verify these properties.

3.5. Numerical Verification of Non-Conforming Summation-by-Parts Schemes for the Two Dimensional Linear Advection Equation

In this section we numerically verify properties of the non-conforming SBP scheme as high-order, primary conservation and energy stability/conservation for the two dimensional LAE. We set $\Omega = [0, 1] \times [0, 1] \subset \mathbb{R}^2$ and $a = b = 1$. As in Section 2.3 we use the five-stage, fourth-order low-storage Runge-Kutta method of Carpenter and Kennedy [14]. The explicit time step Δt is selected by the *CFL* condition [39]

$$\Delta t := CFL \frac{\min_i \left\{ \frac{\Delta x_i}{2}, \frac{\Delta y_i}{2} \right\}}{\max_j \{N_j + 1\} \max\{a, b\}}, \quad (3.80)$$

where Δx_i and Δy_i denote the width and height of the i -th element and $N_j + 1$ denotes the number of nodes in one dimension of the j -th element.

In this section, we verify the experimental order of convergence as well as conservation of the primary quantity and energy for the h/p non-conforming SBP scheme. To obtain the projection operators we need to choose the nodes on the intermediate grid. Here, we consider the idea of Kopriva [69] and choose the intermediate grid to contain the nodes of the element with the maximum number of nodes, denoted by $\boldsymbol{\eta}^{max}$. For p refinement we set

$$\boldsymbol{\eta}^{\Xi} := \boldsymbol{\eta}^{max}. \quad (3.81)$$

We reiterate, that the choice of the nodes on the intermediate grid are nearly arbitrary provided an accurate quadrature rule exists [39]. The intermediate grid

does not affect the properties of the scheme as primary conservation, energy stability or high-order. Investigations of choosing different intermediate grids for p refinement have been done by Friedrich et al. [39]. Here, the maximum time step (CFL number) remained constant for different types of intermediate grids. A slightly better \mathcal{L}_2 error was obtained considering a large number of nodes on the intermediate grid. However, this involved more computational cost. Therefore, we chose $\boldsymbol{\eta}^\Xi := \boldsymbol{\eta}^{max}$ as we can simply copy the data from certain elements to the intermediate grid and at the same time consider a "large" number of nodes.

For h refinement we consider discretizations where an element can have two neighbors on one side as in Figure 3.10.

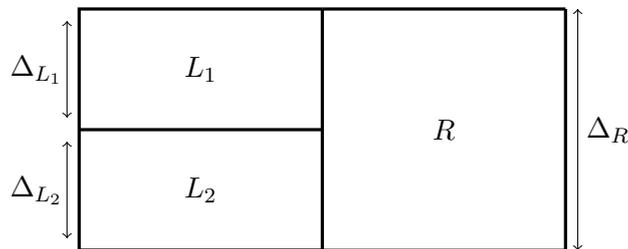


Figure 3.10.: For h refinement we consider "two-to-one" couplings.

Therefore, we set

$$\boldsymbol{\eta}^{\Xi,T} := \left(\frac{\Delta L_1}{\Delta R} (\boldsymbol{\eta}^{max,T} + 1) - 1, \frac{\Delta L_2}{\Delta R} (\boldsymbol{\eta}^{max,T} + 1) - 1 + 2 \frac{\Delta L_1}{\Delta R} \right). \quad (3.82)$$

With these intermediate grids we are able to construct separate projection operators as in Section 3.2 and thus, we can apply non-conforming schemes.

3.5.1. Comparison and Convergence of Summation-by-Parts Schemes

For the numerical convergence experiments, we set $T = 0.1$ and $CFL = 1$. We analyze the experimental order of convergence with the energy stable numerical surface fluxes (3.69) and (3.70). For convergence studies we consider the same exact solution as in Section 2.3 (Equation (2.156)) with periodic boundaries. To examine the convergence order for h/p non-conforming methods we consider a general mesh setup that includes pure p non-conforming interfaces, pure h non-conforming interfaces and h/p non-conforming interfaces. Therefore we define three element types A, B, C . Here, the mesh is prescribed in the following way

- Elements of type A in $\Omega_1 = [0, \frac{1}{2}] \times [0, 1]$,
- Elements of type B in $\Omega_1 = [\frac{1}{2}, 1] \times [0, \frac{1}{2}]$,
- Elements of type C in $\Omega_1 = [\frac{1}{2}, 1] \times [\frac{1}{2}, 1]$.

For each level of the convergence analysis, a single element is divided into three sub-elements. This mesh refinement strategy is sketched in Figure 3.11.

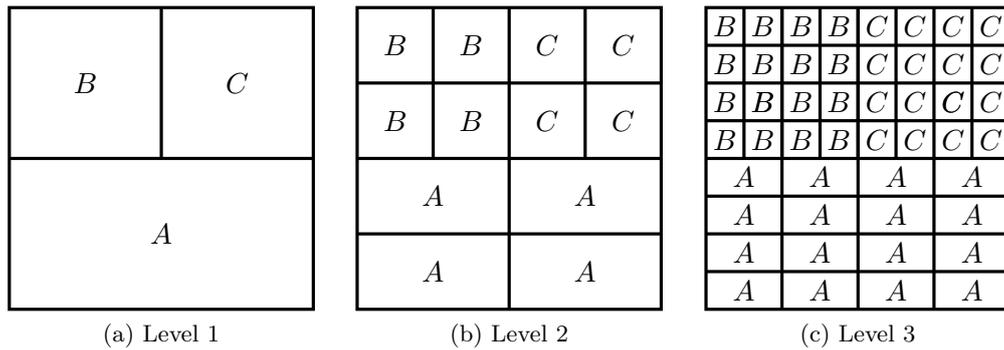


Figure 3.11.: Three levels of mesh refinement used to investigate the EOC for the h/p non-conforming SBP approximation. The figure is reproduced from [38].

In Section 3.2.3, we claimed that non-conforming SBP schemes with differentiation matrices of degree p and norm matrices of degree $2p - 1$ suffer from order loss (in comparison to a conforming scheme) due to projection operators of degree $< p$. To increase the degree of the projection operator we introduced degree preserving SBP operators based on finite difference stencils. In order to verify the order loss or degree preservation we set element A, B and C to be all of the same degree p . Therefore, we consider FD, HGTL and DP operators where

- Element A has 22 nodes in x - and y -direction,
- Element B has 24 nodes in x - and y -direction,
- Element C has 22 nodes in x - and y -direction,

and $p = 2, 3$. With such element distributions we consider p refinement along the line $x = \frac{1}{2}$ for $y \in [\frac{1}{2}, 1]$, h refinement along the line $y = \frac{1}{2}$ for $x \in [\frac{1}{2}, 1]$ and h/p refinement along the line $y = \frac{1}{2}$ for $x \in [0, \frac{1}{2}]$. In addition, we calculate the

maximum CFL number as in Section 2.3. For the non-conforming SBP scheme we obtain EOC rates and maximum CFL numbers presented in Tables 3.1 and 3.2.

EOC and maximum CFL number for the non-conforming SBP discretization with DP, FD and HGTL operators for $p = 2, 3$.

$DOFS$	\mathcal{L}_2DP	$EOCDP$	\mathcal{L}_2FD	$EOCFD$	\mathcal{L}_2HGTL	$EOCHGTL$
1544	3.35E-03		2.62E-03		2.60E-03	
6176	7.76E-04	2.1	5.02E-04	2.4	4.59E-04	2.5
24704	8.96E-05	3.1	9.25E-05	2.4	8.72E-05	2.4
98816	1.02E-05	3.1	1.80E-05	2.4	1.73E-05	2.3
395264	1.18E-06	3.1	3.66E-06	2.3	3.58E-06	2.3
1581056	1.34E-07	3.1	7.64E-07	2.3	7.71E-07	2.2
	Max. CFL	2.57	Max. CFL	2.57	Max. CFL	2.57

Table 3.1.: Setting $p = 2$ for the non-conforming SBP scheme for the LAE.

$DOFS$	\mathcal{L}_2DP	$EOCDP$	\mathcal{L}_2FD	$EOCFD$	\mathcal{L}_2HGTL	$EOCHGTL$
1544	1.42E-03		1.72E-03		1.47E-03	
6176	7.28E-05	4.3	1.45E-04	3.6	1.78E-04	3.0
24704	4.31E-06	4.1	1.43E-05	3.3	1.90E-05	3.2
98816	2.37E-07	4.2	1.58E-06	3.2	2.17E-06	3.1
395264	1.25E-08	4.2	1.78E-07	3.1	2.54E-07	3.1
1581056	6.43E-10	4.3	2.05E-08	3.1	3.01E-08	3.1
	Max. CFL	2.25	Max. CFL	2.24	Max. CFL	2.23

Table 3.2.: Setting $p = 3$ for the non-conforming SBP scheme for the LAE.

In Table 3.1 with $p = 2$ we verify a convergence rate slightly higher than p focusing on FD and HGTL operators. For DP operators however we obtain the desired EOC of $p + 1$. This indicates, that the order reduction considering FD and HGTL operators is presumably because of the degree of the projection operators. Comparing the \mathcal{L}_2 error we verify that the non-conforming scheme with HGTL operators has a better error than using FD operators for all mesh levels, which is the same result we obtained in Section 2.3 when considering a conforming mesh. Considering DP operators and a coarse mesh as in mesh level one with 1544 $DOFS$, we have the highest error. But as the SBP scheme using DP operators has an EOC of $p + 1$ its error is best for more dense meshes (mesh level five, six and seven). Remarkably, all three configurations have the same maximum CFL number of $CFL^* = 2.57$, even though the scheme with DP operators is one order

higher.

When looking at Table 3.2 with $p = 3$ we observe that the maximum CFL numbers for all three configurations are nearly the same ($CFL^* \approx 2.24$). As expected, these CFL numbers are smaller than in Table 3.1 due to a higher order. This also holds for the \mathcal{L}_2 errors. In comparison to Table 3.1 the error of the scheme considering FD operators is smaller than for HGTL operators (for mesh level two and higher). However, the smallest error for all mesh levels is observed when considering DP operators. In total, we observe a degree loss for FD and HGTL operators ($EOC \approx 3$), whereas the EOC for DP operators is here even slightly higher than $p + 1 = 4$.

Note, that within Table 3.1 and 3.2 we did not consider DG operators. This is because we cannot apply p refinement and set all elements to be of the same degree. However, for completeness and to show that non-conforming SBP schemes with norm matrices of degree $2p - 1$ lose an order, we run convergence test using the same mesh levels as in Figure 3.11 with

- Element A with DG operators of degree $p_A = p$ in x - and y -direction,
- Element B with DG operators of degree $p_B = p + 1$ in x - and y -direction,
- Element C with DG operators of degree $p_C = p$ in x - and y -direction,

for $p = 2, 3$. We provide the \mathcal{L}_2 errors, maximum CFL numbers and EOC for the non-conforming SBP method with DG operators in Table 3.3 and 3.4. Here, we again verify a convergence rate slightly higher than p , but not $p + 1$. Also, the maximum CFL numbers are smaller than for the all operators in Table 3.1 and 3.2. This has also been observed in Section 2.3. Therefore the \mathcal{L}_2 error is clearly better when comparing it with all other operators when considering (nearly) the same amount of $DOFS$. However, this result is reasonable as the non-conforming SBP method with DG operators consists of elements with a higher polynomial degree ($p+1$ in element type B) which indicates a smaller \mathcal{L}_2 error and a smaller time step.

To summarize, the non-conforming SBP scheme has the flexibility to chose different nodal distribution as well as elements of different sizes. For classical SBP operators with norm matrices of degree $2p - 1$ the EOC of non-conforming approximations is one degree lower compared to a conforming discretization (EOC

***EOC* and maximum *CFL* number for the non-conforming SBP discretization with DG operators for $p = 2, 3$.**

<i>DOFS</i>	\mathcal{L}_2DG	<i>EOCDG</i>
136	3.47E-02	
544	5.77E-03	2.6
2176	9.13E-04	2.7
8704	1.61E-04	2.5
34816	3.04E-05	2.4
139264	5.75E-06	2.4
557056	1.15E-06	2.3
2228224	2.30E-07	2.3
	Max. <i>CFL</i>	1.63

Table 3.3.: Setting $p = 2$ for the non-conforming SBP scheme with DG operators for the LAE.

<i>DOFS</i>	\mathcal{L}_2DG	<i>EOCDG</i>
228	3.56E-03	
912	2.70E-04	3.7
3648	2.10E-05	3.7
14592	2.17E-06	3.3
58368	1.95E-07	3.5
233472	1.98E-08	3.3
933888	2.01E-09	3.3
3735552	2.21E-10	3.2
	Max. <i>CFL</i>	1.36

Table 3.4.: Setting $p = 3$ for the non-conforming SBP scheme with DG operators for the LAE.

of p instead of $p + 1$). However, when combining a non-conforming scheme with degree preserving SBP operators, we maintain an *EOC* of $p + 1$ on meshes with h and p refinement. Therefore, we recommend DP operators when considering non-conforming SBP schemes for the LAE. Note, that exchanging classical SBP operators with degree preserving SBP operators does effect the maximum time step for the LAE. Besides verifying the high-order property, we show that the scheme is primary conservative and energy stable/conservative in the next section.

3.5.2. Numerical Verification of Primary Conservation and Energy Stability

This section is set up analogous to Section 2.3.2. For all upcoming tests we consider the same configuration as in the last section focusing on mesh level three in Figure 3.11(c). In order to verify that the scheme is energy stable/conservative independent of the corresponding SBP operator we chose the following element types

- Element *A*: FD operator of degree $p = 1$ and 12 nodes in x - and y -direction,
- Element *B*: HGTL operator with $p = 2$ and 16 nodes in x - and y -direction,

- Element C : DP operator of degree $p = 3$ and 24 nodes in x - and y -direction.

Due to this setup all element types consist of a different SBP operator, a different degree and a different number of nodes. We focus on the numerical fluxes (3.69) and (3.70) and control the interface dissipation by introducing a parameter $\sigma \geq 0$

$$\begin{aligned}\tilde{\mathbf{f}}_{ES}^{*,L_i} &= \tilde{\mathbf{f}}_{EC}^{*,L_i} - \sigma \frac{|\tilde{a}_{L_i}|}{2} \left(\mathbf{P}_{R2L_i} \mathbf{u}^R - \mathbf{u}^{L_i} \right), \\ \tilde{\mathbf{f}}_{ES}^{*,R} &= \tilde{\mathbf{f}}_{EC}^{*,R} - \sigma \frac{|\tilde{a}_R|}{2} \sum_{i=1}^{NE} \mathbf{P}_{L_i2R} \left(\mathbf{P}_{R2L_i} \mathbf{u}^R - \mathbf{u}^{L_i} \right).\end{aligned}\tag{3.83}$$

and similar for \mathbf{g}_{ES}^* . For $\sigma = 0$ we proved that the scheme is energy conservative and for $\sigma > 0$ it remains energy stable. As in Section 2.3.2 we verify these properties by calculating

$$\begin{aligned}E_t^{tot} &:= \sum_{q=1}^{N_Q} \sum_{i,j=0}^N u_{ij}^q Res(u_{ij}^q), \\ U_t^{tot} &:= \sum_{q=1}^{N_Q} \sum_{i,j=0}^N Res(u_{ij}^q),\end{aligned}\tag{3.84}$$

where Res is defined in (2.165). In order to verify energy conservation ($\sigma = 0$) we consider random initial conditions (2.171) with periodic boundary conditions. Again, calculating $(E_t^{tot})_k$ and $(U_t^{tot})_k$ for $k = 1, \dots, 1000$ independently we obtain the following results

Calculating the growth in the total discrete energy and primary quantity

$\ U_t^{tot}\ _2$	2.46E-14
$\ E_t^{tot}\ _2$	9.14E-12

Table 3.5.: Calculating the growth in the total discrete primary quantity U_t^{tot} and total discrete energy E_t^{tot} for 1000 different random initial conditions for the non-conforming SBP scheme. Here, we verify conservation of the primary quantity and energy.

Table 3.5 verifies primary and energy conservation even though we consider an arbitrary distribution of the element types. Next, we observe the behaviour of the total discrete primary quantity and total discrete energy in time. Therefore, we consider the initial condition

$$\mathcal{U}(t, x, y) = \begin{cases} 5 & \text{if } 0.3 \leq x \leq 0.7 \text{ and } 0.3 \leq y \leq 0.7, \\ 0 & \text{else.} \end{cases}\tag{3.85}$$

We set $T = 10$ and $CFL = 1$ and consider the same element types as above. The simulation is run twice considering $\sigma = 0$ (energy conservative) and $\sigma = 1$ (energy stable). The total discrete primary quantity and energy is calculated analogously as in (2.173). We plot these values over time in Figure 3.12 and 3.13 respectively.

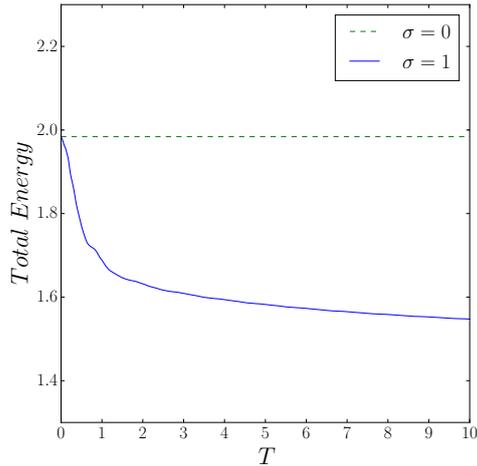


Figure 3.12.: Evolution of the total discrete energy of the solution with and without dissipation. We see that the total discrete energy is conserved when considering the central flux ($\sigma = 0$) and that it decays using the upwind flux ($\sigma = 1$).

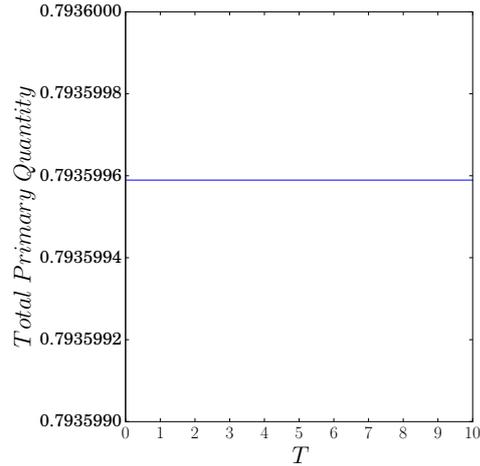


Figure 3.13.: A plot that demonstrates the conservation of the integrated conserved variables. The plot does not depend on the choice of the numerical surface flux.

Here, we verify energy conservation and stability depending on σ as the total discrete energy is not increasing. Also, the total discret primary quantity remains constant independent of an energy conservative or stable surface flux.

3.6. Summary

In this chapter we derived a primary conservative and energy stable SBP method for solving the two dimensional linear advection equation on a non-conforming mesh considering h/p refinement. For such meshes, neither the nodal distributions

nor the interfaces of the elements need to coincide making the mesh more flexible. We reviewed the work on non-conforming SBP methods by considering older and state-of-the-art schemes. All these approximations rely on projection operators. To obtain an energy stable scheme, it is necessary that the projection operators satisfy the \mathbf{M} -compatibility condition (3.2). We described how to construct high-order projection operators which at the same time satisfy this condition. However, the \mathbf{M} -compatibility condition violates that the projection operators are of the same degree as the differentiation matrix of the SBP operator. This is due to the accuracy of the norm matrix, as for classical SBP operators the degree of the norm matrix is $2p - 1$, where p denotes the degree of the differentiation matrix. In order to obtain \mathbf{M} -compatible projection operators of degree p , we introduced the degree preserving SBP operators by Friedrich et al. [39], where the degree of the norm matrix is increased to $\geq 2p$. Given a set of \mathbf{M} -compatible projection operators we introduced the non-conforming method by first focusing on p refinement. We then adapt the results from p refinement to obtain an energy stable scheme even for meshes with h refinement. At last, we implemented the non-conforming SBP method. Here, we verified that the scheme is high-order and looked at the different error terms and maximum CFL numbers. We observed, that the non-conforming SBP scheme with degree preserving operators of Friedrich et al. [39] had a higher EOC as the classical SBP operators. Remarkably, even though the convergence order is one order higher, the maximum time step is not affected when considering classical SBP operators or degree preserving SBP operators. Besides analyzing the time step and errors of the scheme, we verified numerically that the non-conforming SBP scheme is energy stable/conservative and primary conservative considering arbitrary random initial conditions and an arbitrary distributions of SBP operators in the elements.

4. Entropy Stable and Primary Conservative Summation-by-Parts Methods for Non-Linear Hyperbolic Systems on Conforming Meshes

In the last two chapters we introduced SBP methods for solving the linear advection equation. These methods build the foundation of solving more complex hyperbolic systems. In comparison to Chapter 2 and 3 we now consider a general form of systems of hyperbolic conservation laws. In one spatial dimension a system with N_{Eq} equations is defined as

$$\frac{\partial \mathbf{u}}{\partial t} + \frac{\partial \mathcal{F}(\mathbf{u})}{\partial x} = \mathbf{0}, \quad (4.1)$$

with $\mathbf{u} := [u^1, \dots, u^{N_{Eq}}]^T$ and similar for \mathcal{F} . Note that, in contrast to the scalar case, \mathbf{u} and \mathcal{F} are vectors as we consider a system of conservation laws. There are several problems modeled by (4.1), e.g. the Shallow Water equations without bottom topography

$$\frac{\partial}{\partial t} \begin{pmatrix} h \\ h\nu \end{pmatrix} + \frac{\partial}{\partial x} \begin{pmatrix} h\nu \\ h\nu^2 + \frac{1}{2}gh^2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad (4.2)$$

where $h \geq 0$ denotes the water height and ν is the velocity with the gravitational constant $g > 0$. Another representative of (4.1) are the compressible Euler equations

$$\frac{\partial}{\partial t} \begin{pmatrix} \rho \\ \rho\nu \\ \mathcal{E} \end{pmatrix} + \frac{\partial}{\partial x} \begin{pmatrix} \rho\nu \\ \rho\nu^2 + \mathcal{P} \\ (\mathcal{E} + \mathcal{P})\nu \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \quad (4.3)$$

where ρ denotes the density, ν the velocity, \mathcal{E} the total energy, \mathcal{P} the pressure, where $\mathcal{P} = (\gamma - 1)(\mathcal{E} - \frac{1}{2}\rho\nu^2)$ and γ the adiabatic coefficient with $\gamma = 1.4$.

In contrast to the LAE, solutions of non-linear conservation laws can be discontinuous regardless of the initial condition. This violates the model (4.1) as it

assumes a differentiable solution. Therefore, we seek solutions of the weak formulation (e.g. (2.128)) of hyperbolic systems. However, weak solutions are not unique and thus can behave unphysical [85]. Thus, we focus on solutions which obey the second law of thermodynamics, saying that the total entropy of a physical system can only increase. The interpretation of the entropy is problem specific (e.g. energy for the LAE), see [22]. We describe the entropy function by $\mathcal{S}(\mathbf{u})$. Furthermore, we introduce *entropy variables* $\mathbf{v} := \left(\frac{\partial \mathcal{S}}{\partial u^1}, \dots, \frac{\partial \mathcal{S}}{\partial u^{N_{Eq}}} \right)^T$ and thus we get the relationship

$$\frac{\partial \mathcal{S}}{\partial t} = \sum_{i=1}^{N_{Eq}} \frac{\partial \mathcal{S}}{\partial u^i} \frac{\partial u^i}{\partial t} = \mathbf{v}^T \frac{\partial \mathbf{u}}{\partial t}. \quad (4.4)$$

The mathematical definition of \mathcal{S} is adapted from [51].

Definition 10. Entropy Function

A scalar function $\mathcal{S} = \mathcal{S}(\mathbf{u})$ is an entropy function if it is strongly convex and the corresponding entropy variables satisfy the compatibility condition

$$\mathbf{v}^T \frac{\partial \mathcal{F}}{\partial x} = \frac{\partial \mathcal{F}^{ent}}{\partial x}, \quad (4.5)$$

where \mathcal{F}^{ent} denotes the entropy flux.

Remark 11. Here, the entropy variables are obtained by considering a given strongly convex entropy function. However, Mock [86] proved that a strongly convex entropy function exists if the Jacobian matrix $\frac{\partial \mathbf{u}}{\partial \mathbf{v}}$ is symmetric positive definite and $\frac{\partial \mathcal{F}}{\partial \mathbf{v}}$ is symmetric for a given set of variables \mathbf{v} .

Due to the convexity of \mathcal{S} the Hessian $\frac{\partial^2 \mathcal{S}}{\partial u^2} = \frac{\partial \mathbf{v}}{\partial \mathbf{u}}$ is symmetric positive definite and thus, yields to a one-to-one mapping from the conserved variables to the entropy variables. Typically, the entropy function and the entropy flux are denoted as an entropy-entropy flux pair $(\mathcal{S}, \mathcal{F}^{ent})$. A connection between this pair and $(\mathbf{u}, \mathcal{F})$ is given by the potential-potential flux pair (Ψ^u, Ψ^f) [115]

$$\Psi^u = \mathbf{v}^T \mathbf{u} - \mathcal{S}, \quad (4.6)$$

$$\Psi^f = \mathbf{v}^T \mathcal{F} - \mathcal{F}^{ent}. \quad (4.7)$$

With the entropy variables we can contract (4.1) with \mathbf{v}^T and get a scalar conservation law for the mathematical entropy function

$$\begin{aligned} \mathbf{v}^T \frac{\partial \mathbf{u}}{\partial t} + \mathbf{v}^T \frac{\partial \mathcal{F}(\mathbf{u})}{\partial x} &= 0, \\ \Leftrightarrow \frac{\partial \mathcal{S}}{\partial t} + \frac{\partial \mathcal{F}^{ent}}{\partial x} &= 0. \end{aligned} \quad (4.8)$$

Applying the divergence theorem over the domain $\Omega = [x_L, x_R]$ results in a global conservation statement for the entropy

$$\int_{\Omega} \frac{\partial \mathcal{S}}{\partial t} d\Omega + \mathcal{F}^{ent} \Big|_{x_L}^{x_R} = 0, \quad (4.9)$$

so conservation laws are entropy conservative for smooth solutions. However, discontinuities in the form of shocks can develop in finite time for non-linear problems despite smooth initial data. Therefore, the integral statement of (4.9) is not strictly valid and does not account for the dissipation of the entropy at the shock [115]. From a physical point of view, the total entropy of a physical system can only increase. In contrast, from a mathematical view we are concerned of bounding the solution by initial and boundaries conditions in terms of stability (e.g. the energy stability for the LAE). Therefore, we consider the mathematical entropy to be a decreasing quantity in time. So, there is a sign issue and thus different interpretations for mathematical and physical entropy. For more informations see [85]. In this work we interpret the entropy mathematically. Due to dissipation in the entropy we enforce that the entropy satisfies the entropy inequality

$$\int_{\Omega} \frac{\partial \mathcal{S}}{\partial t} d\Omega + \mathcal{F}^{ent} \Big|_{x_L}^{x_R} \leq 0. \quad (4.10)$$

Note, that in comparison to the LAE we avoid discussing the boundary conditions and well-posedness. Details on well-posedness can be found in [35]. Deriving appropriate boundaries conditions depends on the corresponding conservation laws and have been presented in e.g. [34] for the Burgers' equation and in [92, 95] for the Navier-Stokes equations. However, as we consider a general system of hyperbolic equations, we will not focus on physically relevant boundary conditions and set the boundary conditions to be periodic, i.e. for a closed system. Therefore, $\mathcal{F}^{ent}(t, x_L) = \mathcal{F}^{ent}(t, x_R)$ and thus the inequality (4.10) becomes

$$\int_{\Omega} \frac{\partial \mathcal{S}}{\partial t} d\Omega \leq 0. \quad (4.11)$$

Note, that (4.11) is a more general definition of the energy estimate (2.11) for the LAE (with periodic boundary conditions). For the LAE the energy estimate (2.11) bounds the total energy and thus the solution cannot explode, for which we denoted the LAE as energy stable. For the entropy inequality (4.11) the total entropy cannot grow, for which we denote the hyperbolic system (4.1) as *entropy stable*. As the entropy function is convex it follows that all conservative variables are bounded under certain assumptions (e.g. positive water height $h > 0$ for Shallow Water Equations), see [115]. Thus, the continuous entropy analysis provides

an insightful \mathcal{L}_2 stability estimate for the hyperbolic system of conservation laws [30, 52, 114]. Therefore, we are interested in discretely mimicking the inequality (4.11).

In this chapter we introduce numerical schemes which discretely mimic the entropy stability inequality (4.11). These schemes are denoted as *entropy stable schemes*. Such methods were first derived in a finite volume framework by Tadmor in 1987 [113]. Over the ensuing years there have been numerous investigations into entropy stable schemes [10, 33, 47, 62, 68, 102, 113, 115, 122]. We investigate in entropy stable methods as these are more robust compared to methods that do not consider the entropy, especially for turbulent flows [7, 46, 107, 123].

4.1. Entropy Stable Summation-by-Parts Methods for One Dimensional Hyperbolic Systems of Conservation Laws

In this section we will introduce primary conservative and entropy stable discretizations for non-linear systems of hyperbolic conservation laws. As in Chapters 2 and 3 we use SBP operators to discretize derivatives. However, in comparison to the chapters before, we provide a general framework as we do not focus on a specific set of equations, but on an arbitrary non-linear hyperbolic system of conservation laws with an entropy function $\mathcal{S}(\mathbf{u})$. Such a system can be numerically solved by first dividing the domain Ω into N_Q non-overlapping elements as in (2.45) and performing the approximation (2.98).

$$\omega_i \frac{\partial \mathbf{u}_i}{\partial t} + \sum_{k=0}^N Q_{ik} \mathbf{f}_k = \delta_{iN} [\mathbf{f} - \mathbf{f}^*]_N - \delta_{i0} [\mathbf{f} - \mathbf{f}^*]_0, \quad (4.12)$$

where $\mathbf{u}_i = (u_i^1, \dots, u_i^{N_{Eq}})^T$ for $i = 0, \dots, N$ and similar for \mathbf{f} on each element. Note, that (4.12) differs slightly from (2.98) as from now on the entries of \mathbf{u}_i refer to the equations of the hyperbolic system. However, for scalar conservation laws (4.12) and (2.98) are algebraically equivalent. To derive an entropy stable scheme we first define the terms *primary conservative* and *entropy stable*.

Definition 11. Primary Conservative Schemes for One Dimensional Hyperbolic Systems of Conservation Laws

A semi-discrete method approximating a hyperbolic system of conservation laws is said to be primary conservative if

$$\sum_{q=1}^{N_Q} \frac{\Delta x}{2} \sum_{i=0}^N \omega_i \frac{\partial \mathbf{u}_i^q}{\partial t} = \mathbf{0}, \quad (4.13)$$

assuming periodic boundary conditions.

Definition 12. Entropy Stable Schemes for One Dimensional Hyperbolic Systems of Conservation Laws

A semi-discrete method approximating a hyperbolic system of conservation laws is said to be entropy stable if

$$\sum_{q=1}^{N_Q} \frac{\Delta x}{2} \sum_{i=0}^N \omega_i \frac{\partial S_i^q}{\partial t} \leq 0, \tag{4.14}$$

assuming periodic boundary conditions.

Remark 12. Definition 12 is similar to the Definition (2.131) for the LAE when considering periodic boundary conditions and setting $\mathcal{S} = \frac{1}{2}\mathcal{U}^2$. Definition 12 can be interpreted as a generalization of the definition of energy stability for the LAE.

Here, for simplicity, we assumed that all elements have the same width and thus Δx remains constant. Note, that in comparison to Definitions 5 and 6, we set the boundary conditions to be periodic. For the discretization (4.12) the scheme is primary conservative. This can be shown by re-deriving the steps of the proof of Theorem 2 in a one-to-one fashion. However, discretization (4.12) is not entropy stable when considering non-linear physical fluxes. Therefore, we must modify the discretization. To do so, we introduce *skew-symmetric* discretizations in Section 4.1.1. An even more general entropy stable discretization is presented in Section 4.1.2. Both of these methods depend on split forms of the PDE and are equivalent for problems as Burgers' or Shallow Water equations, which will be demonstrated.

4.1.1. Skew-Symmetric Discretization

In this section we introduce a skew-symmetric discretization for non-linear problems. Such discretizations are entropy stable and rely on split forms of the conservation law. The choice of the corresponding split form is problem specific. Therefore, we begin with the simplest prototype of a non-linear conservation law: The Burgers' equation.

$$\frac{\partial \mathcal{U}}{\partial t} + \frac{\partial}{\partial x} \left(\frac{1}{2} \mathcal{U}^2 \right) = 0. \tag{4.15}$$

As for the LAE, the Burgers' equation is a scalar equation with entropy/energy of $\mathcal{S} = \frac{1}{2}\mathcal{U}^2$. Note, that in this chapter we aim to derive entropy stable discretizations for non-linear systems. However, in terms of simplicity and understanding the idea of skew-symmetric discretizations based on split forms, the scalar Burgers'

equation is sufficient. Simply discretizing the Burgers' equation with (4.12) is not energy stable. This can be shown analyzing

$$\frac{\partial u_i}{\partial t} + \sum_{k=0}^N D_{ik} \left(\frac{1}{2} u_k^2 \right) = SAT_i, \quad (4.16)$$

where we divided (4.12) by ω_i for $i = 0, \dots, N$ and replaced the surface contribution by an appropriate SAT . Assuming that norm matrix \mathbf{M} is diagonal ($M_{ij} = \delta_{ij}\omega_i$), we apply the energy method by multiplying the discretization (4.16) with $\omega_i u_i$ and summing over all $i = 0, \dots, N$

$$\sum_{i=0}^N \omega_i \frac{1}{2} \frac{\partial u_i^2}{\partial t} + \sum_{i=0}^N \sum_{k=0}^N u_i Q_{ik} \left(\frac{1}{2} u_k^2 \right) = \sum_{i=0}^N \omega_i u_i SAT_i. \quad (4.17)$$

In the last section the discrete volume integral for the LAE reduced to surface evaluations. However, this does not hold for non-linear flux function, as for the Burgers' equation the term

$$\sum_{i=0}^N \sum_{k=0}^N u_i Q_{ik} u_k^2, \quad (4.18)$$

cannot be simplified to evaluations on the surface of the element. To overcome this difficulty we derive a discrete skew-symmetric discretization. The main idea is rewriting the spatial derivative $\frac{\partial}{\partial x}$ in terms of split forms. For example, in the work of Gassner [42] the spatial derivative of the Burgers' flux is written as

$$\frac{\partial \mathcal{U}}{\partial t} + \alpha \left(\frac{1}{2} \frac{\partial \mathcal{U}^2}{\partial x} \right) + (1 - \alpha) \left(\mathcal{U} \frac{\partial \mathcal{U}}{\partial x} \right) = 0, \quad (4.19)$$

for $\alpha \in [0, 1]$. Note, that (4.15) and (4.19) are algebraically equivalent due to the chain rule $\frac{1}{2} \frac{\partial \mathcal{U}^2}{\partial x} = \mathcal{U} \frac{\partial \mathcal{U}}{\partial x}$. Discretizing (4.19) with SBP operators gives us the discretization

$$\frac{\partial u_i}{\partial t} + \alpha \sum_{k=0}^N D_{ik} \left(\frac{1}{2} u_k^2 \right) + (1 - \alpha) \sum_{k=0}^N u_i D_{ik} u_k = SAT_i, \quad (4.20)$$

where the boundary term SAT remains the same as in (4.16). Note, that even though (4.15) and (4.19) are equivalent at a continuous level, this does not hold at a discrete level for (4.16) and (4.20). Here, for an appropriate choice of α the discretization (4.20) is entropy stable. We show this by applying the energy method

$$\sum_{i=0}^N \omega_i \frac{1}{2} \frac{\partial u_i^2}{\partial t} + \frac{\alpha}{2} \sum_{i=0}^N \sum_{k=0}^N u_i Q_{ik} u_k^2 + (1 - \alpha) \sum_{i=0}^N \sum_{k=0}^N u_i^2 Q_{ik} u_k = \sum_{i=0}^N \omega_i u_i SAT_i. \quad (4.21)$$

Due to the SBP property (2.64) we know

$$\sum_{i=0}^N \sum_{k=0}^N u_i \mathbf{Q}_{ik} u_k^2 = \sum_{i=0}^N \sum_{k=0}^N u_i \mathbf{B}_{ik} u_k^2 - \sum_{i=0}^N \sum_{k=0}^N u_i^2 \mathbf{Q}_{ik} u_k, \quad (4.22)$$

and therefore

$$\sum_{i=0}^N \omega_i \frac{1}{2} \frac{\partial u_i^2}{\partial t} + \frac{\alpha}{2} (u_N^3 - u_0^3) + \left(1 - \frac{3\alpha}{2}\right) \sum_{i=0}^N \sum_{k=0}^N u_i^2 \mathbf{Q}_{ik} u_k = \sum_{i=0}^N \omega_i u_i SAT_i. \quad (4.23)$$

Finally, by setting $\alpha = \frac{2}{3}$ the volume term vanishes and the discrete energy growth reduces to evaluations at the surface of the element. Therefore, the discretization

$$\frac{\partial u_i}{\partial t} + \frac{2}{3} \sum_{k=0}^N \mathbf{D}_{ik} \left(\frac{1}{2} u_k^2\right) + \frac{1}{3} \sum_{k=0}^N u_i \mathbf{D}_{ik} u_k = SAT_i, \quad (4.24)$$

is entropy stable provided an appropriate numerical surface flux.

Remark 13. Discretization (4.24) is entropy conservative considering the numerical surface flux given in [42]

$$f_{EC}^*(u^L, u^R) = \frac{1}{6} \left((u^L)^2 + u^L u^R + (u^R)^2 \right). \quad (4.25)$$

Entropy stability is then ensured by including numerical dissipation.

Remark 14. Discretization (4.24) with the numerical surface flux (4.25) is entropy conservative for diagonal norm matrices. For dense norm matrix an entropy conservative discretization has been purposed by [101].

Skew-symmetric discretizations can be found in numerous papers as in [7, 43, 47, 68, 118]. As mentioned before, the choice of the split form and the resulting discretization depends on the corresponding conservation law. Also, when considering problems such as the compressible Euler equations, an explicit skew-symmetric form to obtain an entropy stable scheme is (up to now) not known. In order to derive an entropy stable scheme for a general system of hyperbolic conservation laws we next introduce the *SBP Flux Difference* scheme by Fisher and Carpenter [33].

4.1.2. Summation-by-Parts Flux Difference Scheme

In the section we discuss the SBP Flux Difference method by Fisher and Carpenter [33]. As for the skew-symmetric discretization the Flux Difference scheme is constructed such that the discrete growth in entropy purely depends on the

surfaces of each element. Here, results from entropy stable finite volume schemes as in [62, 102, 112, 113, 115, 121, 122] are adapted to high-order SBP schemes. For more details on finite volume schemes see [74]. Here, we review entropy stable/conservative finite volume schemes on a domain $\Omega = [x_L, x_R] \subset \mathbb{R}$. A semi-discrete first order finite volume method on N_Q elements with fixed width Δx is defined by

$$\frac{\partial \mathbf{u}_i}{\partial t} + \frac{1}{\Delta x} (\mathbf{f}_{i+1}^* - \mathbf{f}_i^*) = 0, \quad (4.26)$$

where $\mathbf{u}_i = (u_i^1, \dots, u_i^{N_{Eq}})^T$ and

$$\mathbf{f}_i^* := \mathbf{f}^*(\mathbf{u}_i, \mathbf{u}_{i-1}), \quad (4.27)$$

for $i = 2, \dots, N_Q$ and $\mathbf{f}_1^* := \mathbf{f}(\mathbf{u}_1)$, $\mathbf{f}_{N_Q+1}^* := \mathbf{f}(\mathbf{u}_{N_Q})$ needed for primary conservation [115]. As noted in [33] a finite volume method is entropy conservative if it discretely approximates

$$\int_{\Omega} \frac{\partial \mathcal{S}}{\partial t} d\Omega + \mathcal{F}^{ent} \Big|_{x_L}^{x_R} = 0. \quad (4.28)$$

In terms of deriving an entropy conserving scheme we assume that the numerical flux satisfies

$$\mathbf{f}^*(\mathbf{u}_i, \mathbf{u}_j) = \mathbf{f}^*(\mathbf{u}_j, \mathbf{u}_i), \quad (4.29)$$

$$\mathbf{f}^*(\mathbf{u}_i, \mathbf{u}_i) = \mathbf{f}(\mathbf{u}_i). \quad (4.30)$$

Fluxes fulfilling (4.29) and (4.30) are denoted as *symmetric* and *consistent*, respectively.

To obtain the discrete entropy function S we multiply (4.26) with the discrete entropy variables \mathbf{v}_i^T and sum over all elements

$$\begin{aligned} & \sum_{i=1}^{N_Q} \mathbf{v}_i^T \frac{\partial \mathbf{u}_i}{\partial t} + \frac{1}{\Delta x} \sum_{i=1}^{N_Q} (\mathbf{v}_i^T \mathbf{f}^*(\mathbf{u}_{i+1}, \mathbf{u}_i) - \mathbf{v}_i^T \mathbf{f}^*(\mathbf{u}_i, \mathbf{u}_{i-1})) = 0, \\ \Leftrightarrow & \sum_{i=1}^{N_Q} \Delta x \frac{\partial S_i}{\partial t} + \left(\mathbf{v}_{N_Q}^T \mathbf{f}(\mathbf{u}_{N_Q}) - \mathbf{v}_1^T \mathbf{f}(\mathbf{u}_1) - \sum_{i=1}^{N_Q-1} (\mathbf{v}_{i+1} - \mathbf{v}_i)^T \mathbf{f}^*(\mathbf{u}_{i+1}, \mathbf{u}_i) \right) = 0, \end{aligned} \quad (4.31)$$

In order to include the entropy flux we use $f^{ent} = \mathbf{v}^T \mathbf{f} - \Psi^f$ and get

$$\underbrace{\sum_{i=1}^{N_Q} \Delta x \frac{\partial S_i}{\partial t}}_{\approx \int_{\Omega} \frac{\partial \mathcal{S}}{\partial t} d\Omega} + \underbrace{f_{N_Q}^{ent} - f_1^{ent}}_{\approx \mathcal{F}^{ent} \Big|_{x_L}^{x_R}} = \underbrace{-\Psi_{N_Q}^f + \Psi_1^f + \sum_{i=1}^{N_Q-1} (\mathbf{v}_{i+1} - \mathbf{v}_i)^T \mathbf{f}^*(\mathbf{u}_{i+1}, \mathbf{u}_i)}_{= (*)}. \quad (4.32)$$

Here, the left hand side is a first order approximation of the left hand side of (4.28). So the finite volume scheme is said to be entropy conservative if $(*) = 0$. As $-\Psi_{N_Q}^f + \Psi_1^f = -\sum_{i=1}^{N_Q-1} (\Psi_{i+1}^f - \Psi_i^f)$ we get

$$(*) = \sum_{i=1}^{N_Q-1} (\mathbf{v}_{i+1} - \mathbf{v}_i)^T \mathbf{f}^*(\mathbf{u}_{i+1}, \mathbf{u}_i) - (\Psi_{i+1}^f - \Psi_i^f). \quad (4.33)$$

So, in order to obtain an entropy conservative scheme, the entropy conserving flux needs to satisfy the so called *Tadmor Shuffle condition* [115]

$$(\mathbf{v}_{i+1} - \mathbf{v}_i)^T \mathbf{f}^*(\mathbf{u}_{i+1}, \mathbf{u}_i) = \Psi_{i+1}^f - \Psi_i^f, \quad (4.34)$$

for $i = 1, \dots, N_Q$. Using a finite volume method with a numerical flux satisfying the Tadmor Shuffle condition (4.34) gives us a low-order entropy conservative scheme. Over the last decades there have been several publications extending such spatial methods to high-order spatial approximations with particular finite volume reconstruction techniques, e.g. [37, 73].

In 2013, the work of Fisher and Carpenter [33] opened a new avenue for discrete entropy analysis for arbitrary high-order SBP methods by considering low-order entropy conservative fluxes satisfying the Tadmor Shuffle condition (4.34). Their method is referred to as the *Flux Difference method*. The key concept is to rewrite a SBP scheme into an equivalent sub-cell finite volume differencing method using a *staggered grid*. Let $\bar{\xi}_l$ denote the nodes on the staggered grid with the representative flux \bar{f}_l for $l = 0, \dots, N + 1$. The staggered grid consists of nodes of the size $N + 2$ (one more than for the SBP operator) and is illustrated for the LGL nodes with $N = 3$ in Figure 4.1

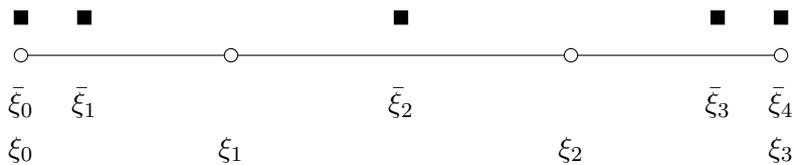


Figure 4.1.: One dimensional staggered grid for four LGL nodes.

We rewrite the SBP scheme in terms of a sub-cell finite volume method by replacing the volume integral (4.12) in terms of

$$\sum_{k=0}^N Q_{ik} \mathbf{f}_k = \sum_{l=0}^{N+1} \Delta_{il} \bar{\mathbf{f}}_l, \quad (4.35)$$

for $i = 0, \dots, N$. Here, the differencing matrix $\Delta \in \mathbb{R}^{N \times N+1}$ is defined by

$$\Delta := \begin{pmatrix} -1 & 1 & 0 & \dots & \dots & 0 \\ 0 & -1 & 1 & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & -1 & 1 & 0 \\ 0 & \dots & \dots & 0 & -1 & 1 \end{pmatrix}. \quad (4.36)$$

The values of \bar{f}_l for $l = 0, \dots, N + 1$ on the staggered grid satisfying (4.35) are calculated by

$$\bar{f}_l := \begin{cases} \mathbf{f}_0 & \text{if } l = 0, \\ \mathbf{f}_0 + \sum_{k=0}^N \sum_{l=0}^{i-1} \mathbf{Q}_{lk} \mathbf{f}_k & \text{for } l = 1, \dots, N, \\ \mathbf{f}_N & \text{if } l = N + 1. \end{cases} \quad (4.37)$$

The corresponding nodes on the staggered grid are obtained by exchanging \mathbf{f}_l with the nodes of the SBP operator ξ_l . So an algebraically equivalent discretization of (4.12) on an element E_q is given by

$$\omega_i \frac{\partial \mathbf{u}_i}{\partial t} + \sum_{k=0}^{N+1} \Delta_{ik} \bar{\mathbf{f}}_k = \delta_{iN} [\mathbf{f} - \mathbf{f}^*]_N - \delta_{i0} [\mathbf{f} - \mathbf{f}^*]_0, \quad (4.38)$$

or

$$\frac{\partial \mathbf{u}_i}{\partial t} + \frac{1}{\omega_i} (\bar{\mathbf{f}}_{i+1} - \bar{\mathbf{f}}_i) = \frac{\delta_{iN}}{\omega_i} [\mathbf{f} - \mathbf{f}^*]_N - \frac{\delta_{i0}}{\omega_i} [\mathbf{f} - \mathbf{f}^*]_0, \quad (4.39)$$

for $i = 0, \dots, N$. Considering (4.37) for $\bar{\mathbf{f}}$, then the scheme is equivalent to the standard SBP discretization (4.12). As pointed out in Section 4.1.1 such a discretization is not necessarily entropy stable/conservative. Therefore, Fisher and Carpenter replaced $\bar{\mathbf{f}}_l$ by a symmetric numerical flux $\bar{\mathbf{f}}_l^* := \bar{\mathbf{f}}^*(\mathbf{u}_l, \mathbf{u}_{l-1})$ for $l = 1, \dots, N$ and $\bar{\mathbf{f}}_0^* := \mathbf{f}_0$, $\bar{\mathbf{f}}_{N+1}^* := \mathbf{f}_N$. Including this flux we obtain the modified discretization

$$\frac{\partial \mathbf{u}_i}{\partial t} + \frac{1}{\omega_i} (\bar{\mathbf{f}}_{i+1}^* - \bar{\mathbf{f}}_i^*) = \frac{\delta_{iN}}{\omega_i} [\mathbf{f} - \mathbf{f}^*]_N - \frac{\delta_{i0}}{\omega_i} [\mathbf{f} - \mathbf{f}^*]_0, \quad (4.40)$$

for $i = 0, \dots, N$. Here, the left hand side is similar to the left hand side of the finite volume discretization (4.26). So, (4.40) is a finite volume differencing type where the element E is divided into $N + 1$ sub-elements of size ω_i for $i = 0, \dots, N$. This is the crux of the Flux Difference scheme as we can use entropy conservative numerical surface fluxes from finite volume methods and thus, set \mathbf{f}^* to satisfy the Tadmor Shuffle condition (4.34). Note, that this modification does not effect

primary conservation as $\sum_{i=0}^N \omega_i \frac{\partial \mathbf{u}_i}{\partial t} = \mathbf{f}_N^* - \mathbf{f}_0^*$ due to a telescopic sum. In order to obtain the discrete growth in entropy we multiply (4.40) by $\omega_i \mathbf{v}_i^T$ and sum over all $i = 0, \dots, N$

$$\sum_{i=0}^N \omega_i \mathbf{v}_i^T \frac{\partial \mathbf{u}_i}{\partial t} + \sum_{i=0}^N \mathbf{v}_i^T (\bar{\mathbf{f}}_{i+1}^* - \bar{\mathbf{f}}_i^*) = \mathbf{v}_N^T [\mathbf{f} - \mathbf{f}^*]_N - \mathbf{v}_0^T [\mathbf{f} - \mathbf{f}^*]_0. \quad (4.41)$$

By applying analogous steps as for the finite volume discretization and using that $f^{ent} = \mathbf{v}^T \mathbf{f} - \Psi^f$ we obtain

$$\begin{aligned} & \sum_{i=0}^N \omega_i \frac{\partial S_i}{\partial t} + [\mathbf{v}^T \mathbf{f} - \Psi^f]_N - [\mathbf{v}^T \mathbf{f} - \Psi^f]_0 = \mathbf{v}_N^T [\mathbf{f} - \mathbf{f}^*]_N - \mathbf{v}_0^T [\mathbf{f} - \mathbf{f}^*]_0, \\ \Leftrightarrow & \sum_{i=0}^N \omega_i \frac{\partial S_i}{\partial t} = -[\mathbf{v}^T \mathbf{f}^* - \Psi^f]_N + [\mathbf{v}^T \mathbf{f}^* - \Psi^f]_0. \end{aligned} \quad (4.42)$$

So the discrete entropy growth reduces to evaluations on the element surface. To finally verify that the Flux Difference scheme is entropy conservative we look at the discrete entropy growth at an interior interface denoted by IS_t with a left and right state.

$$IS_t = (\mathbf{v}^R - \mathbf{v}^L)^T \mathbf{f}^* - (\Psi^{f,R} - \Psi^{f,L}). \quad (4.43)$$

Again, choosing \mathbf{f}^* to satisfy the Tadmor Shuffle condition (4.34), we find that $IS_t = 0$ and thus, we obtain an entropy conservative scheme. In addition, the scheme remains entropy stable by including numerical dissipation at the element interfaces as in [18, 36, 118].

The discretization (4.40) is entropy conservative for systems of hyperbolic conservation laws assuming that the numerical fluxes $\bar{\mathbf{f}}^*$ and \mathbf{f}^* satisfy the Tadmor Shuffle condition (4.34). Many such entropy conservative numerical flux functions are available for systems of hyperbolic conservation laws, e.g. the compressible Euler equations [18, 62] and are denoted by \mathbf{f}_{EC}^* , where the index "EC" refers to "entropy conservation". However, classical FV approximations (without reconstruction) are low-order methods. Therefore, instead of directly setting $\bar{\mathbf{f}}^* = \mathbf{f}_{EC}^*$, Fisher and Carpenter [33] modified the discretization to obtain a high-order scheme.

Focusing on (4.35) we see

$$\frac{1}{\omega_i} (\bar{\mathbf{f}}_{i+1} - \bar{\mathbf{f}}_i) = \sum_{k=0}^N D_{ik} \mathbf{f}_k = \frac{\partial \mathcal{F}(\mathbf{u}_i)}{\partial x} + \mathcal{O}(\Delta x^{p+1}), \quad (4.44)$$

for $i = 0, \dots, N$ and $\bar{\mathbf{f}}$ defined in (4.37). Therefore, we need to choose $\bar{\mathbf{f}}^*$ so that

$$\frac{1}{\omega_i} (\bar{\mathbf{f}}_{i+1}^* - \bar{\mathbf{f}}_i^*) = \frac{\partial \mathcal{F}(\mathbf{u}_i)}{\partial x} + \mathcal{O}(\Delta x^{p+1}). \quad (4.45)$$

Additionally, in order not to violate entropy conservation, the following equation needs to hold

$$\sum_{i=0}^N \mathbf{v}_i^T (\bar{\mathbf{f}}_{i+1}^* - \bar{\mathbf{f}}_i^*) = f_N^{ent} - f_0^{ent}. \quad (4.46)$$

A suitable choice for $\bar{\mathbf{f}}_i^*$ with $i = 1, \dots, N$ has been derived by Fisher and Carpenter [33]. Here, they set

$$\bar{\mathbf{f}}_i^* = \sum_{m=i}^N \sum_{l=0}^{i-1} 2\mathbf{Q}_{lm} \mathbf{f}_{EC}^*(\mathbf{u}_l, \mathbf{u}_m). \quad (4.47)$$

The equation (4.47) is a sufficient for an entropy stable/conservative scheme as shown in Lemma 4.

Lemma 4. *Considering (4.47), then*

$$\frac{1}{\omega_i} (\bar{\mathbf{f}}_{i+1}^* - \bar{\mathbf{f}}_i^*) = \sum_{j=0}^N 2\mathbf{D}_{ij} \mathbf{f}_{EC}^*(\mathbf{u}_i, \mathbf{u}_j), \quad (4.48)$$

for $i = 0, \dots, N$ is a degree p approximation of $\frac{\partial \mathcal{F}(\mathbf{u}_i)}{\partial x}$, meaning

$$\sum_{j=0}^N 2\mathbf{D}_{ij} \mathbf{f}_{EC}^*(\mathbf{u}_i, \mathbf{u}_j) = \frac{\partial \mathcal{F}(\mathbf{u}_i)}{\partial x} + \mathcal{O}(\Delta x^{p+1}). \quad (4.49)$$

Additionally, (4.47) satisfies

$$\sum_{i=0}^N \omega_i \mathbf{v}_i^T \left(\sum_{j=0}^N 2\mathbf{D}_{ij} \mathbf{f}_{EC}^*(\mathbf{u}_i, \mathbf{u}_j) \right) = f_N^{ent} - f_0^{ent}, \quad (4.50)$$

for which discretization (4.40) is entropy stable/conservative.

Proof. The proof can be found in [33]. □

Adapting the results from Lemma 4 we obtain the primary conservative, entropy stable/conservative and high-order Flux Difference discretization

$$\frac{\partial \mathbf{u}_i}{\partial t} + \sum_{k=0}^N 2\mathbf{D}_{ik} \mathbf{f}_{EC}^*(\mathbf{u}_i, \mathbf{u}_k) = \frac{\delta_{iN}}{\omega_i} [\mathbf{f} - \mathbf{f}^*]_N - \frac{\delta_{i0}}{\omega_i} [\mathbf{f} - \mathbf{f}^*]_0. \quad (4.51)$$

for $i = 0, \dots, N$. Remarkably, the Flux Difference discretization does not directly imply a skew-symmetric discretizations based on split forms. Therefore, the discretization holds for a general conservation law with an entropy function \mathcal{S} , assuming that a finite volume numerical flux \mathbf{f}_{EC}^* exists which satisfies the Tadmor Shuffle condition (4.34).

Connection to Split Forms

In the previous sections we introduced the Flux Difference discretization derived from a finite volume background. Remarkably, the discretization in certain cases also recovers a split form like the skew-symmetric discretization in section 4.1.1. We demonstrate this by focusing first on the Burgers' equation. Here, an entropy conservative flux which satisfies the Tadmor Shuffle condition (4.34) is given in (4.25). Analyzing the volume contribution of (4.51) we get

$$\begin{aligned} \sum_{k=0}^N 2D_{ik} \mathbf{f}_{EC}^*(\mathbf{u}_i, \mathbf{u}_k) &= \frac{1}{3} \sum_{k=0}^N D_{ik} (\mathbf{u}_k^2 + \mathbf{u}_i \mathbf{u}_k + \mathbf{u}_i^2), \\ &= \frac{1}{3} \left(\sum_{k=0}^N D_{ik} \mathbf{u}_k^2 + \mathbf{u}_i \sum_{k=0}^N D_{ik} \mathbf{u}_k + \mathbf{u}_i^2 \sum_{k=0}^N D_{ik} \right). \end{aligned} \quad (4.52)$$

The differentiation matrix is of degree $p > 0$, so it holds $\sum_{k=0}^N D_{ik} = 0$ and thus

$$\sum_{k=0}^N 2D_{ik} \mathbf{f}_{EC}^*(\mathbf{u}_i, \mathbf{u}_k) = \frac{1}{3} \sum_{k=0}^N D_{ik} \mathbf{u}_k^2 + \frac{1}{3} \mathbf{u}_i \sum_{k=0}^N D_{ik} \mathbf{u}_k. \quad (4.53)$$

Note that the right hand side is equivalent to the volume contribution of the skew-symmetric discretization in (4.24). Therefore, the Flux Difference and the skew-symmetric discretization are equivalent. So, the Flux Difference discretization can be interpreted as a numerical approximation of the split form

$$\sum_{k=0}^N 2D_{ik} \mathbf{f}_{EC}^*(\mathbf{u}_i, \mathbf{u}_k) \approx \frac{2}{3} \left(\frac{1}{2} \frac{\partial \mathcal{U}^2}{\partial x} \right) + \frac{1}{3} \left(\mathcal{U} \frac{\partial \mathcal{U}}{\partial x} \right). \quad (4.54)$$

This result does not only hold for the Burgers' equation, but also for other systems of conservation laws, as e.g. for the Shallow Water equations with no bottom topography (4.2). Here, a numerical flux satisfying the Tadmor Shuffle condition (4.34) is given in [47] and is defined by

$$\mathbf{f}_{EC}^* = \left(\begin{array}{c} \{h\}\{\nu\} \\ \{h\}\{\nu\}^2 + \frac{1}{2}g\{h^2\} \end{array} \right), \quad (4.55)$$

where $\{\cdot\}$ defines the mean of the two states, e.g. $\{h\} := \frac{1}{2}(h^L + h^R)$. With this numerical flux the Flux Difference scheme has the following volume contribution

$$\begin{aligned} & \sum_{k=0}^N 2D_{ik} \mathbf{f}_{EC}^*(\mathbf{u}_i, \mathbf{u}_k), \\ &= \sum_{k=0}^N D_{ik} \left(\frac{1}{4} h_k \nu_k^2 + \frac{1}{2} h_i \nu_i \nu_k + \frac{1}{2} \nu_i h_k \nu_k + \frac{1}{4} h_i \nu_k^2 + \frac{1}{4} \nu_i^2 h_k + \frac{1}{2} g h_k^2 \right), \end{aligned} \quad (4.56)$$

which is a discrete approximation of the split form

$$\begin{aligned} & \sum_{k=0}^N 2D_{ik} \mathbf{f}_{EC}^*(\mathbf{u}_i, \mathbf{u}_k) \\ & \approx \left(\frac{1}{2} \frac{\partial h \nu}{\partial x} + \frac{1}{2} \left(h \frac{\partial \nu}{\partial x} + \nu \frac{\partial h}{\partial x} \right) \right. \\ & \quad \left. + \frac{1}{4} \frac{\partial h \nu^2}{\partial x} + \frac{1}{2} \left(h \nu \frac{\partial \nu}{\partial x} + \nu \frac{\partial h \nu}{\partial x} \right) + \frac{1}{4} \left(h \frac{\partial \nu^2}{\partial x} + \nu^2 \frac{\partial h}{\partial x} \right) + \frac{1}{2} \frac{\partial h^2}{\partial x} \right), \end{aligned} \quad (4.57)$$

where the right hand side is equivalent to the physical flux of (4.2) due to the product rule.

Considering different numerical fluxes based on the mean $\{\cdot\}$ and its products one can recast different split forms into the Flux Difference framework. A list of such numerical fluxes with their corresponding approximated split forms is provided for the compressible Euler equations in [46]. If it is possible to always recast a split form from a numerical flux, e.g. for the compressible Euler equations using the Ismail-Roe flux [62], is still an open question. Due to the work of Tadmor [112] we know that a skew-symmetric form exists. However, we do not need the explicit skew-symmetric form to obtain an entropy stable scheme.

4.2. Entropy Stable Summation-by-Parts Methods for Two Dimensional Hyperbolic Systems of Conservation Laws

In this section we extend the Flux Difference discretization (4.51) to two dimensional conservation laws. A two dimensional system of hyperbolic conservation laws with N_{Eq} equations is given by

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathcal{F}}{\partial x} + \frac{\partial \mathcal{G}}{\partial y} = 0, \quad (4.58)$$

where $\mathbf{U}, \mathcal{F}, \mathcal{G} \in \mathbb{R}^{N_{Eq}}$, $t \in [0, T] \subset \mathbb{R}$ and $\Omega \subset \mathbb{R}^2$. As for the one dimensional case we assume that (4.58) is equipped with periodic boundary conditions. To derive

the discretization, we subdivide Ω in N_Q non-overlapping quadrilateral elements and focus on the transformed conservation law in Section 2.2 (Equation (2.121)) on a single element E_q . Considering a mapping $\chi : \hat{\Omega} \rightarrow E$ with $(x, y)^T = \chi(\xi, \eta)$, where $\hat{\Omega} = [-1, +1] \times [-1, +1]$ describes the reference space, then the transformed conservation law on $\hat{\Omega}$ is defined by [67]

$$J \frac{\partial \mathcal{U}}{\partial t} + \frac{\partial \tilde{\mathcal{F}}}{\partial \xi} + \frac{\partial \tilde{\mathcal{G}}}{\partial \eta} = 0, \quad (4.59)$$

with $J = \frac{\partial y}{\partial \eta} \frac{\partial x}{\partial \xi} - \frac{\partial x}{\partial \eta} \frac{\partial y}{\partial \xi}$ and the contravariant fluxes

$$\begin{aligned} \tilde{\mathcal{F}} &:= \mathcal{F} \frac{\partial y}{\partial \eta} - \mathcal{G} \frac{\partial x}{\partial \eta}, \\ \tilde{\mathcal{G}} &:= \mathcal{G} \frac{\partial x}{\partial \xi} - \mathcal{F} \frac{\partial y}{\partial \xi}. \end{aligned} \quad (4.60)$$

Next, we discuss the discretization of (4.59). As in Section 4.1.2 we are interested in primary conservative and entropy stable schemes. The definition of primary conservation and entropy stability for two dimensional hyperbolic systems is given in Definitions 13 and 14.

Definition 13. Primary Conservative Schemes for Two Dimensional Hyperbolic Systems of Conservation Laws

A semi-discrete method approximating two dimensional hyperbolic systems of conservation laws is said to be primary conservative if

$$\sum_{q=1}^{N_Q} \sum_{i,j=0}^N J_{ij} \omega_i \omega_j \frac{\partial \mathbf{u}_{ij}}{\partial t} = \mathbf{0}, \quad (4.61)$$

when considering periodic boundary conditions.

Definition 14. Entropy Stable Schemes for Two Dimensional Hyperbolic Systems of Conservation Laws

A semi-discrete method approximating two dimensional hyperbolic systems of conservation laws is said to be entropy stable if

$$\sum_{q=1}^{N_Q} \sum_{i,j=0}^N J_{ij} \omega_i \omega_j \frac{\partial S_{ij}}{\partial t} \leq 0, \quad (4.62)$$

when considering periodic boundary conditions.

The transformed conservation law (4.59) can be solved on a single element using a standard SBP discretization as presented in Section 2.2, which is

$$\begin{aligned} J_{ij}\omega_i\omega_j\frac{\partial\mathbf{u}_{ij}}{\partial t} + \omega_j \left([\tilde{\mathbf{f}}_{Nj}^* - \tilde{\mathbf{f}}_{Nj}] \delta_{iN} - [\tilde{\mathbf{f}}_{0j}^* - \tilde{\mathbf{f}}_{0j}] \delta_{i0} + \sum_{k=0}^N Q_{ik} \tilde{\mathbf{f}}_{kj} \right) \\ + \omega_i \left([\tilde{\mathbf{g}}_{iN}^* - \tilde{\mathbf{g}}_{iN}] \delta_{Nj} - [\tilde{\mathbf{g}}_{i0}^* - \tilde{\mathbf{g}}_{i0}] \delta_{0j} + \sum_{l=0}^N Q_{jl} \tilde{\mathbf{g}}_{il} \right) = 0, \end{aligned} \quad (4.63)$$

for $i, j = 0, \dots, N$. As for the one dimensional case the discretization (4.63) is not entropy stable due to non-linear terms. Therefore, we use the tensor product and obtain the two dimensional Flux Difference discretization

$$\begin{aligned} J_{ij}\omega_i\omega_j\frac{\partial\mathbf{u}_{ij}}{\partial t} + \omega_j \left([\tilde{\mathbf{f}}_{Nj}^* - \tilde{\mathbf{f}}_{Nj}] \delta_{iN} - [\tilde{\mathbf{f}}_{0j}^* - \tilde{\mathbf{f}}_{0j}] \delta_{i0} + 2 \sum_{k=0}^N Q_{ik} \tilde{\mathbf{f}}^\#(\mathbf{u}_{kj}, \mathbf{u}_{ij}) \right) \\ + \omega_i \left([\tilde{\mathbf{g}}_{iN}^* - \tilde{\mathbf{g}}_{iN}] \delta_{Nj} - [\tilde{\mathbf{g}}_{i0}^* - \tilde{\mathbf{g}}_{i0}] \delta_{0j} + 2 \sum_{l=0}^N Q_{jl} \tilde{\mathbf{g}}^\#(\mathbf{u}_{il}, \mathbf{u}_{ij}) \right) = 0, \end{aligned} \quad (4.64)$$

for $i, j = 0, \dots, N$. Here, we present the Flux Difference discretization in a general form on a quadrilateral element. The numerical fluxes $\tilde{\mathbf{f}}^\#, \tilde{\mathbf{g}}^\#$ are denoted as numerical *volume* fluxes whereas $\tilde{\mathbf{f}}^*, \tilde{\mathbf{g}}^*$ are the numerical *surface* fluxes. The numerical volume and surface fluxes are both symmetric and consistent ($\tilde{\mathbf{f}}^\#(a, b) = \tilde{\mathbf{f}}^\#(b, a)$ and $\tilde{\mathbf{f}}^\#(a, a) = \tilde{\mathbf{f}}(a)$ for $a, b \in \mathbb{R}$). Choosing different volume fluxes $\tilde{\mathbf{f}}^\#, \tilde{\mathbf{g}}^\#$ one can reconstruct different split forms [46]. For the purpose constructing entropy stable/conservative schemes we choose $\tilde{\mathbf{f}}^\# = \tilde{\mathbf{f}}_{EC}^*$ and $\tilde{\mathbf{g}}^\# = \tilde{\mathbf{g}}_{EC}^*$ such that the volume fluxes satisfy the Tadmor Shuffle conditions (4.34)

$$\begin{aligned} (\mathbf{v}_i - \mathbf{v}_j)^T \tilde{\mathbf{f}}^\#(\mathbf{u}_i, \mathbf{u}_j) &= \tilde{\Psi}_i^f - \tilde{\Psi}_j^f, \\ (\mathbf{v}_i - \mathbf{v}_j)^T \tilde{\mathbf{g}}^\#(\mathbf{u}_i, \mathbf{u}_j) &= \tilde{\Psi}_i^g - \tilde{\Psi}_j^g, \end{aligned} \quad (4.65)$$

with

$$\tilde{\Psi}^f := \Psi^f \frac{\partial y}{\partial \eta} - \Psi^g \frac{\partial x}{\partial \eta}, \quad \tilde{\Psi}^g := \Psi^g \frac{\partial x}{\partial \xi} - \Psi^f \frac{\partial y}{\partial \xi}. \quad (4.66)$$

The choice of the numerical surface fluxes $\tilde{\mathbf{f}}^*$ and $\tilde{\mathbf{g}}^*$ will be discussed later. When focusing on a Cartesian mesh the contravariant fluxes and potentials reduce to the physical fluxes and potentials scaled by the width and height respectively

$$\begin{aligned} \tilde{\mathbf{f}} &= \frac{\Delta y}{2} \mathbf{f}, & \tilde{\Psi}^f &= \frac{\Delta y}{2} \Psi^f, \\ \tilde{\mathbf{g}} &= \frac{\Delta x}{2} \mathbf{g}, & \tilde{\Psi}^g &= \frac{\Delta x}{2} \Psi^g. \end{aligned} \quad (4.67)$$

Therefore, we denote contravariant fluxes and potentials on Cartesian elements as *scaled fluxes* and *scaled potentials*. Based on a Cartesian mesh we prove that (4.64) is a primary conservative and entropy stable/conservative discretization. To do so, we first derive the discrete growth in the primary quantity and entropy on a single element in Lemma 5.

Lemma 5. *The discrete growths of the primary quantities and entropy on a single element of discretization (4.64) are*

$$\sum_{i,j=0}^N J_{ij} \omega_i \omega_j \frac{\partial \mathbf{u}_{ij}}{\partial t} = - \sum_{j=0}^N \omega_j \left(\tilde{\mathbf{f}}_{Nj} - \tilde{\mathbf{f}}_{0j} \right) - \sum_{i=0}^N \omega_i \left(\tilde{\mathbf{g}}_{iN} - \tilde{\mathbf{g}}_{i0} \right). \quad (4.68)$$

and

$$\begin{aligned} \sum_{i,j=0}^N J_{ij} \omega_i \omega_j \frac{\partial S_{ij}}{\partial t} &= - \sum_{j=0}^N \omega_j \left(\left[\mathbf{v}^T \tilde{\mathbf{f}}^* - \tilde{\Psi}^f \right]_{Nj} - \left[\mathbf{v}^T \tilde{\mathbf{f}}^* - \tilde{\Psi}^f \right]_{0j} \right) \\ &\quad - \sum_{i=0}^N \omega_i \left(\left[\mathbf{v}^T \tilde{\mathbf{g}}^* - \tilde{\Psi}^g \right]_{iN} - \left[\mathbf{v}^T \tilde{\mathbf{g}}^* - \tilde{\Psi}^g \right]_{i0} \right). \end{aligned} \quad (4.69)$$

Proof. We first prove (4.68). Therefore, we sum over all nodes of the Flux Difference discretization (4.64) for $i, j = 0, \dots, N$.

$$\begin{aligned} &\sum_{i,j=0}^N J_{ij} \omega_i \omega_j \frac{\partial \mathbf{u}_{ij}}{\partial t} \\ &= - \sum_{i,j=0}^N \omega_j \left(\left[\tilde{\mathbf{f}}_{Nj}^* - \tilde{\mathbf{f}}_{Nj} \right] \delta_{iN} - \left[\tilde{\mathbf{f}}_{0j}^* - \tilde{\mathbf{f}}_{0j} \right] \delta_{i0} + 2 \sum_{k=0}^N \mathbf{Q}_{ik} \tilde{\mathbf{f}}^\#(\mathbf{u}_{kj}, \mathbf{u}_{ij}) \right) \\ &\quad - \sum_{i,j=0}^N \omega_i \left(\left[\tilde{\mathbf{g}}_{iN}^* - \tilde{\mathbf{g}}_{iN} \right] \delta_{Nj} - \left[\tilde{\mathbf{g}}_{i0}^* - \tilde{\mathbf{g}}_{i0} \right] \delta_{0j} + 2 \sum_{l=0}^N \mathbf{Q}_{jl} \tilde{\mathbf{g}}^\#(\mathbf{u}_{il}, \mathbf{u}_{ij}) \right). \end{aligned} \quad (4.70)$$

Rearranging the sums gives

$$\begin{aligned} &\sum_{i,j=0}^N J_{ij} \omega_i \omega_j \frac{\partial \mathbf{u}_{ij}}{\partial t} \\ &= - \sum_{j=0}^N \omega_j \left(\left[\tilde{\mathbf{f}}_{Nj}^* - \tilde{\mathbf{f}}_{Nj} \right] - \left[\tilde{\mathbf{f}}_{0j}^* - \tilde{\mathbf{f}}_{0j} \right] + 2 \sum_{i=0}^N \sum_{k=0}^N \mathbf{Q}_{ik} \tilde{\mathbf{f}}^\#(\mathbf{u}_{kj}, \mathbf{u}_{ij}) \right) \\ &\quad - \sum_{i=0}^N \omega_i \left(\left[\tilde{\mathbf{g}}_{iN}^* - \tilde{\mathbf{g}}_{iN} \right] - \left[\tilde{\mathbf{g}}_{i0}^* - \tilde{\mathbf{g}}_{i0} \right] + 2 \sum_{j=0}^N \sum_{l=0}^N \mathbf{Q}_{jl} \tilde{\mathbf{g}}^\#(\mathbf{u}_{il}, \mathbf{u}_{ij}) \right). \end{aligned} \quad (4.71)$$

Due to the SBP property (2.64) we know that $2\mathbf{Q} = \mathbf{Q} + \mathbf{Q}^T + \mathbf{B}$ and thus

$$\begin{aligned}
 & 2 \sum_{i=0}^N \sum_{k=0}^N \mathbf{Q}_{ik} \tilde{\mathbf{f}}^\#(\mathbf{u}_{kj}, \mathbf{u}_{ij}) \\
 = & \sum_{i=0}^N \sum_{k=0}^N \mathbf{Q}_{ik} \tilde{\mathbf{f}}^\#(\mathbf{u}_{kj}, \mathbf{u}_{ij}) - \sum_{i=0}^N \sum_{k=0}^N \mathbf{Q}_{ki} \tilde{\mathbf{f}}^\#(\mathbf{u}_{kj}, \mathbf{u}_{ij}) \\
 & + \sum_{i=0}^N \sum_{k=0}^N \mathbf{B}_{ik} \tilde{\mathbf{f}}^\#(\mathbf{u}_{kj}, \mathbf{u}_{ij}),
 \end{aligned} \tag{4.72}$$

As $\tilde{\mathbf{f}}^\#$ is symmetric and consistent and due to $\mathbf{B} = \text{diag}(-1, 0, \dots, 0, +1)$ we obtain

$$\begin{aligned}
 & 2 \sum_{i=0}^N \sum_{k=0}^N \mathbf{Q}_{ik} \tilde{\mathbf{f}}^\#(\mathbf{u}_{kj}, \mathbf{u}_{ij}) \\
 = & \sum_{i=0}^N \sum_{k=0}^N \mathbf{Q}_{ik} \tilde{\mathbf{f}}^\#(\mathbf{u}_{kj}, \mathbf{u}_{ij}) - \sum_{i=0}^N \sum_{k=0}^N \mathbf{Q}_{ik} \tilde{\mathbf{f}}^\#(\mathbf{u}_{kj}, \mathbf{u}_{ij}) \\
 & + \tilde{\mathbf{f}}^\#(\mathbf{u}_{Nj}, \mathbf{u}_{Nj}) - \tilde{\mathbf{f}}^\#(\mathbf{u}_{0j}, \mathbf{u}_{0j}), \\
 = & \tilde{\mathbf{f}}(\mathbf{u}_{Nj}) - \tilde{\mathbf{f}}(\mathbf{u}_{0j}) = \tilde{\mathbf{f}}_{Nj} - \tilde{\mathbf{f}}_{0j},
 \end{aligned} \tag{4.73}$$

and with an analogous derivation for the volume integral with respect to $\tilde{\mathbf{g}}^\#$ we find

$$2 \sum_{l=0}^N \mathbf{Q}_{jl} \tilde{\mathbf{g}}^\#(\mathbf{u}_{il}, \mathbf{u}_{ij}) = \tilde{\mathbf{g}}_{iN} - \tilde{\mathbf{g}}_{i0}. \tag{4.74}$$

In total this gives

$$\begin{aligned}
 \sum_{i,j=0}^N J_{ij} \omega_i \omega_j \frac{\partial \mathbf{u}_{ij}}{\partial t} = & - \sum_{j=0}^N \omega_j \left([\tilde{\mathbf{f}}_{Nj}^* - \tilde{\mathbf{f}}_{Nj}] - [\tilde{\mathbf{f}}_{0j}^* - \tilde{\mathbf{f}}_{0j}] + \tilde{\mathbf{f}}_{Nj} - \tilde{\mathbf{f}}_{0j} \right) \\
 & - \sum_{i=0}^N \omega_i \left([\tilde{\mathbf{g}}_{iN}^* - \tilde{\mathbf{g}}_{iN}] - [\tilde{\mathbf{g}}_{i0}^* - \tilde{\mathbf{g}}_{i0}] + \tilde{\mathbf{g}}_{iN} - \tilde{\mathbf{g}}_{i0} \right),
 \end{aligned} \tag{4.75}$$

and thus we obtain (4.68)

$$\sum_{i,j=0}^N \omega_i \omega_j J_{ij} \frac{\partial \mathbf{u}_{ij}}{\partial t} = - \sum_{j=0}^N \omega_j \left(\tilde{\mathbf{f}}_{Nj}^* - \tilde{\mathbf{f}}_{0j}^* \right) - \sum_{i=0}^N \omega_i \left(\tilde{\mathbf{g}}_{iN}^* - \tilde{\mathbf{g}}_{i0}^* \right). \tag{4.76}$$

Next, we derive the discrete growth in entropy on an element and show that the growth reduces to evaluations at the element surface due to the SBP property

(2.64) and the symmetric, consistent fluxes $\tilde{\mathbf{f}}^\#$ and $\tilde{\mathbf{g}}^\#$. Therefore, we contract the Flux Difference discretization (4.64) with the entropy variables \mathbf{v}_{ij} as in [34, 45] and sum over all nodes

$$\begin{aligned}
 & \sum_{i,j=0}^N J_{ij} \omega_i \omega_j \mathbf{v}_{ij}^T \frac{\partial \mathbf{u}_{ij}}{\partial t}, \\
 &= - \sum_{i,j=0}^N \omega_j \left([\mathbf{v}^T \tilde{\mathbf{f}}^* - \mathbf{v}^T \tilde{\mathbf{f}}]_{Nj} \delta_{iN} - [\mathbf{v}^T \tilde{\mathbf{f}}^* - \mathbf{v}^T \tilde{\mathbf{f}}]_{0j} \delta_{i0} \right) \\
 & \quad - \sum_{i,j=0}^N \omega_j \left(2 \sum_{k=0}^N Q_{ik} \mathbf{v}_{ij}^T \tilde{\mathbf{f}}^\#(\mathbf{u}_{kj}, \mathbf{u}_{ij}) \right), \\
 & \quad - \sum_{i,j=0}^N \omega_i \left([\mathbf{v}^T \tilde{\mathbf{g}}^* - \mathbf{v}^T \tilde{\mathbf{g}}]_{iN} \delta_{Nj} - [\mathbf{v}^T \tilde{\mathbf{g}}^* - \mathbf{v}^T \tilde{\mathbf{g}}]_{i0} \delta_{0j} \right) \\
 & \quad - \sum_{i,j=0}^N \omega_i \left(2 \sum_{l=0}^N Q_{jl} \mathbf{v}_{ij}^T \tilde{\mathbf{g}}^\#(\mathbf{u}_{il}, \mathbf{u}_{ij}) \right).
 \end{aligned} \tag{4.77}$$

As we focus on semi-discrete schemes we assume continuity in time. Due to the definition of the entropy function it holds $\frac{\partial S_{ij}}{\partial t} = \mathbf{v}_{ij}^T \frac{\partial \mathbf{u}_{ij}}{\partial t}$. Rearranging the sums we obtain

$$\begin{aligned}
 & \sum_{i,j=0}^N J_{ij} \omega_i \omega_j \frac{\partial S_{ij}}{\partial t}, \\
 &= - \sum_{j=0}^N \omega_j \left([\mathbf{v}^T \tilde{\mathbf{f}}^* - \mathbf{v}^T \tilde{\mathbf{f}}]_{Nj} - [\mathbf{v}^T \tilde{\mathbf{f}}^* - \mathbf{v}^T \tilde{\mathbf{f}}]_{0j} + 2 \sum_{i=0}^N \sum_{k=0}^N Q_{ik} \mathbf{v}_{ij}^T \tilde{\mathbf{f}}^\#(\mathbf{u}_{kj}, \mathbf{u}_{ij}) \right) \\
 & \quad - \sum_{i=0}^N \omega_i \left([\mathbf{v}^T \tilde{\mathbf{g}}^* - \mathbf{v}^T \tilde{\mathbf{g}}]_{iN} - [\mathbf{v}^T \tilde{\mathbf{g}}^* - \mathbf{v}^T \tilde{\mathbf{g}}]_{i0} + 2 \sum_{j=0}^N \sum_{l=0}^N Q_{jl} \mathbf{v}_{ij}^T \tilde{\mathbf{g}}^\#(\mathbf{u}_{il}, \mathbf{u}_{ij}) \right),
 \end{aligned} \tag{4.78}$$

Focusing on the volume contribution with respect to $\tilde{\mathbf{f}}^\#$ and using the SBP property (2.64) we get

$$\begin{aligned}
 & 2 \sum_{i=0}^N \sum_{k=0}^N Q_{ik} \mathbf{v}_{ij}^T \tilde{\mathbf{f}}^\#(\mathbf{u}_{kj}, \mathbf{u}_{ij}) \\
 &= \sum_{i=0}^N \sum_{k=0}^N Q_{ik} \mathbf{v}_{ij}^T \tilde{\mathbf{f}}^\#(\mathbf{u}_{kj}, \mathbf{u}_{ij}) - \sum_{i=0}^N \sum_{k=0}^N Q_{ki} \mathbf{v}_{ij}^T \tilde{\mathbf{f}}^\#(\mathbf{u}_{kj}, \mathbf{u}_{ij}) \\
 & \quad + \sum_{i=0}^N \sum_{k=0}^N B_{ik} \mathbf{v}_{ij}^T \tilde{\mathbf{f}}^\#(\mathbf{u}_{kj}, \mathbf{u}_{ij}).
 \end{aligned} \tag{4.79}$$

Due to the symmetry and consistency of $\tilde{\mathbf{f}}^\#$ and the definition of the boundary matrix \mathbf{B} we obtain

$$\begin{aligned}
 & 2 \sum_{i=0}^N \sum_{k=0}^N \mathbf{Q}_{ik} \mathbf{v}_{ij}^T \tilde{\mathbf{f}}^\#(\mathbf{u}_{kj}, \mathbf{u}_{ij}) \\
 &= \sum_{i=0}^N \sum_{k=0}^N \mathbf{Q}_{ik} \mathbf{v}_{ij}^T \tilde{\mathbf{f}}^\#(\mathbf{u}_{kj}, \mathbf{u}_{ij}) - \sum_{i=0}^N \sum_{k=0}^N \mathbf{Q}_{ij} \mathbf{v}_{kj}^T \tilde{\mathbf{f}}^\#(\mathbf{u}_{kj}, \mathbf{u}_{ij}) \\
 &\quad + \mathbf{v}_{Nj}^T \tilde{\mathbf{f}}(\mathbf{u}_{Nj}) - \mathbf{v}_{0j}^T \tilde{\mathbf{f}}(\mathbf{u}_{0j}), \\
 &= \sum_{i=0}^N \sum_{k=0}^N \mathbf{Q}_{ik} (\mathbf{v}_{ij} - \mathbf{v}_{kj})^T \tilde{\mathbf{f}}^\#(\mathbf{u}_{kj}, \mathbf{u}_{ij}) + \mathbf{v}_{Nj}^T \tilde{\mathbf{f}}_{Nj} - \mathbf{v}_{0j}^T \tilde{\mathbf{f}}_{0j}.
 \end{aligned} \tag{4.80}$$

As $\tilde{\mathbf{f}}^\#$ satisfies the Tadmor Shuffle condition (4.34) we arrive at

$$\begin{aligned}
 & 2 \sum_{i=0}^N \sum_{k=0}^N \mathbf{Q}_{ik} \mathbf{v}_{ij}^T \tilde{\mathbf{f}}^\#(\mathbf{u}_{kj}, \mathbf{u}_{ij}) \\
 &= \sum_{i=0}^N \sum_{k=0}^N \mathbf{Q}_{ik} (\tilde{\Psi}_{ij}^f - \tilde{\Psi}_{kj}^f) + \mathbf{v}_{Nj}^T \tilde{\mathbf{f}}_{Nj} - \mathbf{v}_{0j}^T \tilde{\mathbf{f}}_{0j}, \\
 &= \sum_{i=0}^N \left(\tilde{\Psi}_{ij}^f \sum_{k=0}^N \mathbf{Q}_{ik} - \sum_{k=0}^N \mathbf{Q}_{ik} \tilde{\Psi}_{kj}^f \right) + \mathbf{v}_{Nj}^T \tilde{\mathbf{f}}_{Nj} - \mathbf{v}_{0j}^T \tilde{\mathbf{f}}_{0j}.
 \end{aligned} \tag{4.81}$$

Assuming that the differentiation matrix \mathbf{D} is of degree $p > 0$ (consistency of the differentiation matrix), then $\mathbf{D}\mathbf{1} = \mathbf{0} \Leftrightarrow \mathbf{Q}\mathbf{1} = \mathbf{0}$ and thus $\sum_{k=0}^N \mathbf{Q}_{ik} = \mathbf{0}$. Therefore (4.81) reduces to

$$2 \sum_{i=0}^N \sum_{k=0}^N \mathbf{Q}_{ik} \mathbf{v}_{ij}^T \tilde{\mathbf{f}}^\#(\mathbf{u}_{kj}, \mathbf{u}_{ij}) = - \sum_{i=0}^N \sum_{k=0}^N \mathbf{Q}_{ik} \tilde{\Psi}_{kj}^f + \mathbf{v}_{Nj}^T \tilde{\mathbf{f}}_{Nj} - \mathbf{v}_{0j}^T \tilde{\mathbf{f}}_{0j}. \tag{4.82}$$

Due to the SBP property we know $\mathbf{Q}_{ik} = \mathbf{B}_{ik} - \mathbf{Q}_{ki}$ which yields

$$\begin{aligned}
 2 \sum_{i=0}^N \sum_{k=0}^N \mathbf{Q}_{ik} \mathbf{v}_{ij}^T \tilde{\mathbf{f}}^\#(\mathbf{u}_{kj}, \mathbf{u}_{ij}) &= \sum_{k=0}^N \tilde{\Psi}_{kj}^f \sum_{i=0}^N \mathbf{Q}_{ki} - \sum_{i=0}^N \sum_{k=0}^N \mathbf{B}_{ik} \tilde{\Psi}_{kj}^f \\
 &\quad + \mathbf{v}_{Nj}^T \tilde{\mathbf{f}}_{Nj} - \mathbf{v}_{0j}^T \tilde{\mathbf{f}}_{0j}.
 \end{aligned} \tag{4.83}$$

Again, using the consistency of the differentiation matrix $\sum_{i=0}^N \mathbf{Q}_{ki} = \mathbf{0}$ and due to the definition of \mathbf{B} we find

$$2 \sum_{i=0}^N \sum_{k=0}^N \mathbf{Q}_{ik} \mathbf{v}_{ij}^T \tilde{\mathbf{f}}^\#(\mathbf{u}_{kj}, \mathbf{u}_{ij}) = -\tilde{\Psi}_{Nj}^f + \tilde{\Psi}_{0j}^f + \mathbf{v}_{Nj}^T \tilde{\mathbf{f}}_{Nj} - \mathbf{v}_{0j}^T \tilde{\mathbf{f}}_{0j}. \tag{4.84}$$

Similarly, doing the same derivations for the volume contribution with respect to $\tilde{\mathbf{g}}^\#$ we get

$$2 \sum_{j=0}^N \sum_{l=0}^N \mathbf{Q}_{jl} \mathbf{v}_{ij}^T \tilde{\mathbf{g}}^\#(\mathbf{u}_{il}, \mathbf{u}_{ij}) = -\tilde{\Psi}_{iN}^g + \tilde{\Psi}_{i0}^g + \mathbf{v}_{iN}^T \tilde{\mathbf{g}}_{iN} - \mathbf{v}_{i0}^T \tilde{\mathbf{f}}_{i0}. \quad (4.85)$$

In total, we obtain the following discrete growth in discrete entropy

$$\begin{aligned} & \sum_{i,j=0}^N J_{ij} \omega_i \omega_j \frac{\partial S_{ij}}{\partial t}, \\ &= - \sum_{j=0}^N \omega_j \left([\mathbf{v}^T \tilde{\mathbf{f}}^* - \mathbf{v}^T \tilde{\mathbf{f}}]_{Nj} - [\mathbf{v}^T \tilde{\mathbf{f}}^* - \mathbf{v}^T \tilde{\mathbf{f}}]_{0j} \right) \\ &+ \sum_{j=0}^N \omega_j \left(\tilde{\Psi}_{Nj}^f - \tilde{\Psi}_{0j}^f - \mathbf{v}_{Nj}^T \tilde{\mathbf{f}}_{Nj} + \mathbf{v}_{0j}^T \tilde{\mathbf{f}}_{0j} \right) \\ &- \sum_{i=0}^N \omega_i \left([\mathbf{v}^T \tilde{\mathbf{g}}^* - \mathbf{v}^T \tilde{\mathbf{g}}]_{iN} - [\mathbf{v}^T \tilde{\mathbf{g}}^* - \mathbf{v}^T \tilde{\mathbf{g}}]_{i0} \right) \\ &+ \sum_{i=0}^N \omega_i \left(\tilde{\Psi}_{iN}^g - \tilde{\Psi}_{i0}^g - \mathbf{v}_{iN}^T \tilde{\mathbf{g}}_{iN} + \mathbf{v}_{i0}^T \tilde{\mathbf{f}}_{i0} \right), \end{aligned} \quad (4.86)$$

which finally leads to (4.69)

$$\begin{aligned} \sum_{i,j=0}^N J_{ij} \omega_i \omega_j \frac{\partial S_{ij}}{\partial t} &= - \sum_{j=0}^N \omega_j \left([\mathbf{v}^T \tilde{\mathbf{f}}^* - \tilde{\Psi}^f]_{Nj} - [\mathbf{v}^T \tilde{\mathbf{f}}^* - \tilde{\Psi}^f]_{0j} \right) \\ &- \sum_{i=0}^N \omega_i \left([\mathbf{v}^T \tilde{\mathbf{g}}^* - \tilde{\Psi}^g]_{iN} - [\mathbf{v}^T \tilde{\mathbf{g}}^* - \tilde{\Psi}^g]_{i0} \right). \end{aligned} \quad (4.87)$$

□

In Lemma 5 we found that the growth in primary quantities and in entropy is reduced to evaluations at the element surface when considering a Cartesian mesh. The key of proving this result was using SBP operators and special numerical volume fluxes $\tilde{\mathbf{f}}^\# = \tilde{\mathbf{f}}_{EC}^*$, $\tilde{\mathbf{g}}^\# = \tilde{\mathbf{g}}_{EC}^*$ for the Flux Difference discretization. However, the proof of Lemma 5 is not valid for curvilinear elements as the non-constant metric terms introduce further non-linearities. Therefore, a modification of the discretization needs to be considered. The modified discretization is presented in Appendix A.2.

Focusing on Cartesian meshes and applying the results of Lemma 5 we prove primary conservation and entropy stability. Therefore, we need to specify the numerical surface fluxes $\tilde{\mathbf{f}}^*$. On a Cartesian mesh these are defined by $\tilde{\mathbf{f}}^* = \frac{\Delta y}{2} \mathbf{f}^*$ and $\tilde{\mathbf{g}}^* = \frac{\Delta x}{2} \mathbf{g}^*$. The entropy stable surface fluxes \mathbf{f}_{ES}^* and \mathbf{g}_{ES}^* are

$$\begin{aligned}\mathbf{f}_{ES}^*(\mathbf{u}^L, \mathbf{u}^R) &= \mathbf{f}_{EC}^*(\mathbf{u}^L, \mathbf{u}^R) - \frac{\lambda_f}{2} \mathbf{K} (\mathbf{v}^R - \mathbf{v}^L), \\ \mathbf{g}_{ES}^*(\mathbf{u}^L, \mathbf{u}^R) &= \mathbf{g}_{EC}^*(\mathbf{u}^L, \mathbf{u}^R) - \frac{\lambda_g}{2} \mathbf{K} (\mathbf{v}^R - \mathbf{v}^L).\end{aligned}\tag{4.88}$$

We interpret the fluxes (4.88) as a combination of an entropy conserving flux minus a dissipation term. As pointed out in Section 4.1.2 the numerical surface fluxes \mathbf{f}_{EC}^* and \mathbf{g}_{EC}^* are entropy conservative, which will be proved later in Theorem 8.

In order to dissipate the total discrete entropy we need to consider a dissipation term. This is indispensable for non-linear problem as these usually develop shocks in finite time even when considering smooth initial data, for which the total entropy should dissipate. Thus, we include an interface dissipation term in the sense of Lax-Friedrichs. Here, $\lambda_f > 0$ and $\lambda_g > 0$ denote the maximum absolute eigenvalue of the Jacobians of the physical fluxes \mathcal{F} and \mathcal{G} respectively to approximate the wave speed [116]. For a classical Lax-Friedrichs dissipation term one typically includes the jump in the conservative variables $\mathbf{u}^R - \mathbf{u}^L$. However, when considering the jump in the conservative variables the numerical surface fluxes are not necessarily entropy stable. Therefore, we introduce a positive definite dissipation matrix \mathbf{K} to approximate the jump in the entropy variables $\mathbf{v}^R - \mathbf{v}^L$

$$\mathbf{u}^R - \mathbf{u}^L \approx \mathbf{K} (\mathbf{v}^R - \mathbf{v}^L).\tag{4.89}$$

So \mathbf{K} approximates the dissipation term of a classical Lax-Friedrichs flux. The positive definite dissipation matrix \mathbf{K} depends on the corresponding system of conservation laws. Such matrices are given for numerous conservation laws, e.g. for the compressible Euler equations [46] or the Ideal Magnetohydrodynamics (MHD) equations [120].

With the numerical surface fluxes (4.88) we will prove in Theorem 8 that the two dimensional Flux Difference discretization (4.64) is primary conservative and entropy stable.

Theorem 8. *Considering a Cartesian mesh for the discretization (4.64), where $\mathbf{f}^\#, \mathbf{g}^\#$ satisfy the Tadmor Shuffle condition (4.34). Then the scheme is primary conservative and entropy stable when using the numerical surface fluxes (4.88).*

Proof. The proof is done in a similar fashion to Theorem 3. We first sum (4.68) and (4.69) over all elements to obtain the discrete growth of the primary quantities and entropy. Due to Lemma 5 the total discrete growths depends on the interfaces of the elements. We look at a vertical interior interface. The contribution for the horizontal interfaces is done in an analogous fashion. The terms referring to a vertical interior interfaces are

$$\mathbf{IU}_t := \sum_{j=0}^N \omega_j \tilde{\mathbf{f}}_{0j}^{*,R} - \sum_{j=0}^N \omega_j \tilde{\mathbf{f}}_{Nj}^{*,L}, \quad (4.90)$$

$$IS_t := \sum_{j=0}^N \omega_j \left[\mathbf{v}^{R,T} \tilde{\mathbf{f}}^{*,R} - \tilde{\Psi}^{f,R} \right]_{0j} - \sum_{j=0}^N \omega_j \left[\mathbf{v}^{L,T} \tilde{\mathbf{f}}^{*,L} - \tilde{\Psi}^{f,L} \right]_{Nj}, \quad (4.91)$$

where the superscripts L and R refer to the left and right element, respectively. Here, \mathbf{IU}_t and IS_t are an approximation the surface integral of $\frac{\partial \mathbf{u}}{\partial t}$ and $\frac{\partial \mathcal{S}}{\partial t}$ on a single interface. From now on we ignore the first index of the numerical surface, entropy variables and potentials as these remain constant throughout this proof.

$$\mathbf{IU}_t := \sum_{j=0}^N \omega_j \tilde{\mathbf{f}}_j^{*,R} - \sum_{j=0}^N \omega_j \tilde{\mathbf{f}}_j^{*,L}, \quad (4.92)$$

$$IS_t := \sum_{j=0}^N \omega_j \left[\mathbf{v}^{R,T} \tilde{\mathbf{f}}^{*,R} - \tilde{\Psi}^{f,R} \right]_j - \sum_{j=0}^N \omega_j \left[\mathbf{v}^{L,T} \tilde{\mathbf{f}}^{*,L} - \tilde{\Psi}^{f,L} \right]_j, \quad (4.93)$$

For a conforming mesh the numerical flux is unique at an interface, meaning $\tilde{\mathbf{f}}^* := \tilde{\mathbf{f}}^{*,L} = \tilde{\mathbf{f}}^{*,R}$ which directly gives us

$$\mathbf{IU}_t = \sum_{j=0}^N \omega_j \tilde{\mathbf{f}}_j^* - \sum_{j=0}^N \omega_j \tilde{\mathbf{f}}_j^* = \mathbf{0}, \quad (4.94)$$

which indicates primary conservation. Note, that the derivation holds independent of the choice of the numerical surface flux. For entropy stability however we need to consider the fluxes (4.88). We focus on (4.93) and include these fluxes

$$\begin{aligned} IS_t = & \frac{\Delta y}{2} \sum_{j=0}^N \omega_j \left(\mathbf{v}^{R,T} \left(\mathbf{f}_{EC}^* - \frac{\lambda_f}{2} \mathbf{K} (\mathbf{v}^R - \mathbf{v}^L) \right) - \Psi^{f,R} \right)_j \\ & - \frac{\Delta y}{2} \sum_{j=0}^N \omega_j \left(\mathbf{v}^{L,T} \left(\mathbf{f}_{EC}^* - \frac{\lambda_f}{2} \mathbf{K} (\mathbf{v}^R - \mathbf{v}^L) \right) - \Psi^{f,L} \right)_j. \end{aligned} \quad (4.95)$$

Rearranging terms gives us

$$\begin{aligned}
 IS_t = & \frac{\Delta y}{2} \sum_{j=0}^N \omega_j \left((\mathbf{v}^R - \mathbf{v}^L)^T \mathbf{f}_{EC}^* - (\Psi^{f,R} - \Psi^{f,L}) \right)_j \\
 & - \frac{\Delta y \lambda_f}{2} \sum_{j=0}^N \omega_j \left((\mathbf{v}^R - \mathbf{v}^L)^T \mathbf{K} (\mathbf{v}^R - \mathbf{v}^L) \right)_j.
 \end{aligned} \tag{4.96}$$

Due to the Tadmor Shuffle condition (4.34) the first sum vanishes and thus

$$IS_t = -\frac{\Delta y \lambda_f}{4} \sum_{j=0}^N \omega_j \left((\mathbf{v}^R - \mathbf{v}^L)^T \mathbf{K} (\mathbf{v}^R - \mathbf{v}^L) \right)_j. \tag{4.97}$$

The remaining term intervenes due to the dissipation term. When ignoring the dissipation term, then $IS_t = 0$ and thus we obtain an entropy conservative scheme. As we consider positive quadrature weights and \mathbf{K} is symmetric positive definite we arrive at

$$IS_t = -\underbrace{\frac{\Delta y \lambda_f}{4}}_{\geq 0} \sum_{j=0}^N \underbrace{\omega_j}_{> 0} \underbrace{\left((\mathbf{v}^R - \mathbf{v}^L)^T \mathbf{K} (\mathbf{v}^R - \mathbf{v}^L) \right)_j}_{\geq 0} \leq 0. \tag{4.98}$$

and thus indicates entropy stability. \square

Note, that this proof relied on Cartesian quadrilateral elements. However, the Flux Difference discretization can be applied to different geometries, e.g. triangular/tetrahedral elements [17, 20, 21].

To summarize, we extended the SBP method from linear to non-linear problems. For non-linear problems the standard SBP discretization is not entropy stable. Therefore, we modified the SBP method by the high-order Flux Difference discretization of Fisher and Carpenter [33]. Here, we introduced an additional numerical flux on the volume part of the SBP method to adapt derivations from finite volume schemes. As long as the numerical volume flux satisfies the Tadmor Shuffle condition (4.34) we can prove that the discrete growth of entropy is reduced to evaluations on the surface of the element. Finally, by choosing an appropriate numerical surface fluxes we obtain a provably primary conservative and entropy stable scheme for arbitrary SBP operators.

4.3. Numerical Verification of Summation-by-Parts Schemes for Non-Linear Problems

In this section we verify high-order convergence as well as primary conservation and entropy stability/conservation of the SBP Flux Difference discretization (4.64) for non-linear systems of conservation laws. For all numerical results presented in this section we consider the two dimensional compressible Euler equations

$$\begin{pmatrix} \rho \\ \rho v_1 \\ \rho v_2 \\ \mathcal{E} \end{pmatrix}_t + \begin{pmatrix} \rho v_1 \\ \rho v_1^2 + \mathcal{P} \\ \rho v_1 v_2 \\ v_1(\mathcal{E} + \mathcal{P}) \end{pmatrix}_x + \begin{pmatrix} \rho v_2 \\ \rho v_1 v_2 \\ \rho v_2^2 + \mathcal{P} \\ v_2(\mathcal{E} + \mathcal{P}) \end{pmatrix}_y = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad (4.99)$$

on $\Omega \subset \mathbb{R}^2$ and $t \in [0, T] \subset \mathbb{R}^+$ with $\mathcal{E} = \frac{1}{2}\rho(v_1^2 + v_2^2) + \frac{\mathcal{P}}{\gamma-1}$ and the adiabatic coefficient $\gamma = 1.4$. For the Euler equations the entropy function is defined by [38]

$$\mathcal{S} = -\frac{\rho}{\gamma-1} \log\left(\frac{\mathcal{P}}{\rho^\gamma}\right), \quad (4.100)$$

The entropy variables are given by

$$\mathbf{v} = \frac{\partial \mathcal{S}}{\partial \mathbf{u}} = \begin{pmatrix} \frac{\gamma-s}{\gamma-1} - \beta(v_1^2 + v_2^2) \\ 2\beta v_1 \\ 2\beta v_2 \\ -2\beta \end{pmatrix}, \quad (4.101)$$

with $\beta = \frac{\rho}{2\mathcal{P}}$ and $s = \log(\mathcal{P}) - \gamma \log(\rho)$. For the compressible Euler equations we consider the Ismail and Roe entropy conserving flux for \mathbf{f}_{EC}^* and \mathbf{g}_{EC}^* [62]. These fluxes have been constructed to satisfy the Tadmor Shuffle condition (4.34) and so they are used to obtain an entropy stable numerical scheme. We set the volume fluxes $\mathbf{f}^\# = \mathbf{f}_{EC}^*$ and $\mathbf{g}^\# = \mathbf{g}_{EC}^*$. The numerical surface fluxes are calculated by (4.88)

$$\begin{aligned} \mathbf{f}^*(\mathbf{u}^L, \mathbf{u}^R) &= \mathbf{f}_{EC}^*(\mathbf{u}^L, \mathbf{u}^R) - \sigma \frac{\lambda_f}{2} \mathbf{K}(\mathbf{v}^R - \mathbf{v}^L), \\ \mathbf{g}^*(\mathbf{u}^L, \mathbf{u}^R) &= \mathbf{g}_{EC}^*(\mathbf{u}^L, \mathbf{u}^R) - \sigma \frac{\lambda_g}{2} \mathbf{K}(\mathbf{v}^R - \mathbf{v}^L), \end{aligned} \quad (4.102)$$

where we included $\sigma \in [0, 1]$ to control the amount of dissipation. The dissipation matrix \mathbf{K} can be found in [46]. The maximum eigenvalue λ_f on an interface in

y -direction is given by

$$\begin{aligned}\lambda_{f,L} &= \max \left\{ \|\boldsymbol{\nu}_1^L + \mathbf{c}^L\|_\infty, \|\boldsymbol{\nu}_1^L\|_\infty, \|\boldsymbol{\nu}_1^L - \mathbf{c}^L\|_\infty \right\}, \\ \lambda_{f,R} &= \max \left\{ \|\boldsymbol{\nu}_1^R + \mathbf{c}^R\|_\infty, \|\boldsymbol{\nu}_1^R\|_\infty, \|\boldsymbol{\nu}_1^R - \mathbf{c}^R\|_\infty \right\}, \\ \lambda_f &= \max \{ \lambda_{f,L}; \lambda_{f,R} \},\end{aligned}\tag{4.103}$$

where the superscript L and R refer to left and right element connected through the vertical interface and $c = \sqrt{\frac{\gamma \mathcal{P}}{\rho}}$, see [38]. The maximum eigenvalue λ_g on an interface in x -direction is obtained in a similar fashion to (4.103) by replacing the velocity ν_1 by ν_2 . The choice of σ in (4.102) will vary in this section. In terms of verifying high-order convergence of the entropy stable scheme we set $\sigma = 1$. When verifying entropy conservation or stability we consider $\sigma = 0$ as well as $\sigma = 1$.

As in Section 2.3 and 3.5 we use the five-stage, fourth-order low-storage Runge-Kutta method of Carpenter and Kennedy [14]. The explicit time step Δt is selected by the CFL condition [38]

$$\Delta t := CFL \frac{\min_i \left\{ \frac{\Delta x_i}{2}, \frac{\Delta y_i}{2} \right\}}{(N+1)\lambda_{\max}},\tag{4.104}$$

where Δx_i and Δy_i denote the width in x - and y -direction of the i -th element, $N+1$ the number of nodes and λ_{\max} denotes the maximum eigenvalue of the flux Jacobians $\lambda_{\max} = \max\{\lambda_f, \lambda_g\}$ over every node in the whole domain. Note, that this CFL condition is equivalent to the CFL condition in Section 3.5 for the LAE as $\lambda_{\max} = \max\{a, b\}$.

In this section, we verify the EOC as well as conservation of the primary quantities and entropy for the SBP scheme. In addition, we numerically demonstrate the robustness of entropy stable schemes.

4.3.1. Comparison and Convergence of Summation-by-Parts Schemes

For the numerical convergence experiments we consider the isentropic vortex advection problem taken from [19]. Here, we set the domain to be $\Omega = [0, 10] \times [0, 10]$. The initial conditions are

$$\begin{pmatrix} \rho \\ \nu_1 \\ \nu_2 \\ \mathcal{P} \end{pmatrix} (0, x, y) = \begin{pmatrix} \mathcal{T}^{\frac{1}{\gamma-1}} \\ 1 - (y-5)\kappa(r) \\ 1 + (x-5)\kappa(r) \\ \mathcal{T}^{\frac{\gamma}{\gamma-1}} \end{pmatrix},\tag{4.105}$$

where

$$\begin{aligned} r(x, y) &= \sqrt{(x - 5)^2 + (y - 5)^2}, \\ \mathcal{T}(x, y) &= 1 - \frac{\gamma - 1}{2\gamma} \kappa(r^2), \\ \kappa(r) &= \varepsilon e^{\alpha(1-r^2)}, \end{aligned} \tag{4.106}$$

with $\varepsilon = \frac{5}{2\pi}$ and $\alpha = 0.5$. With these initial condition the vortex is advected along the diagonal of the domain. We impose Dirichlet boundary conditions using the exact solution which is

$$\mathbf{u}(t, x, y) = \mathbf{u}_0(x - t, y - t). \tag{4.107}$$

We run the simulation to $T = 1$ with $CFL = 1$. We consider SBP operators as DG, FD and HGTL with polynomial degrees $p = 2, 3$. As in Section 2.3.1 we focus on different mesh levels where we increase the number of elements in each level to obtain the *EOC*. We choose $N + 1 = 4(p + 1)$ nodes in one dimension and 2^{2d} elements for FD and HGTL operators in the d -th mesh level. As DG operators consist of $N + 1 = p + 1$ nodes in one dimension we choose 2^{2d+4} elements in the d -th mesh level. Therefore, we get $DOFS_d = (p + 1)^2 2^{2d+4}$ in the d -th level and thus we can compare the errors for the DG, FD and HGTL operators while $DOFS_d = (p + 1)^2 2^{2d+4}$ remains constant. For systems of conservation laws *DOFS* refers to the degrees of freedom per equation. Note, that we consider the two dimensional compressible Euler equations and thus we obtain a \mathcal{L}_2 error for all four conservative variables. In Table 4.1-4.2 we present the average \mathcal{L}_2 error in terms of readability.

As for the results produced for the LAE in Section 2.3.1 the FD operator has a better error than the DG operator for $p = 2$. When increasing the degree to $p = 3$ we see that the error of the DG operator is smaller than for FD. However, for $p = 2, 3$ the error of the HGTL operator is smaller than for FD and DG.

For this test case the HGTL operator behaved the best. However, all schemes have a convergence rate near $p + 1$ and therefore are high-order methods. This is the same result as verified with the standard SBP discretization (4.63) for the LAE on a conforming mesh in Section 2.3.1. Applying the same convergence test with the standard SBP discretization for the compressible Euler equations would also obtain a convergence rate of $p + 1$. However, such a scheme would not be entropy stable.

EOC for the SBP Flux Difference discretization with DG, FD and HGTL operators for $p = 2, 3$.

$DOFS$	\mathcal{L}_2DG	$EOCDG$	\mathcal{L}_2FD	$EOCFD$	\mathcal{L}_2HGTL	$EOCHGTL$
576	1.61E-01		1.10E-01		8.71E-02	
2304	3.51E-02	2.2	1.96E-02	2.5	1.41E-02	2.6
9216	6.03E-03	2.5	3.51E-03	2.5	2.42E-03	2.5
36864	1.07E-03	2.5	4.96E-04	2.8	3.38E-04	2.8
147456	2.04E-04	2.4	8.38E-05	2.6	5.74E-05	2.6
589824	3.45E-05	2.6	1.23E-05	2.8	8.24E-06	2.8

Table 4.1.: Setting $p = 2$, $T = 1$, $CFL = 1$ and the number of nodes per element in one dimension for FD and HGTL operators to $N + 1 = 12$.

$DOFS$	\mathcal{L}_2DG	$EOCDG$	\mathcal{L}_2FD	$EOCFD$	\mathcal{L}_2HGTL	$EOCHGTL$
1024	4.77E-02		5.45E-02		2.09E-02	
4096	4.72E-03	3.3	5.02E-03	3.4	1.56E-03	3.7
16384	4.13E-04	3.5	4.03E-04	3.6	8.80E-05	4.1
65536	2.58E-05	4.0	2.58E-05	4.0	5.85E-06	3.9
262144	1.58E-06	4.0	1.81E-06	3.8	4.12E-07	3.8
1048576	9.87E-08	4.0	1.38E-07	3.7	2.97E-08	3.8

Table 4.2.: Setting $p = 3$, $T = 1$, $CFL = 1$ and the number of nodes per element in one dimension for FD and HGTL operators to $N + 1 = 16$.

In the next section we numerically verify primary conservation as well as entropy stability/conservation. In addition, we demonstrate the increased robustness of entropy stable schemes by comparing a simulation with the SBP Flux Difference discretization (4.64) and the standard SBP discretization (4.63).

4.3.2. Numerical Verification of Primary Conservation and Entropy Stability

In this section we numerically verify primary conservation and entropy stability/conservation for the SBP Flux Difference scheme. We first demonstrate primary and entropy conservation which was the result of Theorem 8 when setting $\sigma = 0$, so the numerical surface flux reduces to the entropy conservative flux of Ismail and Roe [62] without dissipation. We consider a mesh with 10×10 elements and periodic boundary conditions. Each element is fitted with a HGTL operator of degree $p = 2$ and $N + 1 = 12$ nodes. However, the choice of the operator is arbitrary provided that the operator satisfies the SBP property. To calculate the

growth in the total discrete primary quantities and entropy we rewrite the Flux Difference discretization (4.64) by

$$J_{ij}\omega_i\omega_j\frac{\partial\mathbf{u}_{ij}}{\partial t} = \mathbf{Res}(\mathbf{u})_{ij}, \quad (4.108)$$

where

$$\begin{aligned} \mathbf{Res}(\mathbf{u})_{ij} = & -\omega_j \left([\tilde{\mathbf{f}}_{Nj}^* - \tilde{\mathbf{f}}_{Nj}] \delta_{iN} - [\tilde{\mathbf{f}}_{0j}^* - \tilde{\mathbf{f}}_{0j}] \delta_{i0} + 2 \sum_{k=0}^N \mathbf{Q}_{ik} \tilde{\mathbf{f}}^\#(\mathbf{u}_{kj}, \mathbf{u}_{ij}) \right) \\ & - \omega_i \left([\tilde{\mathbf{g}}_{iN}^* - \tilde{\mathbf{g}}_{iN}] \delta_{Nj} - [\tilde{\mathbf{g}}_{i0}^* - \tilde{\mathbf{g}}_{i0}] \delta_{0j} + 2 \sum_{l=0}^N \mathbf{Q}_{jl} \tilde{\mathbf{g}}^\#(\mathbf{u}_{il}, \mathbf{u}_{ij}) \right). \end{aligned} \quad (4.109)$$

The growth in the discrete entropy is computed by contracting (4.108) with the vector of entropy variables

$$\begin{aligned} J_{ij}\omega_i\omega_j\mathbf{v}_{ij}^T\frac{\partial\mathbf{u}_{ij}}{\partial t} &= \mathbf{v}_{ij}^T\mathbf{Res}(\mathbf{u})_{ij} \\ \Leftrightarrow J_{ij}\omega_i\omega_j\frac{\partial S_{ij}}{\partial t} &= \mathbf{v}_{ij}^T\mathbf{Res}(\mathbf{u})_{ij} \end{aligned} \quad (4.110)$$

as $\frac{\partial}{\partial t}S = \mathbf{v}^T\frac{\partial\mathbf{u}}{\partial t}$ (continuity in time) we obtain the temporal derivative of the entropy at each node. As shown in Theorem 8, the scheme is primary and entropy conservative when no interface dissipation is included, meaning that $\sigma = 0$. In addition, when assuming periodic boundary conditions the terms on the boundary of the domain vanish and thus we get

$$\begin{aligned} \sum_{q=1}^{N_Q} \left(\sum_{i,j=0}^N J_{ij}\omega_i\omega_j\frac{\partial\mathbf{u}_{ij}^q}{\partial t} \right) &= 0, \\ \sum_{q=1}^{N_Q} \left(\sum_{i,j=0}^N J_{ij}\omega_i\omega_j\frac{\partial S_{ij}^q}{\partial t} \right) &= 0, \end{aligned} \quad (4.111)$$

for all time. To verify this result numerically we insert (4.108) and (4.110) and calculate

$$\begin{aligned} \mathbf{U}_t^{tot} &:= \sum_{q=1}^{N_Q} \left(\sum_{i,j=0}^N \mathbf{Res}(\mathbf{u}_{ij}^q) \right), \\ S_t^{tot} &:= \sum_{q=1}^{N_Q} \left(\sum_{i,j=0}^N \mathbf{v}_{ij}^T \mathbf{Res}(\mathbf{u}_{ij}^q) \right). \end{aligned} \quad (4.112)$$

We claim that with this setup the scheme is fully conservative independent of the initial condition. Therefore, we consider randomly constructed discontinuous initial conditions

$$\begin{pmatrix} \rho \\ u \\ v \\ p \end{pmatrix} = \begin{pmatrix} \vartheta_{1,1} \\ \vartheta_{1,2} \\ \vartheta_{1,3} \\ \vartheta_{1,4} \end{pmatrix} \quad \text{if } x \leq y, \quad \begin{pmatrix} \rho \\ u \\ v \\ p \end{pmatrix} = \begin{pmatrix} \vartheta_{2,1} \\ \vartheta_{2,2} \\ \vartheta_{2,3} \\ \vartheta_{2,4} \end{pmatrix} \quad \text{if } x > y. \quad (4.113)$$

Here $\vartheta_{k,l}$ are uniformly generated random numbers in $[0, 1]$. We calculate \mathbf{U}_t^{tot} and S_t^{tot} for 1000 different initial conditions which gives us $(\mathbf{U}_t^{tot})_{lk}$ and $(S_t^{tot})_k$ for $k = 1, \dots, 1000$ and $l = 1, \dots, 4$. Within the euclidean norm we obtain the results in Table 4.3.

**Verification of primary and entropy conservation for the SBP Flux
Difference scheme**

$\ \mathbf{S}_t^{tot}\ _2$	$\ (\mathbf{U}_t^{tot})_{1,:}\ _2$	$\ (\mathbf{U}_t^{tot})_{2,:}\ _2$	$\ (\mathbf{U}_t^{tot})_{3,:}\ _2$	$\ (\mathbf{U}_t^{tot})_{4,:}\ _2$
9.63E-14	4.93E-15	2.17E-14	5.31E-14	8.18E-15

Table 4.3.: Calculating the growth of the total discrete primary quantities \mathbf{U}_t^{tot} and entropy S_t^{tot} for 1000 different random initial conditions with the Flux Difference discretization. The average growth is presented within the euclidean norm. All values are near machine precision which demonstrates primary and entropy conservation.

In Table 4.3 we verify primary and entropy conservation. In comparison, when considering the same setup with the standard SBP discretization (4.63) we verify primary conservation but not entropy conservation, see Table 4.4.

**Calculating the total discrete growth in the primary quantities and entropy
with the standard SBP discretization**

$\ \mathbf{S}_t^{tot}\ _2$	$\ (\mathbf{U}_t^{tot})_{1,:}\ _2$	$\ (\mathbf{U}_t^{tot})_{2,:}\ _2$	$\ (\mathbf{U}_t^{tot})_{3,:}\ _2$	$\ (\mathbf{U}_t^{tot})_{4,:}\ _2$
6.70E-01	8.61E-15	2.83E-14	6.93E-15	9.86E-14

Table 4.4.: Calculating the growth of the total discrete primary quantities \mathbf{U}_t^{tot} and entropy \mathbf{S}_t^{tot} for 1000 different random initial conditions with the standard SBP discretization. The growth is presented within the euclidean norm. Here, we verify conservation of the primary quantities but not of the entropy.

Next, we run a simulation in time to demonstrate the increased robustness of the entropy stable SBP Flux Difference scheme. Therefore, we approximate the total entropy as a function of time

$$S^{tot} := \sum_{q=1}^{N_Q} \left(\sum_{i,j=0}^N J_{ij} \omega_i \omega_j S_{ij}^q \right), \quad (4.114)$$

over the time domain $t \in [0, T]$, where we choose $T = 25$ and $CFL = 0.5$. We adapt the discontinuous initial condition from [38]

$$\begin{pmatrix} \rho \\ \nu_1 \\ \nu_2 \\ \mathcal{P} \end{pmatrix} = \begin{pmatrix} 1.08 \\ 0.2 \\ 0.01 \\ 0.95 \end{pmatrix} \quad \text{if } x \leq y, \quad \begin{pmatrix} \rho \\ \nu_1 \\ \nu_2 \\ \mathcal{P} \end{pmatrix} = \begin{pmatrix} 1 \\ 10^{-12} \\ 10^{-12} \\ 1 \end{pmatrix} \quad \text{if } x > y, \quad (4.115)$$

and consider periodic boundaries. We solve the compressible Euler equations with the Flux Difference scheme and the standard SBP method. In Figure 4.2 we plot the temporal evolution of the total entropy for the for both schemes.

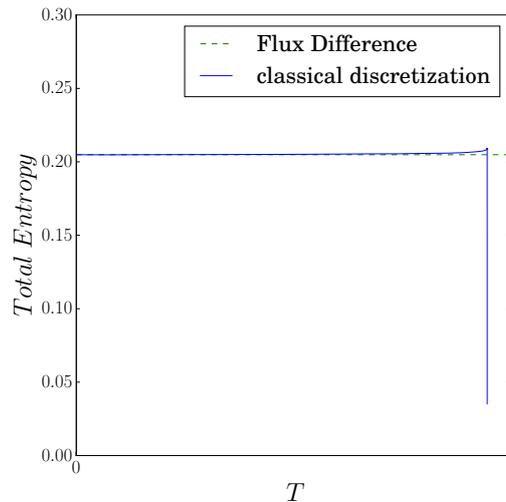


Figure 4.2.: Evolution of the total discrete entropy. The SBP Flux Difference scheme conserves the total discrete entropy whereas the entropy explodes for the standard SBP method. At $t \approx 0.95$ the standard SBP scheme crashes.

The Flux Difference scheme conserves the total discrete entropy. However, for the classical SBP method we observe an unpredictable behavior of the total discrete

entropy at $t \approx 0.95$ and the simulation crashes. This has been verified for $CFL = 0.5$; 0.25 ; 0.125 ; 0.0625 and demonstrates the enhanced robustness of entropy conserving/stable schemes. As the Flux Difference method does not crash we run the simulation until $T = 25$. Moreover, we also verify entropy stability by including dissipation setting $\sigma = 1$. For this test we use the same configuration as for verifying entropy conservation (HGTL operator with $p = 2$ and $N + 1 = 12$) and set $CFL = 0.5$. The total discrete primary quantities and entropy for $\sigma = 0$ and $\sigma = 1$ are plotted in Figure 4.3 and 4.4.

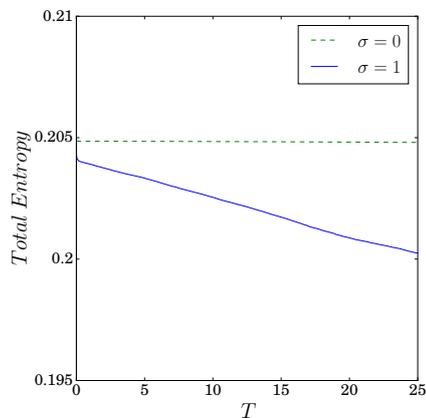


Figure 4.3.: Evolution of the total discrete entropy of the solution with and without dissipation. We see that the total discrete entropy is conserved when no interface dissipation is included and decays with interface dissipation.

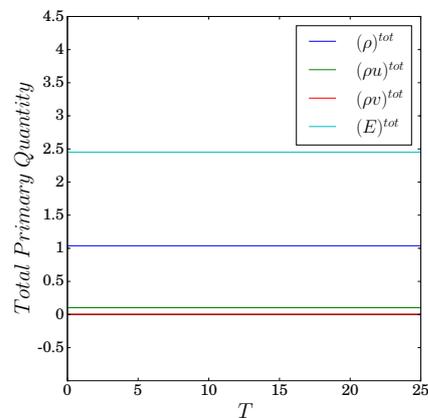


Figure 4.4.: A plot that demonstrates the discrete conservation of the primary quantities. The plot does not depend on interface dissipation.

Here, the total primary quantities are conserved while at the same time the total entropy decays/remains constant (depending on the choice of σ). We note, that the plot of the total discrete primary quantities remains the same whether or not interface dissipation is included.

4.4. Summary

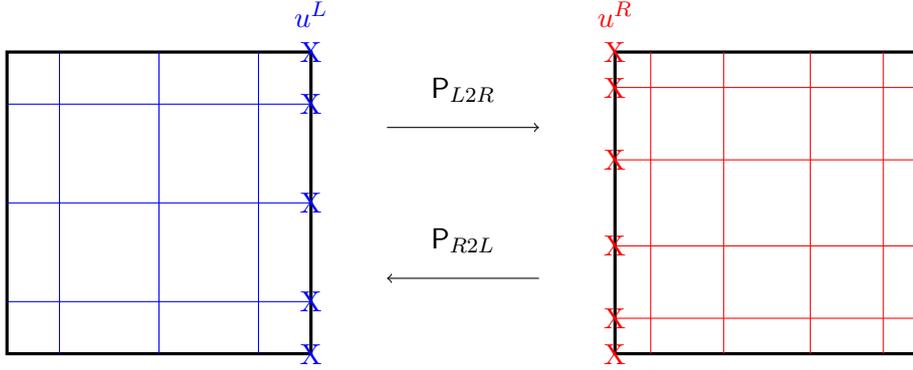
In this chapter we introduced primary conservative and entropy stable SBP methods for non-linear hyperbolic systems of conservation laws. For simplicity we first focused on problems with one dimensional space. When solving such systems using standard SBP methods introduced in Chapter 2 we do not necessarily obtain a physical relevant solution. Therefore, we seek numerical solutions which discretely obey the second law of thermodynamics. Here, we introduced the concept of entropy stability by introducing the entropy function \mathcal{S} which can be interpreted as an extension of the theory for linear stability described in Chapter 2. The goal of the chapter was to derive high-order primary conservative and entropy stable schemes. Hence, we introduced skew-symmetric discretizations based on split forms of the corresponding conservation laws. A more general approach deriving entropy stable schemes was given by Fisher and Carpenter [33]. They invented the high-order Flux Difference discretization adapting knowledge from finite volume schemes and extending these results to high-order SBP methods. The skew-symmetric and the Flux Difference discretizations are equivalent. However, the Flux Difference method is not instructed to know the explicit split form of the conservation laws and therefore can be applied to, e.g. the compressible Euler equations. We applied the Flux Difference discretization to problems with a two dimensional spatial domain and mathematically proved that this method is primary conservative and entropy stable. Next, we implemented the Flux Difference method and applied it to the Euler equations. We observed that this method is high-order accurate considering DG, FD and HGTL operators. Besides the high-order property we numerically verified primary conservation and entropy stability/conservation. In addition, we demonstrated that entropy stable/conservative schemes are more robust by setting up runs with a discontinuous initial condition. Here, the standard SBP scheme crashes whereas the SBP Flux Difference method remains stable.

All the schemes and results presented in this chapter relied on a conforming mesh. As for the LAE we will extend the Flux Difference discretization to a non-conforming mesh in the next chapter.

5. Entropy Stable and Primary Conservative Summation-by-Parts Methods for Non-Linear Hyperbolic Systems on Non-Conforming Meshes

In this chapter we introduce an entropy stable and primary conservative SBP method for solving non-linear systems of conservation laws on non-conforming meshes. Solving such non-linear problems on non-conforming meshes was first investigated for DG methods by Kopriva [69]. Kopriva introduced the mortar method coupling non-conforming elements through an intermediate grid. The method was provably primary conservative on curvilinear h/p non-conforming meshes, but not entropy stable. After the breakthrough of Fisher and Carpenter [33] deriving the Flux Difference discretization (4.64) in 2013, high-order entropy stable become more popular for SBP methods. In 2016, Carpenter et al. [16] were the first researchers to extend the SBP Flux Difference method to a curvilinear p non-conforming mesh. Focusing on a Cartesian mesh, Friedrich et al. [38] were the first to extend the SBP Flux Difference scheme on a h/p non-conforming mesh. The investigation of primary conservative and entropy stable SBP schemes on curvilinear meshes with h refinement is an ongoing research topic.

Here, we will introduce the entropy stable h/p non-conforming SBP method by Friedrich et al. [38]. As described in Chapter 3, the fundamental difference between a conforming and non-conforming scheme is the calculation of the numerical surface flux. For non-conforming meshes a point-to-point transfer is not possible and therefore projection operators must be introduced. Here, we consider the same projection operators as in Section 3.2. Unfortunately, the construction of non-conforming numerical surface fluxes for non-linear problems is not a simple extension of the linear surface fluxes for the LAE in (3.43). To point out the difficulties we focus on a scalar conservation law and an algebraic non-conforming mesh (p refinement), see Figure 5.1.


 Figure 5.1.: Non-conforming mesh with p refinement.

In Chapter 3 the construction of the numerical surface fluxes on the left and right element $f^{*,L}$ and $f^{*,R}$ has been done by projecting the conservative variables from one side to another. For the scalar LAE on a conforming mesh the central flux $f_{EC}^* = \frac{1}{2} (f(u^L) + f(u^R))$ is energy conservative. On a non-conforming mesh the energy conservative flux for the LAE is given by

$$\begin{aligned} f_{EC,i}^{*,L} &= f_{EC}^* \left(\sum_{k=0}^{N_R} [P_{R2L}]_{ik} u_k^R, u_i^L \right), \\ f_{EC,j}^{*,R} &= f_{EC}^* \left(u_j^R, \sum_{k=0}^{N_L} [P_{L2R}]_{jk} u_k^L \right). \end{aligned} \quad (5.1)$$

for $i = 0, \dots, N_L$ and $j = 0, \dots, N_R$. The key to prove energy conservation was the M-compatibility condition (3.2)

$$P_{R2L}^T M_L = M_R P_{L2R}. \quad (5.2)$$

This condition depends linearly on the projection operators and thus, we can recast the proof of entropy conservation for linear problems on a conforming mesh to a non-conforming. However, for non-linear problems we replace f_{EC}^* with a more complicated entropy conservative numerical flux that features possible polynomial or rational non-linearities as well as strong cross coupling between the left and right solution states, e.g. [18, 36, 42]. This introduces complications which do not necessarily maintain that the scheme is entropy conservative. A simple example, consider the Burgers' equation. For a conforming mesh the entropy conservative

numerical flux is given by [42]

$$f_{EC}^* = \frac{1}{6} \left((u^L)^2 + u^L u^R + (u^R)^2 \right). \quad (5.3)$$

Constructing the non-conforming numerical surface flux using (5.1) we obtain

$$\begin{aligned} f_{EC,i}^{*,L} &= \frac{1}{6} \left((u_i^L)^2 + u_i^L \left(\sum_{k=0}^{N_R} [P_{R2L}]_{ik} u_k^R \right) + \left(\sum_{k=0}^{N_R} [P_{R2L}]_{ik} u_k^R \right)^2 \right), \\ f_{EC,j}^{*,R} &= \frac{1}{6} \left(\left(\sum_{k=0}^{N_L} [P_{L2R}]_{jk} u_k^L \right)^2 + \left(\sum_{k=0}^{N_L} [P_{L2R}]_{jk} u_k^L \right) u_j^R + (u_j^R)^2 \right). \end{aligned} \quad (5.4)$$

for $i = 0, \dots, N_L$ and $j = 0, \dots, N_R$. However, it is not obvious how to treat non-linear projections, e.g. the operation to compute the square of $\sum_{k=0}^{N_L} [P_{L2R}]_{jk} u_k^L$ and $\sum_{k=0}^{N_R} [P_{R2L}]_{ik} u_k^R$ will change the total discrete entropy. Recasting the entropy conservation proof from a conforming to a non-conforming mesh as for linear problems does not work as the M-compatibility condition (5.2) depends linearly on the projection operators and thus the non-linear projection terms within the proof do not vanish. Therefore, we need a further modification to the surface fluxes to guarantee entropy stability.

In Section 5.1 we introduce how such fluxes are calculated to obtain entropy stable SBP schemes for non-conforming meshes. To conclude, we verify the theoretical derivations through a variety of numerical test cases in Section 5.2.

5.1. Non-Conforming Discrete Analysis for Non-Linear Problems

This section is structured in a one-to-one fashion to Section 3.4. Throughout we focus on a Cartesian mesh. First, in Section 5.1.1, we address the issues associated with p refinement. We build on the p refinement result to construct entropy conservative numerical surface fluxes for h refinement in Section 5.1.2. Then, Section 5.1.3 describes how additional dissipation can be included at non-conforming interfaces to guarantee entropy stability.

5.1.1. Non-Conforming Nodal Distributions

In this section we introduce how to create a fully conservative scheme on a mesh with p refinement. As shown in Chapter 4, Lemma 5, the discrete growth of the

primary quantities and entropy on a single element reduces to

$$\sum_{i,j=0}^N J_{ij} \omega_i \omega_j \frac{\partial \mathbf{u}_{ij}}{\partial t} = - \sum_{j=0}^N \omega_j (\tilde{\mathbf{f}}_{Nj} - \tilde{\mathbf{f}}_{0j}) - \sum_{i=0}^N \omega_i (\tilde{\mathbf{g}}_{iN} - \tilde{\mathbf{g}}_{i0}). \quad (5.5)$$

and

$$\begin{aligned} \sum_{i,j=0}^N J_{ij} \omega_i \omega_j \frac{\partial S_{ij}}{\partial t} = & - \sum_{j=0}^N \omega_j \left([\mathbf{v}^T \tilde{\mathbf{f}}^* - \Psi^f]_{Nj} - [\mathbf{v}^T \tilde{\mathbf{f}}^* - \Psi^f]_{0j} \right) \\ & - \sum_{i=0}^N \omega_i \left([\mathbf{v}^T \tilde{\mathbf{g}}^* - \Psi^g]_{iN} - [\mathbf{v}^T \tilde{\mathbf{g}}^* - \Psi^g]_{i0} \right). \end{aligned} \quad (5.6)$$

Due to the Flux Difference form and the use of SBP operators the discrete growth of the quantities reduce to evaluations at the element interfaces. In order to obtain a primary and entropy conservative or, in other words, fully conservative scheme, the terms on the interior interfaces need to vanish. These terms are defined by

$$\mathbf{I}U_t := \sum_{j=0}^{N_R} \omega_j^R \tilde{\mathbf{f}}_j^{*,R} - \sum_{j=0}^{N_L} \omega_j^L \tilde{\mathbf{f}}_j^{*,L}, \quad (5.7)$$

$$IS_t := \sum_{j=0}^{N_R} \omega_j^R [\mathbf{v}^{R,T} \tilde{\mathbf{f}}_j^{*,R} - \Psi^{f,R}]_j - \sum_{j=0}^{N_L} \omega_j^L [\mathbf{v}^{L,T} \tilde{\mathbf{f}}_j^{*,L} - \Psi^{f,L}]_j, \quad (5.8)$$

which are almost the same terms as in (4.92) and (4.93). However, as we consider non-conforming elements we introduce the index L and R on N and ω .

Here we address the importance of choosing an appropriate numerical surface flux $\tilde{\mathbf{f}}^{*,R}$ and $\tilde{\mathbf{f}}^{*,L}$. Assume we have an entropy conservative flux \mathbf{f}_{EC}^* for a conforming mesh satisfying the Tadmor Shuffle Condition

$$(\mathbf{v}^R - \mathbf{v}^L)^T \mathbf{f}_{EC}^*(\mathbf{u}^R, \mathbf{u}^L) = \tilde{\Psi}^{f,R} - \tilde{\Psi}^{f,L}. \quad (5.9)$$

Simply applying (5.1) is insufficient for an entropy conservative scheme due to non-linear projection terms. The key to obtain non-conforming entropy conservative fluxes is not to project the conservative variables \mathbf{u} but projecting the surface flux \mathbf{f}_{EC}^* in terms of

$$\begin{aligned} \mathbf{f}_{EC,i}^{*,L} &= \sum_{k=0}^{N_R} [P_{R2L}]_{ik} \mathbf{f}_{EC}^*(\mathbf{u}_i^L, \mathbf{u}_k^R), \\ \mathbf{f}_{EC,j}^{*,R} &= \sum_{k=0}^{N_L} [P_{L2R}]_{jk} \mathbf{f}_{EC}^*(\mathbf{u}_k^L, \mathbf{u}_j^R), \end{aligned} \quad (5.10)$$

for $i = 0, \dots, N_L$ and $j = 0, \dots, N_R$. For a Cartesian mesh the scaled numerical fluxes are given by $\tilde{\mathbf{f}}_{EC,i}^{*,L} = \frac{\Delta y}{2} \mathbf{f}_{EC,i}^{*,L}$ and $\tilde{\mathbf{f}}_{EC,j}^{*,R} = \frac{\Delta y}{2} \mathbf{f}_{EC,j}^{*,R}$. Note, that in (5.10) we observe only linear projections.

Remark 15. Focusing on the LAE with $f_{EC}^*(u^R, u^L) = \frac{1}{2} (f(u^R) + f(u^L))$ we see that (5.10) and (5.1) are equivalent as

$$\begin{aligned} f_{EC,i}^{*,L} &= \sum_{k=0}^{N_R} [\mathbf{P}_{R2L}]_{ik} f_{EC}^*(u_i^L, u_k^R), \\ &= \frac{1}{2} \left(\sum_{k=0}^{N_R} [\mathbf{P}_{R2L}]_{ik} f(u_k^R) + f(u_i^L) \sum_{k=0}^{N_R} [\mathbf{P}_{R2L}]_{ik} \right). \end{aligned} \quad (5.11)$$

Assuming that the \mathbf{P}_{R2L} can project a constant exactly ($\sum_{k=0}^{N_R} [\mathbf{P}_{R2L}]_{ik} = 1$) and including $f(u) = au$ we get

$$f_{EC,i}^{*,L} = \frac{1}{2} \left(a \sum_{k=0}^{N_R} [\mathbf{P}_{R2L}]_{ik} u_k^R + au_i^L \right) = f_{EC}^* \left(u_i^L, \sum_{k=0}^{N_R} [\mathbf{P}_{R2L}]_{ik} u_k^R \right), \quad (5.12)$$

which is precisely (5.1). However, considering the non-linear Burgers' equation we notice a difference. Setting $f_{EC}^* = \frac{1}{6} ((u^L)^2 + u^L u^R + (u^R)^2)$ in (5.10) we get

$$f_{EC,i}^{*,L} = \frac{1}{6} \left((u_i^L)^2 \sum_{k=0}^{N_R} [\mathbf{P}_{R2L}]_{ik} + u_i^L \sum_{k=0}^{N_R} [\mathbf{P}_{R2L}]_{ik} u_k^R + \sum_{k=0}^{N_R} [\mathbf{P}_{R2L}]_{ik} (u_k^R)^2 \right). \quad (5.13)$$

Again, assuming that \mathbf{P}_{R2L} can project a constant exactly we obtain

$$f_{EC,i}^{*,L} = \frac{1}{6} \left((u_i^L)^2 + u_i^L \sum_{k=0}^{N_R} [\mathbf{P}_{R2L}]_{ik} u_k^R + \sum_{k=0}^{N_R} [\mathbf{P}_{R2L}]_{ik} (u_k^R)^2 \right). \quad (5.14)$$

Here, (5.1) and (5.10) are nearly the same, but for (5.10) the projection operator is outside of the square and therefore we only have a linear projection.

With this construction of a linear projection we prove that the non-conforming SBP Flux Difference scheme is fully conservative in Theorem 9.

Theorem 9. *The SBP Flux Difference discretization (4.64) with the numerical surface fluxes (5.10) is fully conservative on an algebraic non-conforming Cartesian mesh, provided that \mathbf{f}_{EC}^* satisfies the Tadmor Shuffle condition (5.9) and the projection operators satisfy the M-compatibility condition (5.2).*

Proof. First, we prove primary conservation. Therefore, equation (5.7) needs to vanish. We include the numerical fluxes derived from (5.10) in (5.7) and obtain

$$\mathbf{I}U_t = \sum_{j=0}^{N_R} \omega_j^R \sum_{k=0}^{N_L} [\mathbf{P}_{L2R}]_{jk} \tilde{\mathbf{f}}_{EC}^*(\mathbf{u}_k^L, \mathbf{u}_j^R) - \sum_{j=0}^{N_L} \omega_j^L \sum_{k=0}^{N_R} [\mathbf{P}_{R2L}]_{jk} \tilde{\mathbf{f}}_{EC}^*(\mathbf{u}_j^L, \mathbf{u}_k^R). \quad (5.15)$$

Due to M-compatibility condition (5.2) we know $\omega_j^R [\mathbf{P}_{L2R}]_{jk} = \omega_k^L [\mathbf{P}_{R2L}]_{kj}$. This gives

$$\mathbf{I}U_t = \sum_{j=0}^{N_R} \sum_{k=0}^{N_L} \omega_j^R [\mathbf{P}_{L2R}]_{jk} \tilde{\mathbf{f}}_{EC}^*(\mathbf{u}_k^L, \mathbf{u}_j^R) - \sum_{j=0}^{N_L} \sum_{k=0}^{N_R} \omega_k^R [\mathbf{P}_{L2R}]_{kj} \tilde{\mathbf{f}}_{EC}^*(\mathbf{u}_j^L, \mathbf{u}_k^R), \quad (5.16)$$

and rearranging indices we obtain

$$\mathbf{I}U_t = \mathbf{0}, \quad (5.17)$$

which indicates primary conservation.

Next, we show that the non-conforming scheme is entropy conservative. To do so, IS_t needs to be zero. Therefore, we substitute the scaled fluxes from (5.10) in (5.8).

$$\begin{aligned} IS_t = & \sum_{j=0}^{N_R} \omega_j^R \left[\mathbf{v}_j^{R,T} \sum_{k=0}^{N_L} [\mathbf{P}_{L2R}]_{jk} \tilde{\mathbf{f}}_{EC}^*(\mathbf{u}_k^L, \mathbf{u}_j^R) - \tilde{\Psi}_j^{f,R} \right] \\ & - \sum_{j=0}^{N_L} \omega_j^L \left[\mathbf{v}_j^{L,T} \sum_{k=0}^{N_R} [\mathbf{P}_{R2L}]_{jk} \tilde{\mathbf{f}}_{EC}^*(\mathbf{u}_j^L, \mathbf{u}_k^R) - \tilde{\Psi}_j^{f,L} \right]. \end{aligned} \quad (5.18)$$

Assuming that the projection operators can project a constant exactly, (5.18) is equivalent to

$$\begin{aligned} IS_t = & \sum_{j=0}^{N_R} \omega_j^R \left[\mathbf{v}_j^{R,T} \sum_{k=0}^{N_L} [\mathbf{P}_{L2R}]_{jk} \tilde{\mathbf{f}}_{EC}^*(\mathbf{u}_k^L, \mathbf{u}_j^R) - \tilde{\Psi}_j^{f,R} \sum_{k=0}^{N_L} [\mathbf{P}_{L2R}]_{jk} \right] \\ & - \sum_{j=0}^{N_L} \omega_j^L \left[\mathbf{v}_j^{L,T} \sum_{k=0}^{N_R} [\mathbf{P}_{R2L}]_{jk} \tilde{\mathbf{f}}_{EC}^*(\mathbf{u}_j^L, \mathbf{u}_k^R) - \tilde{\Psi}_j^{f,L} \sum_{k=0}^{N_R} [\mathbf{P}_{R2L}]_{jk} \right]. \end{aligned} \quad (5.19)$$

Rearranging sums we obtain

$$\begin{aligned} IS_t = & \sum_{j=0}^{N_R} \sum_{k=0}^{N_L} \omega_j^R [\mathbf{P}_{L2R}]_{jk} \left[\mathbf{v}_j^{R,T} \tilde{\mathbf{f}}_{EC}^*(\mathbf{u}_k^L, \mathbf{u}_j^R) - \tilde{\Psi}_j^{f,R} \right] \\ & - \sum_{j=0}^{N_L} \sum_{k=0}^{N_R} \omega_j^L [\mathbf{P}_{R2L}]_{jk} \left[\mathbf{v}_j^{L,T} \tilde{\mathbf{f}}_{EC}^*(\mathbf{u}_j^L, \mathbf{u}_k^R) - \tilde{\Psi}_j^{f,L} \right]. \end{aligned} \quad (5.20)$$

Due to the linear dependence of the projection operators we apply the M-compatibility condition $\omega_j^R [\mathbf{P}_{L2R}]_{jk} = \omega_k^L [\mathbf{P}_{R2L}]_{kj}$ and find

$$\begin{aligned} IS_t = & \sum_{j=0}^{N_R} \sum_{k=0}^{N_L} \omega_j^R [\mathbf{P}_{L2R}]_{jk} \left[\mathbf{v}_j^{R,T} \tilde{\mathbf{f}}_{EC}^*(\mathbf{u}_k^L, \mathbf{u}_j^R) - \tilde{\Psi}_j^{f,R} \right] \\ & - \sum_{j=0}^{N_L} \sum_{k=0}^{N_R} \omega_k^R [\mathbf{P}_{L2R}]_{kj} \left[\mathbf{v}_j^{L,T} \tilde{\mathbf{f}}_{EC}^*(\mathbf{u}_j^L, \mathbf{u}_k^R) - \tilde{\Psi}_j^{f,L} \right]. \end{aligned} \quad (5.21)$$

By rearranging the indices we arrive at

$$IS_t = \sum_{j=0}^{N_R} \sum_{k=0}^{N_L} \omega_j^R [\mathbf{P}_{L2R}]_{jk} \left[(\mathbf{v}_j^{R,T} - \mathbf{v}_k^{L,T}) \tilde{\mathbf{f}}_{EC}^*(\mathbf{u}_k^L, \mathbf{u}_j^R) - (\tilde{\Psi}_j^{f,R} - \tilde{\Psi}_k^{f,L}) \right]. \quad (5.22)$$

Finally, as \mathbf{f}_{EC}^* satisfies the Tadmor Shuffle condition (5.9) we obtain

$$IS_t = \sum_{j=0}^{N_R} \sum_{k=0}^{N_L} \omega_j^R [\mathbf{P}_{L2R}]_{jk} 0 = 0, \quad (5.23)$$

which indicates entropy conservation. As IS_t and IU_t are both zero we have derived an fully conservative algebraic non-conforming SBP scheme. \square

With this proof, we can construct entropy conservative schemes on algebraic non-conforming meshes (p refinement). To introduce additional flexibility, we next consider geometric non-conforming discretizations (h refinement).

5.1.2. Non-Conforming Interfaces with Hanging Corners

Building on the fully conservative, algebraic non-conforming SBP method for non-linear problems in Section 5.1.1 we now focus on handling meshes with non-conforming interfaces (h refinement). Therefore, we focus on the same mesh setup as in Section 3.4.2 in Figure 3.9, where we discussed h refinement for linear problems. Here, we considered an element R with height Δ_R connected to NE elements denoted by L_i with height Δ_{L_i} , for $i = 1, \dots, NE$. Following the analogous steps as in Section 3.4.2 we obtain projection matrices satisfying the modified M-compatibility condition

$$\Delta_{L_i} \mathbf{P}_{R2L_i}^T \mathbf{M}_{L_i} = \Delta_R \mathbf{M}_R \mathbf{P}_{L_i2R}, \quad i = 1, \dots, NE, \quad (5.24)$$

or

$$\Delta_{L_i} [\mathbf{P}_{R2L_i}]_{kj} \omega_k^{L_i} = \Delta_R \omega_j^R [\mathbf{P}_{L_i2R}]_{jk}, \quad i = 1, \dots, NE, \quad (5.25)$$

for $j = 0, \dots, N_R$ and $k = 0, \dots, N_{L_i}$. Due to the construction of the projection operators it holds that

$$\begin{aligned} \mathbf{P}_{R2L_i} \mathbf{1}^R &= \mathbf{1}^{L_i}, \quad \text{for } i = 1, \dots, NE, \\ \sum_{i=1}^{NE} \mathbf{P}_{L_i 2R} \mathbf{1}^{L_i} &= \mathbf{1}^R, \end{aligned} \quad (5.26)$$

where $\mathbf{1}^{L_i}$ denotes a one vector of size $N_{L_i} + 1$. A component-wise expression of (5.26) is

$$\begin{aligned} \sum_{l=0}^{N_R} [\mathbf{P}_{R2L_i}]_{jl} &= 1, \quad \text{for } i = 1, \dots, NE, \\ \sum_{i=1}^{NE} \left(\sum_{l=0}^{N_{L_i}} [\mathbf{P}_{L_i 2R}]_{kl} \right) &= 1, \end{aligned} \quad (5.27)$$

with $j = 0, \dots, N_R$ and $k = 0, \dots, N_{L_i}$.

Next, we require a numerical surface flux to couple the elements L_i for $i = 1, \dots, NE$ with R . In order to obtain a fully conservative scheme we focus on all terms defined on the interface connecting L_i and R . As for (5.7) and (5.8) these terms are

$$\mathbf{I}U_t := \sum_{j=0}^{N_R} \omega_j^R \tilde{\mathbf{f}}_j^{*,R} - \sum_{i=1}^{NE} \sum_{j=0}^{N_{L_i}} \omega_j^{L_i} \tilde{\mathbf{f}}_j^{*,L_i}, \quad (5.28)$$

$$IS_t := \sum_{j=0}^{N_R} \omega_j^R \left[\mathbf{v}^{R,T} \tilde{\mathbf{f}}^{*,R} - \tilde{\Psi}^{f,R} \right]_j - \sum_{i=1}^{NE} \sum_{j=0}^{N_{L_i}} \omega_j^{L_i} \left[\mathbf{v}^{L_i,T} \tilde{\mathbf{f}}^{*,L_i} - \tilde{\Psi}^{f,L_i} \right]_j. \quad (5.29)$$

In order to obtain a fully conservative scheme (5.28) and (5.29) should be zero. The appropriate numerical surface flux is given by extending the geometric non-conforming fluxes (5.10) for non-linear problems in Section 5.1.2 by

$$\begin{aligned} \mathbf{f}_{EC,k}^{*,L_i} &= \sum_{l=0}^{N_R} [\mathbf{P}_{R2L_i}]_{kl} \mathbf{f}_{EC}^*(\mathbf{u}_k^{L_i}, \mathbf{u}_l^R), \\ \mathbf{f}_{EC,j}^{*,R} &= \sum_{m=1}^{NE} \left(\sum_{l=0}^{N_{L_i}} [\mathbf{P}_{L_m 2R}]_{jl} \mathbf{f}_{EC}^*(\mathbf{u}_l^{L_m}, \mathbf{u}_j^R) \right), \end{aligned} \quad (5.30)$$

for $j = 0, \dots, N_R$ and $k = 0, \dots, N_{L_i}$ where $i = 1, \dots, NE$. As we focus on

Cartesian meshes we obtain the scaled fluxes by multiplying (5.30) with the corresponding metric terms or, in this case, with half the element height. So

$$\begin{aligned}\tilde{\mathbf{f}}_{EC,k}^{*,L_i} &= \frac{\Delta_{L_i}}{2} \mathbf{f}_{EC,k}^{*,L_i}, \\ \tilde{\mathbf{f}}_{EC,j}^{*,R} &= \frac{\Delta_R}{2} \mathbf{f}_{EC,j}^{*,R}.\end{aligned}\tag{5.31}$$

Note, that as we focus on a geometric non-conforming mesh it holds $\Delta_{L_i} \neq \Delta_R$. We will show that these fluxes give us a fully conservative scheme for non-linear problems on geometric non-conforming meshes.

Theorem 10. *Consider the non-conforming fluxes (5.30) and (5.31). Then the SBP Flux Difference discretization (4.64) is fully conservative on a geometric non-conforming Cartesian mesh.*

Proof. As for the proof of Theorem 9 we first prove primary conservation. Inserting the fluxes (5.30) in \mathbf{IU}_t we get

$$\begin{aligned}\mathbf{IU}_t &= \sum_{j=0}^{N_R} \omega_j^R \sum_{i=1}^{NE} \left(\sum_{l=0}^{N_{L_i}} [\mathbf{P}_{L_i 2R}]_{jl} \frac{\Delta_R}{2} \tilde{\mathbf{f}}_{EC}^*(\mathbf{u}_l^{L_i}, \mathbf{u}_j^R) \right) \\ &\quad - \sum_{i=1}^{NE} \sum_{j=0}^{N_{L_i}} \omega_j^{L_i} \sum_{l=0}^{N_R} [\mathbf{P}_{R 2L_i}]_{jl} \frac{\Delta_{L_i}}{2} \tilde{\mathbf{f}}_{EC}^*(\mathbf{u}_j^{L_i}, \mathbf{u}_l^R).\end{aligned}\tag{5.32}$$

Rearranging terms gives

$$\begin{aligned}\mathbf{IU}_t &= \sum_{i=1}^{NE} \sum_{j=0}^{N_R} \sum_{l=0}^{N_{L_i}} \frac{\Delta_R}{2} \omega_j^R [\mathbf{P}_{L_i 2R}]_{jl} \mathbf{f}_{EC}^*(\mathbf{u}_l^{L_i}, \mathbf{u}_j^R) \\ &\quad - \sum_{i=1}^{NE} \sum_{j=0}^{N_{L_i}} \sum_{l=0}^{N_R} \frac{\Delta_{L_i}}{2} \omega_j^{L_i} [\mathbf{P}_{R 2L_i}]_{jl} \mathbf{f}_{EC}^*(\mathbf{u}_j^{L_i}, \mathbf{u}_l^R),\end{aligned}\tag{5.33}$$

and by indexing we arrive at

$$\mathbf{IU}_t = \frac{1}{2} \sum_{i=1}^{NE} \sum_{j=0}^{N_{L_i}} \sum_{l=0}^{N_R} \left(\Delta_R \omega_l^R [\mathbf{P}_{L_i 2R}]_{lj} - \Delta_{L_i} \omega_j^{L_i} [\mathbf{P}_{R 2L_i}]_{jl} \right) \mathbf{f}_{EC}^*(\mathbf{u}_j^{L_i}, \mathbf{u}_l^R).\tag{5.34}$$

Finally, due to the the modified M-compatibility condition (5.27) we obtain

$$\mathbf{IU}_t = \frac{1}{2} \sum_{i=1}^{NE} \sum_{j=0}^{N_{L_i}} \sum_{l=0}^{N_R} 0 \mathbf{f}_{EC}^*(\mathbf{u}_j^{L_i}, \mathbf{u}_l^R) = \mathbf{0},\tag{5.35}$$

for which primary conservation holds.

Focusing on entropy conservation we look at IS_t (5.29) and include the numerical fluxes (5.30)

$$\begin{aligned}
 IS_t = & \sum_{j=0}^{N_R} \frac{\Delta_R}{2} \omega_j^R \left[\mathbf{v}_j^{R,T} \sum_{i=1}^{NE} \left(\sum_{l=0}^{N_{L_i}} [\mathbf{P}_{L_i 2R}]_{jl} \mathbf{f}_{EC}^*(\mathbf{u}_l^{L_i}, \mathbf{u}_j^R) \right) - \Psi_j^{f,R} \right] \\
 & - \sum_{i=1}^{NE} \sum_{j=0}^{N_{L_i}} \frac{\Delta_{L_i}}{2} \omega_j^L \left[\mathbf{v}_j^{L_i,T} \sum_{l=0}^{N_R} [\mathbf{P}_{R 2L_i}]_{jl} \mathbf{f}_{EC}^*(\mathbf{u}_j^L, \mathbf{u}_l^R) - \Psi_j^{f,L_i} \right], \tag{5.36}
 \end{aligned}$$

where we rewrite the scaled entropy potentials in terms of $\tilde{\Psi}^{f,R} = \frac{\Delta_R}{2} \Psi^{f,R}$. As for the proof of Theorem 9 we include (5.27) (sums of the projection operators are equal to one) for the further course of the proof. Therefore, (5.36) is equivalent to

$$\begin{aligned}
 IS_t = & \sum_{j=0}^{N_R} \frac{\Delta_R}{2} \omega_j^R \left[\mathbf{v}_j^{R,T} \sum_{i=1}^{NE} \left(\sum_{l=0}^{N_{L_i}} [\mathbf{P}_{L_i 2R}]_{jl} \mathbf{f}_{EC}^*(\mathbf{u}_l^{L_i}, \mathbf{u}_j^R) \right) - \Psi_j^{f,R} \sum_{i=1}^{NE} \left(\sum_{l=0}^{N_{L_i}} [\mathbf{P}_{L_i 2R}]_{jl} \right) \right] \\
 & - \sum_{i=1}^{NE} \sum_{j=0}^{N_{L_i}} \frac{\Delta_{L_i}}{2} \omega_j^L \left[\mathbf{v}_j^{L_i,T} \sum_{l=0}^{N_R} [\mathbf{P}_{R 2L_i}]_{jl} \mathbf{f}_{EC}^*(\mathbf{u}_j^L, \mathbf{u}_l^R) - \Psi_j^{f,L_i} \sum_{l=0}^{N_R} [\mathbf{P}_{R 2L_i}]_{jl} \right]. \tag{5.37}
 \end{aligned}$$

Analogously to Theorem 9 we rewrite the sums

$$\begin{aligned}
 IS_t = & \sum_{i=1}^{NE} \sum_{j=0}^{N_R} \sum_{l=0}^{N_{L_i}} \frac{\Delta_R}{2} \omega_j^R [\mathbf{P}_{L_i 2R}]_{jl} \left(\mathbf{v}_j^{R,T} \mathbf{f}_{EC}^*(\mathbf{u}_l^{L_i}, \mathbf{u}_j^R) - \Psi_j^{f,R} \right) \\
 & - \sum_{i=1}^{NE} \sum_{j=0}^{N_{L_i}} \sum_{l=0}^{N_R} \frac{\Delta_{L_i}}{2} \omega_j^L [\mathbf{P}_{R 2L_i}]_{jl} \left(\mathbf{v}_j^{L_i,T} \mathbf{f}_{EC}^*(\mathbf{u}_j^L, \mathbf{u}_l^R) - \Psi_j^{f,L_i} \right), \tag{5.38}
 \end{aligned}$$

and perform a re-indexing to find

$$\begin{aligned}
 IS_t = & \sum_{i=1}^{NE} \sum_{j=0}^{N_R} \sum_{l=0}^{N_{L_i}} \frac{\Delta_R}{2} \omega_j^R [\mathbf{P}_{L_i 2R}]_{jl} \left(\mathbf{v}_j^{R,T} \mathbf{f}_{EC}^*(\mathbf{u}_l^{L_i}, \mathbf{u}_j^R) - \Psi_j^{f,R} \right) \\
 & - \sum_{i=1}^{NE} \sum_{j=0}^{N_R} \sum_{l=0}^{N_{L_i}} \frac{\Delta_{L_i}}{2} \omega_l^L [\mathbf{P}_{R 2L_i}]_{lj} \left(\mathbf{v}_l^{L_i,T} \mathbf{f}_{EC}^*(\mathbf{u}_l^{L_i}, \mathbf{u}_j^R) - \Psi_l^{f,L_i} \right). \tag{5.39}
 \end{aligned}$$

Next, we apply the modified M-compatibility condition (5.27) and simplify to

obtain

$$IS_t = \sum_{i=1}^{NE} \sum_{j=0}^{N_R} \sum_{l=0}^{N_{L_i}} \frac{\Delta_R}{2} \omega_j^R [\mathbf{P}_{L_i 2R}]_{jl} \left((\mathbf{v}_j^R - \mathbf{v}_l^{L_i})^T \mathbf{f}_{EC}^*(\mathbf{u}_l^{L_i}, \mathbf{u}_j^R) - (\Psi_j^{f,R} - \Psi_l^{f,L_i}) \right), \quad (5.40)$$

and, finally, apply the Tadmor Shuffle condition (5.9) to show the desired result

$$IS_t = \sum_{i=1}^{NE} \sum_{j=0}^{N_R} \sum_{l=0}^{N_{L_i}} \frac{\Delta_R}{2} \omega_j^R [\mathbf{P}_{L_i 2R}]_{jl} 0 = 0, \quad (5.41)$$

and thus obtain entropy conservation. \square

We derived a fully conservative scheme for h/p non-conforming meshes. Note, that the proofs of Theorems 9 and 10 are general for any hyperbolic PDE with physical fluxes \mathbf{f}, \mathbf{g} where we have a mathematical entropy function. Next, we include interface dissipation to obtain an entropy stable scheme.

5.1.3. Including Dissipation within the Numerical Surface Flux

Considering shocks, the mathematical entropy should decay, which the numerical scheme should reflect. Therefore, we modify the numerical fluxes in Section 5.1.1 and 5.1.2 by a dissipation term. This term provides an entropy stable numerical flux which at the same time maintains primary conservative. As such, we seek a dissipation term for the entropy stable flux such that

$$\begin{aligned} IU_t &= \mathbf{0}, \\ IS_t &\leq 0. \end{aligned} \quad (5.42)$$

We focus on the general h/p non-conforming mesh, where neither nodal distribution nor the interfaces coincide. Therefore, we focus on a mesh like that in Figure 3.9. We construct the entropy stable flux by combining the results of including dissipation for non-conforming fluxes for linear problems in Section 3.4.3 and for conforming fluxes for non-linear problems in Section 4.2. Merging the fluxes (3.69), (3.70) and (4.88) together we have

$$\mathbf{f}_{ES,k}^{*,L_i} = \mathbf{f}_{EC,k}^{*,L_i} - \frac{\lambda_f}{2} \mathbf{K}_k \left(\sum_{l=0}^{N_R} [\mathbf{P}_{R2L_i}]_{kl} \mathbf{v}_l^R - \mathbf{v}_k^{L_i} \right), \quad (5.43)$$

$$\mathbf{f}_{ES,j}^{*,R} = \mathbf{f}_{EC,j}^{*,R} - \frac{\lambda_f}{2} \sum_{i=1}^{NE} \left(\sum_{l=0}^{N_{L_i}} [\mathbf{P}_{L_i 2R}]_{jl} \mathbf{K}_l \left(\sum_{m=0}^{N_R} [\mathbf{P}_{R2L_i}]_{lm} \mathbf{v}_m^R - \mathbf{v}_l^{L_i} \right) \right), \quad (5.44)$$

where λ_f denotes the maximum eigenvalue of the Jacobian of \mathcal{F} , see (4.103). The positive definite dissipation matrix is constructed based on $\mathbf{P}_{R2L_i} \mathbf{v}^R$ and \mathbf{v}^{L_i} , so $\mathbf{K} := \mathbf{K}(\mathbf{P}_{R2L_i} \mathbf{v}^R, \mathbf{v}^{L_i})$. Again, the scaled fluxes are given by (5.31), meaning $\tilde{\mathbf{f}}_{ES,k}^{*,L_i} = \frac{\Delta_{L_i}}{2} \mathbf{f}_{ES,k}^{*,L_i}$ and $\tilde{\mathbf{f}}_{ES,j}^{*,R} = \frac{\Delta_R}{2} \mathbf{f}_{ES,j}^{*,R}$. We prove primary conservation and entropy stability in Theorem 11.

Theorem 11. *Consider the numerical surface fluxes (5.43) and (5.44), then we obtain a primary conservative and entropy stable discretization on a h/p non-conforming mesh.*

Proof. In order to obtain a primary conservative and entropy stable discretization on a h/p non-conforming mesh we need to satisfy the conditions (5.42). Therefore, we include the scaled numerical surface fluxes of (5.43) and (5.44) in (5.28)

$$\begin{aligned} \mathbf{I}\mathbf{U}_t = & \sum_{j=0}^{N_R} \omega_j^R \left(\tilde{\mathbf{f}}_{EC,j}^{*,R} - \frac{\lambda_f \Delta_R}{4} \sum_{i=1}^{NE} \left(\sum_{l=0}^{N_{L_i}} [\mathbf{P}_{L_i 2R}]_{jl} \mathbf{K}_l \left(\sum_{m=0}^{N_R} [\mathbf{P}_{R2L_i}]_{lm} \mathbf{v}_m^R - \mathbf{v}_l^{L_i} \right) \right) \right) \\ & - \sum_{i=1}^{NE} \sum_{k=0}^{N_{L_i}} \omega_k^{L_i} \left(\tilde{\mathbf{f}}_{EC,k}^{*,L_i} - \frac{\lambda_f \Delta_{L_i}}{4} \mathbf{K}_k \left(\sum_{l=0}^{N_R} [\mathbf{P}_{R2L_i}]_{kl} \mathbf{v}_l^R - \mathbf{v}_k^{L_i} \right) \right). \end{aligned} \quad (5.45)$$

As proven in the Theorem 10 the fluxes $\tilde{\mathbf{f}}_{EC,j}^{*,R}$ and $\tilde{\mathbf{f}}_{EC,k}^{*,L_i}$ are primary conservative and thus cancel each other out. This reduces $\mathbf{I}\mathbf{U}_t$ to

$$\begin{aligned} \mathbf{I}\mathbf{U}_t = & - \frac{\lambda_f}{4} \sum_{i=1}^{NE} \sum_{j=0}^{N_R} \sum_{l=0}^{N_{L_i}} \Delta_R \omega_j^R [\mathbf{P}_{L_i 2R}]_{jl} \mathbf{K}_l \left(\sum_{m=0}^{N_R} [\mathbf{P}_{R2L_i}]_{lm} \mathbf{v}_m^R - \mathbf{v}_l^{L_i} \right) \\ & + \frac{\lambda_f}{4} \sum_{i=1}^{NE} \sum_{k=0}^{N_{L_i}} \Delta_{L_i} \omega_k^{L_i} \mathbf{K}_k \left(\sum_{l=0}^{N_R} [\mathbf{P}_{R2L_i}]_{kl} \mathbf{v}_l^R - \mathbf{v}_k^{L_i} \right). \end{aligned} \quad (5.46)$$

Applying the modified M-compatibility equation (5.25) we get

$$\begin{aligned} \mathbf{I}\mathbf{U}_t = & - \frac{\lambda_f}{4} \sum_{i=1}^{NE} \sum_{j=0}^{N_R} \sum_{l=0}^{N_{L_i}} \Delta_{L_i} \omega_l^{L_i} [\mathbf{P}_{R2L_i}]_{lj} \mathbf{K}_l \left(\sum_{m=0}^{N_R} [\mathbf{P}_{R2L_i}]_{lm} \mathbf{v}_m^R - \mathbf{v}_l^{L_i} \right) \\ & + \frac{\lambda_f}{4} \sum_{i=1}^{NE} \sum_{k=0}^{N_{L_i}} \Delta_{L_i} \omega_k^{L_i} \mathbf{K}_k \left(\sum_{l=0}^{N_R} [\mathbf{P}_{R2L_i}]_{kl} \mathbf{v}_l^R - \mathbf{v}_k^{L_i} \right). \end{aligned} \quad (5.47)$$

Rearranging terms and using the fact that the projection operator \mathbb{P}_{R2L_i} can project a constant exactly, see (5.27), we obtain

$$\begin{aligned} \mathbf{I}U_t = & -\frac{\lambda_f}{4} \sum_{i=1}^{NE} \sum_{l=0}^{N_{L_i}} \Delta_{L_i} \omega_l^{L_i} \mathbf{K}_l \left(\sum_{m=0}^{N_R} [\mathbb{P}_{R2L_i}]_{lm} \mathbf{v}_m^R - \mathbf{v}_l^{L_i} \right) \underbrace{\sum_{j=0}^{N_R} [\mathbb{P}_{R2L_i}]_{lj}}_{=1} \\ & + \frac{\lambda_f}{4} \sum_{i=1}^{NE} \sum_{k=0}^{N_{L_i}} \Delta_{L_i} \omega_k^{L_i} \mathbf{K}_k \left(\sum_{l=0}^{N_R} [\mathbb{P}_{R2L_i}]_{kl} \mathbf{v}_l^R - \mathbf{v}_k^{L_i} \right). \end{aligned} \quad (5.48)$$

Finally, by re-indexing we get

$$\begin{aligned} \mathbf{I}U_t = & -\frac{\lambda_f}{4} \sum_{i=1}^{NE} \sum_{l=0}^{N_{L_i}} \Delta_{L_i} \omega_l^{L_i} \mathbf{K}_l \left(\sum_{m=0}^{N_R} [\mathbb{P}_{R2L_i}]_{lm} \mathbf{v}_m^R - \mathbf{v}_l^{L_i} \right) \\ & + \frac{\lambda_f}{4} \sum_{i=1}^{NE} \sum_{l=0}^{N_{L_i}} \Delta_{L_i} \omega_l^{L_i} \mathbf{K}_l \left(\sum_{m=0}^{N_R} [\mathbb{P}_{R2L_i}]_{lm} \mathbf{v}_m^R - \mathbf{v}_l^{L_i} \right) = \mathbf{0}, \end{aligned} \quad (5.49)$$

which proves that the numerical surface fluxes are primary conservative. Next, we prove entropy stability and include the interface fluxes in (5.29). As $\mathbf{f}_{EC,k}^{*,L_i}$ and $\mathbf{f}_{EC,j}^{*,R}$ are provable entropy conservative we only pay attention to the dissipation terms in (5.29).

$$\begin{aligned} IS_t = & -\frac{\lambda_f}{4} \sum_{j=0}^{N_R} \Delta_R \omega_j^R \mathbf{v}_j^{R,T} \sum_{i=1}^{NE} \left(\sum_{l=0}^{N_{L_i}} [\mathbb{P}_{L_i2R}]_{jl} \mathbf{K}_l \left(\sum_{m=0}^{N_R} [\mathbb{P}_{R2L_i}]_{lm} \mathbf{v}_m^R - \mathbf{v}_l^{L_i} \right) \right) \\ & + \frac{\lambda_f}{4} \sum_{i=1}^{NE} \sum_{k=0}^{N_{L_i}} \Delta_{L_i} \omega_k^L \mathbf{v}_k^{L_i,T} \mathbf{K}_k \left(\sum_{l=0}^{N_R} [\mathbb{P}_{R2L_i}]_{kl} \mathbf{v}_l^R - \mathbf{v}_k^{L_i} \right). \end{aligned} \quad (5.50)$$

First, we reorder the sums

$$\begin{aligned} IS_t = & -\frac{\lambda_f}{4} \sum_{i=1}^{NE} \sum_{j=0}^{N_R} \sum_{l=0}^{N_{L_i}} \Delta_R \omega_j^R [\mathbb{P}_{L_i2R}]_{jl} \mathbf{v}_j^{R,T} \mathbf{K}_l \left(\sum_{m=0}^{N_R} [\mathbb{P}_{R2L_i}]_{lm} \mathbf{v}_m^R - \mathbf{v}_l^{L_i} \right) \\ & + \frac{\lambda_f}{4} \sum_{i=1}^{NE} \sum_{k=0}^{N_{L_i}} \Delta_{L_i} \omega_k^L \mathbf{v}_k^{L_i,T} \mathbf{K}_k \left(\sum_{l=0}^{N_R} [\mathbb{P}_{R2L_i}]_{kl} \mathbf{v}_l^R - \mathbf{v}_k^{L_i} \right). \end{aligned} \quad (5.51)$$

Again, using the modified M-compatibility equation (5.25) we obtain

$$\begin{aligned}
 IS_t = & -\frac{\lambda_f}{4} \sum_{i=1}^{NE} \sum_{j=0}^{N_R} \sum_{l=0}^{N_{L_i}} \Delta_{L_i} \omega_l^{L_i} [\mathbf{P}_{R2L_i}]_{lj} \mathbf{v}_j^{R,T} \mathbf{K}_l \left(\sum_{m=0}^{N_R} [\mathbf{P}_{R2L_i}]_{lm} \mathbf{v}_m^R - \mathbf{v}_l^{L_i} \right) \\
 & + \frac{\lambda_f}{4} \sum_{i=1}^{NE} \sum_{k=0}^{N_{L_i}} \Delta_{L_i} \omega_k^L \mathbf{v}_k^{L_i,T} \mathbf{K}_k \left(\sum_{l=0}^{N_R} [\mathbf{P}_{R2L_i}]_{kl} \mathbf{v}_l^R - \mathbf{v}_k^{L_i} \right).
 \end{aligned} \tag{5.52}$$

Next, we rearrange terms

$$\begin{aligned}
 IS_t = & -\frac{\lambda_f}{4} \sum_{i=1}^{NE} \sum_{l=0}^{N_{L_i}} \Delta_{L_i} \omega_l^{L_i} \left(\sum_{j=0}^{N_R} [\mathbf{P}_{R2L_i}]_{lj} \mathbf{v}_j^R \right)^T \mathbf{K}_l \left(\sum_{m=0}^{N_R} [\mathbf{P}_{R2L_i}]_{lm} \mathbf{v}_m^R - \mathbf{v}_l^{L_i} \right) \\
 & + \frac{\lambda_f}{4} \sum_{i=1}^{NE} \sum_{l=0}^{N_{L_i}} \Delta_{L_i} \omega_l^L \left(\mathbf{v}_l^{L_i} \right)^T \mathbf{K}_l \left(\sum_{m=0}^{N_R} [\mathbf{P}_{R2L_i}]_{lm} \mathbf{v}_m^R - \mathbf{v}_l^{L_i} \right).
 \end{aligned} \tag{5.53}$$

Thus, we have the expression

$$IS_t = -\frac{\lambda_f}{4} \sum_{i=1}^{NE} \sum_{l=0}^{N_{L_i}} \Delta_{L_i} \omega_l^{L_i} \left(\sum_{m=0}^{N_R} [\mathbf{P}_{R2L_i}]_{lm} \mathbf{v}_m^R - \mathbf{v}_l^{L_i} \right)^T \mathbf{K}_l \left(\sum_{m=0}^{N_R} [\mathbf{P}_{R2L_i}]_{lm} \mathbf{v}_m^R - \mathbf{v}_l^{L_i} \right). \tag{5.54}$$

Finally, as \mathbf{K}_l is a symmetric positive definite matrix we know

$$\left(\sum_{m=0}^{N_R} [\mathbf{P}_{R2L_i}]_{lm} \mathbf{v}_m^R - \mathbf{v}_l^{L_i} \right)^T \mathbf{K}_l \left(\sum_{m=0}^{N_R} [\mathbf{P}_{R2L_i}]_{lm} \mathbf{v}_m^R - \mathbf{v}_l^{L_i} \right) \geq 0, \tag{5.55}$$

and as $\lambda_f, \Delta_{L_i}, \omega_l^{L_i} > 0$ for $l = 0, \dots, N_{L_i+1}$ and $i = 1, \dots, NE$ we obtain

$$IS_t \leq 0. \tag{5.56}$$

Thus, the numerical surface fluxes (5.43) and (5.44) lead to a primary conservative and entropy stable scheme for non-linear problems on h/p non-conforming meshes. \square

Note, that the proof of Theorem 11 referred to a mesh with hanging corners. Obtaining the entropy stable flux for pure p refinement is done by setting $NE = 1$. Also, we point out that the dissipation term in (5.43) and (5.44) is non-symmetric. A discussion on non-symmetric dissipation terms can be found in Section 3.4.3 beneath (3.78) and (3.79).

To summarize, in this section we derived a primary conservative and entropy stable scheme for non-linear problems on h/p non-conforming meshes. The derivations combined the machinery for non-conforming SBP methods for the LAE in Section 3.4 with the SBP Flux Difference scheme in Section 4.1.2. First, we described how to create such entropy conservative non-conforming schemes on algebraic non-conforming meshes. Based on these results we extended the scheme for geometric non-conforming meshes. In order to obtain entropy stability we introduced a dissipation term within the numerical flux.

5.2. Numerical Verification of Non-Conforming Summation-by-Parts Schemes for Non-Linear Problems

In this section we verify properties as high-order, primary conservation and entropy stability/conservation of the non-conforming SBP Flux Difference scheme. As in Section 4.3 we consider the two dimensional compressible Euler equations. As in all numerical result sections before (Section 2.3, 3.5 and 4.3), we use the five-stage, fourth-order low-storage Runge-Kutta method of Carpenter and Kennedy[14] to integrate in time. The time step is selected by the CFL condition

$$\Delta t := CFL \frac{\min_i \left\{ \frac{\Delta x_i}{2}, \frac{\Delta y_i}{2} \right\}}{\max_j \{N_j + 1\} \lambda_{\max}}, \quad (5.57)$$

where Δx_i and Δy_i denote the width in x - and y -direction of the i -th element, $N_j + 1$ denotes the number of nodes in the j -th element and λ_{\max} denotes the maximum eigenvalue of the flux Jacobians $\lambda_{\max} = \max\{\lambda_f, \lambda_g\}$ over every node in the whole domain. Note, that this CFL condition is equivalent to (4.104) for a conforming mesh as $N_j = N$ would remain constant for all elements.

In order to apply the SBP Flux Difference scheme we will consider a h/p non-conforming mesh. Therefore, we focus on meshes with a "two-to-one" coupling, see Figure 3.10, and the same projection operators as in Section 3.5. With the projection operators we discuss the construction of the numerical fluxes. As the numerical volume flux is independent of the mesh structure (conforming or non-conforming), we set $\mathbf{f}^\# = \mathbf{f}_{EC}^*$ and $\mathbf{g}^\# = \mathbf{g}_{EC}^*$. For the numerical surface flux, however, we need to consider projection operators for the interface coupling. We set these fluxes to be

$$\mathbf{f}_{ES,k}^{*,L_i} = \mathbf{f}_{EC,k}^{*,L_i} - \sigma \frac{\lambda}{2} \mathbf{K}_k \left(\sum_{l=0}^{N_R} [\mathbf{P}_{R2L_i}]_{kl} \mathbf{v}_l^R - \mathbf{v}_k^{L_i} \right), \quad (5.58)$$

$$\mathbf{f}_{ES,j}^{*,R} = \mathbf{f}_{EC,j}^{*,R} - \sigma \frac{\lambda}{2} \sum_{i=1}^{NE} \left(\sum_{l=0}^{N_{L_i}} [\mathbf{P}_{L_i 2R}]_{jl} \mathbf{K}_l \left(\sum_{m=0}^{N_R} [\mathbf{P}_{R 2L_i}]_{lm} \mathbf{v}_m^R - \mathbf{v}_l^{L_i} \right) \right), \quad (5.59)$$

where these fluxes are equivalent to (5.43) and (5.44) with $\sigma = 1$. We introduced σ to switch between an entropy conservative ($\sigma = 0$) and an entropy stable scheme ($\sigma = 1$). For h refinement we have $NE = 2$ as we focus on a "two-to-one" coupling and for p refinement we have $NE = 1$.

As for all numerical result sections before, we calculate the *EOC* and verify conservation of the primary quantities and entropy for the non-conforming Flux Difference SBP scheme. In addition, we compare this scheme with the Mortar element method [69] to demonstrate the enhanced robustness of entropy stable/conservative schemes.

5.2.1. Comparison and Convergence of Summation-by-Parts Schemes for Non-Linear Problems

In this section we combine the setup of Section 3.5 and 4.3. For the convergence study we consider the isentropic vortex advection problem (4.105) from Section 4.3 with $\Omega = [0, 10] \times [0, 10]$, final time $T = 1$ with $CFL = 0.4$ and Dirichlet boundary conditions. We set $\sigma = 1$ to obtain an entropy stable scheme. The non-conforming mesh for the convergence tests are adapted from Section 3.5, see Figure 3.11, with elements types *A*, *B* and *C*. Considering this mesh we have h , p and h/p non-conforming interfaces. For the first test we analyse the convergence rate of the scheme where all elements have SBP operators with a degree p differentiation matrix. As in Section 3.5 we focus on DP, FD and HGTL operators where

- Element *A* has 22 nodes in x - and y -direction,
- Element *B* has 24 nodes in x - and y -direction,
- Element *C* has 22 nodes in x - and y -direction,

and $p = 2, 3$. As an reminder, FD and HGTL operators consist of a norm matrix of degree $2p - 1$. Therefore, all projection operators are not of degree p , see Section (3.2.3). For the DP operators however the corresponding norm matrix is of degree $2p$ and thus all projection operators are of degree p . Therefore, we intuitively expect a better convergence rate for such SBP operators. We obtain the following \mathcal{L}_2 errors and experimental convergence rates in Table 5.1 and 5.2

EOC for the non-conforming SBP Flux Difference discretization with DP, FD and HGTL operators for $p = 2, 3$.

$DOFS$	\mathcal{L}_2DP	$EOCDP$	\mathcal{L}_2FD	$EOCFD$	\mathcal{L}_2HGTL	$EOCHGTL$
1544	2.42E-01		6.09E-01		6.72E-01	
6176	4.98E-02	2.3	1.65E-01	1.9	1.78E-01	1.9
24704	6.67E-03	2.9	2.75E-02	2.6	2.86E-02	2.6
98816	8.30E-04	3.0	3.61E-03	2.9	3.80E-03	2.9
395264	1.03E-04	3.0	4.84E-04	2.9	5.37E-04	2.8

Table 5.1.: Set $p = 2$, $CFL = 0.4$ and $T = 1$ for the non-conforming SBP Flux Difference scheme.

$DOFS$	\mathcal{L}_2DP	$EOCDP$	\mathcal{L}_2FD	$EOCFD$	\mathcal{L}_2HGTL	$EOCHGTL$
1544	1.06E-01		2.86E-01		2.73E-01	
6176	1.78E-02	2.6	3.83E-02	2.9	3.79E-02	2.8
24704	1.45E-03	3.6	2.01E-03	4.3	1.78E-03	4.4
98816	8.08E-05	4.2	1.17E-04	4.1	7.38E-05	4.6
395264	4.34E-06	4.2	8.06E-06	3.9	5.54E-06	3.7

Table 5.2.: Set $p = 3$, $CFL = 0.4$ and $T = 1$ for the non-conforming SBP Flux Difference scheme.

The presented \mathcal{L}_2 errors in Table 5.1 and 5.2 are averaged errors over all four conserved variables. Here, $DOFS$ refers to the degrees of freedom for each Euler equation. As in Section 3.5 we obtain an EOC of about $p + 1$ for DP operators. The convergence rate for FD and HGTL operators are slightly smaller than $p + 1$ and the errors are larger as for the DP operator. A better EOC and \mathcal{L}_2 error was expected for DP operators, due to projection operators of a higher degree. Comparing FD and HGTL we observe a slightly better error for the FD operators for $p = 2$ and a better error for the HGTL operators for $p = 3$.

Another interesting discussion is the choice of the CFL number. The choice of $CFL = 0.4$ was chosen as the SBP Flux Difference Scheme with DP operators of degree $p = 2, 3$ crashes for $CFL = 0.5$. This even happens for HGTL with $p = 3$. Increasing the CFL number in increments of 0.1 the non-conforming SBP Flux Difference method with FD operators crashes at $CFL = 0.8$ for $p = 2$ and at $CFL = 0.7$ for $p = 3$. Therefore, the non-conforming scheme enjoys the flexibility of taking bigger time steps when considering FD operators.

For considering a mixture of polynomial degrees within the non-conforming SBP Flux Difference scheme we focus on DG operators with the following setup

- Element A with DG operators of degree $p_A = p$ in x - and y -direction,
- Element B with DG operators of degree $p_B = p + 1$ in x - and y -direction,
- Element C with DG operators of degree $p_C = p$ in x - and y -direction,

for $p = 2, 3$. The \mathcal{L}_2 errors and EOC for the non-conforming SBP Flux Difference method with DG operators is provided in Table 5.3 and 5.4.

***EOC* for the non-conforming SBP Flux Difference discretization with DG operators for $p = 2, 3$.**

<i>DOFS</i>	$\mathcal{L}_2 DG$	<i>EOCDG</i>
136	7.75E-02	
544	1.51E-02	2.4
2176	3.30E-03	2.2
8704	7.72E-04	2.1
34816	1.75E-04	2.1
	<i>CFL</i> = 0.5	

Table 5.3.: Set $p = 2$, $CFL = 0.4$ and $T = 1$ for the non-conforming SBP Flux Difference scheme with DG operators.

<i>DOFS</i>	$\mathcal{L}_2 DG$	<i>EOCDG</i>
228	1.99E-02	
912	2.32E-03	3.1
3648	1.95E-04	3.6
14592	1.71E-05	3.5
58368	1.68E-06	3.4
	<i>CFL</i> = 0.5	

Table 5.4.: Set $p = 3$, $CFL = 0.4$ and $T = 1$ for the non-conforming SBP Flux Difference scheme with DG operators.

In Table 5.3 and 5.4 the non-conforming SBP Flux Difference scheme obtains $EOC \approx p$ for an even p and $EOC \approx p + \frac{1}{2}$ for an odd p . Not obtaining a convergence rate of $p + 1$ matches the result for the non-conforming SBP method with DG operators for the LAE in Section 3.5. However, the \mathcal{L}_2 error is significantly small compared with those of the DP, FD and HGTL operator despite the fact we have a smaller EOC . As considering elements where the differentiation matrix is of higher degree (element type B is of degree $p + 1$) a smaller error is reasonable. More interesting is the fact that this setup, where certain DG operators are of degree $p + 1$, is not restricted to a CFL number of 0.4 like for the DP

and HGTL operators. When increasing the CFL number in increments of 0.1, the non-conforming SBP Flux Difference scheme with DG operators crashes at $CFL = 0.7$ for $p = 2$ and at $CFL = 0.6$ for $p = 3$.

To summarize, considering DP operators for the non-conforming SBP Flux Difference discretization we obtain an $EOC \approx p + 1$. For FD and HGTL operators the EOC is slightly smaller. However, considering the FD operators we can take larger time steps. For DG operators the time step needs to be chosen smaller than for the FD operators, but larger than for DP and HGTL operators. Even though the DG operators suffer from a loss in the EOC , their \mathcal{L}_2 error is significantly small. Due to all these different observations (EOC , \mathcal{L}_2 error and CFL number) we cannot claim which operator is the "best", but all operators can be taken into consideration for the entropy stable non-conforming SBP Flux Difference method.

5.2.2. Numerical Verification of Primary Conservation and Entropy Stability

In this section we numerically verify primary conservation and entropy stability/conservation for the SBP Flux Difference scheme and compare this method with the state-of-the-art Mortar element approach developed by Kopriva [69]. The Mortar element method was constructed based on DG operators. Therefore, we focus on such operators throughout this section.

First, we verify that the scheme is fully conservative. Therefore we set $\sigma = 0$ within (5.58) and (5.59). We consider the mesh in Figure 3.11(c) with DG operators of degree $p_A = p_C = 3$ and $p_B = 4$. The methodology of verifying that the scheme is fully conservative is done in an analogous fashion as in Section 4.3. We calculated the growth of the total discrete primary quantity \mathbf{U}_t^{tot} and entropy S_t^{tot} by (4.112) with the random initial condition (4.113). We repeat these calculations 1000 times and obtain $(\mathbf{U}_t^{tot})_{lk}$ and $(S_t^{tot})_k$ for $k = 1, \dots, 1000$ and $l = 1, \dots, 4$. Within the euclidean norm we obtain the results in Table 5.5. Here, we verify primary and entropy conservation. In comparison, when considering the same setup with the Mortar element method we verify primary conservation but not entropy conservation, see Table 5.6. To demonstrate the increased robustness of entropy stable schemes we calculate the discrete entropy in time by (4.114) with the initial condition (4.115). We plot the total discrete entropy for the SBP Flux Difference scheme and the Mortar element method in Figure 5.2. Here, the SBP Flux Difference scheme conserves the total entropy. However, for the mortar method we observe an unpredictable behavior of the entropy and note that at $t \approx 0.82$ it even crashes. This has been verified for $CFL = 0.5; 0.25; 0.125; 0.0625$ and demonstrates the enhanced robustness of entropy conserving/stable schemes.

Verification of primary and entropy conservation for the non-conforming SBP Flux Difference method

$\ \mathbf{S}_t^{tot}\ _2$	$\ (\mathbf{U}_t^{tot})_{1,:}\ _2$	$\ (\mathbf{U}_t^{tot})_{2,:}\ _2$	$\ (\mathbf{U}_t^{tot})_{3,:}\ _2$	$\ (\mathbf{U}_t^{tot})_{4,:}\ _2$
6.31e-14	3.98E-15	2.38E-14	1.22E-15	8.44E-14

Table 5.5.: Calculating the growth of the total discrete primary quantities \mathbf{U}_t^{tot} and entropy \mathbf{S}_t^{tot} for 1000 different random initial conditions with the non-conforming SBP Flux Difference discretization. The average growth is presented within the euclidean norm. All values are near machine precision which demonstrates primary and entropy conservation.

Calculating the growth in the primary quantities and entropy with the non-conforming Mortar method

$\ \mathbf{S}_t^{tot}\ _2$	$\ (\mathbf{U}_t^{tot})_{1,:}\ _2$	$\ (\mathbf{U}_t^{tot})_{2,:}\ _2$	$\ (\mathbf{U}_t^{tot})_{3,:}\ _2$	$\ (\mathbf{U}_t^{tot})_{4,:}\ _2$
1.36	3.01E-14	1.84E-14	2.30E-14	8.41E-14

Table 5.6.: Calculating the growth of the total discrete primary quantities \mathbf{U}_t^{tot} and entropy \mathbf{S}_t^{tot} for 1000 different random initial conditions with the Mortar method. The growth is presented within the euclidean norm. Here, we verify conservation of the primary quantities but not of the entropy.

For the robust non-conforming SBP Flux Difference scheme we run the simulation to $T = 25$ with $CFL = 0.5$. Besides conservation, we also run this test case for an entropy stable flux setting $\sigma = 1$. The total discrete entropy and primary quantities of the non-conforming SBP Flux Difference scheme are plotted in Figure 5.3 and 5.4. Here, we see that the total discrete entropy is conserved when no interface dissipation is included ($\sigma = 0$) and that the total discrete entropy decays with interface dissipation ($\sigma = 1$). The total discrete primary quantities in Figure 5.4 remains constant over time no matter if dissipation is included or not. Thus we numerically verified primary conservation and entropy stability/conservation of the h/p non-conforming SBP Flux Difference method.

5.3. Summary

In this chapter we derived the h/p non-conforming primary conservative and entropy stable SBP Flux Difference scheme for non-linear problems. The idea was to combine the non-conforming SBP method for linear problems from Chapter 3 and

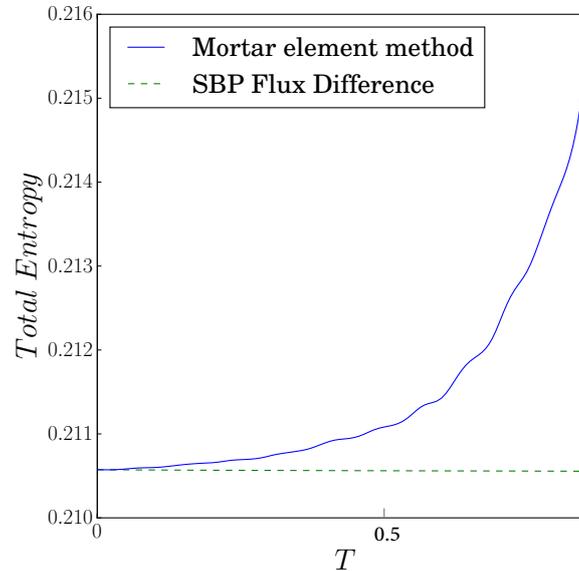


Figure 5.2.: Evolution of the total discrete entropy. The SBP Flux Difference scheme conserves the total discrete entropy whereas the entropy becomes unstable for the Mortar Method at $t \approx 0.82$.

the conforming SBP Flux Difference method for non-linear problems from Chapter 4. We presented the modification to extend the projection procedure in the context of non-conforming linear problems to non-conforming non-linear problems in a stable manner. For linear problems the numerical surface flux was calculated by projecting the conserved variables from the neighbors into the corresponding element and then including this projected neighbor value into the numerical surface flux (5.1). Using this strategy for non-linear problems gives us non-linear projection terms which do not necessary lead to an entropy stable scheme. The key of obtaining such a scheme for non-linear problems is not projecting the conserved variables, but the numerical flux itself (5.10). Based on this modification we only consider linear projections and thus we obtained the primary conservative and entropy stable SBP Flux Difference scheme for h/p non-conforming meshes, where neither the nodes nor the interface of two neighboring elements need to coincide. We implemented the method and applied it to the two dimensional compressible Euler equations for different SBP operators. Here, we demonstrated high-order accuracy of the scheme. Throughout this work, we are interested in primary conservation and entropy stability. We demonstrate these properties for

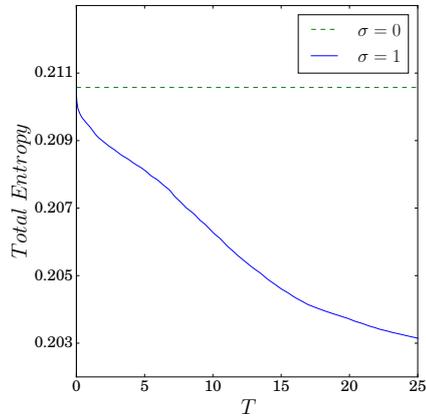


Figure 5.3.: Evolution of the total entropy of the solution using the non-conforming Ismail-Roe flux with and without interface dissipation. Here, the total entropy is constant or decays depending on σ .

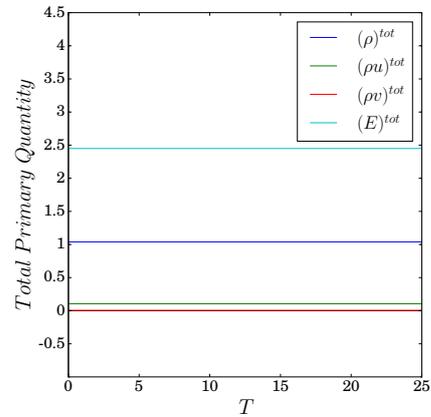


Figure 5.4.: A plot that demonstrates the conservation of the primary quantities using the non-conforming Ismail-Roe flux. We obtain the same plots with and without interface dissipation.

the non-conforming SBP Flux Difference scheme. Also, we compared this method with the well-known Mortar element method [69], which is designed to be primary conservative but is not entropy stable/conservative. Through numerical tests we verified that the entropy stable SBP Flux Difference scheme is more robust. This, again, shows the benefit of considering entropy stable schemes.

6. Entropy Stable and Primary Conservative Summation-by-Parts Space-Time Methods for Non-Linear Hyperbolic Systems

In the last chapters we derived entropy stable schemes on non-conforming meshes. All introduced schemes were built on semi-discrete discretizations (method of lines), where we discretized the spatial domain and assumed continuity in time. Focusing on a one dimensional conservation law, we constructed entropy conservative discretizations such that

$$\int_{\Omega} \frac{\partial \mathcal{S}}{\partial t} d\Omega \approx \sum_{q=1}^{N_Q} \frac{\Delta x}{2} \sum_{i=0}^N \omega_i \frac{\partial S_i^q}{\partial t} = 0. \quad (6.1)$$

Focusing on a smooth solution we know that the continuous total entropy is conserved over time due to the fundamental theorem of calculus

$$\int_{\Omega} \mathcal{S}(T, \cdot) d\Omega = \int_{\Omega} \mathcal{S}(0, \cdot) d\Omega. \quad (6.2)$$

When calculating the total discrete entropy with a semi-discrete scheme we do not necessarily obtain the discrete version of (6.2) to machine precision. For semi-discrete discretization we get a discrete system of ODEs for the conserved variables which are then solved by a Runge-Kutta time integration method. Due to the time integration method we introduce temporal errors within the entropy. When numerically verifying entropy conservation of a scheme we need to shrink the time step as in [7, 10, 17, 20, 21, 36] to reduce the time integration error. So in order to conserve the total discrete entropy we are limited to a small time step.

In this chapter we introduce a fully-discrete numerical scheme which bounds the total discrete entropy in a space-time slab. In the literature there are high-order entropy conservative (or stable) space-time schemes available, e.g. [29, 37], however they assume exact integration of the variational forms in the space-time formulation. This is problematic because aliasing errors introduced by inexact

quadratures are unavoidable (or at least cannot be avoided in practical simulations) for conservation laws like the compressible Euler equations, e.g. [46, 88]. So, the design of numerical methods that are entropy stable/conservative in a fully-discrete space-time domain is an important step in the development of entropy stable numerical methods.

Here, we introduce the SBP space-time method developed by Friedrich et al. [40]. This scheme referred to discontinuous Galerkin operators which satisfy the SBP property. However, it can be applied to any SBP operator. The crux is to approximate the discrete derivative in time with SBP operators, as we previously did in space, which leads to a fully-discrete space-time scheme. There is past work applying SBP operators to temporal derivatives to prove energy stability for linear problems [8, 79, 91]. Friedrich et al. [40] generalized this SBP stability analysis to non-linear problems and derived a fully-discrete entropy analysis to approximate solutions of a general non-linear conservation law with inexact numerical integration.

6.1. Discrete Analysis for Summation-by-Parts Space-Time Methods

In this section we derivative the SBP space-time method. As the main focus is on the temporal discretization we consider (for simplicity) a one dimensional conservation law

$$\frac{\partial \mathbf{u}}{\partial t} + \frac{\partial \mathcal{F}(\mathbf{u})}{\partial x} = 0, \quad (6.3)$$

on $\Omega = [x_L, x_R]$ and $t \in [0, T]$. The extension to multiple spatial dimensions is straight forward as presented in [40]. The SBP space-time method relies on the Flux Difference discretization by Fisher and Carpenter [33]. Therefore, we review the theory on entropy conservative finite volume schemes in Section 6.1.1. Based on the finite volume result we derive the high order SBP space-time method in Section 6.1.2. With the space-time method we discuss the advantages of applying such a scheme to a non-conforming mesh in Section 6.1.3.

6.1.1. Review of Finite Volume Methods

We review the theory on entropy conservative finite volume methods. For a semi-discrete analysis this has been done in Section 4.1.2. The semi-discrete low-order finite volume discretization is given by integration of (6.3) in space and considering N_Q elements with the solution vector \mathbf{u}_j for $j = 1, \dots, N_Q$. We obtain

$$\frac{\partial \mathbf{u}_{j+1}}{\partial t} + \frac{1}{\Delta x} (\mathbf{f}_{j+1}^* - \mathbf{f}_j^*) = 0, \quad (6.4)$$

with $j = 0, \dots, N_Q - 1$ and

$$\mathbf{f}_j^* := \mathbf{f}^*(\mathbf{u}_{j+1}, \mathbf{u}_j), \quad (6.5)$$

for $j = 1, \dots, N_Q - 1$ and $\mathbf{f}_0^* = \mathbf{f}(\mathbf{u}_1)$ and $\mathbf{f}_{N_Q}^* = \mathbf{f}(\mathbf{u}_{N_Q})$. For simplicity we assume that all elements are uniformly distributed with width Δx . The semi-discrete finite volume method (6.4) is provable entropy conservative if the numerical surface flux satisfies the Tadmor Shuffle condition (5.9), see Section 4.1.2.

Fully-discrete finite volume methods are obtained by not only integrating (6.3) in space, but also in time. We describe such a scheme by

$$\frac{1}{\Delta t} (\mathbf{u}_{i+1,j+1}^* - \mathbf{u}_{i,j+1}^*) + \frac{1}{\Delta x} (\mathbf{f}_{i,j+1}^* - \mathbf{f}_{i,j}^*) = 0, \quad (6.6)$$

with $j = 0, \dots, N_Q - 1, i = 1, \dots, N_t$ where N_t denotes the number of elements in time and

$$\mathbf{u}_{i,j}^* := \mathbf{u}^*(\mathbf{u}_{i+1,j}, \mathbf{u}_{i,j}), \quad (6.7)$$

for $i = 2, \dots, N_t$ and $\mathbf{u}_{1,j}^* = \mathbf{u}_{1,j}$ and $\mathbf{u}_{N_t+1,j}^* = \mathbf{u}_{N_t,j}$. Here, we note that (6.6) is an atypical formulation of a finite volume method as we introduced a numerical state function for the conserved variables \mathbf{u}^* as in [49]. The origin of introducing such states was to handle discontinuous solutions by approximating a Riemann problem. Here, the solution to the Riemann problem is given by an upwind state, see [116]. Even from a physical point of view this is logical as time moves linearly forward. With the upwind state $\mathbf{u}_{i,j}^* = \mathbf{u}_{i,j}$ the discretization remains to be

$$\frac{1}{\Delta t} (\mathbf{u}_{i+1,j+1} - \mathbf{u}_{i,j+1}) + \frac{1}{\Delta x} (\mathbf{f}_{i,j+1}^* - \mathbf{f}_{i,j}^*) = 0, \quad (6.8)$$

which is equivalent to

$$\mathbf{u}_{i+1,j+1} = \mathbf{u}_{i,j+1} - \frac{\Delta t}{\Delta x} (\mathbf{f}_{i,j+1}^* - \mathbf{f}_{i,j}^*). \quad (6.9)$$

Here (6.9) is a more commonly used description of a fully-discrete finite volume method and can be found in e.g. [74]. Also (6.9) algebraically equivalent to the semi-discrete formulation (6.4) when applying the explicit Euler time integration method (index $i + 1$ on the left hand side, index i on the right hand side). Still, for further extension we focus on discretization (6.6) with a general numerical state.

In order to derive a fully-discrete entropy conservative scheme, we want the scheme to mimic (6.2). Therefore, we need to specify the numerical state \mathbf{u}^* . In the entropy analysis we contract (6.6) with the entropy variables and sum over all nodes in space and time

$$\Delta x \sum_{i=1}^{N_t} \sum_{j=1}^{N_Q} \mathbf{v}_{i,j}^T (\mathbf{u}_{i+1,j}^* - \mathbf{u}_{i,j}^*) + \Delta t \sum_{i=1}^{N_t} \sum_{j=1}^{N_Q} \mathbf{v}_{i,j}^T (\mathbf{f}_{i,j}^* - \mathbf{f}_{i,j-1}^*) = 0. \quad (6.10)$$

Assuming that \mathbf{f}^* satisfies the Tadmor Shuffle condition (5.9) then we found in Section 4.1.2 that

$$\sum_{j=1}^{N_Q} \mathbf{v}_{i,j}^T (\mathbf{f}_{i,j}^* - \mathbf{f}_{i,j-1}^*) = f_{i,N_Q}^{ent} - f_{i,1}^{ent}, \quad (6.11)$$

so the sum reduces to the evaluation of the entropy flux on the boundary of the domain. Considering periodic boundary conditions it holds $f_{i,N_Q}^{ent} = f_{i,1}^{ent}$ and thus (6.10) reduces to

$$\Delta x \sum_{j=1}^{N_Q} \left(\underbrace{\sum_{i=1}^{N_t} \mathbf{v}_{i,j}^T (\mathbf{u}_{i+1,j}^* - \mathbf{u}_{i,j}^*)}_{(*)} \right) = 0. \quad (6.12)$$

For simplicity we analyse the term within the brackets. As the second index j remains constant we ignore it for now.

$$\begin{aligned} (*) &= \sum_{i=1}^{N_t} \mathbf{v}_i^T (\mathbf{u}_{i+1}^* - \mathbf{u}_i^*) \\ &= \mathbf{v}_{N_t}^T \mathbf{u}_{N_t} - \mathbf{v}_1^T \mathbf{u}_1 - \sum_{i=1}^{N_t-1} (\mathbf{v}_{i+1} - \mathbf{v}_i)^T \mathbf{u}_i^*. \end{aligned} \quad (6.13)$$

Next, to obtain the entropy function we use the definition of entropy potential Ψ^u in (4.6) where $\Psi^u = \mathbf{v}^T \mathbf{u} - S$ to find

$$(*) = S_{N_t} + \Psi_{N_t}^u - S_1 - \Psi_1^u - \sum_{i=1}^{N_t-1} (\mathbf{v}_{i+1} - \mathbf{v}_i)^T \mathbf{u}^*(\mathbf{u}_{i+1}, \mathbf{u}_i). \quad (6.14)$$

Finally, as $\Psi_{N_t}^u - \Psi_1^u = \sum_{i=1}^{N_t-1} (\Psi_{i+1}^u - \Psi_i^u)$ we arrive at

$$(*) = S_{N_t} - S_1 - \left(\sum_{i=1}^{N_t-1} (\mathbf{v}_{i+1} - \mathbf{v}_i)^T \mathbf{u}^*(\mathbf{u}_{i+1}, \mathbf{u}_i) - (\Psi_{i+1}^u - \Psi_i^u) \right). \quad (6.15)$$

Including this in (6.12) and incorporating the index j we have

$$\begin{aligned}
 & - \Delta x \sum_{j=1}^{N_Q} \left(\sum_{i=1}^{N_t-1} (\mathbf{v}_{i+1,j} - \mathbf{v}_{i,j})^T \mathbf{u}^*(\mathbf{u}_{i+1,j}, \mathbf{u}_{i,j}) - (\Psi_{i+1,j}^u - \Psi_{i,j}^u) \right) \\
 & + \Delta x \sum_{j=1}^{N_Q} (S_{N_t,j} - S_{1,j}) = 0.
 \end{aligned} \tag{6.16}$$

Assuming that the numerical state \mathbf{u}^* satisfies the condition

$$\sum_{i=1}^{N_t-1} (\mathbf{v}_{i+1} - \mathbf{v}_i)^T \mathbf{u}^*(\mathbf{u}_{i+1}, \mathbf{u}_i) = \Psi_{i+1}^u - \Psi_i^u, \tag{6.17}$$

then (6.16) reduces to

$$\Delta x \sum_{j=1}^{N_Q} S_{N_t,j} = \Delta x \sum_{j=1}^{N_Q} S_{1,j}. \tag{6.18}$$

which is a discrete mimic of (6.2). Thus, choosing the numerical state of the conservative variables to satisfy (6.17) we obtain a fully-discrete entropy conservative scheme.

Obviously, the condition (6.17) is closely related to the Tadmor Shuffle condition (5.9). This is not unexpected as the derivations of the flux has been done in a one-to-one fashion as in Section 4.1.2. Here, we denote (6.17) as the *time dependent Tadmor Shuffle condition*.

A disadvantage of the fully-discrete finite volume scheme (6.10) is that we have to solve all degrees of freedom simultaneously, as the system is fully coupled. For the fully coupled finite volume scheme (6.10) this gives a high computational cost. Therefore, the practicality of such an entropy conservative scheme is questionable. In comparison, a semi-discrete scheme calculates a numerical solution time step by time step. Thus, information is only needed from the last time step which avoids the global coupling. In [40], Friedrich et al. developed a SBP scheme considering numerical volume and surface states for \mathbf{u} as for the Flux Difference method of Fisher and Carpenter [33]. In order to calculate the numerical solution time slab by time slab, the numerical surface state is chosen to be an upwind state. The volume flux, however, is set to satisfy the time dependent Tadmor Shuffle condition (6.17). Therefore, we construct a fully-discrete entropy stable scheme where the computational cost is competitive with a semi-discrete scheme using an implicit Runge-Kutta time integration method in time [8].

6.1.2. Summation-by-Parts Space-Time Methods

In this section we will derive the primary conservative and entropy stable SBP space-time scheme of Friedrich et al. [40]. Therefore, we focus on a one dimensional system of conservation laws (6.3) on $\Omega = [x_L, x_R]$ with the time dimension $t \in [0, T]$. We treat the temporal dimension like a spatial dimension. As such, we discretize the space-time domain $[0, T] \times \Omega$ into non-overlapping quadrilateral elements is illustrated in Figure 6.1.

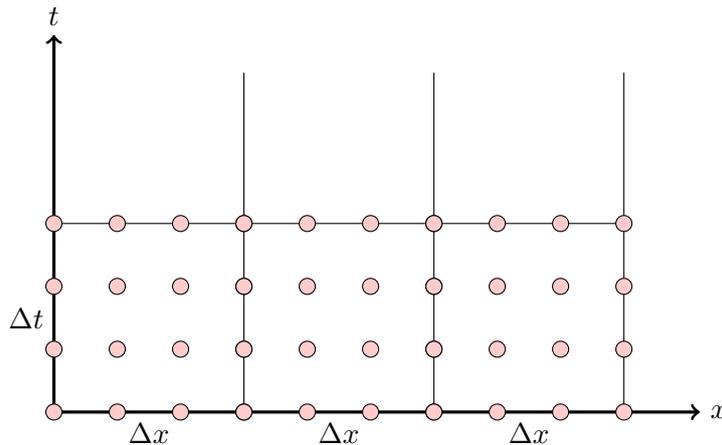


Figure 6.1.: Space-time discretization with red marked *DOFS* on a conforming mesh in a space-time slab. The figure is reproduced from [41].

Let N_t denote the number of elements in time and N_Q denote the number of elements in space. For simplicity, we assume that all elements are distributed uniformly with width Δx and height Δt . Note, that even though we focus on a conservation law with only one spatial dimension the space-time scheme is a two dimensional discretization as we also discretize in time. We denote the (m, n) -th element for $m = 1, \dots, N_t$ and $n = 1, \dots, N_Q$ with $E_{m,n} := [t_m, t_{m+1}] \times [x_n, x_{n+1}]$. We derive the space-time discretization using the DG ansatz as in [40]. Thus, we multiply the conservation law with an arbitrary discontinuous test function $\phi := \phi(t, x)$ and integrate in space and time over the element $E_{m,n}$

$$\int_{E_{m,n}} \left(\frac{\partial \mathbf{u}}{\partial t} + \frac{\partial \mathcal{F}(\mathbf{u})}{\partial x} \right) \phi \, dE_{m,n} = 0. \quad (6.19)$$

As for semi-discrete schemes we transform (6.19) from the space-time element $E_{m,n}$ into the reference space $\hat{\Omega} = [-1, +1]^2$. The transformation is given by the affine

maps

$$\begin{aligned}\tau(t) &= 2\frac{t - t_m}{t_{m+1} - t_m} - 1, \\ \xi(x) &= 2\frac{x - x_n}{x_{n+1} - x_n} - 1.\end{aligned}\tag{6.20}$$

Note, that in comparison to a semi-discrete discretization we also transform the time variable into the $[-1 + 1]$. This gives

$$\frac{2}{\Delta t} \int_{-1}^1 \int_{-1}^1 \frac{\partial \mathbf{u}}{\partial \tau} \phi \, d\tau d\xi + \frac{2}{\Delta x} \int_{-1}^1 \int_{-1}^1 \frac{\partial \mathcal{F}}{\partial \xi} \phi \, d\tau d\xi = 0.\tag{6.21}$$

To obtain the weak formulation as in Section 2.1.3 we perform integration-by-parts in t and x to obtain

$$\begin{aligned}& \frac{2}{\Delta t} \int_{-1}^1 \left(\mathbf{u} \phi \Big|_{-1}^{+1} - \int_{-1}^1 \mathbf{u} \frac{\partial \phi}{\partial \tau} \, d\tau \right) d\xi \\ & + \frac{2}{\Delta x} \int_{-1}^1 \left(\mathcal{F} \phi \Big|_{-1}^{+1} - \int_{-1}^1 \mathcal{F} \frac{\partial \phi}{\partial \xi} \, d\xi \right) d\tau = 0.\end{aligned}\tag{6.22}$$

Here, we remove derivatives from \mathbf{u} and \mathcal{F} onto the test function ϕ . For the space-time DGSEM ansatz we approximate the solution \mathbf{u} and the flux \mathcal{F} on the element $E_{m,n}$ by polynomials

$$\begin{aligned}\mathbf{u}(\tau, \xi) &\approx \sum_{i,j=0}^N \mathbf{u}_{ij} \ell_i(\tau) \ell_j(\xi), \\ \mathcal{F}(\tau, \xi) &\approx \sum_{i,j=0}^N \mathbf{f}_{ij} \ell_i(\tau) \ell_j(\xi),\end{aligned}\tag{6.23}$$

where ℓ_i is a Lagrange polynomial of degree $p = N$ defined as in (2.75). Here, we set the nodal distribution τ_i and ξ_i for $i = 0, \dots, N$ to be the LGL nodes. In general the number of nodes can be chosen differently and, therefore, it is possible for the Lagrange polynomials in τ - and ξ -directions to be polynomials of different degrees. However, for simplicity, we set the nodes in τ - and ξ -direction to be the same. To derive a space-time scheme of degree p we assume that ϕ consists of a basis of the Lagrange polynomials in terms of

$$\phi := \sum_{i,j=0}^N \phi_{ij} \ell_i(\tau) \ell_j(\xi).\tag{6.24}$$

As the choice of ϕ_{ij} is arbitrary we seek a numerical solution which satisfies

$$\begin{aligned} & \frac{2}{\Delta t} \int_{-1}^1 \left(\mathbf{u} \ell_i \Big|_{-1}^{+1} - \int_{-1}^1 \mathbf{u} \frac{\partial \ell_i}{\partial \tau} d\tau \right) \ell_j(\xi) d\xi \\ & + \frac{2}{\Delta x} \int_{-1}^1 \left(\mathcal{F} \ell_j \Big|_{-1}^{+1} - \int_{-1}^1 \mathcal{F} \frac{\partial \ell_j}{\partial \xi} d\xi \right) \ell_i(\tau) d\tau = 0, \end{aligned} \quad (6.25)$$

for all $i, j = 0, \dots, N$. Inserting the DG approximation (6.23), numerical surface flux \mathbf{f}^* and state \mathbf{u}^* and approximating the integrals with LGL quadrature as in Section 2.2 we obtain the implicit weak space-time discretization

$$\begin{aligned} & \frac{2\omega_j}{\Delta t} \left(\mathbf{u}_{Nj}^* \delta_{iN} - \mathbf{u}_{0j}^* \delta_{i0} - \sum_{l=0}^N \mathbf{Q}_{li} \mathbf{u}_{lj} \right) \\ & + \frac{2\omega_i}{\Delta x} \left(\mathbf{f}_{iN}^* \delta_{Nj} - \mathbf{f}_{i0}^* \delta_{0j} - \sum_{l=0}^N \mathbf{Q}_{lj} \mathbf{f}_{il} \right) = 0. \end{aligned} \quad (6.26)$$

Performing summation-by-parts we get the algebraically equivalent implicit strong space-time formulation

$$\begin{aligned} & \frac{2\omega_j}{\Delta t} \left([\mathbf{u}^* - \mathbf{u}]_{Nj} \delta_{iN} - [\mathbf{u}^* - \mathbf{u}]_{0j} \delta_{i0} + \sum_{l=0}^N \mathbf{Q}_{il} \mathbf{u}_{lj} \right) \\ & + \frac{2\omega_i}{\Delta x} \left([\mathbf{f}^* - \mathbf{f}]_{iN} \delta_{Nj} - [\mathbf{f}^* - \mathbf{f}]_{i0} \delta_{0j} + \sum_{l=0}^N \mathbf{Q}_{jl} \mathbf{f}_{il} \right) = 0. \end{aligned} \quad (6.27)$$

However, from the semi-discrete analysis we know that this discretization is not entropy stable. Therefore, we replace the volume flux in time and space by the numerical volume state $\mathbf{u}^\#$ and flux $\mathbf{f}^\#$. By following the same idea as for Flux Difference discretization (4.64) we now have

$$\begin{aligned} & \frac{2\omega_j}{\Delta t} \left([\mathbf{u}^* - \mathbf{u}]_{Nj} \delta_{iN} - [\mathbf{u}^* - \mathbf{u}]_{0j} \delta_{i0} + 2 \sum_{l=0}^N \mathbf{Q}_{il} \mathbf{u}^\#(\mathbf{u}_{lj}, \mathbf{u}_{ij}) \right) \\ & + \frac{2\omega_i}{\Delta x} \left([\mathbf{f}^* - \mathbf{f}]_{iN} \delta_{Nj} - [\mathbf{f}^* - \mathbf{f}]_{i0} \delta_{0j} + 2 \sum_{l=0}^N \mathbf{Q}_{jl} \mathbf{f}^\#(\mathbf{u}_{il}, \mathbf{u}_{ij}) \right) = 0, \end{aligned} \quad (6.28)$$

where we set the numerical volume fluxes to $\mathbf{f}^\# = \mathbf{f}_{EC}^*$ and $\mathbf{u}^\# = \mathbf{u}_{EC}^*$ with

$$(\mathbf{v}_{il} - \mathbf{v}_{ij})^T \mathbf{f}_{EC}^*(\mathbf{u}_{il}, \mathbf{u}_{ij}) = \Psi_{il}^f - \Psi_{ij}^f, \quad (6.29)$$

$$(\mathbf{v}_{lj} - \mathbf{v}_{ij})^T \mathbf{u}_{EC}^*(\mathbf{u}_{lj}, \mathbf{u}_{ij}) = \Psi_{lj}^u - \Psi_{ij}^u, \quad (6.30)$$

where \mathbf{v} represent the entropy variables and Ψ^u and Ψ^f are the entropy potential and entropy flux potential. There are numerous fluxes for \mathbf{f}_{EC}^* that satisfy (6.29) as pointed out in Section 4.1.2. Numerical volume states for which (6.30) holds have been derived for the compressible Euler, Shallow Water and Ideal MHD equations and can be found in [40]. The modification of including the numerical volume flux and state does not violate the high-order accuracy as shown by Fisher and Carpenter [33]. The surface flux will be specified later.

Next, we define primary conservation and entropy stability for a fully-discrete system. We motivate the definition of primary conservation by focusing on the one dimensional conservation law (6.3). Integrating in space $[x_L, x_R]$ and time $[0, T]$ we get

$$\begin{aligned} & \int_0^T \int_{x_L}^{x_R} \frac{\partial \mathbf{u}}{\partial t} dx dt + \int_0^T \int_{x_L}^{x_R} \frac{\partial \mathcal{F}}{\partial x} dx dt = \mathbf{0}, \\ \Leftrightarrow & \int_{x_L}^{x_R} (\mathbf{u}(T, x) - \mathbf{u}(0, x)) dx + \int_0^T (\mathcal{F}(t, x_R) - \mathcal{F}(t, x_L)) dt = \mathbf{0}. \end{aligned} \quad (6.31)$$

Assuming periodic boundary conditions in space ($\mathcal{F}(t, x_R) = \mathcal{F}(t, x_L)$) we arrive at

$$\int_{x_L}^{x_R} \mathbf{u}(T, x) dx = \int_{x_L}^{x_R} \mathbf{u}(0, x) dx. \quad (6.32)$$

We want the scheme to mimic (6.32). Therefore, we need to include further notation. As we consider N_t elements in time and N_Q elements in space, we denote the solution on the element $E_{m,n}$ for $m = 1, \dots, N_t$ and $n = 1, \dots, N_Q$ by \mathbf{u}^{mn} , see Figure 6.2. The definition of primary conservation for space-time methods is given in Definition 15.

Definition 15. Primary Conservative Schemes for Space-Time Methods

A fully-discrete method approximating one dimensional hyperbolic systems of conservation laws is primary conservative if

$$\sum_{n=1}^{N_Q} \frac{\Delta x}{2} \sum_{j=0}^N \omega_j \mathbf{u}_{N_j}^{N_t, n} = \sum_{n=1}^{N_Q} \frac{\Delta x}{2} \sum_{j=0}^N \omega_j \mathbf{u}(t=0, x^n(\xi_j)), \quad (6.33)$$

assuming periodic boundary conditions in space. Here x^n refers to a mapping from the reference space into the physical spatial space of $E_{1,n}$.

The definition of entropy stability is done in a similar way. By contracting the conservation law (6.3) with the entropy variables and integrating in space and time we get

$$\int_{x_L}^{x_R} (\mathcal{S}(T, x) - \mathcal{S}(0, x)) dx + \int_0^T (\mathcal{F}^{ent}(t, x_R) - \mathcal{F}^{ent}(t, x_L)) dt = 0. \quad (6.34)$$

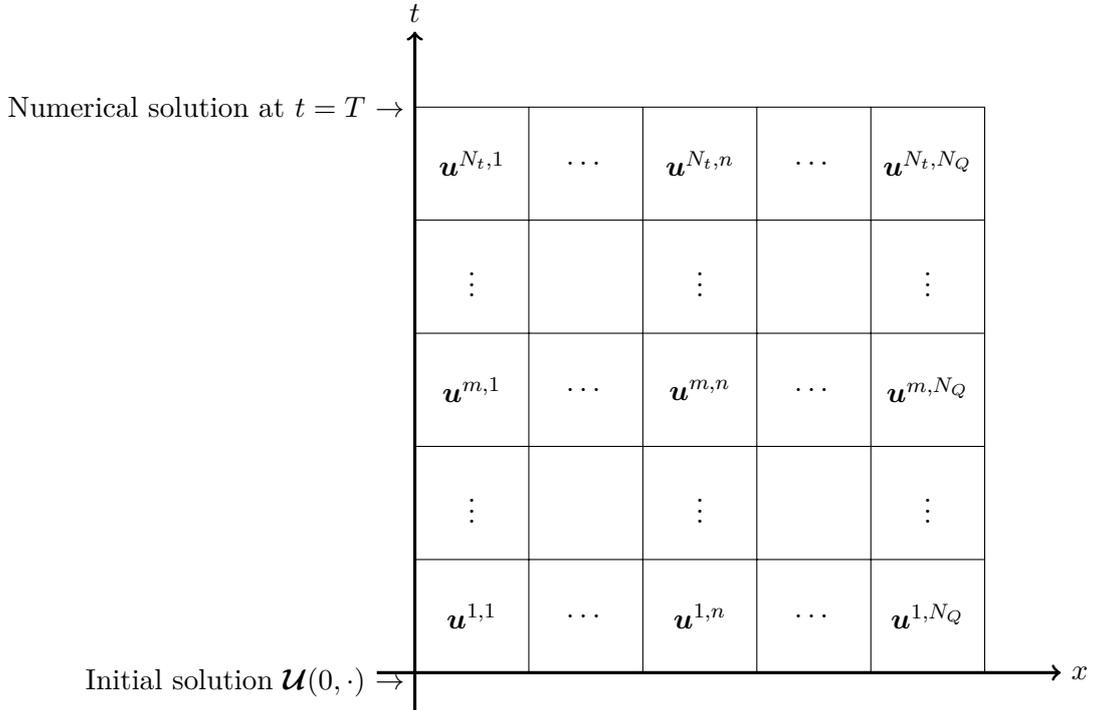


Figure 6.2.: Notation for space-time discretization. The figure is reproduced from [41].

Considering periodic boundary conditions in space and allowing the total entropy to dissipate to reflect the correct physical behaviour we want the numerical scheme to mimic

$$\int_{x_L}^{x_R} \mathcal{S}(T, x) dx \leq \int_{x_L}^{x_R} \mathcal{S}(0, x) dx. \quad (6.35)$$

With this condition we define entropy stability for a fully-discrete scheme.

Definition 16. Entropy Stable Schemes for Space-Time Methods

A fully-discrete method approximating one dimensional hyperbolic systems of conservation laws is entropy stable if

$$\sum_{n=1}^{N_Q} \frac{\Delta x}{2} \sum_{j=0}^N \omega_j S_{N_j}^{N_t, n} \leq \sum_{n=1}^{N_Q} \frac{\Delta x}{2} \sum_{j=0}^N \omega_j \mathcal{S}(t=0, x^n(\xi_j)), \quad (6.36)$$

assuming periodic boundary conditions in space. Again, x^n refers to a mapping from the reference space into the physical spatial space of $E_{1,n}$.

To prove primary conservation we sum over all nodes of the space-time discretization (6.28) in order to mimic the temporal and spatial integration over the conserved variables. Mimicking the integration over the entropy function to prove entropy stability is done by contracting (6.28) with the entropy variables \mathbf{v}_{ij} and summing over all nodes. The resulting terms are presented in Corollary 1. As we focus on a single element we drop the indices (m, n) on \mathbf{u}^* for the sake of clarity.

Corollary 1. *Summing over all nodes of (6.28) gives*

$$\frac{\Delta x}{2} \sum_{j=0}^N \omega_j (\mathbf{u}_{Nj}^* - \mathbf{u}_{0j}^*) + \frac{\Delta t}{2} \sum_{i=0}^N \omega_i (\mathbf{f}_{iN}^* - \mathbf{f}_{i0}^*) = \mathbf{0}. \quad (6.37)$$

In addition, contracting (6.28) with the entropy variables \mathbf{v}_{ij} and summing over all nodes yields

$$\begin{aligned} & \frac{\Delta x}{2} \sum_{j=0}^N \omega_j \left([\mathbf{v}^T \mathbf{u}^* - \Psi^u]_{Nj} - [\mathbf{v}^T \mathbf{u}^* - \Psi^u]_{0j} \right) \\ & + \frac{\Delta t}{2} \sum_{i=0}^N \omega_i \left([\mathbf{v}^T \mathbf{f}^* - \Psi^f]_{iN} - [\mathbf{v}^T \mathbf{f}^* - \Psi^f]_{i0} \right) = 0. \end{aligned} \quad (6.38)$$

Proof. The proof follows identical steps as given for Lemma 5. \square

As for the continuous analysis the approximated integrals reduce to surface evaluations. It is important to note, that this happens due to SBP operators. Next, we examine the surface contributions. Here, we need to specify the numerical surface fluxes. The surface flux \mathbf{f}^* is constructed using $\mathbf{f}^\#$ and including interface dissipation as in Chapter 4, Equation (4.88). We denote this surface flux by \mathbf{f}_{ES}^* . As time always moves forward, it is reasonable to choose \mathbf{u}^* to be an upwind state. Therefore, we can solve for \mathcal{U} time slab by time slab and need only store the numerical solution at the current time slab due to the weak coupling in time. This leads to a non fully coupled system. As we consider a SBP operator for the time integration, the SBP space-time scheme is (from a computational efficiency standpoint) compatible with a semi-discrete scheme with an implicit Runge-Kutta method with N Runge-Kutta steps [8]. We denote the upwind state as \mathbf{u}_{ES}^* .

We reiterate, that numerical volume states satisfying the time dependent Tadmor Shuffle condition (6.17) have been derived for compressible Euler, Shallow Water and Ideal MHD equations in [40]. In addition, Friedrich et al. have shown that for these set of equations the upwind state satisfies the inequality

$$(\mathbf{v}_{lj} - \mathbf{v}_{ij})^T \mathbf{u}_{ES}^*(\mathbf{u}_{lj}, \mathbf{u}_{ij}) \leq \Psi_{lj}^u - \Psi_{ij}^u. \quad (6.39)$$

With this inequality we prove entropy stability of the fully-discrete SBP space-time method.

Theorem 12. *Consider the SBP space-time scheme (6.28) with Dirichlet boundary conditions in time. Assume that the temporal numerical surface flux is the upwind state $\mathbf{u}^* = \mathbf{u}_{ES}^*$ satisfying the inequality (6.39) and assume that the spatial numerical surface flux $\mathbf{f}^* = \mathbf{f}_{ES}^*$ is computed as in (4.88). Then the SBP space-time method is primary conservative and entropy stable.*

Proof. For proving primary conservation we need to sum (6.37) over all elements to find

$$\sum_{m=1}^{N_t} \sum_{n=1}^{N_Q} \left(\frac{\Delta x}{2} \sum_{j=0}^N \omega_j (\mathbf{u}_{Nj}^{m,n,*} - \mathbf{u}_{0j}^{m,n,*}) + \frac{\Delta t}{2} \sum_{i=0}^N \omega_i (\mathbf{f}_{iN}^{m,n,*} - \mathbf{f}_{i0}^{m,n,*}) \right) = \mathbf{0}. \quad (6.40)$$

Rearranging the sums gives us

$$\begin{aligned} & \sum_{n=1}^{N_Q} \frac{\Delta x}{2} \sum_{j=0}^N \omega_j \sum_{m=1}^{N_t} (\mathbf{u}_{Nj}^{m,n,*} - \mathbf{u}_{0j}^{m,n,*}) \\ & + \sum_{m=1}^{N_t} \frac{\Delta t}{2} \sum_{i=0}^N \omega_i \sum_{n=1}^{N_Q} (\mathbf{f}_{iN}^{m,n,*} - \mathbf{f}_{i0}^{m,n,*}) = \mathbf{0}. \end{aligned} \quad (6.41)$$

As we focus on a conforming mesh the numerical surface flux and state are uniquely defined on the interface and therefore, it holds $\mathbf{u}_{Nj}^{m-1,n,*} = \mathbf{u}_{0j}^{m,n,*}$ for $m = 2, \dots, N_t$ and $\mathbf{f}_{iN}^{m,n-1,*} = \mathbf{f}_{i0}^{m,n,*}$ for $n = 2, \dots, N_Q$. Thus, due to telescoping we find

$$\sum_{n=1}^{N_Q} \frac{\Delta x}{2} \sum_{j=0}^N \omega_j (\mathbf{u}_{Nj}^{N_t,n,*} - \mathbf{u}_{0j}^{1,n,*}) + \sum_{m=1}^{N_t} \frac{\Delta t}{2} \sum_{i=0}^N \omega_i (\mathbf{f}_{iN}^{m,N_Q,*} - \mathbf{f}_{i0}^{m,1,*}) = \mathbf{0}. \quad (6.42)$$

Considering periodic boundary conditions in space we get

$$\sum_{n=1}^{N_Q} \frac{\Delta x}{2} \sum_{j=0}^N \omega_j \mathbf{u}_{Nj}^{N_t,n,*} = \sum_{n=1}^{N_Q} \frac{\Delta x}{2} \sum_{j=0}^N \omega_j \mathbf{u}_{0j}^{1,n,*}. \quad (6.43)$$

As we consider an upwind state and Dirichlet boundary conditions in time we know $\mathbf{u}_{Nj}^{N_t,n,*} = \mathbf{u}_{Nj}^{N_t,n}$ and $\mathbf{u}_{0j}^{1,n,*} = \mathbf{u}(0, x_j^n)$ with $x_j^n := x^n(\xi_j)$ and thus, obtain

$$\sum_{n=1}^{N_Q} \frac{\Delta x}{2} \sum_{j=0}^N \omega_j \mathbf{u}_{Nj}^{N_t,n} = \sum_{n=1}^{N_Q} \frac{\Delta x}{2} \sum_{j=0}^N \omega_j \mathbf{u}(t=0, x_j^n). \quad (6.44)$$

So the approximation of the spatial integral of \mathbf{U} at $t = 0$ is the same as the approximated spatial integral of the numerical solution at $t = T$. This gives us the definition of primary conservation (6.33).

To prove entropy stability for the SBP space-time scheme we sum (6.38) over all elements in space and time

$$\begin{aligned} & \sum_{m=1}^{N_t} \sum_{n=1}^{N_Q} \frac{\Delta x}{2} \sum_{j=0}^N \omega_j \left([\mathbf{v}^T \mathbf{u}^* - \Psi^u]_{Nj}^{m,n} - [\mathbf{v}^T \mathbf{u}^* - \Psi^u]_{0j}^{m,n} \right) \\ & + \sum_{m=1}^{N_t} \sum_{n=1}^{N_Q} \frac{\Delta t}{2} \sum_{i=0}^N \omega_i \left([\mathbf{v}^T \mathbf{f}^* - \Psi^f]_{iN}^{m,n} - [\mathbf{v}^T \mathbf{f}^* - \Psi^f]_{i0}^{m,n} \right) = 0. \end{aligned} \quad (6.45)$$

Rearranging the sums gives

$$\begin{aligned} & \sum_{n=1}^{N_Q} \frac{\Delta x}{2} \sum_{j=0}^N \omega_j \underbrace{\left(\sum_{m=1}^{N_t} [\mathbf{v}^T \mathbf{u}^* - \Psi^u]_{Nj}^{m,n} - \sum_{m=1}^{N_t} [\mathbf{v}^T \mathbf{u}^* - \Psi^u]_{0j}^{m,n} \right)}_{=:(DT)_j^n} \\ & + \sum_{m=1}^{N_t} \frac{\Delta t}{2} \sum_{i=0}^N \omega_i \underbrace{\left(\sum_{n=1}^{N_Q} [\mathbf{v}^T \mathbf{f}^* - \Psi^f]_{iN}^{m,n} - \sum_{n=1}^{N_Q} [\mathbf{v}^T \mathbf{f}^* - \Psi^f]_{i0}^{m,n} \right)}_{=:(DS)_j^m} = 0. \end{aligned} \quad (6.46)$$

We next analyse $(DT)_j^n$ and $(DS)_j^m$ individually. We start with $(DS)_j^m$ which is algebraically equivalent to

$$\begin{aligned} (DS)_j^m &= \sum_{n=1}^{N_Q-1} [\mathbf{v}^T \mathbf{f}^* - \Psi^f]_{iN}^{m,n} - \sum_{n=2}^{N_Q} [\mathbf{v}^T \mathbf{f}^* - \Psi^f]_{i0}^{m,n} \\ &+ [\mathbf{v}^T \mathbf{f}^* - \Psi^f]_{iN}^{m,N_Q} - [\mathbf{v}^T \mathbf{f}^* - \Psi^f]_{i0}^{m,1}. \end{aligned} \quad (6.47)$$

Again, as we focus on a conforming mesh, it holds $\mathbf{f}_{iN}^{m,n-1,*} = \mathbf{f}_{i0}^{m,n,*}$ for $n = 2, \dots, N_Q$ and $m = 1, \dots, N_t$ and $\mathbf{f}_{iN}^{m,1,*} = \mathbf{f}_{i0}^{m,N_Q,*}$ due to periodic boundary conditions in space. Thus, we obtain

$$\begin{aligned} (DS)_j^m &= - \sum_{n=2}^{N_Q} \left((\mathbf{v}_{i0}^{m,n} - \mathbf{v}_{iN}^{m,n-1})^T \mathbf{f}_{i0}^{m,n,*} - (\Psi_{i0}^{f,m,n} - \Psi_{iN}^{f,m,n-1}) \right) \\ &- \left((\mathbf{v}_{i0}^{m,1} - \mathbf{v}_{iN}^{m,N_Q})^T \mathbf{f}_{i0}^{m,1,*} - (\Psi_{i0}^{f,m,1} - \Psi_{iN}^{f,m,N_Q}) \right). \end{aligned} \quad (6.48)$$

Due to the proof of Theorem 8, where we showed that (4.91) is ≤ 0 , we get

$$\begin{aligned} (\mathbf{v}_{i0}^{m,n} - \mathbf{v}_{iN}^{m,n-1})^T \mathbf{f}_{i0}^{m,n,*} - (\Psi_{i0}^{f,m,n} - \Psi_{iN}^{f,m,n-1}) &\leq 0 \\ (\mathbf{v}_{i0}^{m,1} - \mathbf{v}_{iN}^{m,N_Q})^T \mathbf{f}_{i0}^{m,1,*} - (\Psi_{i0}^{f,m,1} - \Psi_{iN}^{f,m,N_Q}) &\leq 0. \end{aligned} \quad (6.49)$$

Thus, we arrive at

$$(DS)_j^m \geq 0. \quad (6.50)$$

Next, we analyse $(DT)_j^n$. Here, $(DT)_j^n$ is

$$\begin{aligned} (DT)_j^n &= \sum_{m=1}^{N_t-1} [\mathbf{v}^T \mathbf{u}^* - \Psi^u]_{Nj}^{m,n} - \sum_{m=2}^{N_t} [\mathbf{v}^T \mathbf{u}^* - \Psi^u]_{0j}^{m,n}, \\ &+ [\mathbf{v}^T \mathbf{u}^* - \Psi^u]_{Nj}^{N_t,n} - [\mathbf{v}^T \mathbf{u}^* - \Psi^u]_{0j}^{1,n}. \end{aligned} \quad (6.51)$$

We choose Dirichlet boundary conditions such that $\mathbf{u}_{Nj}^{N_t,n,*} = \mathbf{u}_{Nj}^{N_t,n}$ and $\mathbf{u}_{0j}^{1,n,*} = \mathbf{u}(0, x_j^n)$.

$$\begin{aligned} (DT)_j^n &= - \sum_{m=2}^{N_t} \left((\mathbf{v}_{0j}^{m,n} - \mathbf{v}_{Nj}^{m-1,n})^T \mathbf{u}_{0j}^{m,n,*} - (\Psi_{0j}^{u,m,n} - \Psi_{Nj}^{u,m-1,n}) \right) \\ &+ [\mathbf{v}_{Nj}^{N_t,n,T} \mathbf{u}_{Nj}^{N_t,n} - \Psi_{Nj}^{u,N_t,n}] - [\mathbf{v}^{1,n,T} \mathbf{u}(0, x_j^n) - \Psi_{0j}^{u,1,n}]. \end{aligned} \quad (6.52)$$

Assuming that the upwind state satisfies inequality (6.39), then the sum in (6.52) is less than or equal zero. Thus, we find

$$(DT)_j^n \geq (\mathbf{v}_{Nj}^{N_t,n,T} \mathbf{u}_{Nj}^{N_t,n} - \Psi_{Nj}^{u,N_t,n}) - (\mathbf{v}_{0j}^{1,n,T} \mathbf{u}(0, x_j^n) - \Psi_{0j}^{u,1,n}). \quad (6.53)$$

As the entropy is defined by $S = \mathbf{v}^T \mathbf{u} - \Psi^u$ (4.6) we know

$$(DT)_j^n \geq S_{Nj}^{N_t,n} - (\mathbf{v}_{0j}^{1,n,T} \mathbf{u}(0, x_j^n) - \Psi_{0j}^{u,1,n}). \quad (6.54)$$

In order to include the exact initial entropy at $t = 0$ we add and subtract $\mathcal{S}(0, x_j^n)$ to obtain

$$\begin{aligned} (DT)_j^n &\geq S_{Nj}^{N_t,n} - \mathcal{S}(0, x_j^n) \\ &+ \mathcal{S}(0, x_j^n) - \mathbf{v}_{0j}^{1,n,T} \mathbf{u}(0, x_j^n) + \Psi_{0j}^{u,1,n}. \end{aligned} \quad (6.55)$$

As $\mathcal{S}(0, x_j^n) = \mathbf{v}(0, x_j^n)^T \mathbf{u}(0, x_j^n) - \Psi^u(0, x_j^n)$, Equation (6.55) becomes

$$\begin{aligned} (DT)_j^n &\geq S_{Nj}^{N_t,n} - \mathcal{S}(0, x_j^n) \\ &- \left((\mathbf{v}_{0j}^{1,n} - \mathbf{v}(0, x_j^n))^T \mathbf{u}(0, x_j^n) - (\Psi_{0j}^{u,1,n} - \Psi^u(0, x_j^n)) \right). \end{aligned} \quad (6.56)$$

Here, $\mathbf{u}(0, x_j^n)$ can be interpreted as an upwind state at $t = 0$

$$\mathbf{u}(0, x_j^n) = \mathbf{u}_{ES}^* \left(\mathbf{u}(0, x_j^n), \mathbf{u}_{0j}^{1,n} \right), \quad (6.57)$$

and, therefore, due to (6.39) it holds

$$\left(\mathbf{v}(0, x_j^n) - \mathbf{v}_{0j}^{1,n} \right)^T \mathbf{u}(0, x_j^n) - \left(\Psi^u(0, x_j^n) - \Psi_{0j}^{u,1,n} \right) \leq 0. \quad (6.58)$$

Including this result in (6.56) we obtain

$$(DT)_j^n \geq S_{Nj}^{N_t, n} - \mathcal{S}(0, x_j^n). \quad (6.59)$$

Inserting the estimates of $(DS)_j^m$, Equation (6.50), and $(DT)_j^n$, Equation (6.59), in (6.46) we arrive at

$$\begin{aligned} 0 &= \sum_{n=1}^{N_Q} \frac{\Delta x}{2} \sum_{j=0}^N \omega_j \underbrace{(DT)_j^n}_{\geq S_{Nj}^{N_t, n} - \mathcal{S}(0, x_j^n)} + \sum_{m=1}^{N_t} \frac{\Delta t}{2} \sum_{i=0}^N \omega_i \underbrace{(DS)_j^n}_{\geq 0}, \\ &\geq \sum_{n=1}^{N_Q} \frac{\Delta x}{2} \sum_{j=0}^N \omega_j \left(S_{Nj}^{N_t, n} - \mathcal{S}(0, x_j^n) \right). \end{aligned} \quad (6.60)$$

Finally, we rearrange the inequality and get

$$\sum_{n=1}^{N_Q} \frac{\Delta x}{2} \sum_{j=0}^N \omega_j S_{Nj}^{N_t, n} \leq \sum_{n=1}^{N_Q} \frac{\Delta x}{2} \sum_{j=0}^N \omega_j \mathcal{S}(0, x_j^n), \quad (6.61)$$

which is precisely the definition of entropy stability (6.36). \square

Here, we note that this proof holds independent of the size of the time step. In comparison, a semi-discrete entropy conservative scheme needs to consider a small time step in order to observe discrete entropy conservation in practice [118].

To summarize, we proved primary conservation and entropy stability requiring a numerical flux in space \mathbf{f}_{EC}^* satisfying the Tadmor Shuffle condition (5.9) and a numerical state in time \mathbf{u}_{EC}^* satisfying the time dependent Tadmor Shuffle condition (6.17). Provided that the upwind state satisfies the time dependent inequality (6.39) we can apply the entropy stable SBP space-time scheme to an arbitrary conservation law. The extension to higher spatial dimension problems is straightforward using the tensor product ansatz [40].

Remark 16. In terms of numerical testing we are interested in deriving entropy conservative schemes. Here, as for the semi-discrete analysis, we chose the numerical surface flux to be the entropy conserving flux $\mathbf{f}^* = \mathbf{f}_{EC}^*$ and chose periodic boundaries in space. In time we do the same approach setting $\mathbf{u}^* = \mathbf{u}_{EC}^*$ at all interior interfaces. We note, that this choice couples all elements of the fully-discrete system. For the numerical state at the boundary we still consider an upwind state and Dirichlet boundary conditions. Thus, the initial condition is incorporated at $t = 0$. Re-deriving the proof of Theorem 12 with these assumptions we obtain

$$\begin{aligned} (DS)_j^m &= 0, \\ (DT)_j^n &= S_{N_j}^{N_t, n} - \left(\mathbf{v}_{0_j}^{1, n, T} \mathbf{u}(0, x_j^n) - \Psi_{0_j}^{u, 1, n} \right). \end{aligned} \quad (6.62)$$

Including this in (6.46) it holds that

$$\sum_{n=1}^{N_Q} \frac{\Delta x}{2} \sum_{j=0}^N \omega_j \left(S_{N_j}^{N_t, n} - \left(\mathbf{v}_{0_j}^{1, n, T} \mathbf{u}(0, x_j^n) - \Psi_{0_j}^{u, 1, n} \right) \right) = 0. \quad (6.63)$$

Here, $\mathbf{v}_{0_j}^{1, n, T} \mathbf{u}(0, x_j^n) - \Psi_{0_j}^{u, 1, n}$ is an approximation to the entropy function at $t = 0$. For verifying a correct implementation of the SBP space-time method we will numerically verify that (6.63) holds in Section 6.2.

In the next section we extend the entropy stable and primary conservative SBP space-time scheme to a non-conforming mesh and discuss the advantages of combining non-conforming elements with space-time schemes.

6.1.3. Extension to Non-Conforming Elements

In this section we extend the SBP space-time scheme to non-conforming meshes. As described for semi-discrete schemes, non-conforming elements have the flexibility to distribute arbitrary number of nodes within an element without considering the nodal distribution of its neighbors. For fully-discrete space-time schemes we additionally have the flexibility to chose different nodal distributions in time. For a conservation law with one spatial dimension such a mesh is presented in Figure 6.3. The numerical fluxes \mathbf{f}^* at the non-conforming interfaces are constructed as in (4.88) where we need to incorporate projection operators satisfying the \mathbf{M} -compatibility condition (3.2). Including these fluxes within the SBP space-time scheme we obtain a non-conforming primary conservative and entropy stable method. This can be proven in a one-to-one fashion by applying the proof of Theorem 12 and using the \mathbf{M} -compatibility condition as in the Theorem 9. Thus, we obtain an entropy stable scheme which allows us to take a different number of

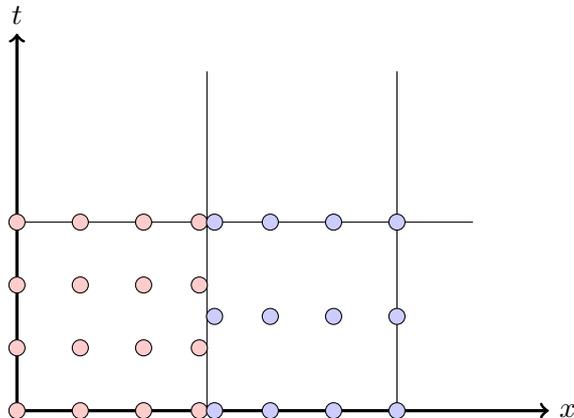


Figure 6.3.: Example of a non-conforming mesh where the nodal distributions in time do not coincide.

nodes in time for each element locally and thus to control the error introduced by the time integration method.

This method is similar to the *local time stepping approach* investigated by Winters [119], where different time steps were considered for different elements. However, the local time stepping method is more coding intensive. In comparison, a non-conforming SBP space-time scheme is simpler to implement as we only need to consider projection operators at non-conforming interfaces. Therefore, non-conforming space-time methods are an attractive alternative when considering different temporal discretizations for different elements.

6.1.4. Implicit Solver

The SBP space-time method described considers an implicit solver in each time slab. Therefore, we rewrite discretization (6.28) with a compact notation

$$\mathcal{H}(\tilde{\mathbf{u}}) = \mathbf{0}, \tag{6.64}$$

with $\tilde{\mathbf{u}} \in \mathbb{R}^{N_Q(N+1)^2}$. The vector $\tilde{\mathbf{u}}$ includes all degrees of freedom over the spatial elements within a time step. Here, $\mathcal{H}(\cdot)$ describes the SBP space-time discretization. Note, that this system is non-linear as we consider non-linear physical fluxes \mathcal{F} . One possible solving technique is using Newton's method

$$\begin{aligned} \nabla \mathcal{H}(\tilde{\mathbf{u}}^n) \tilde{\mathbf{w}}^{n+1} &= -\mathcal{H}(\tilde{\mathbf{u}}^n), \\ \tilde{\mathbf{u}}^{n+1} &= \tilde{\mathbf{w}}^{n+1} + \tilde{\mathbf{u}}^n, \end{aligned} \tag{6.65}$$

where $n \geq 0$ describes the number of iteration steps. Newton's method is instructed to the Jacobian $\nabla\mathcal{H}(\cdot)$ in each iteration. The evaluation of the Jacobian can be very cumbersome for highly non-linear fluxes. To avoid these complexities we use Jacobian-free Newton-Krylov solvers as in [6]. Newton-Krylov methods, e.g. Generalized minimal residual method (GMRES), are not instructed to know the analytic expression of the matrix $\nabla\mathcal{H}(\cdot)$. They only need the evaluation of the Jacobian multiplied with a vector $\nabla\mathcal{H}(\tilde{\mathbf{u}}^n)\tilde{\mathbf{z}}$. This evaluation can be approximated using the Fréchet derivative

$$\nabla\mathcal{H}(\tilde{\mathbf{u}}^n)\tilde{\mathbf{z}} \approx \frac{\mathcal{H}(\tilde{\mathbf{u}}^n + \varepsilon\tilde{\mathbf{z}}) - \mathcal{H}(\tilde{\mathbf{u}}^n)}{\varepsilon}, \quad (6.66)$$

for $\varepsilon > 0$. With this approximation we do not need the explicit form of the matrix $\nabla\mathcal{H}$. So this approach is free of the Jacobian matrix and therefore is denoted as *Jacobian-free*. With this machinery we generated numerical results as presented in the next section.

6.2. Numerical Verification of Summation-by-Parts Space-Time Schemes

In this section we verify properties as high-order, primary conservation and entropy stability of the SBP space-time scheme. We focus on the one dimensional compressible Euler equations as defined in (4.3). For all presented results we used the implicit solver in Section 6.1.4 with the GMRES method and $\varepsilon = 10^{-3}$. As in all numerical result sections before we first verify high-order convergence for different SBP operators in Section 6.2.1. Then, we numerically verify primary conservation and entropy stability in Section 6.2.2.

6.2.1. Comparison and Convergence of Summation-by-Parts Space-Time Schemes

To verify that the SBP space-time method is high-order we examine the convergence rate to an exact solution of the PDE. Instead of deriving an exact solution for the one dimensional compressible Euler equations we use the method of manufactured solutions and introduce the resulting system

$$\frac{\partial}{\partial t} \begin{pmatrix} \rho \\ \rho\nu \\ \mathcal{E} \end{pmatrix} + \frac{\partial}{\partial x} \begin{pmatrix} \rho\nu \\ \rho\nu^2 + \mathcal{P} \\ (\mathcal{E} + \mathcal{P})\nu \end{pmatrix} = \begin{pmatrix} \zeta_1(t, x) \\ \zeta_2(t, x) \\ \zeta_3(t, x) \end{pmatrix}, \quad (6.67)$$

on $\Omega = [0, 1]$ and $t \in [0, T]$ with $T = 1$. We set the manufactured solution to be

$$\mathbf{u} = \begin{pmatrix} 2 + \sin(2\pi(x - t)) \\ 2 + \sin(2\pi(x - t)) \\ (2 + \sin(2\pi(x - t)))^2 \end{pmatrix}. \quad (6.68)$$

This state is a smooth analytical solution of (6.67) if we consider the source term

$$\boldsymbol{\zeta} = \begin{pmatrix} 0 \\ \frac{\pi(\gamma-1)}{2} \cos(2\pi(x - t)) \\ \frac{\pi(\gamma-1)}{2} \cos(2\pi(x - t)) \end{pmatrix}. \quad (6.69)$$

Due to the solution (6.68) we consider periodic boundary conditions in space and Dirichlet boundary conditions in time. In order to avoid a global coupling of all *DOFS* we choose the temporal numerical surface state \mathbf{u}^* to be the upwind state. The entropy stable volume state $\mathbf{u}^\#$ for the Euler equations is taken from Friedrich et al. [40]. For the spatial volume flux $\mathbf{f}^\#$ we consider the one dimensional Ismail-Roe flux [62]. The choice of the spatial surface flux \mathbf{f}^* depends on the corresponding mesh. Here, we differ between conforming and non-conforming meshes.

Conforming Mesh

On a conforming mesh we set the numerical surface flux \mathbf{f}^* to be (4.88) with interface dissipation to obtain an entropy stable scheme. Let p denote the degree of the SBP operators defined on $N + 1$ nodes in space and time. For the convergence tests we set the number of elements in space to be equal the number of elements in time $N_Q = N_t$. Therefore, we solve for $N_Q(N + 1)^2$ *DOFS* in each time step. For the convergence tests we set $N_Q = 1, 2, 4, 8$. The averaged \mathcal{L}_2 errors and *EOC* for FD and HGTL operators are given in Table 6.1-6.2. For $p = 2$ we obtain a better \mathcal{L}_2 error for the HGTL operators. This result is the same as for the semi-discrete schemes on a conforming mesh described in Section 2.3 and 4.3. For $p = 3$ we observe an *EOC* slightly higher than $p + 1$. For $p = 2$ however the *EOC* moves around $p + \frac{1}{2}$. Supposedly, the mesh resolution needs to be more dense for the *EOC* to converge against $p + 1$. Note, that we only considered four convergence runs, whereas in the other numerical result sections for semi-discrete schemes we had more runs for even more *DOFS*. This is due to the increased CPU time for the space-time scheme. For the SBP space-time method the CPU time is increased due to the temporal discretization and the (non-preconditioned) implicit solver. Therefore, we only present results for runs which have finished within 200 hours.

EOC for the SBP space-time discretization with FD and HGTL operators for $p = 2, 3$.

$DOFS$	\mathcal{L}_2FD	$EOCFD$	\mathcal{L}_2HGTL	$EOCHGTL$
144	8.48E-03		4.45E-03	
288	8.33E-04	3.3	5.50E-04	3.0
576	6.52E-05	3.7	4.52E-05	3.6
1152	1.30E-05	2.3	7.98E-06	2.5

Table 6.1.: Setting $p = 2$, $T = 1$ and the number of nodes to $N + 1 = 12$ for the entropy stable SBP space-time scheme.

$DOFS$	\mathcal{L}_2FD	$EOCFD$	\mathcal{L}_2HGTL	$EOCHGTL$
256	1.35E-03		2.84E-04	
512	4.76E-05	4.8	1.37E-05	4.4
1024	5.37E-06	3.1	1.20E-06	3.5
2048	3.20E-07	4.1	5.97E-08	4.3

Table 6.2.: Setting $p = 3$, $T = 1$ and the number of nodes to $N + 1 = 16$ for the entropy stable SBP space-time scheme.

The DG operator only consists of $N + 1 = p + 1$ nodes in one dimension. When setting the degree of the SBP operator and the number of elements in space and time to be the same we generate more runs when doubling the number of $DOFS$. The results for DG are presented in Table 6.3 and 6.4. Here, we observe significantly better \mathcal{L}_2 errors. For $p = 2$ we are close to an EOC of $p + 1$ and for $p = 3$ we hit $p + 1$. Note, that for the DG operator with $p = 3$ we did not generate a run for 2048 $DOFS$ as we did for the FD and HGTL operator due to the high CPU time. Therefore, the run time in our code is higher for DG operators when considering the same number of $DOFS$.

Remark 17. Note, that these numbers in Table 6.3 and 6.4 differ from those in [40] as Friedrich et al. considered the kinetic energy preserving flux by Ranocha [99].

Non-Conforming Mesh

To obtain an entropy stable flux on a non-conforming mesh we consider the surface flux \mathbf{f}^* as in (5.43). We focus on a similar setup as for the conforming mesh by setting the degree of the SBP operators and the number of elements in space and time to be the same, so p remains constant and the number of elements in each dimension are the same $N_t = N_Q$. Moreover, since we focus on a non-conforming

EOC for the SBP space-time discretization with DG operators for $p = 2, 3$.

$DOFS$	$\mathcal{L}_2 DG$	$EOCDG$
18	5.98E-02	
36	9.51E-03	2.7
72	9.74E-04	3.3
144	1.13E-04	3.1
288	1.60E-05	2.8
576	2.24E-06	2.8
1152	3.39E-07	2.7

Table 6.3.: DG operator of degree $p = 2$ and $T = 1$ for the entropy stable SBP space-time scheme.

$DOFS$	$\mathcal{L}_2 DG$	$EOCDG$
32	1.63E-02	
64	5.58E-04	4.9
128	5.15E-05	3.4
256	6.79E-06	2.9
512	3.14E-07	4.4
1024	2.02E-08	4.0

Table 6.4.: DG operator of degree $p = 3$ and $T = 1$ for the entropy stable SBP space-time scheme.

mesh we consider different nodal distributions. First, we focus on DP, FD and HGTL operators. Let the first $\frac{N_Q}{2}$ spatial elements defined on the spatial domain $\Omega_1 = [0, \frac{1}{2}]$ consist of 22 nodes in space and of 24 nodes in time. The second half of spatial elements defined on $\Omega_2 = [\frac{1}{2}, 1]$ has 24 nodes in space and 22 nodes in time. Therefore, we have non-conforming interfaces at $[0, T] \times \{\frac{1}{2}\}$. The construction of the projection operators for these interfaces can be found in Section 3.2. We solve for $DOFS = 22 \times 24N_Q$ in each time slab. For the convergence tests we set $N_Q = 2, 4, 8$. The averaged \mathcal{L}_2 errors and EOC for DP, FD and HGTL operators are given in Table 6.5-6.6. Here, we obtain the best \mathcal{L}_2 error for the DP operators. This result is not too surprising as the projection operators for DP operators are of one degree higher than for FD and HGTL operators due to a norm matrix of degree $\geq 2p$, see Section 3.2.3. The EOC for DP operators is close to $p + 1$. Comparing FD and HGTL operators we obtain a better error for FD operators for $p = 2$, whereas the error for HGTL operators is smaller for $p = 3$. For $p = 2$ the FD operator is close to an EOC of $p + 1$, where the EOC of the HGTL operator is near $p + \frac{1}{2}$. For $p = 3$ however the HGTL operators obtains an EOC of $p + 1$ even though we consider projection operators of degree $p - 1$, whereas the FD operator is near an EOC of p .

Next, we couple different polynomial degrees within the non-conforming SBP space-time scheme considering DG operators. Let p_t and p_x denote the polynomial degree of the DG operator in time and space, respectively. We set these

EOC for the non-conforming SBP space-time discretization with DP, FD and HGTL operators for $p = 2, 3$.

$DOFS$	\mathcal{L}_2DP	$EOCDP$	\mathcal{L}_2FD	$EOCFD$	\mathcal{L}_2HGTL	$EOCHGTL$
1056	2.78E-03		3.94E-03		6.00E-03	
2112	4.59E-04	2.6	5.51E-04	2.8	1.01E-03	2.6
4224	6.75E-05	2.8	7.74E-05	2.8	1.65E-04	2.6

Table 6.5.: Setting $p = 2$ and $T = 1$ for the non-conforming entropy stable SBP space-time scheme. The errors have been produced considering the one dimensional Euler equations and using different SBP operators.

$DOFS$	\mathcal{L}_2DP	$EOCDP$	\mathcal{L}_2FD	$EOCFD$	\mathcal{L}_2HGTL	$EOCHGTL$
1056	5.44E-05		1.34E-04		8.22E-05	
2112	6.47E-06	3.1	2.61E-05	2.4	1.73E-05	2.2
4224	4.72E-07	3.8	2.89E-06	3.2	1.09E-06	4.0

Table 6.6.: Setting $p = 3$ and $T = 1$ for the non-conforming entropy stable SBP space-time scheme. The errors have been produced considering the one dimensional Euler equations and using different SBP operators.

degrees for all elements in Ω_1 to be $p_t = p$ and $p_x = p + 1$. All elements in Ω_2 are set to have $p_t = p + 1$ and $p_x = p$. The results for non-conforming SBP space-time scheme with DG operators for $p = 2, 3$ and averaged \mathcal{L}_2 errors are presented in Table 6.7-6.8. For both tables we observe an EOC of $p + \frac{1}{2}$, supposedly due to projection operators of degree $p - 1$. The final runs for $p = 2, 3$ consist of $DOFS = 2^6(p + 1)p$. For FD, HGTL and DP operators however, we obtained results for $DOFS = 4224$, which is three times more $DOFS$ than $2^6(p + 1)p$ for $p = 2, 3$. Therefore, the SBP space-time scheme performs faster for FD, HGTL and DP operators. Still, we observe a better \mathcal{L}_2 error when considering DG operators. For $p = 2$ and $DOFS = 768$ the error is smaller than for all other runs in Table 6.6 for $DOFS = 1056$. This result is remarkable, as the DG space-time scheme consists of differentiation matrices of degree $p = 2$ in certain elements, whereas the differentiation matrices of the runs in Table 6.6 are all of degree $p = 3$. So even though we consider certain elements with lower degree differentiation matrices and a lower number of DOFS, the SBP space-time scheme with DG operators has a smaller \mathcal{L}_2 error.

Here, we do not want to claim which operator is the best. The DG operator

EOC for the non-conforming SBP space-time discretization with DG operators for $p = 2, 3$.

<i>DOFS</i>	$\mathcal{L}_2 DG$	<i>EOCDG</i>
24	5.96E-02	
48	6.27E-03	3.2
96	7.43E-04	3.1
192	9.06E-05	3.0
384	1.37E-05	2.7
768	2.22E-06	2.6

Table 6.7.: Non-Conforming entropy stable SBP space-time scheme with $T = 1$ for the one dimensional Euler equations with DG operators and $p = 2$.

<i>DOFS</i>	$\mathcal{L}_2 DG$	<i>EOCDG</i>
40	8.73E-03	
80	5.21E-04	4.1
160	5.53E-05	3.2
320	6.87E-06	3.0
640	5.42E-07	3.7
1280	4.89E-08	3.5

Table 6.8.: Non-conforming entropy stable SBP space-time scheme with $T = 1$ for the one dimensional Euler equations with DG operators and $p = 3$.

is the most accurate in this test, but it has the highest cost in terms of CPU time in our implementation. The SBP space-time scheme is faster considering FD, HGTL or DP operators, but therefore they are less accurate. This result holds for conforming and non-conforming meshes.

6.2.2. Numerical Verification of Primary Conservation and Entropy Stability

In this section we numerically verify primary conservation and entropy stability/conservation. Here, we purely focus on DG operators. Again, we set the domain to be $\Omega = [0, 1]$ and $t \in [0, T]$ with $T = 1$. The boundary conditions are set to be periodic in space and Dirichlet in time. For all tests we consider a non-conforming mesh with four elements in space E_1, \dots, E_4 . Each element E_i is defined on the spatial domain $[(i-1)\frac{1}{4}, i\frac{1}{4}]$ for $i = 1, \dots, 4$. The elements E_1 and E_2 consist of DG operators with degree $p_x = 1$ in space and with degree $p_t = 2$ in time. For elements E_3 and E_4 the DG operators are of degree $p_x = 2$ in space and with degree $p_t = 1$ in time.

First we will perform an entropy conservation check. This is done by setting the temporal numerical states to be $\mathbf{u}^* = \mathbf{u}^\# = \mathbf{u}_{EC}^*$ and the spatial numerical fluxes to be $\mathbf{f}^* = \mathbf{f}^\# = \mathbf{f}_{EC}^*$. This leads to a fully coupled system. With this setup it

should hold that

$$\Delta \mathcal{S} := \sum_{n=1}^{N_Q} \frac{\Delta x}{2} \sum_{j=0}^N \omega_j \left(S_{N_j}^{N_t, n} - \left(\mathbf{v}_{0_j}^{1, n, T} \mathbf{u}(0, x_j^n) - \Psi_{0_j}^{u, 1, n} \right) \right), \quad (6.70)$$

is equal to zero, see (6.63). A scheme satisfying this condition is denoted as entropy conservative as $\mathbf{v}_{0_j}^{1, n, T} \mathbf{u}(0, x_j^n) - \Psi_{0_j}^{u, 1, n}$ is a high-order approximation of $\mathcal{S}_{0_j}^{1, n}$. In addition, to verify primary conservation we define

$$\Delta \mathbf{u} := \sum_{n=1}^{N_Q} \frac{\Delta x}{2} \sum_{j=0}^N \omega_j \left(\mathbf{u}_{N_j}^{N_t, n} - \mathbf{u}(t = 0, x^n(\xi_j)) \right), \quad (6.71)$$

which should be $\mathbf{0}$ due to the definition of primary conservation (6.33). In terms of verifying a correct implementation we calculate $\Delta \mathcal{S}$ and $\Delta \mathbf{u}$ with the discontinuous initial condition

$$\begin{pmatrix} \rho \\ \nu \\ \mathcal{P} \end{pmatrix} = \begin{pmatrix} 1 + \vartheta_{1,1} \\ 1 + \vartheta_{1,2} \\ 1 + \vartheta_{1,3} \end{pmatrix} \quad \text{if } x \leq \frac{1}{2}, \quad \begin{pmatrix} \rho \\ \nu \\ \mathcal{P} \end{pmatrix} = \begin{pmatrix} 1 + \vartheta_{2,1} \\ 1 + \vartheta_{2,2} \\ 1 + \vartheta_{2,3} \end{pmatrix} \quad \text{if } x > \frac{1}{2}. \quad (6.72)$$

Here $\vartheta_{k,l}$ are uniformly generated random numbers in $[0, 0.1]$. We calculate $\Delta \mathbf{u}$ and $\Delta \mathcal{S}$ for 1000 different initial conditions which gives us $(\Delta \mathbf{u})_{lk}$ and $(\Delta \mathcal{S})_k$ for $k = 1, \dots, 1000$ and $l = 1, \dots, 3$. In each run we consider three elements in space $N_t = 3$. Within the euclidean norm we obtain the results in Table 6.9.

Verification of primary and entropy conservation for the SBP space-time scheme

$\ \Delta \mathcal{S}\ _2$	$\ (\Delta \mathbf{u})_{1,:}\ _2$	$\ (\Delta \mathbf{u})_{2,:}\ _2$	$\ (\Delta \mathbf{u})_{3,:}\ _2$
-1.81E-15	6.30266E-16	5.84621E-16	1.93807E-17

Table 6.9.: Calculating $\Delta \mathbf{u}$ and $\Delta \mathcal{S}$ for 1000 different random initial conditions with the SBP space-time scheme. The average growth is presented within the euclidean norm. All values are near machine precision which demonstrates primary and entropy conservation.

In Table 6.9 we verify primary and entropy conservation. Finally, we will verify entropy stability. Therefore, we consider the initial condition

$$\begin{pmatrix} \rho \\ \nu \\ \mathcal{P} \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} \quad \text{if } x \leq \frac{1}{2}, \quad \begin{pmatrix} \rho \\ \nu \\ \mathcal{P} \end{pmatrix} = \begin{pmatrix} 1.125 \\ 1 \\ 1.1 \end{pmatrix} \quad \text{if } x > \frac{1}{2}. \quad (6.73)$$

In terms of including interface dissipation we set the numerical surface state \mathbf{u}^* to be the upwind state and \mathbf{f}^* to be calculated by (5.43). Considering $N_t = 128$ we calculate the total discrete primary quantity and entropy at $t = 0$ after each time slab by

$$\begin{aligned}
 S_0^{tot} &:= \sum_{n=1}^{N_Q} \frac{\Delta x}{2} \sum_{j=0}^N \omega_j \mathcal{S}(0, x_j^n), \\
 S_m^{tot} &:= \sum_{n=1}^{N_Q} \frac{\Delta x}{2} \sum_{j=0}^N \omega_j S_{Nj}^{m,n}, \\
 U_0^{tot} &:= \sum_{n=1}^{N_Q} \frac{\Delta x}{2} \sum_{j=0}^N \omega_j \mathbf{u}(0, x_j^n), \\
 U_m^{tot} &:= \sum_{n=1}^{N_Q} \frac{\Delta x}{2} \sum_{j=0}^N \omega_j \mathbf{u}_{Nj}^{m,n},
 \end{aligned} \tag{6.74}$$

for $m = 1, \dots, N_t$. The plot of these quantities is given in Figure 6.4 and 6.5. Here, we see that the total discrete entropy decays whereas the total discrete primary quantities remain constant. This indicates that the SBP space-time scheme by Friedrich et al. [40] is primary conservative and entropy stable.

6.3. Summary

In this chapter we presented the primary conservative and entropy stable SBP space-time method for non-linear conservation laws first presented by Friedrich et al. [40]. Here, special attention was given to the temporal approximation as discrete entropy analysis is typically done on the semi-discrete level. The derivative in time was approximated with a high-order differentiation matrix satisfying the SBP property. As for the discretization in space the temporal part of the variational formulation consists of a volume and surface part. As in the Flux Difference discretization by Fisher and Carpenter [33] we introduced numerical volume and surface states. The volume state should satisfy a time dependent Tadmor Shuffle condition (6.17) and the surface state is chosen to be an upwind state provided that it satisfies the time dependent inequality (6.39). Such states have been derived by Friedrich et al. [40] for the compressible Euler, Shallow Water or Ideal MHD equations. In particular for the Euler equations we provide numerical results to verify the proven properties as high-order, primary conservation and entropy stability of the SBP space-time scheme. In addition, we extended the SBP space-time scheme to a non-conforming mesh. This advantageous approach gives us the

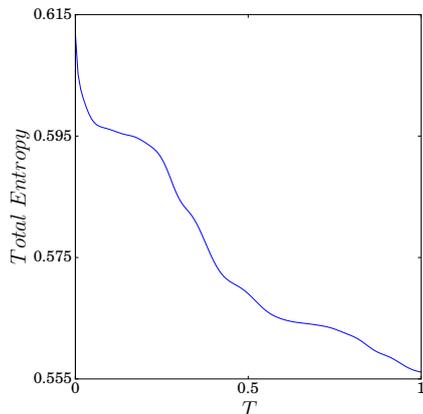


Figure 6.4.: Evolution of the total discrete entropy S_m^{tot} for $m = 0, \dots, N_t$ with the SBP space-time scheme using the upwind state in time and the Ismail-Roe flux with interface dissipation in space.

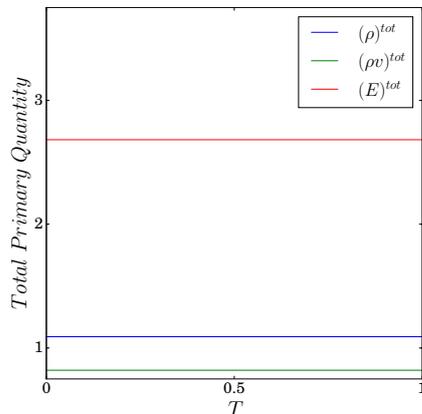


Figure 6.5.: Evolution of the total discrete primary quantities U_m^{tot} for $m = 0, \dots, N_t$. These remain constant over time verifying that the SBP space-time scheme is primary conservative.

flexibility of choosing different nodal distributions in space and at the same time maintaining a high-order scheme which is primary conservative and entropy stable as verified in the numerical result section.

Besides of these nice theoretical properties the space-time method has a drawback. In comparison to a semi-discrete code the SBP space-time method has a high CPU run time. One reason for a higher run time is because the space-time scheme is instructed to an implicit solver in each time slab. Such solvers are run time intensive, especially without considering preconditioners. For this reason we only presented results for the one dimensional compressible Euler equations. In order to apply the SBP space-time method to higher dimensions, more investigation is necessary to reduce the CPU time. The development of efficient high dimensional SBP space-time schemes is left for future work. Therefore, we focus on semi-discrete approaches in the next chapter.

7. Mesh Generation and Applications

For the multitude of real world applications we want to simulate the behaviour of fluids around complex geometries. In order to obtain such a mesh we are instructed to the use of mesh generators. Many of these mesh generators have been developed over the last decades. Generators relying on triangular/tetrahedral elements are given in [48, 64, 106]. In order to apply the tensor product based SBP Flux Difference scheme we are instructed to meshes with quadrilateral/hexahedral elements. For such meshes there are generators, e.g. HOPR (High Order Preprocessor) by Hindenlang, Bolemann and Munz [59] or the mesh generator by Yuan and Kopriva [124]. These generators rely on complex algorithms for the generation of meshes [104]. In Figure 7.1 we present a conforming quadrilateral mesh around an air foil constructed by HOPR.

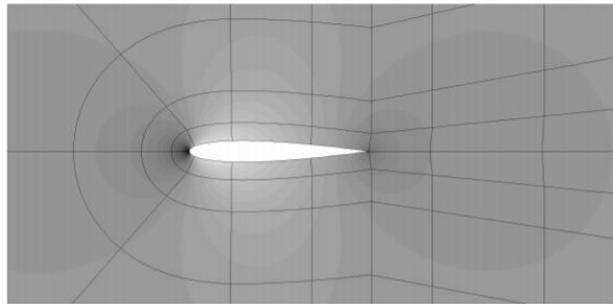


Figure 7.1.: Two dimensional conforming quadrilateral mesh around an air foil constructed by HOPR [59]. The picture is taken from the HOPR manual.

However, the construction of these meshes, especially for three dimensional problems, can be computationally very expensive for complex geometries. Thus, it can take more time to generate the mesh than to simulate the fluid flow [61]. Another issue is that the resulting mesh on conforming grids is unstructured, for which it is more complicated to find the neighbours of a given element. As described in the chapters before, we have the theory and machinery for high-order primary conservative and entropy stable schemes on non-conforming meshes. Therefore, we focus on the construction of "more simpler" non-conforming meshes.

A simple approach for the adaptive construction of quadrilateral meshes around complex geometries is the *Cartesian cut cell method*. The conceptual idea is to "cut" the geometry out of a Cartesian background mesh. An example focusing on a cylinder is illustrated in Figure 7.2 and 7.3.

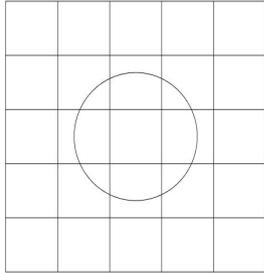


Figure 7.2.: Cartesian background mesh considering a cylinder.

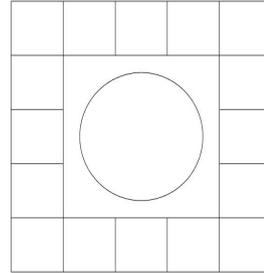


Figure 7.3.: "Cut out" the subdomains which are inside the cylinder.

We define Cartesian elements on all remaining subdomains in Figure 7.3. In order to resolve the geometry surface with quadrilateral elements more refinement is necessary. Here, we subdivide all subdomains within Figure 7.2 which interact with the geometry by a two-refinement step [104]. Here, we again perform the strategy cutting out subdomains if they interact with the geometry.

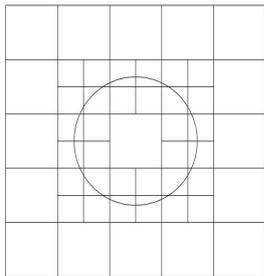


Figure 7.4.: Subdivide all subdomains which interact with the cylinder.

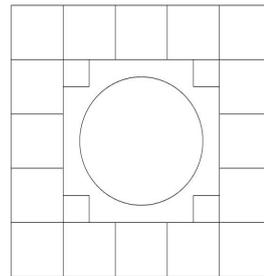


Figure 7.5.: "Cut out" the subdomains which hit or are inside the cylinder.

Again, we define Cartesian elements on all remaining subdomains in Figure 7.5. Here, the meshes in Figure 7.3 and 7.5 are denoted as Cartesian cut cell meshes at the first and second refinement level, respectively. This strategy can be repeated recursively depending on the number of refinement levels defined by the user. In Figure 7.6 and 7.7 we present the cut cell meshes at the third and fourth refinement level, respectively.

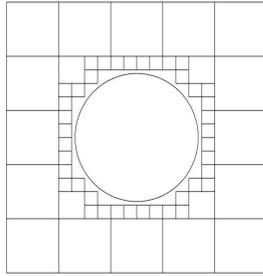


Figure 7.6.: Cartesian cut cell mesh at the third refinement level.

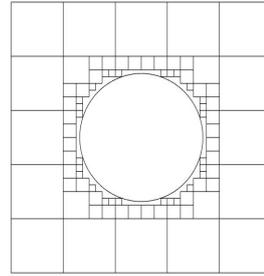


Figure 7.7.: Cartesian cut cell mesh at the fourth refinement level.

Depending on the number of refinement levels we obtain a structured non-conforming mesh. The structure is explained in Section 7.1.1. The resulting cut cell mesh has a fine resolution near the surface of the geometry and a coarse resolution else where. Such meshes are of special interest when one is purely interested in the fluid flow around an object and not in the far field. An advantageous feature of cut cell methods is that they are non-body-fitted [1], i.e. the volume mesh structure is independent of the surface discretization. Thus, the mesh results in Cartesian elements for which we can use tensor product SBP operators. Also, the cut cell technique of generating meshes is fast and simple. Therefore, such mesh generation schemes have gained popularity over the last years as in [1, 5, 61, 63, 76, 117].

A disadvantage of Cartesian cut cell methods is that they have been designed for low-order methods as curved surfaces of the geometries are approximated by straight interfaces of the Cartesian elements, see Figure 7.7. In order to construct high-order schemes on cut cell meshes, we need to pay special attention to the subdomains interacting with the geometry surface in the final refinement level. These subdomains are denoted as *final-cut domains*.

High-order cut cell methods on triangular elements have been purposed by

[32, 87, 110]. Depending on the geometry the final-cut domains do not necessarily have to be triangles. Therefore, depending on the element shape, new quadrature rules are constructed on the curved polyhedral final-cut domains. Thus, the final-cut domains are set to be elements. To adapt this strategy, while at the same time ensuring an entropy stable simulation, we would need to construct multidimensional SBP operators on these curved polyhedral final-cut domains [25, 28, 55, 56, 57]. However, this process is cumbersome as we need to construct such a multidimensional SBP operator for each final-cut domain individually with arbitrary polyhedral elements.

In this work we decompose the resulting final-cut domains into curved quadrilateral elements. Therefore, we do not have to construct a new set of SBP operators and simply apply tensor product SBP operators from Section 2.1. In two dimensional space the decomposition of a polyhedral domain into quadrilateral elements is always possible [98]. In three dimensional space it is still an open question if a polyhedral can be decomposed by hexahedral elements [77, 78, 105]. Focusing on the cylinder as in Figure 7.7 we present the resulting mesh in Figure 7.8.

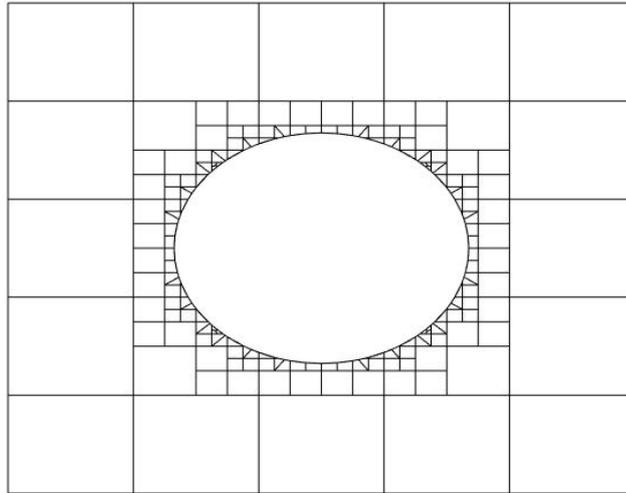


Figure 7.8.: Generated non-conforming mesh with quadrilateral elements based on the cut cell technique. The elements at the cylinder surface are curved in order to obtain a high-order mesh.

Note, that the resulting mesh in Figure 7.8 is a h/p non-conforming mesh with curved elements as the interfaces do not coincide and the nodal distribution of each element can differ. As mentioned in Chapter 5 the development of primary conservative and entropy stable SBP schemes with non-conforming curved interfaces is an ongoing research topic. Therefore, we ensure that all curved interfaces are conforming interfaces. This can be ensured by decomposing final-cut domains with the same SBP operator and nodes.

In this chapter we discuss the the implementation of the mesh generator in Section 7.1. Based on this mesh generator, we apply the final simulations of this work in Section 7.2; the flow around a NACA0012 air foil.

7.1. Mesh Generator Implementation

In this section we discuss the implementation of the mesh generation strategy based on the cut cell approach and final-cut domain decomposition. In order to obtain a structured cut cell mesh we first introduce the *quadtree* data structure in Section 7.1.1. The decomposition of the final-cut domains is discussed in Section 7.1.2. Without any further modification we are likely to obtain badly shaped elements, e.g. elements with a very small Jacobian. Therefore, we introduce an element merging strategy in Section 7.1.3. Also, within the cut cell approach, we obtain non-conforming interfaces with hanging corners, see Figure 3.9 with an arbitrary amount of hanging corners. In theory, we know how to couple this elements to obtain a primary conservative and entropy stable simulation. However, most open source codes, e.g. FLUXO [46], and the self implemented code within this work is limited to a "two-to-one" coupling, see Figure 3.10. Therefore, we propose an additional strategy to ensure a "two-to-one" coupling in Section 7.1.4. With all these strategies and approaches we present a set of meshes for different types of complex geometries in Section 7.1.5.

7.1.1. Quadtree Data Structure

The adaptive cut cell mesh generation technique begins with a Cartesian background mesh. Each subdomain of the background mesh is assigned with a quadtree. Such a tree consists of leaves with four children. Provided a subdomain of the background mesh does not interact with the geometry, we set the subdomain to be a Cartesian element. The quadtree of this subdomain consists of a single leaf without any children, where the leaf is equipped with the element number. More interesting is the structure when the subdomain of the background mesh interacts

with the geometry. As the background subdomain is divided into four smaller subdomains, we set the root of the quadtree to have four children where each child is a representative of one smaller subdomain. If a smaller child subdomain hits the geometry, we equip the corresponding leaf with another four children. Otherwise, the smaller child subdomain does not touch the geometry and we define an element. Therefore, the corresponding leaf does not have any children and is attached with the element number (or a default value if the subdomain is completely within the geometry). We repeat this recursive process depending on the number of refinement levels. Thus, the height of the corresponding quadtree depends on the number of refinement levels. An example of a mesh and its resulting quadtree with two refinement levels is given in Figure 7.9 and 7.10.

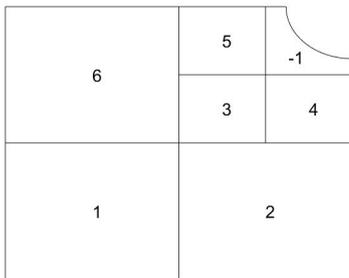


Figure 7.9.: Generated non-conforming mesh equipped with element/subdomain numbers.

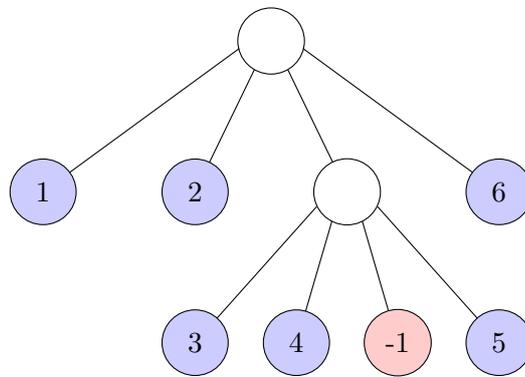


Figure 7.10.: Corresponding structured quadtree. The blue nodes denote the Cartesian elements. The red nodes denote a curved subdomain.

Here, we need to pay special attention to the curved subdomain attached with the number -1 . The negative sign denotes that we are on a final-cut domain where each final-cut domain is attached with a different negative number. As such domains are not necessarily quadrilaterals, as in Figure 7.9, we decompose the domain as described in Section 7.1.2. Therefore, during the mesh generation process, we store all curved elements within a *Curvlist*. This *Curvlist* maps the negative final-cut domain number to the numbers of the corresponding curved elements. Additionally, the curved elements store the negative number of the final-cut domain and the leaf position within the quadtree. Based on the *Curvlist*, we obtain the element neighbors by simply moving along the structured quadtree,

which is a benefit of structured meshes.

7.1.2. Final-Cut Domain Decomposition

In this section we discuss possible curved quadrilateral element decomposition strategies of final-cut domains. Here, we make the following assumptions

- A geometry, e.g. a cylinder, is not fully inside a final-cut domain. Thus, the geometry and the final-cut domain always intersect.
- When considering a flow around more than one geometry, e.g. air foil with multiple components (see Section 7.1.5), we assume that only one geometry cuts a final-cut domain.

Both of these assumptions are reasonable considering a fine mesh resolution near the geometry surface and thus, small final-cut domains.

Within our code we implement numerous decomposition strategies depending on how the geometry hits the final-cut domain. Here, we present some of the most common decomposition techniques in Figure 7.11 based on a smooth geometry.

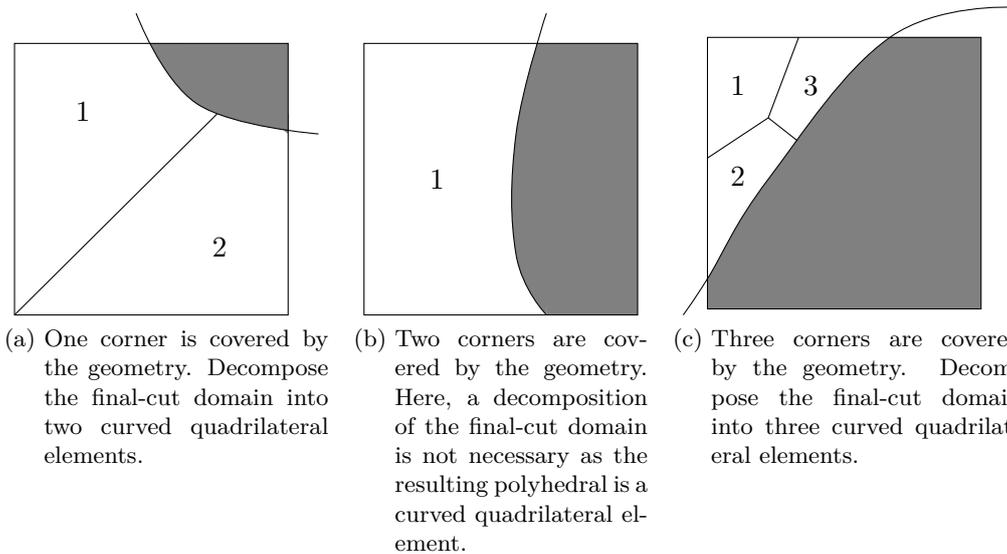


Figure 7.11.: The three most common decomposition strategies. The geometry is colored grey. All resulting elements within the final-cut domain are curved quadrilateral elements.

For small final-cut domains the geometry generally covers one, two or three corners of the Cartesian subdomain. Depending on the the number of corners the remaining final-cut domain represents a different type of polyhedral with five, four or three corners. As our code is restricted to elements with four corners, we decompose the final-cut domains as presented in Figure 7.11 such that only curved quadrilateral elements remain. Additionally, we implement decomposition strategies when the geometry interacts with the subdomain but does not cover a single corner or covers all corners. However, this scenario is unlikely when considering small final-cut domains.

Remark 18. In Figure 7.11 (c) we obtain hanging corners which can lead to h non-conforming interfaces. However, even though the elements are curved, the interface with the hanging corner is straight. Therefore, the metric terms on the interface remain constant and thus, we can couple elements in an entropy stable way as done for Cartesian elements.

Besides checking if the corners of the Cartesian subdomain are in- or outside of the geometry, we need to check if the interfaces between two non-covered corners are cut by the geometry. Also, it is necessary to check if a interface between two covered corners is not fully covered by the geometry, see Figure 7.12.

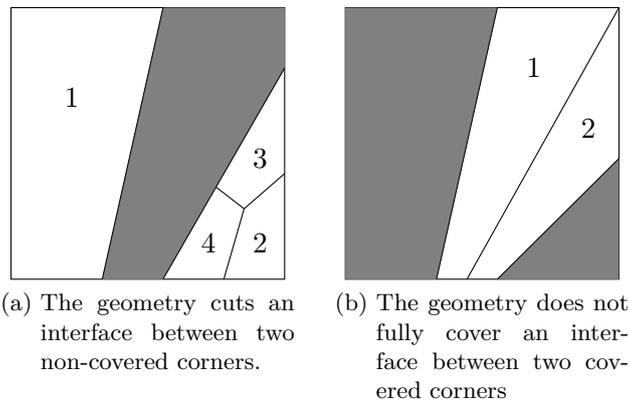


Figure 7.12.: Special cases of the geometry cutting a Cartesian subdomain.

All the above presented decomposition strategies assume a smooth geometry. For geometries with sharp corners, e.g. the tail of an air foil, we need to consider different decomposition techniques. Within our code we implemented decomposition strategies where the final-cut domain can have up to one sharp corner. An example is given in Figure 7.13.

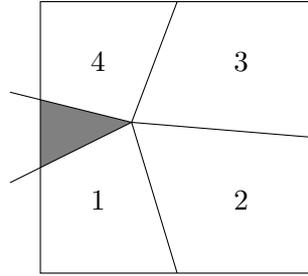


Figure 7.13.: The geometry with a sharp corner cuts the Cartesian subdomain. The final-cut domain is decomposed into four quadrilateral elements.

In our code we stored the sharp corners in a *Cornerlist*. So, if a sharp corner from the *Cornerlist* is within a final-cut domain, we consider a different decomposition strategy. In this work we only considered geometries where the exact position of a sharp corner is known. For arbitrary geometries, however, the amount of sharp corners can be numerous and a priori not known. Algorithms for detecting sharp corners can be found in [75, 125].

The interface curves of the remaining curved quadrilateral elements are approximated by a polynomial of degree p , where p denotes the degree of the differentiation matrix within the curved element. For more details see Appendix A.1. Based on the presented strategies we obtain non-conforming meshes for flows around complex geometries as a cylinder, NACA0012 air foil and an air foil with multiple components. However, within this mesh generation strategy we can obtain badly shaped elements. Therefore, we discuss the element merging technique to improve the overall mesh quality in the next section.

7.1.3. Element Merging

For meshes with badly shaped elements a very small explicit time is necessary to obtain a stable simulation. In this section we focus on avoiding such badly shaped elements within our mesh. Let E denote an arbitrary element. To indicate if an element is badly shaped, we introduce an indicator function Φ with $\Phi : E \rightarrow [0, 1]$. An element is said to be badly shaped if the indicator function is smaller than a certain tolerance Φ_{min} . Mathematically speaking, an element is badly shaped if

$$\Phi(E) \leq \Phi_{min}. \quad (7.1)$$

Here, we discuss the choice of the indicator function Φ . On curved meshes the time step is defined by the *CFL* condition

$$\Delta t := \frac{CFL}{\max_j \{N_j + 1\}} \frac{J_{min}}{\lambda_{max}}, \quad (7.2)$$

where J_{min} denotes the minimum Jacobian over all elements, $N_j + 1$ denotes the number of nodes in the j -th element and λ_{max} denotes the maximum eigenvalue of the flux Jacobians over all elements. In order to increase the time step defined by (7.2) we seek for curved elements with a large minimum Jacobian. However, small Jacobians are indispensable when considering a fine discretization at the geometry surface. Therefore, we define the indicator function Φ by the ratio of the minimum and maximum value of the Jacobian $J(E)$ on the element E

$$\Phi(E) := \frac{\min_{i,j=0,\dots,N} J(E)_{ij}}{\max_{i,j=0,\dots,N} J(E)_{ij}}. \quad (7.3)$$

This indicator function can also be found in the verdict library [108]. When focusing on a Cartesian element it holds $\Phi(E) = 1$ independent of the element size, as the Jacobian remains constant on all nodes within the element. For curved elements it holds that $\Phi(E) \leq 1$.

Remark 19. Within the verdict library [108] there are numerous different types of indicator functions based on triangular/tetrahedral or quadrilateral/hexahedral elements.

With the indicator function (7.3) and a given limit Φ_{min} we discuss how to avoid badly shaped elements in the mesh. Each curved element has stored the position of its final-cut leaf within the quadtree, see Section 7.1.1. Assuming that we consider more than one refinement level within the mesh generation process, the final-cut leaf always has a parent, see Figure 7.14 and 7.15. The idea of avoiding badly shaped elements is to merge the corresponding final-cut domain with all its siblings into one element. The new final-cut domain is stored on the parent node. As this domain is larger than before, the corresponding decomposed elements are more likely to have a better shape, see Figure 7.16 and 7.17. If the new elements are still badly shaped, we apply the same strategy again by merging elements. This recursive process is repeated until all elements are not badly shaped or until the whole quadtree is merged together to a single root.

Remark 20. When merging a final-cut domain special attention must be paid if the neighbor is also a final-cut domain. In order to obtain at least a "two-to-one" coupling between the elements of these two domains, it can be necessary to merge the neighbored final-cut domain even if it is well shaped. Even though we focus

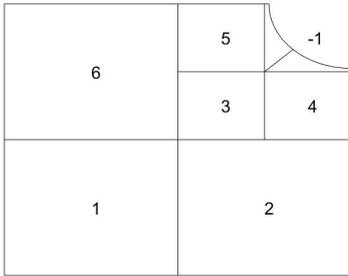


Figure 7.14.: A non-conforming mesh with two refinement levels. We obtain badly shaped elements in the final-cut domain with the number -1 .

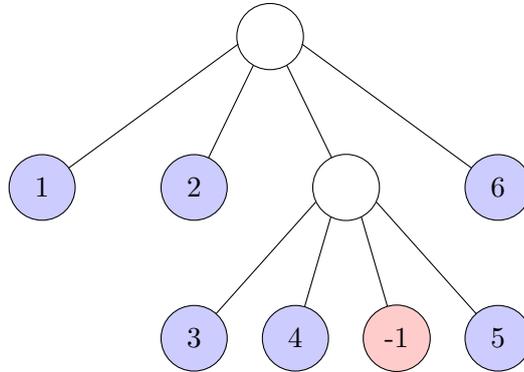


Figure 7.15.: The badly shaped elements are represented by the leaf number -1 in the quadtree.

on curved elements we can handle "two-to-one" couplings in an entropy stable way as two neighbored final-cut domains are connected by a straight line, see Remark 18.

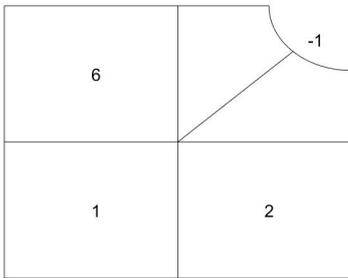


Figure 7.16.: Generated non-conforming mesh with better shaped elements due to the merging procedure.

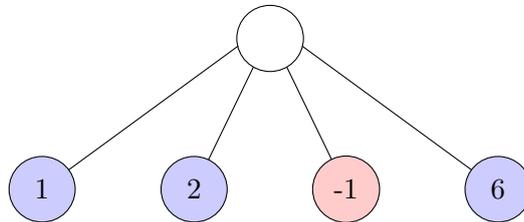


Figure 7.17.: Corresponding structured quadtree. The former final-cut domain and its siblings have been merged to a larger final-cut domain with better shaped elements.

7.1.4. Two-to-One Coupling

In this section we discuss how to ensure a "two-to-one" coupling as in Figure 3.10. With such a coupling an element can only have up to two neighbors on an interface. This simplifies the implementation of the non-conforming entropy stable simulation. Here, the quadtree structure of the mesh is helpful to guarantee a "two-to-one" coupling.

We demonstrate the procedure by focusing on an element E_x . This element is represented by a leaf with a corresponding height within the quadtree. Due to the structured quadtree we can easily identify all the neighbor elements. If a neighbor element has the same height, than E_x and its neighbor element have a conforming interface or, in other words, we have a "one-to-one" coupling. However, if the height of E_x and its neighbor differs by one refinement level, then the elements are connected by a "two-to-one" coupling. Again, we can see the advantages of cut cell mesh as the quadtree describes how elements are coupled with each other.

Here, we discuss the issue when the element E_x and its neighbor differs by more than one refinement level. Here, we do not have a "one-to-one" and "two-to-one" coupling, see Figure 7.18 and 7.19.

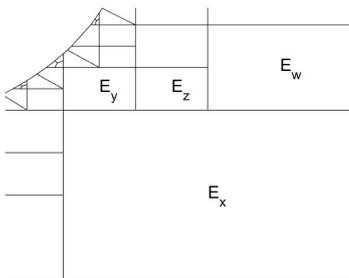


Figure 7.18.: Generated non-conforming where the element E_x has more than two neighbors at the top interface.

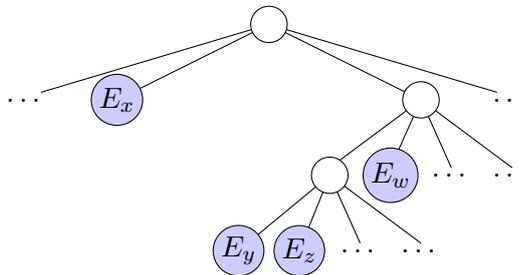


Figure 7.19.: The corresponding structured quadtree. Here, the height of E_x and the neighbor elements E_y and E_z differ by two.

In Figure 7.18 we see that E_x has three neighbors at the top interface (E_y , E_z and E_w). Therefore, a "two-to-one" coupling is not possible. This can also be observed by looking at the corresponding quadtree in Figure 7.19 as the leaf height of E_x and the neighbors E_y , E_z differ by two. Therefore, we need to modify our mesh.

The modification is obtained by subdividing the element E_x into four subelements as in Figure 7.20 and 7.21.

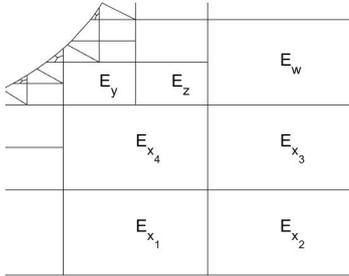


Figure 7.20.: Subdivide E_x into four smaller elements. Here, we obtain a mesh with a "two-to-one" and "one-to-one" coupling.

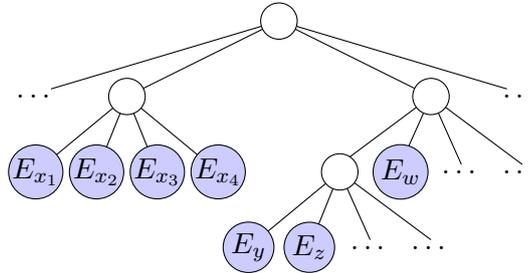


Figure 7.21.: The corresponding structured quadtree. Here, the height of E_{x_1}, \dots, E_{x_4} and the neighbor elements E_y, E_z, E_w only differ by one refinement level or are the same.

Due to the subdivision we obtain four new elements E_{x_1}, \dots, E_{x_4} in Figure 7.20. We interpret this subdivision in the quadtree in Figure 7.21 by attaching four children to the leaf of E_x . Therefore, the height of E_{x_4} and the neighbor elements E_y, E_z only differ by one, which indicates a "two-to-one" coupling. In addition, the height of E_{x_3} and E_w are the same, which indicates a conforming interface ("one-to-one" coupling) between the two elements, see Figure 7.20.

For this example we subdivided the element E_x once into four smaller elements. In general, a subdivision can be done multiple times depending on the height of the element leaf and its neighbour. With this recursive process we ensure a "one-to-one" or "two-to-one" coupling.

7.1.5. Non-Conforming Meshes for Complex Geometries

Based on the presented mesh generation strategies in Sections 7.1.1-7.1.4 we generate numerous meshes. First we present meshes around a cylinder in $\Omega = [0, 1] \times [0, 1]$. The center of the cylinder is set to $(\frac{1}{2}, \frac{1}{2})$ with a radius of 0.15. The Cartesian background mesh is constructed by $N_{Q_x} \times N_{Q_y}$ subdomains with $N_{Q_x} = N_{Q_y} = 4$. We chose to have eight refinement levels for the mesh. On each

element of the mesh we define DG operators of degree $p = 3$. To finally obtain the resulting mesh we need to specify the limit Φ_{min} in order to determine if an element is badly shaped (7.1). Here, we generated three different meshes with $\Phi_{min} = 0; 0.1; 0.2$ in Figure 7.22

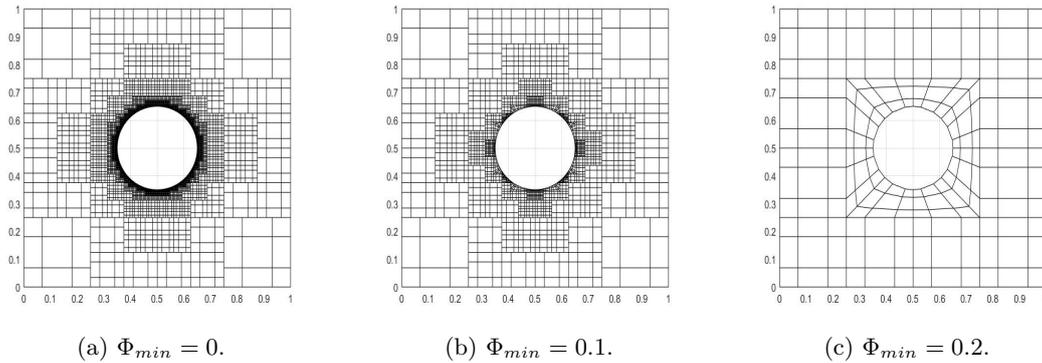


Figure 7.22.: Generated non-conforming meshes with LGL nodes surrounding a cylinder with three different Φ_{min} values.

In Figure 7.22(a) no element merging has been done. Here, we have a very finer resolution near the cylinder surface. However, as we set $\Phi_{min} = 0$, elements can occur which drastically reduce the time step for explicit time marching. When increasing $\Phi_{min} = 0.1$ as in Figure 7.22(b) we obtain a less fine mesh at the cylinder surface, but therefore the elements are better shaped. Finally, considering $\Phi_{min} = 0.2$ as in Figure 7.22(c) we obtain a coarse mesh with even better shaped elements. For all the elements E in Figure 7.22(a)-(c) it holds $\Phi(E) > \Phi_{min}$.

Next, we presented meshes around a NACA0012 air foil in $\Omega = [-1.05, 1.95] \times [-\frac{2}{3}, \frac{2}{3}]$. As for the cylinder we set the Cartesian background mesh to have $N_{Q_x} \times N_{Q_y}$ subdomains with $N_{Q_x} = N_{Q_y} = 4$. Again, we chose eight refinement levels for the mesh and consider DG operators of degree $p = 3$ on each element. The three presented meshes in Figure 7.23 are constructed by setting $\Phi_{min} = 0; 0.04; 0.1$. As for the cylinder we obtain a very fine mesh in Figure 7.23(a), a less finer mesh in Figure 7.23(b) and a coarse mesh in Figure 7.23(c). All three meshes consist of well shaped elements (based on the Φ_{min} value).

Finally, we present meshes around an air foil with three components. We set $\Omega = [-1.5, 1.5] \times [-3, 3]$. Here, we choose a larger background mesh to ensure that all subdomains do not cut two geometries at the same time. We set $N_{Q_x} = N_{Q_y} = 20$. Here, we fix the number of refinement levels to six. Considering DG operators

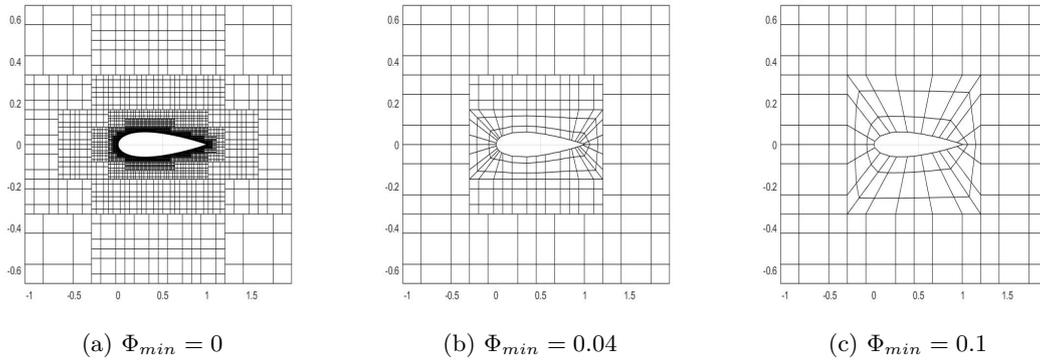


Figure 7.23.: Generated non-conforming meshes with LGL nodes surrounding a NACA0012 air foil with three different Φ_{min} values.

of degree $p = 3$ on each element and $\Phi_{min} = 0; 0.01$ we obtain the meshes in Figure 7.24(a)-(b).

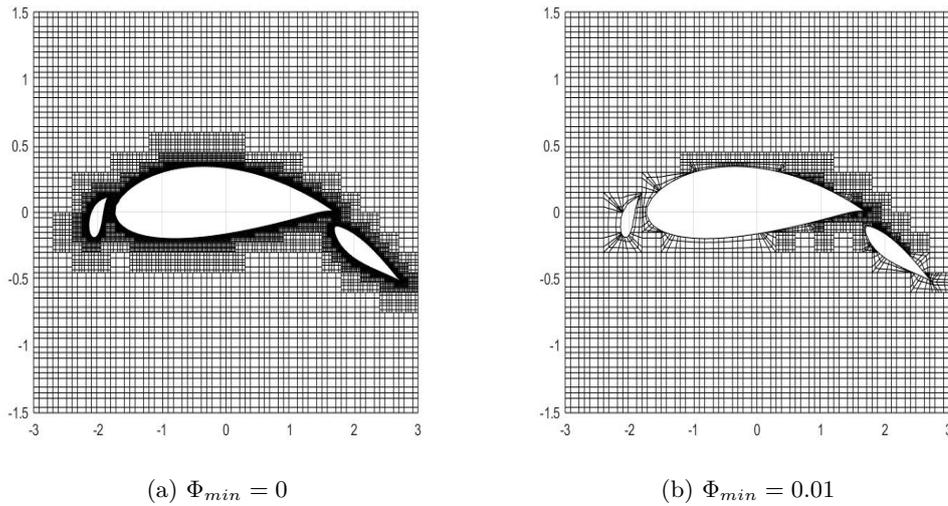


Figure 7.24.: Generated non-conforming meshes with LGL nodes surrounding air foil with three components. Here, we considered two different Φ_{min} values.

The mesh in Figure 7.24(a) is very fine at the element surface. However, when increasing Φ_{min} to 0.01 we obtain final cut domains which consists of curved elements which are conforming with the background mesh in Figure 7.24(b). Also, our code could not ensure that these curved elements are well shaped (some elements E exist with $\Phi(E) \leq \Phi_{min}$). Further element merging of the final-cut domains is

not possible as the corresponding quadtree only consists of a single root. Here, different decomposition strategies need to be taken into account to obtain a mesh with well shaped elements.

Here, we demonstrated the speed and the simplicity of the mesh generator. We highlight, that all the meshes have been generated using the same code by only changing the geometry data. This again shows that the generator is fully adaptive independent of the corresponding geometry. For meshes, e.g. air foil with three components, we need to consider different decomposition techniques to obtain a mesh with well shaped element. Defining these decomposition strategies is left for future work.

7.2. Application: Flow around a NACA0012 Air Foil

In this section we apply the primary conservative and entropy stable SBP Flux Difference scheme on the generated non-conforming cut cell mesh with final-cut domains decomposed into curved elements. Here, the discretization on Cartesian elements is given by (4.64), where as the discretization on curved elements is presented in Appendix A.2. Here, we point out the advantageous features of non-conforming meshes. Therefore, we focus on the two dimensional compressible Euler equations defined in (4.99). The numerical volume fluxes are set to be the Ismail-Roe fluxes and the surface fluxes are set to be the non-conforming Ismail-Roe fluxes with interface dissipation (4.88).

For the final application, we consider the NACA0012 air foil. We are interested in the fluid flow at the surface of the air foil. Therefore, the cut cell mesh with curved elements is well suited as it has a fine resolution at the air foil surface and a coarse resolution in the far field. We consider the domain $\Omega = [-4; 4.55] \times [-4; 4]$ and choose the Cartesian background mesh to consist of 4×4 subdomains. The number of refinement levels is set to four. We detect badly shaped elements by choosing $\Phi_{min} = 0.02$. On all elements we consider DG operators. However, it would be possible to consider any other SBP operator to obtain a primary conservative and entropy stable simulation. For all elements defined on the Cartesian background mesh (elements at the zero-th refinement level) we set the operators to have a degree of $p = 3$. All refined elements have a degree of $p = 6$. Thus, we obtain a fully h/p non-conforming mesh. The resulting mesh is given in Figure 7.25. The mesh in Figure 7.25 consists of $DOFS = 1760$. Here, we observe coarse elements near the boundary of the domain and fine elements near the geometry surface. Generating a conforming mesh with the same properties is cumbersome

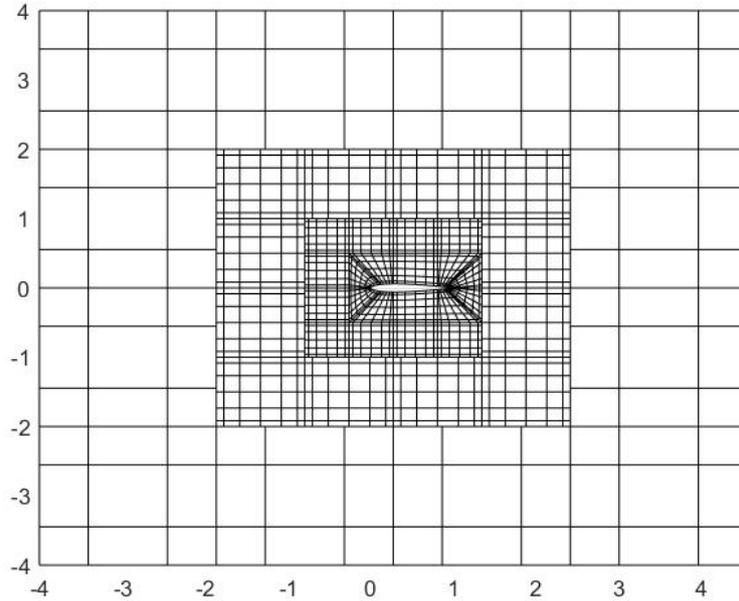


Figure 7.25.: Non-conforming mesh to simulate a flow around a NACA0012 air foil. The mesh consists of $DOFS = 1760$.

as they are instructed to remove any hanging corners. In addition, conforming meshes need to consider the same SBP operators with the same degree and thus, considering lower degree SBP operators on elements in the far field is not possible. The simplest approach of generating a conforming mesh around the NACA0012 air foil is given by considering the cut cell technique without any refinement levels. Decomposing the final-cut domains into curved elements at the zero-th refinement level gives us a conforming mesh as in Figure 7.26. In order to obtain the same resolution at the surface of the air foil as in Figure 7.25 we set the background mesh to consist of 16×16 Cartesian subdomains. By defining DG operators of degree $p = 6$ on each element the resolution of the surface elements in Figure 7.25 and in Figure 7.26 are the same. Here, the conforming mesh consists of $DOFS = 12740$ which is seven times more $DOFS$ as for the non-conforming mesh. With these two meshes we simulate a flow around a NACA0012 air foil.

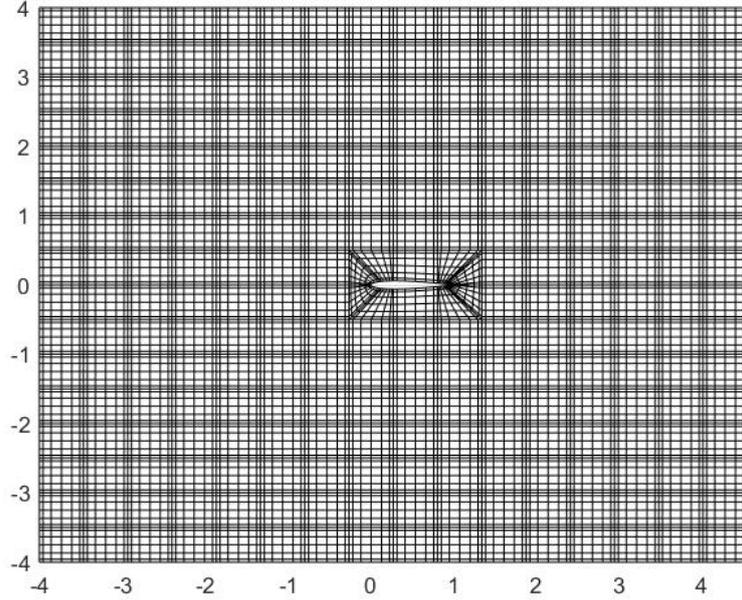


Figure 7.26.: Conforming mesh to simulate a flow around a NACA0012 air foil with $DOFS = 12740$.

We set the constant initial conditions to be

$$\begin{pmatrix} \rho \\ \nu_1 \\ \nu_2 \\ \mathcal{P} \end{pmatrix} = \begin{pmatrix} 1 \\ \cos(\theta) \\ \sin(\theta) \\ \frac{1}{\gamma(Ma)^2} \end{pmatrix}, \quad (7.4)$$

with adiabatic coefficient $\gamma = 1.4$, Mach number $Ma = 0.3$ and $\theta = 0.0349066$. We consider Dirichlet boundary conditions defined by (7.4). At the surface of the air foil we define reflecting boundary conditions

$$\begin{pmatrix} \rho \\ \nu_1 \\ \nu_2 \\ \mathcal{P} \end{pmatrix}^{reflect} := \begin{pmatrix} \rho^- \\ \nu_1^- - 2(n_1\nu_1^- + n_2\nu_2^-)n_1 \\ \nu_2^- - 2(n_1\nu_1^- + n_2\nu_2^-)n_2 \\ \mathcal{P}^- \end{pmatrix}, \quad (7.5)$$

where $(\rho^-, \nu_1^-, \nu_2^-, \mathcal{P}^-)$ denote the inner state and $\mathbf{n} = (n_1, n_2)^T$ denotes the outward pointing approximated normal vector. We set $T = 0.5$ and $CFL = 1$. Note,

that we do not reach a steady state. However, reaching a steady state is not necessary to demonstrate the advantages of non-conforming meshes. The simulation results of a flow around the NACA0012 air foil are given in Fig 7.27.

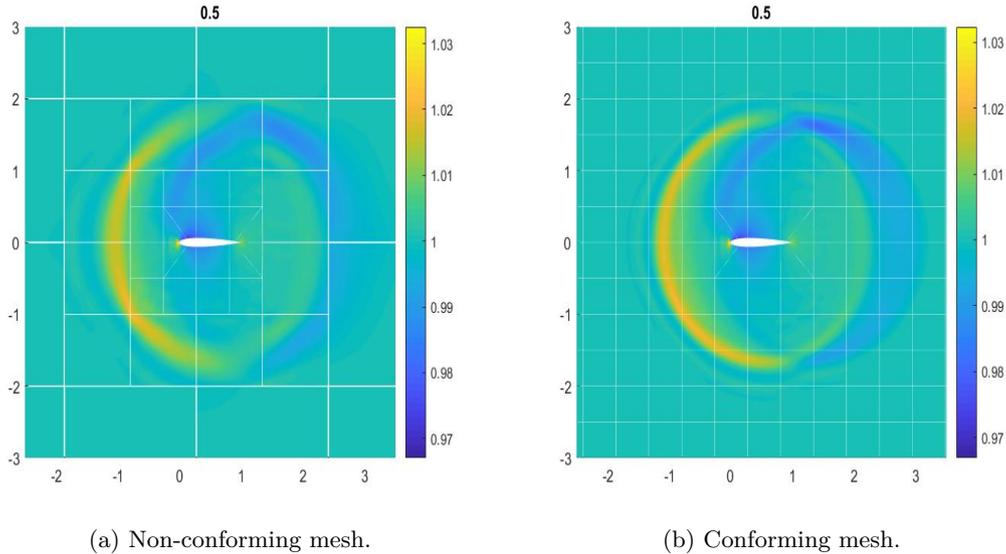


Figure 7.27.: Plot of the density ρ at $T = 0.5$ on two different meshes.

Here, we plotted the density with the mesh to distinguish between the conforming and non-conforming simulation. For both simulations we obtained the following CPU run times

- Non-conforming simulation: 2873 seconds,
- Conforming simulation: 17632 seconds.

Here, the CPU run time for the conforming simulation is six times more expensive than the non-conforming simulation, presumably due to the reduced number of *DOFS*. A factor of six is a significant speed up which demonstrates the beneficial properties of non-conforming meshes.

Besides focusing on the CPU run time we are interested in the fluid behaviour. Here, comparing these two plots we notice slightly more smearing of the density in the far field on the non-conforming mesh. This result was predictable as the non-conforming mesh has a coarser resolution in the far field. For analyzing the flow behaviour at the surface of the air foil we zoom in the plots of Figure 7.27. The zoomed in plots are presented in Figure 7.28.

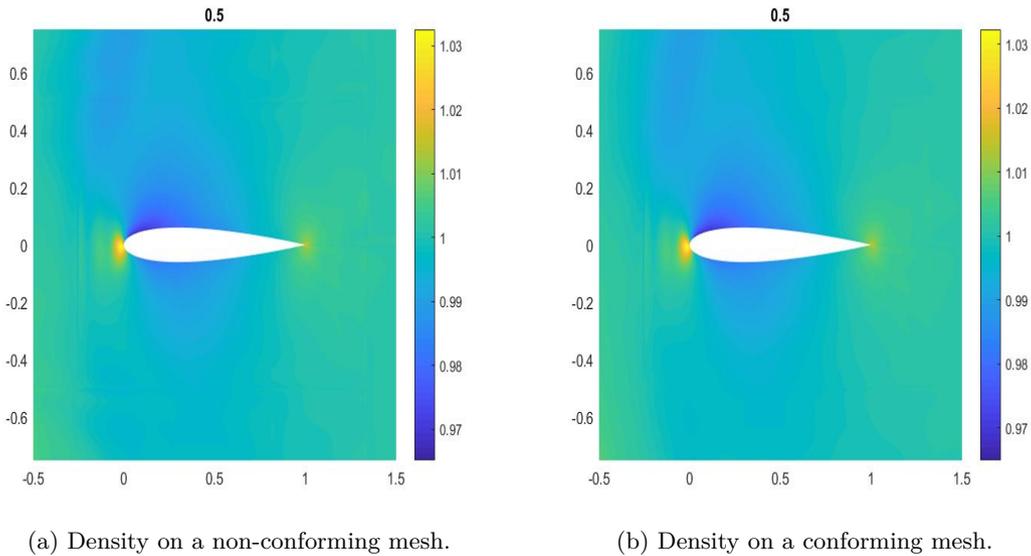


Figure 7.28.: Plot of the density ρ at $T = 0.5$ on two different meshes. The plot is zoomed in on $[-0.5, 1.5] \times [-0.75, 0.75]$.

Here, we do not plot the interface lines of the mesh in Figure 7.28(a) and (b) as these are the same for both meshes near the geometry surface. For both plots we cannot see any difference in the eyeball norm, despite the fact that the non-conforming mesh has a coarse mesh resolution in the far field. So, we obtain the same results at the air foil surface (with the naked eye) for both meshes. However, we generated these results more than six times faster on a non-conforming mesh.

7.3. Summary

In this chapter we introduced a modified cut cell approach for generating non-conforming meshes around complex geometries, e.g. a cylinder or an air foil. Typically, cut cell meshes are constructed for low-order schemes on Cartesian meshes. Therefore, we introduced curved elements near the surface of the complex geometry to obtain a high-order mesh representation. Here, the surface is approximated by degree p polynomials, where p denotes the degree of the SBP operator. The cut cell approach is advantageous, as it is a fully adaptive and fast method generating structured meshes around complex geometries.

We discussed the implementation of the cut cell method using a quadtree which gives a fully structured Cartesian mesh. Decomposing the final-cut domains into

curved elements we obtained a mesh with a high-order approximation of the surface of a complex geometry. Additionally, we introduced further mesh modification to obtain better shaped elements and an easier implementation. With the generator we presented non-conforming meshes around cylinders and air foils with multiple components.

Finally, we applied the non-conforming primary conservative and entropy stable SBP scheme to simulate a flow around a NACA0012 air foil. Here, we consider a conforming and a non-conforming mesh. Due to the cut cell approach the non-conforming mesh had a coarse resolution in the far field and a fine resolution at the geometry surface for the non-conforming mesh. As conforming meshes are less flexible all elements are instructed to have the same fine mesh resolution. Thus, by constructing a conforming mesh in the same simple fashion as the non-conforming mesh, we obtain seven times more degrees of freedom. However, when comparing the flow at the air foil surface both simulations show the same fluid behaviour, see Figure 7.28. Comparing the CPU run time, the non-conforming simulation was six times faster than the conforming simulation. Thus, we obtain a large reduction of computational costs considering non-conforming elements.

8. Global Summary

In this work we introduced high-order primary conservative and entropy stable summation-by-parts (SBP) schemes for systems of hyperbolic conservation laws. Such methods have been developed over the last years. However, most methods rely on a point-to-point transfer at the interfaces between two elements for which the mesh flexibility is limited. Therefore, we described how to extend high-order primary conservative and entropy stable schemes to a non-conforming mesh.

This work began focusing on the simplest conservation law: the linear advection equation (or LAE). Based on the LAE we introduced finite difference SBP operators (FD operators) and the multi-block SBP-SAT discretization to obtain a primary conservative and energy stable scheme in Chapter 2. Besides focusing on finite differences we introduced a discretization based on a discontinuous Galerkin approach. Even though finite difference and discontinuous Galerkin follow a different motivation, both presented approaches in this work belonged to the class of SBP schemes. This indicates the broad variety of SBP methods. Besides focusing on FD and DG operators we introduced the HGTL operators.

We extended the SBP scheme to a h/p non-conforming mesh in Chapter 3. As the construction of entropy stable methods on curved meshes with h non-conforming interfaces is still not known, we purely focus on Cartesian elements. In order to connect the non-conforming elements we introduced projection operators. Focusing on the construction of these operators we discussed the importance of the M -compatibility condition to ensure energy/entropy stability. Unfortunately, the compatibility condition occurs a degree reduction of the projection operator due to the accuracy of SBP norm matrices. Therefore, we introduced degree preserving SBP operators for which the corresponding projection operators is of the same degree as the differentiation matrix. Thus, considering such SBP operators the non-conforming scheme obtained the same convergence rate as on a conforming mesh.

For non-linear conservation laws deriving high-order primary conservative and entropy stable schemes is more complicated, as discussed in Chapter 4. Here, we introduced split forms and corresponding skew-symmetric discretizations. A

breakthrough within the research field of high-order entropy stable methods is the SBP Flux Difference method which requires no knowledge of an explicit skew-symmetric form. However, in certain cases, as for the Burgers' or the Shallow Water equations, the SBP Flux Difference method recovers known skew-symmetric discretizations. Additionally, we verified the robustness of entropy stable schemes by comparing simulations of the SBP Flux Difference method with the standard SBP discretization.

Next, we discussed non-conforming schemes for non-linear conservation laws. A state-of-the-art scheme is the Mortar method which is primary conservative, but not entropy stable. To obtain an entropy stable scheme for non-linear conservation laws on non-conforming meshes, we extended the SBP Flux Difference scheme to h/p non-conforming Cartesian meshes in Chapter 5. Here, we combined the ideas of Chapters 3 and 4. Additionally, we paid special attention to the interface coupling. In order to avoid non-linear projection terms we consider pure linear projections of the numerical surface flux. Thus, the proof of primary conservation and entropy stability was done in a one-to-one fashion as for linear conservation laws. Again, we demonstrated the enhanced robustness of entropy stable schemes by comparing simulations of the non-conforming SBP Flux Difference scheme with the Mortar method.

All presented schemes in Chapters 2-5 concerned semi-discrete schemes. Thus, when applying time integration methods with a large time step we introduce temporal errors which can lead to an unpredictable behaviour of the entropy. Therefore, we introduced the fully-discrete space-time entropy stable method in Chapter 6. The main idea is to discretize the temporal derivative in a similar fashion as for the SBP Flux Difference scheme specifying numerical volume and surface states for the numerical solution. The high-order primary conservative and entropy stable SBP space-time method can be applied on conforming and non-conforming elements. For non-conforming elements we obtain the additional flexibility to consider different nodal distributions in time for different elements. Besides all these advantageous results the SBP space-time method has high computational costs because it is fully implicit. Thus, from a today's point of view we recommend to consider semi-discrete schemes.

In order to obtain non-conforming meshes around complex geometries we introduced a mesh generation strategy based on a cut cell approach in Chapter 6. By introducing conforming curved elements at the geometry surface and element merging/refinement strategies we obtained meshes around cylinders and air foils.

As a final application we simulated a flow around a NACA0012 air foil using the high-order primary conservative and entropy stable SBP Flux Difference scheme. Here, we focused on a non-conforming as well as a conforming mesh, where both meshes had the same refinement at the geometry surface. For both simulations we could not see any differences of the flow behaviour near the surface of the air foil. However, the non-conforming simulation was about six times faster and its mesh had seven times less degrees of freedom than the conforming mesh.

8.1. Conclusion

Entropy stable schemes are beneficial as they have enhanced robustness. When deriving high-order primary conservative and entropy stable schemes it is a suitable choice to consider SBP operators. These operators discretely mimic the integration-by-parts rule. Here, we introduced FD, DG, HGTL and DP operators. All these operators had advantageous features as small errors, large stable time steps or a higher accurate norm matrix demonstrated in Sections 2.3, 3.5, 4.3 and 6.2. The choice of the "best" operator depends on the user preferences (CPU run time versus accuracy), problem (linear or non-linear problem) and mesh (conforming or non-conforming mesh). However, as long as the operators are SBP operators we obtain a high-order primary conservative and entropy stable scheme.

At the beginning of this work, we asked the question if it is possible to maintain the properties of a conforming SBP scheme (e.g. SBP Flux Difference scheme) as convergence order, primary conservation and entropy stability to a non-conforming mesh. The answer to this question is: yes!

Non-conforming elements are advantageous. We obtain the ability to arbitrary refine elements independent of their neighbours. Therefore, we can control the mesh resolution, e.g. a fine resolution near a geometry surface and a coarse resolution in the far field as in Chapter 7. Compared with simulations on conforming meshes the use of non-conforming methods can lead to a massive savings of CPU time. Additionally, as the interfaces of non-conforming elements do not necessarily have to coincide the issue of mesh generation becomes much simpler. Further flexibility is given as we can switch between different SBP operators on different elements within the same simulation. Therefore, it is reasonable to consider non-conforming elements in future work.

8.2. Outlook

Within all numerical schemes presented in this work we relied on the use of SBP operators. The development of such operators is ongoing research, especially constructing degree preserving operators for non-conforming applications. Due to the work of Ranocha [99] and Chan [17] it is possible to consider SBP operators where the boundary points are not included and thus, additional degrees of freedom are given which can be optimized.

From a practical point of view a large range of problems are defined in three dimensional space. Therefore, further implementations must be done, e.g. for the mesh generation strategy we need to extend the quadtree to an octree and to consider decomposition strategies for final-cut domains in a three dimensional domain. In general, further decomposition strategies should be considered even for two dimensional meshes in order to obtain better shaped elements. However, due to use of the tensor product, the theory on entropy stable schemes and non-conforming elements does not change. Especially, the development of a higher dimensional primary conservative and entropy stable code is necessary for space-time applications. In order to speed up the space-time code further investigations need to be done. A possibility is the use of suitable preconditioners to solve the linear system for the numerical solution.

In the research field of non-conforming elements the next milestone is to derive primary conservative and entropy stable schemes which can handle curved interfaces with hanging corners (h refinement). This would include further flexibility for the mesh generation, e.g. by including a non-conforming boundary layer. Thus, investigations on non-conforming elements is still ongoing research.

A. Appendix

A.1. Primary Conservative and Energy Stable Discretization for the Linear Advection Equation on Conforming Curved Elements

Herein, we provide a primary conservative and entropy stable discretization of the two dimensional LAE on a curvilinear, conforming mesh. The standard SBP discretization (2.129) is insufficient as the non-constant metric terms introduce non-linearities as e.g. $\tilde{\mathcal{F}} = \tilde{a}\mathcal{U}$ can be interpreted as a non-linear term. Therefore, we consider a skew-symmetric discretization as in Section 4.1.1. We focus on the transformed two dimensional conservation law (2.121) with $\tilde{\mathcal{F}} = \tilde{a}\mathcal{U}$ and $\tilde{\mathcal{G}} = \tilde{b}\mathcal{U}$

$$J \frac{\partial \mathcal{U}}{\partial t} + \frac{\partial (\tilde{a}\mathcal{U})}{\partial \xi} + \frac{\partial (\tilde{b}\mathcal{U})}{\partial \eta} = 0. \quad (\text{A.1})$$

with $\tilde{a} := \frac{\partial y}{\partial \eta} a - \frac{\partial x}{\partial \eta} b$ and $\tilde{b} := \frac{\partial x}{\partial \xi} b - \frac{\partial y}{\partial \xi} a$. Due to the product rule an algebraically equivalent form of (A.1) is given by

$$J \frac{\partial \mathcal{U}}{\partial t} + \frac{1}{2} \frac{\partial \tilde{a}\mathcal{U}}{\partial \xi} + \frac{1}{2} \left(\tilde{a} \frac{\partial \mathcal{U}}{\partial \xi} + \mathcal{U} \frac{\partial \tilde{a}}{\partial \xi} \right) + \frac{1}{2} \frac{\partial \tilde{b}\mathcal{U}}{\partial \eta} + \frac{1}{2} \left(\tilde{b} \frac{\partial \mathcal{U}}{\partial \eta} + \mathcal{U} \frac{\partial \tilde{b}}{\partial \eta} \right) = 0. \quad (\text{A.2})$$

Instead of discretizing (A.1) we focus on (A.2). Using the tensor product the two dimensional discretization on an element with $N + 1$ nodes in ξ - and η -direction we get

$$J_{ij} \omega_i \omega_j \frac{\partial u_{ij}}{\partial t} + \omega_j (Dx)_{ij} + \omega_i (Dy)_{ij} = 0, \quad (\text{A.3})$$

A. Appendix

where u_{ij} denotes the discrete solution for $i, j = 0, \dots, N$ and

$$\begin{aligned}
 (Dx)_{ij} &:= \frac{1}{2} \left(\sum_{m=0}^N \mathbf{Q}_{im} \tilde{a}_{mj}^p u_{mj} + \tilde{a}_{ij}^p \sum_{m=0}^N \mathbf{Q}_{im} u_{mj} + u_{ij} \sum_{m=0}^N \mathbf{Q}_{im} \tilde{a}_{mj}^p \right) \\
 &\quad - \left(\delta_{iN} [\tilde{a}^p u - \tilde{f}^*]_{Nj} - \delta_{i0} [\tilde{a}^p u - \tilde{f}^*]_{0j} \right), \\
 (Dy)_{ij} &:= \frac{1}{2} \left(\sum_{m=0}^N \mathbf{Q}_{jm} \tilde{b}_{im}^p u_{im} + \tilde{b}_{ij}^p \sum_{m=0}^N \mathbf{Q}_{jm} u_{im} + u_{ij} \sum_{m=0}^N \mathbf{Q}_{jm} \tilde{b}_{im}^p \right) \\
 &\quad - \left(\delta_{jN} [\tilde{b}^p u - \tilde{g}^*]_{iN} - \delta_{j0} [\tilde{b}^p u - \tilde{g}^*]_{i0} \right),
 \end{aligned} \tag{A.4}$$

with $i, j = 0, \dots, N$. Here, \tilde{a}^p and \tilde{b}^p are defined by

$$\tilde{a}^p := \frac{\partial y^p}{\partial \eta} a - \frac{\partial x^p}{\partial \eta} b \quad \tilde{b}^p := \frac{\partial x^p}{\partial \xi} b - \frac{\partial y^p}{\partial \xi} a. \tag{A.5}$$

The terms x^p, y^p are physical coordinates of the curved element approximated by a polynomial of degree p , where p denotes the degree of the differentiation matrix \mathbf{D} [65]. As x^p, y^p are of degree p we can evaluate their derivatives exactly using the differentiation matrix \mathbf{D}

$$\begin{aligned}
 \frac{\partial x_{ij}^p}{\partial \xi} &= \sum_{m=0}^N \mathbf{D}_{im}(x^p)_{mj}, & \frac{\partial x_{ji}^p}{\partial \eta} &= \sum_{m=0}^N \mathbf{D}_{jm}(x^p)_{im}, \\
 \frac{\partial y_{ij}^p}{\partial \xi} &= \sum_{m=0}^N \mathbf{D}_{im}(y^p)_{mj}, & \frac{\partial y_{ji}^p}{\partial \eta} &= \sum_{m=0}^N \mathbf{D}_{jm}(y^p)_{im},
 \end{aligned} \tag{A.6}$$

with $i, j = 0, \dots, N$. In terms of readability we set $x_\xi = \frac{\partial x^p}{\partial \xi}, x_\eta = \frac{\partial x^p}{\partial \eta}, y_\xi = \frac{\partial y^p}{\partial \xi}, y_\eta = \frac{\partial y^p}{\partial \eta}$, which are denoted as *approximated metric terms*. As x^p, y^p are smooth functions it holds due to Schwarz' theorem

$$\begin{aligned}
 \frac{\partial}{\partial \xi} \frac{\partial y^p}{\partial \eta} &= \frac{\partial}{\partial \eta} \frac{\partial y^p}{\partial \xi}, \\
 \frac{\partial}{\partial \xi} \frac{\partial x^p}{\partial \eta} &= \frac{\partial}{\partial \eta} \frac{\partial x^p}{\partial \xi}.
 \end{aligned} \tag{A.7}$$

In addition, as x^p, y^p are of degree p an algebraically equivalent representation of (A.7) is given by

$$\sum_{m=0}^N \mathbf{D}_{im}(y_\eta)_{mj} = \sum_{m=0}^N \mathbf{D}_{jm}(y_\xi)_{im}, \tag{A.8}$$

and

$$\sum_{m=0}^N \mathbf{D}_{im}(x_\eta)_{mj} = \sum_{m=0}^N \mathbf{D}_{jm}(x_\xi)_{im}, \tag{A.9}$$

A. Appendix

with $i, j = 0, \dots, N$. The equations (A.8) and (A.9) are denoted as *metric identities*, and are indispensable for proving free stream preservation and energy stability as shown below.

Remark 21. We do not consider the exact metric terms, as these can be described by an arbitrary function (and not by a polynomial of degree p). Thus, they do not necessarily satisfy the metric identities (A.8) and (A.9).

First, we verify that discretization (A.3) is free stream preservative, meaning that when the initial solution is constant, then the solution should remain constant over time. Mathematically speaking, $\frac{\partial u_{ij}}{\partial t}$ should be zero when setting u_{ij} constant for all $i, j = 0, \dots, N$. Setting all $u_{ij} = 1$ we look at Dx, Dy :

$$\begin{aligned} (Dx)_{ij} &= \frac{1}{2} \left(\sum_{m=0}^N \mathbf{Q}_{im} \tilde{a}_{mj}^p + \tilde{a}_{ij}^p \sum_{m=0}^N \mathbf{Q}_{im} + \sum_{m=0}^N \mathbf{Q}_{im} \tilde{a}_{mj}^p \right), \\ (Dy)_{ij} &= \frac{1}{2} \left(\sum_{m=0}^N \mathbf{Q}_{jm} \tilde{b}_{im}^p + \tilde{b}_{ij}^p \sum_{m=0}^N \mathbf{Q}_{jm} + \sum_{m=0}^N \mathbf{Q}_{jm} \tilde{b}_{im}^p \right). \end{aligned} \quad (\text{A.10})$$

The surface terms vanish as we assume that the numerical flux is consistent, meaning $\tilde{f}^*(1, 1) = f(1) = \tilde{a}^p$. As we consider SBP operators of degree $p > 0$, we know that $\sum_{m=0}^N \mathbf{Q}_{im} = 0$ which yields

$$(Dx)_{ij} = \sum_{m=0}^N \mathbf{Q}_{im} \tilde{a}_{mj}^p, \quad (Dy)_{ij} = \sum_{m=0}^N \mathbf{Q}_{jm} \tilde{b}_{im}^p. \quad (\text{A.11})$$

Including this in (A.3) we get

$$J_{ij} \omega_i \omega_j \frac{\partial u_{ij}}{\partial t} + \omega_j \sum_{m=0}^N \mathbf{Q}_{im} \tilde{a}_{mj}^p + \omega_i \sum_{m=0}^N \mathbf{Q}_{jm} \tilde{b}_{im}^p = 0. \quad (\text{A.12})$$

Since $\mathbf{Q}_{im} = \omega_i \mathbf{D}_{im}$ we divide by $\omega_i \omega_j$ and obtain

$$J_{ij} \frac{\partial u_{ij}}{\partial t} + \sum_{m=0}^N \mathbf{D}_{im} \tilde{a}_{mj}^p + \sum_{m=0}^N \mathbf{D}_{jm} \tilde{b}_{im}^p = 0. \quad (\text{A.13})$$

As $\tilde{a}^p = y_\eta a - x_\eta b$ and $\tilde{b}^p = x_\xi b - y_\xi a$ we get

$$\begin{aligned} J_{ij} \frac{\partial u_{ij}}{\partial t} &+ a \left(\sum_{m=0}^N \mathbf{D}_{im} (y_\eta)_{mj} - \sum_{m=0}^N \mathbf{D}_{jm} (y_\xi)_{im} \right) \\ &+ b \left(\sum_{m=0}^N \mathbf{D}_{jm} (x_\xi)_{im} - \sum_{m=0}^N \mathbf{D}_{im} (x_\eta)_{mj} \right) = 0. \end{aligned} \quad (\text{A.14})$$

A. Appendix

Due to the construction of the discrete metric terms the metric identities (A.8) and (A.9) hold and we obtain

$$\frac{\partial u_{ij}}{\partial t} = 0, \quad (\text{A.15})$$

for $i, j = 0, \dots, N$. Therefore, a constant solution remains constant over time.

Secondly we will proof conservation for the primary quantity. Therefore, we consider discretization (A.3) and sum over all nodes $i, j = 0, \dots, N$ to approximate the integral over $\frac{\partial \mathcal{U}}{\partial t}$

$$\sum_{i,j=0}^N J_{ij} \omega_i \omega_j \frac{\partial u_{ij}}{\partial t} + \sum_{j=0}^N \omega_j \sum_{i=0}^N (Dx)_{ij} + \sum_{i=0}^N \omega_i \sum_{j=0}^N (Dy)_{ij} = 0, \quad (\text{A.16})$$

We focus on

$$\begin{aligned} \sum_{i=0}^N (Dx)_{ij} &= \frac{1}{2} \sum_{i=0}^N \left(\sum_{m=0}^N \mathbf{Q}_{im} \tilde{a}_{mj} u_{mj} + \tilde{a}_{ij} \sum_{m=0}^N \mathbf{Q}_{im} u_{mj} + u_{ij} \sum_{m=0}^N \mathbf{Q}_{im} \tilde{a}_{mj} \right) \\ &\quad - \sum_{i=0}^N \left(\delta_{iN} [\tilde{a}u - \tilde{f}^*]_{Nj} - \delta_{i0} [\tilde{a}u - \tilde{f}^*]_{0j} \right). \end{aligned} \quad (\text{A.17})$$

Setting $\mathbf{u}_{:j} := (u_{0j}, \dots, u_{Nj})^T$ and $\tilde{\mathbf{a}}_{:j} := (\tilde{a}_{0j}, \dots, \tilde{a}_{Nj})^T$ we can rewrite (A.17) by

$$\begin{aligned} \sum_{i=0}^N (Dx)_{ij} &= \frac{1}{2} \left(\mathbf{1}^T \mathbf{Q} (\tilde{\mathbf{a}}_{:j} \circ \mathbf{u}_{:j}) + \tilde{\mathbf{a}}_{:j}^T \mathbf{Q} \mathbf{u}_{:j} + \mathbf{u}_{:j}^T \mathbf{Q} \tilde{\mathbf{a}}_{:j} \right) \\ &\quad - [\tilde{a}u - \tilde{f}^*]_{Nj} + [\tilde{a}u - \tilde{f}^*]_{0j}, \end{aligned} \quad (\text{A.18})$$

where \circ denotes the Hadamard product (elementwise multiplication). We focus on the volume parts of (A.18) and as we consider SBP operators we know $\mathbf{Q} + \mathbf{Q}^T = \mathbf{B}$ with $\mathbf{B} = \text{diag}(-1, \dots, +1)$ which gives us

$$\mathbf{1}^T \mathbf{Q} (\tilde{\mathbf{a}}_{:j} \circ \mathbf{u}_{:j}) = \tilde{a}_{Nj} u_{Nj} - \tilde{a}_{0j} u_{0j} + (\tilde{\mathbf{a}}_{:j} \circ \mathbf{u}_{:j})^T \underbrace{\mathbf{Q} \mathbf{1}}_{=0} = \tilde{a}_{Nj} u_{Nj} - \tilde{a}_{0j} u_{0j}, \quad (\text{A.19})$$

and

$$\tilde{\mathbf{a}}_{:j}^T \mathbf{Q} \mathbf{u}_{:j} + \mathbf{u}_{:j}^T \mathbf{Q} \tilde{\mathbf{a}}_{:j} = \tilde{\mathbf{a}}_{:j}^T \mathbf{Q} \mathbf{u}_{:j} + \tilde{\mathbf{a}}_{:j}^T \mathbf{Q}^T \mathbf{u}_{:j} = \tilde{a}_{Nj} u_{Nj} - \tilde{a}_{0j} u_{0j}. \quad (\text{A.20})$$

Including this in (A.18) we get

$$\begin{aligned} \sum_{i=0}^N (Dx)_{ij} &= \tilde{a}_{Nj} u_{Nj} - \tilde{a}_{0j} u_{0j} - [\tilde{a}u - \tilde{f}^*]_{Nj} + [\tilde{a}u - \tilde{f}^*]_{0j}, \\ &= \tilde{f}_{Nj}^* - \tilde{f}_{0j}^*. \end{aligned} \quad (\text{A.21})$$

A. Appendix

By doing the same derivations for $\sum_{j=0}^N (Dy)_{ij}$ we get

$$\sum_{j=0}^N (Dy)_{ij} = \tilde{g}_{iN}^* - \tilde{g}_{i0}^*. \quad (\text{A.22})$$

All together this gives us

$$\sum_{i,j=0}^N J_{ij} \omega_i \omega_j \frac{\partial u_{ij}}{\partial t} + \sum_{j=0}^N \omega_j (\tilde{f}_{Nj}^* - \tilde{f}_{0j}^*) + \sum_{i=0}^N \omega_i (\tilde{g}_{iN}^* - \tilde{g}_{i0}^*) = 0, \quad (\text{A.23})$$

Next, we focus on the growth of the primary quantity on the interior interfaces. Focusing on an interface with respect to \tilde{f}^* we get

$$IU_t = - \sum_{j=0}^N \omega_j \tilde{f}_{Nj}^{*,L} + \sum_{j=0}^N \omega_j \tilde{f}_{0j}^{*,R}. \quad (\text{A.24})$$

Note, that ω_j does not have a superscript index L or R as we focus on a conforming mesh and thus all SBP operators in each element are the same. On a conforming mesh we set $\tilde{f}_j^* := \tilde{f}_{Nj}^{*,L} = \tilde{f}_{0j}^{*,R}$ and obtain

$$IU_t = - \sum_{j=0}^N \omega_j \tilde{f}_j^* + \sum_{j=0}^N \omega_j \tilde{f}_j^* = 0. \quad (\text{A.25})$$

So the terms which remain to describe the growth of the primary quantity are defined on the boundary. This indicates primary conservation. Note, that this holds independent of the choice of the numerical surface flux.

Next, we will proof energy stability. Therefore we multiply (A.3) by the solution u_{ij} sum over all $i, j = 0, \dots, N$ to approximate the integral of $\frac{\partial}{\partial t} \left(\frac{1}{2} \mathcal{U}^2 \right)$

$$\sum_{i,j=0}^N J_{ij} \omega_i \omega_j \frac{\partial}{\partial t} \left(\frac{1}{2} u_{ij}^2 \right) + \sum_{j=0}^N \omega_j \sum_{i=0}^N u_{ij} (Dx)_{ij} + \sum_{i=0}^N \omega_i \sum_{j=0}^N u_{ij} (Dy)_{ij} = 0. \quad (\text{A.26})$$

We focus on

$$\begin{aligned} & \sum_{i=0}^N u_{ij} (Dx)_{ij} \\ &= \frac{1}{2} \left(\sum_{i=0}^N u_{ij} \sum_{m=0}^N \mathbf{Q}_{im} \tilde{a}_{mj} u_{mj} + \sum_{i=0}^N u_{ij} \tilde{a}_{ij} \sum_{m=0}^N \mathbf{Q}_{im} u_{mj} + \sum_{i=0}^N u_{ij}^2 \sum_{m=0}^N \mathbf{Q}_{im} \tilde{a}_{mj} \right) \\ & \quad - \sum_{i=0}^N u_{ij} \left(\delta_{iN} [\tilde{a}u - \tilde{f}^*]_{Nj} - \delta_{i0} [\tilde{a}u - \tilde{f}^*]_{0j} \right), \end{aligned} \quad (\text{A.27})$$

A. Appendix

or in a more compact matrix vector notation we get

$$\begin{aligned} & \sum_{i=0}^N u_{ij}(Dx)_{ij} \\ &= \frac{1}{2} \left(\mathbf{u}_{:j}^T \mathbf{Q} (\tilde{\mathbf{a}}_{:j} \circ \mathbf{u}_{:j}) + (\mathbf{u}_{:j} \circ \tilde{\mathbf{a}}_{:j})^T \mathbf{Q} \mathbf{u}_{:j} + (\mathbf{u}_{:j}^2)^T \mathbf{Q} \tilde{\mathbf{a}}_{:j} \right) \\ & \quad - [\tilde{a}u^2 - u\tilde{f}^*]_{Nj} - [\tilde{a}u^2 - u\tilde{f}^*]_{0j}. \end{aligned} \quad (\text{A.28})$$

Due to the SBP property we know

$$\mathbf{u}_{:j}^T \mathbf{Q} (\tilde{\mathbf{a}}_{:j} \circ \mathbf{u}_{:j}) + (\mathbf{u}_{:j} \circ \tilde{\mathbf{a}}_{:j})^T \mathbf{Q} \mathbf{u}_{:j} = \tilde{a}_{Nj} u_{Nj}^2 - \tilde{a}_{0j} u_{0j}^2, \quad (\text{A.29})$$

which leads to

$$\sum_{i=0}^N u_{ij}(Dx)_{ij} = \frac{1}{2} (\mathbf{u}_{:j}^2)^T \mathbf{Q} \tilde{\mathbf{a}}_{:j} - \left[\frac{1}{2} \tilde{a}u^2 - u\tilde{f}^* \right]_{Nj} + \left[\frac{1}{2} \tilde{a}u^2 - u\tilde{f}^* \right]_{0j}. \quad (\text{A.30})$$

Doing the same derivations we get

$$\sum_{j=0}^N u_{ij}(Dy)_{ij} = \frac{1}{2} (\mathbf{u}_{:i}^2)^T \mathbf{Q} \tilde{\mathbf{b}}_{:i} - \left[\frac{1}{2} \tilde{b}u^2 - u\tilde{g}^* \right]_{iN} + \left[\frac{1}{2} \tilde{b}u^2 - u\tilde{g}^* \right]_{i0}. \quad (\text{A.31})$$

In total this gives us

$$\begin{aligned} & \sum_{i=0}^N \sum_{j=0}^N J_{ij} \omega_i \omega_j \frac{\partial}{\partial t} \left(\frac{1}{2} u_{ij}^2 \right) \\ &= \sum_{j=0}^N \omega_j \frac{1}{2} (\mathbf{u}_{:j}^2)^T \mathbf{Q} \tilde{\mathbf{a}}_{:j} - \left[\frac{1}{2} \tilde{a}u^2 - u\tilde{f}^* \right]_{Nj} + \left[\frac{1}{2} \tilde{a}u^2 - u\tilde{f}^* \right]_{0j} \\ & \quad + \sum_{i=0}^N \omega_i \frac{1}{2} (\mathbf{u}_{:i}^2)^T \mathbf{Q} \tilde{\mathbf{b}}_{:i} - \left[\frac{1}{2} \tilde{b}u^2 - u\tilde{g}^* \right]_{iN} + \left[\frac{1}{2} \tilde{b}u^2 - u\tilde{g}^* \right]_{i0}. \end{aligned} \quad (\text{A.32})$$

Next, we focus on the volume parts of (A.32) denoted by VP , where

$$VP = \sum_{j=0}^N \omega_j \frac{1}{2} (\mathbf{u}_{:j}^2)^T \mathbf{Q} \tilde{\mathbf{a}}_{:j} + \sum_{i=0}^N \omega_i \frac{1}{2} (\mathbf{u}_{:i}^2)^T \mathbf{Q} \tilde{\mathbf{b}}_{:i}. \quad (\text{A.33})$$

Rewriting VP and using $\mathbf{Q} = \mathbf{M}\mathbf{D}$ gives us

$$VP = \frac{1}{2} \sum_{j,k=0}^N \omega_j \omega_k \mathbf{u}_{kj}^2 \sum_{m=0}^N \mathbf{D}_{km} \tilde{a}_{mj} + \frac{1}{2} \sum_{i,k=0}^N \omega_i \omega_k \mathbf{u}_{ik}^2 \sum_{m=0}^N \mathbf{D}_{km} \tilde{b}_{im}. \quad (\text{A.34})$$

A. Appendix

Rearranging the sums we get

$$VP = \frac{1}{2} \sum_{i,j=0}^N \omega_i \omega_j \mathbf{u}_{ij}^2 \left(\sum_{m=0}^N \mathbf{D}_{im} \tilde{a}_{mj} + \sum_{m=0}^N \mathbf{D}_{jm} \tilde{b}_{im} \right). \quad (\text{A.35})$$

Focusing on the term within in brackets and using $\tilde{a} = y_\eta a - x_\eta b$ and $\tilde{b} = x_\xi b - y_\xi a$ we get

$$\begin{aligned} \sum_{m=0}^N \mathbf{D}_{im} \tilde{a}_{mj} + \sum_{m=0}^N \mathbf{D}_{jm} \tilde{b}_{im} &= a \left(\sum_{m=0}^N \mathbf{D}_{im} (y_\eta)_{mj} - \sum_{m=0}^N \mathbf{D}_{jm} (y_\xi)_{im} \right) \\ &\quad + b \left(\sum_{m=0}^N \mathbf{D}_{jm} (x_\xi)_{im} - \sum_{m=0}^N \mathbf{D}_{im} (x_\eta)_{mj} \right), \quad (\text{A.36}) \\ &= 0. \end{aligned}$$

as we assume that the metric terms satisfy the metric identities (A.8) and (A.9). Therefore, we get $VP = 0$ and thus the discrete growth for the total energy reduces to surface contributions

$$\begin{aligned} \sum_{i=0}^N \sum_{j=0}^N J_{ij} \omega_i \omega_j \frac{\partial}{\partial t} \left(\frac{1}{2} u_{ij}^2 \right) &= - \left[\frac{1}{2} \tilde{a} u^2 - u \tilde{f}^* \right]_{Nj} + \left[\frac{1}{2} \tilde{a} u^2 - u \tilde{f}^* \right]_{0j} \\ &\quad - \left[\frac{1}{2} \tilde{b} u^2 - u \tilde{g}^* \right]_{iN} + \left[\frac{1}{2} \tilde{b} u^2 - u \tilde{g}^* \right]_{i0}. \quad (\text{A.37}) \end{aligned}$$

Again, we focus on an interior interface. Due to the conformity of the mesh and get

$$IE_t = \sum_{j=0}^N \omega_j \left[\frac{1}{2} \tilde{a} (u^L)^2 - u^L \tilde{f}^{*,L} \right]_{Nj} - \sum_{j=0}^N \omega_j \left[\frac{1}{2} \tilde{a} (u^R)^2 - u^R \tilde{f}^{*,R} \right]_{0j}, \quad (\text{A.38})$$

setting $\tilde{f}_j^* := \tilde{f}_{Nj}^{*,L} = \tilde{f}_{0j}^{*,R}$ and ignoring the first index within u^L, u^R, \tilde{a} and \tilde{b} (as these are always constant) we get

$$IE_t = \sum_{j=0}^N \omega_j \left(\frac{1}{2} \tilde{a} (u^L)^2 - u^L \tilde{f}^* - \frac{1}{2} \tilde{a} (u^R)^2 + u^R \tilde{f}^* \right)_j. \quad (\text{A.39})$$

A. Appendix

Rewriting the contravariant fluxes in terms of f^* and g^* we obtain

$$\begin{aligned}
IE_t &= \sum_{j=0}^N \omega_j \left(\frac{1}{2} (y_\eta a - x_\eta b) (u^L)^2 - u^L (y_\eta f^* - x_\eta g^*) \right) \\
&\quad - \sum_{j=0}^N \omega_j \left(\frac{1}{2} (y_\eta a - x_\eta b) (u^R)^2 + u^R (y_\eta f^* - x_\eta g^*) \right)_j, \\
&= \sum_{j=0}^N \omega_j \left(y_\eta \left(\frac{1}{2} a (u^L)^2 - u^L f^* - \frac{1}{2} a (u^R)^2 + u^R f^* \right) \right)_j \\
&\quad - \sum_{j=0}^N \omega_j \left(x_\eta \left(\frac{1}{2} b (u^L)^2 - u^L g^* - \frac{1}{2} b (u^R)^2 + u^R g^* \right) \right)_j.
\end{aligned} \tag{A.40}$$

We set the numerical fluxes to be

$$\begin{aligned}
f^* &:= a \frac{u^R + u^L}{2} - \left(\frac{y_\eta}{\sqrt{y_\eta^2 + x_\eta^2}} \right) \frac{|a|}{2} (u^R - u^L), \\
g^* &:= b \frac{u^R + u^L}{2} - \left(\frac{-x_\eta}{\sqrt{y_\eta^2 + x_\eta^2}} \right) \frac{|b|}{2} (u^R - u^L).
\end{aligned} \tag{A.41}$$

Including these fluxes in (A.40) we get

$$\begin{aligned}
IE_t &= - \frac{|a|}{2} \sum_{j=0}^N \omega_j \left(\frac{y_\eta^2}{\sqrt{y_\eta^2 + x_\eta^2}} (u^R - u^L)^2 \right)_j \\
&\quad - \frac{|b|}{2} \sum_{j=0}^N \omega_j \left(\frac{x_\eta^2}{\sqrt{y_\eta^2 + x_\eta^2}} (u^R - u^L)^2 \right)_j, \\
&\leq - \frac{\min\{|a|, |b|\}}{2} \sum_{j=0}^N \omega_j \left(\frac{y_\eta^2}{\sqrt{y_\eta^2 + x_\eta^2}} (u^R - u^L)^2 + \frac{x_\eta^2}{\sqrt{y_\eta^2 + x_\eta^2}} (u^R - u^L)^2 \right)_j, \\
&\leq - \sigma \underbrace{\frac{\min\{|a|, |b|\}}{2} \sum_{j=0}^N \omega_j (\sqrt{y_\eta^2 + x_\eta^2})_j (u^R - u^L)_j^2}_{\geq 0} \leq 0,
\end{aligned} \tag{A.42}$$

So the scheme is provable energy stable and primary conservative on a curvilinear mesh. To summarize, in comparison to the discretization on a Cartesian mesh,

we considered a skew-symmetric discretization (A.3). Assuming that the discrete metric terms satisfy the metric identities (A.8) and (A.9) we proved that the discretization is primary conservative and energy stable.

A.2. Primary Conservative and Entropy Stable Discretization for Hyperbolic Systems of Conservation Laws on Conforming Curved Elements

Herein, we provide a discretization of two dimensional hyperbolic systems with N_{Eq} equations on a curvilinear, conforming mesh. The standard Flux Difference discretization (4.64) is insufficient as the non-constant metric terms introduce further non-linearities. Therefore, we consider the two dimensional tensor product discretization on an arbitrary element with N nodes in x - and y -direction, where \mathbf{u}_{ij} denotes the discrete solution for $i, j = 0, \dots, N$ (compare Appendix A.1)

$$J_{ij}\omega_i\omega_j\frac{\partial\mathbf{u}_{ij}}{\partial t} + \omega_j(\mathbf{D}\mathbf{x})_{ij} + \omega_i(\mathbf{D}\mathbf{y})_{ij} = 0, \quad (\text{A.43})$$

with

$$\begin{aligned} (\mathbf{D}\mathbf{x})_{ij} &:= \sum_{m=0}^N \mathbf{Q}_{im} \left(\tilde{\mathbf{f}}^\#(\mathbf{u}_{ij}, \mathbf{u}_{mj}) + \tilde{\mathbf{f}}^\#(\mathbf{u}_{mj}, \mathbf{u}_{ij}) \right) \\ &\quad - \left(\delta_{iN}[\tilde{\mathbf{f}} - \tilde{\mathbf{f}}^*]_{Nj} - \delta_{i0}[\tilde{\mathbf{f}} - \tilde{\mathbf{f}}^*]_{0j} \right), \\ (\mathbf{D}\mathbf{y})_{ij} &:= \sum_{m=0}^N \mathbf{Q}_{jm} \left(\tilde{\mathbf{g}}^\#(\mathbf{u}_{ij}, \mathbf{u}_{im}) + \tilde{\mathbf{g}}^\#(\mathbf{u}_{im}, \mathbf{u}_{ij}) \right) \\ &\quad - \left(\delta_{jN}[\tilde{\mathbf{g}} - \tilde{\mathbf{g}}^*]_{iN} - \delta_{j0}[\tilde{\mathbf{g}} - \tilde{\mathbf{g}}^*]_{i0} \right). \end{aligned} \quad (\text{A.44})$$

Here, \mathbf{u} , \mathbf{f} and \mathbf{g} denote vectors of size N_{Eq} . The contravariant fluxes are defined by

$$\begin{aligned} \tilde{\mathbf{f}} &:= y_\eta \mathbf{f} - x_\eta \mathbf{g}, \\ \tilde{\mathbf{g}} &:= x_\xi \mathbf{g} - y_\xi \mathbf{f}, \end{aligned} \quad (\text{A.45})$$

where the approximated metric terms $y_\eta, x_\eta, y_\xi, x_\xi$ are constructed as in Appendix A.1. Thus, they satisfy the metric identities

$$\sum_{m=0}^N \mathbf{D}_{im}(y_\eta)_{mj} = \sum_{m=0}^N \mathbf{D}_{jm}(y_\xi)_{im}, \quad (\text{A.46})$$

and

$$\sum_{m=0}^N \mathbf{D}_{im}(x_\eta)_{mj} = \sum_{m=0}^N \mathbf{D}_{jm}(x_\xi)_{im}, \quad (\text{A.47})$$

A. Appendix

Within discretization (A.43) the two point volume flux is defined by

$$\begin{aligned}\tilde{\mathbf{f}}^\#(\mathbf{u}_{ij}, \mathbf{u}_{mj}) &:= (y_\eta)_{ij} \mathbf{f}^\#(\mathbf{u}_{ij}, \mathbf{u}_{mj}) - (x_\eta)_{ij} \mathbf{g}^\#(\mathbf{u}_{ij}, \mathbf{u}_{mj}), \\ \tilde{\mathbf{g}}^\#(\mathbf{u}_{ij}, \mathbf{u}_{mj}) &:= (x_\xi)_{ij} \mathbf{g}^\#(\mathbf{u}_{ij}, \mathbf{u}_{mj}) - (y_\xi)_{ij} \mathbf{f}^\#(\mathbf{u}_{ij}, \mathbf{u}_{mj}).\end{aligned}\tag{A.48}$$

We note that in comparison to the Cartesian case the two point volume flux is not symmetric as the metric terms refer to the first entry. Here, we set the volume fluxes to $\mathbf{f}^\# = \mathbf{f}_{EC}^*$ and $\mathbf{g}^\# = \mathbf{g}_{EC}^*$ where these fluxes satisfy the Tadmor Shuffle condition

$$\begin{aligned}(\mathbf{v}_{ij} - \mathbf{v}_{mj})^T \mathbf{f}_{EC}^*(\mathbf{u}_{ij}, \mathbf{u}_{mj}) &= (\Psi_{ij}^f - \Psi_{mj}^f), \\ (\mathbf{v}_{ij} - \mathbf{v}_{mj})^T \mathbf{g}_{EC}^*(\mathbf{u}_{ij}, \mathbf{u}_{mj}) &= (\Psi_{ij}^g - \Psi_{mj}^g),\end{aligned}\tag{A.49}$$

where \mathbf{v} denotes the entropy variables and Ψ^f, Ψ^g denote the flux potentials.

First we will prove free stream preservation, see Appendix A.1. Therefore we set all values of \mathbf{f} and \mathbf{g} to be constant vectors denoted by \mathbf{C}_f and \mathbf{C}_g . Looking at $\mathbf{D}\mathbf{x}, \mathbf{D}\mathbf{y}$ we get

$$\begin{aligned}(\mathbf{D}\mathbf{x})_{ij} &= \sum_{m=0}^N \mathbf{Q}_{im} ((y_\eta)_{ij} \mathbf{C}_f - (x_\eta)_{ij} \mathbf{C}_g + (y_\eta)_{mj} \mathbf{C}_f - (x_\eta)_{mj} \mathbf{C}_g), \\ (\mathbf{D}\mathbf{y})_{ij} &= \sum_{m=0}^N \mathbf{Q}_{jm} ((x_\xi)_{ij} \mathbf{C}_g - (y_\xi)_{ij} \mathbf{C}_f + (x_\xi)_{im} \mathbf{C}_g - (y_\xi)_{im} \mathbf{C}_f).\end{aligned}\tag{A.50}$$

Due to consistency of the the numerical fluxes $\tilde{\mathbf{f}}^\#, \tilde{\mathbf{g}}^\#, \tilde{\mathbf{f}}^*, \tilde{\mathbf{g}}^*$ the surface parts vanish and the volume fluxes remain constant. Rewriting $\mathbf{Q} = \mathbf{M}\mathbf{D}$ we get (A.43)

$$\begin{aligned}& J_{ij} \omega_i \omega_j \frac{\partial \mathbf{u}_{ij}}{\partial t} \\ &= -\omega_i \omega_j \sum_{m=0}^N \mathbf{D}_{im} ((y_\eta)_{ij} \mathbf{C}_f - (x_\eta)_{ij} \mathbf{C}_g + (y_\eta)_{mj} \mathbf{C}_f - (x_\eta)_{mj} \mathbf{C}_g) \\ &\quad - \omega_i \omega_j \sum_{m=0}^N \mathbf{D}_{jm} ((x_\xi)_{ij} \mathbf{C}_g - (y_\xi)_{ij} \mathbf{C}_f + (x_\xi)_{im} \mathbf{C}_g - (y_\xi)_{im} \mathbf{C}_f).\end{aligned}\tag{A.51}$$

Dividing by $\omega_i \omega_j$ and rearranging terms we obtain

$$\begin{aligned}J_{ij} \frac{\partial \mathbf{u}_{ij}}{\partial t} &= -\mathbf{C}_f \sum_{m=0}^N \mathbf{D}_{im} ((y_\eta)_{ij} + (y_\eta)_{mj} - (y_\xi)_{ij} - (y_\xi)_{im}) \\ &\quad - \mathbf{C}_g \sum_{m=0}^N \mathbf{D}_{im} ((x_\xi)_{ij} + (x_\xi)_{im} - (x_\eta)_{ij} - (x_\eta)_{mj}).\end{aligned}\tag{A.52}$$

A. Appendix

As the differentiation matrix is of degree $p > 0$ we know $\sum_{m=0}^N D_{im} = 0$ and thus

$$\begin{aligned} J_{ij} \frac{\partial \mathbf{u}_{ij}}{\partial t} = & -\mathbf{C}_f \left(\sum_{m=0}^N D_{im}(y_\eta)_{mj} - \sum_{m=0}^N D_{im}(y_\xi)_{im} \right) \\ & -\mathbf{C}_g \left(\sum_{m=0}^N D_{im}(x_\xi)_{im} - \sum_{m=0}^N D_{im}(x_\eta)_{mj} \right). \end{aligned} \quad (\text{A.53})$$

Due to the metric identities (A.46) and (A.47) we obtain

$$\frac{\partial \mathbf{u}_{ij}}{\partial t} = 0, \quad (\text{A.54})$$

which indicates free stream preservation.

Secondly we will proof conservation for the primary quantities. Therefore we sum (A.43) over all $i, j = 0, \dots, N$ to approximate the discrete integral of $\frac{\partial \mathbf{u}}{\partial t}$

$$\sum_{i,j=0}^N J_{ij} \omega_i \omega_j \frac{\partial \mathbf{u}_{ij}}{\partial t} + \sum_{j=0}^N \omega_j \sum_{i=0}^N (\mathbf{D}\mathbf{x})_{ij} + \sum_{i=0}^N \omega_i \sum_{j=0}^N (\mathbf{D}\mathbf{y})_{ij} = 0, \quad (\text{A.55})$$

Focusing on $\sum_{i,j=0}^N \omega_j (\mathbf{D}\mathbf{x})_{ij}$ we get

$$\begin{aligned} \sum_{i=0}^N (\mathbf{D}\mathbf{x})_{ij} = & \sum_{i=0}^N \sum_{m=0}^N \mathbf{Q}_{im} \tilde{\mathbf{f}}^\#(\mathbf{u}_{ij}, \mathbf{u}_{mj}) + \sum_{i=0}^N \sum_{m=0}^N \mathbf{Q}_{im} \tilde{\mathbf{f}}^\#(\mathbf{u}_{mj}, \mathbf{u}_{ij}) \\ & - \sum_{i=0}^N \left(\delta_{iN} [\tilde{\mathbf{f}} - \tilde{\mathbf{f}}^*]_{Nj} - \delta_{i0} [\tilde{\mathbf{f}} - \tilde{\mathbf{f}}^*]_{0j} \right), \end{aligned} \quad (\text{A.56})$$

Using the SBP property $\mathbf{Q} = \mathbf{B} - \mathbf{Q}^T$ we obtain

$$\begin{aligned} \sum_{i=0}^N (\mathbf{D}\mathbf{x})_{ij} = & \sum_{i=0}^N \sum_{m=0}^N \mathbf{Q}_{im} \tilde{\mathbf{f}}^\#(\mathbf{u}_{ij}, \mathbf{u}_{mj}) - \sum_{i=0}^N \sum_{m=0}^N \mathbf{Q}_{mi} \tilde{\mathbf{f}}^\#(\mathbf{u}_{mj}, \mathbf{u}_{ij}) \\ & + \sum_{i=0}^N \sum_{m=0}^N \mathbf{B}_{im} \tilde{\mathbf{f}}^\#(\mathbf{u}_{mj}, \mathbf{u}_{ij}) - [\tilde{\mathbf{f}} - \tilde{\mathbf{f}}^*]_{Nj} + [\tilde{\mathbf{f}} - \tilde{\mathbf{f}}^*]_{0j}, \end{aligned} \quad (\text{A.57})$$

By rearranging indices the two sum with \mathbf{Q} included cancel each other out and due to the structure of $\mathbf{B} = \text{diag}(-1, 0, \dots, 0, +1)$ and the consistency of $\tilde{\mathbf{f}}^\#$ it remains

$$\begin{aligned} \sum_{i=0}^N (\mathbf{D}\mathbf{x})_{ij} = & \tilde{\mathbf{f}}(\mathbf{u}_{Nj}) - \tilde{\mathbf{f}}(\mathbf{u}_{0j}) - [\tilde{\mathbf{f}} - \tilde{\mathbf{f}}^*]_{Nj} + [\tilde{\mathbf{f}} - \tilde{\mathbf{f}}^*]_{0j}, \\ = & \tilde{\mathbf{f}}_{Nj}^* - \tilde{\mathbf{f}}_{0j}^*. \end{aligned} \quad (\text{A.58})$$

A. Appendix

By doing the same derivations for $\sum_{j=0}^N (\mathbf{D}\mathbf{y})_{ij}$ we get

$$\sum_{j=0}^N (\mathbf{D}\mathbf{y})_{ij} = \tilde{\mathbf{g}}_{iN}^* - \tilde{\mathbf{g}}_{i0}^* \quad (\text{A.59})$$

All together this gives us

$$\sum_{i=0}^N \sum_{j=0}^N J_{ij} \omega_i \omega_j \frac{\partial \mathbf{u}_{ij}}{\partial t} = - \sum_{j=0}^N \omega_j \tilde{\mathbf{f}}_{Nj}^* + \sum_{j=0}^N \omega_j \tilde{\mathbf{f}}_{0j}^* - \sum_{i=0}^N \omega_i \tilde{\mathbf{g}}_{iN}^* + \sum_{j=0}^N \omega_j \tilde{\mathbf{g}}_{i0}^*. \quad (\text{A.60})$$

Focusing on a single interior interface we get

$$IU_t = - \sum_{j=0}^N \omega_j \tilde{\mathbf{f}}_{Nj}^{*,L} + \sum_{j=0}^N \omega_j \tilde{\mathbf{f}}_{0j}^{*,R}. \quad (\text{A.61})$$

Note, that ω_j does not have a superscript index L or R as we focus on a conforming mesh and thus all SBP operators in each element are the same. Due to the conforming mesh we set $\tilde{\mathbf{f}}_j^* := \tilde{\mathbf{f}}_{Nj}^{*,L} = \tilde{\mathbf{f}}_{0j}^{*,R}$ and arrive at

$$IU_t = - \sum_{j=0}^N \omega_j \tilde{\mathbf{f}}_j^* + \sum_{j=0}^N \omega_j \tilde{\mathbf{f}}_j^* = 0, \quad (\text{A.62})$$

which indicates primary conservation.

Next, we will proof entropy stability. Therefore we contract (A.43) by the entropy variables \mathbf{v}_{ij} and sum over all $i, j = 0, \dots, N$ to approximate the discrete integral of $\frac{\partial \mathcal{S}}{\partial t}$

$$\sum_{i,j=0}^N J_{ij} \omega_i \omega_j \mathbf{v}_{ij}^T \frac{\partial \mathbf{u}_{ij}}{\partial t} + \sum_{j=0}^N \omega_j \sum_{i=0}^N \mathbf{v}_{ij}^T (\mathbf{D}\mathbf{x})_{ij} + \sum_{i=0}^N \omega_i \sum_{j=0}^N \mathbf{v}_{ij}^T (\mathbf{D}\mathbf{y})_{ij} = 0 \quad (\text{A.63})$$

We focus on $\sum_{i=0}^N \mathbf{v}_{ij}^T (\mathbf{D}\mathbf{x})_{ij}$ an get

$$\begin{aligned} \sum_{i=0}^N \mathbf{v}_{ij}^T (\mathbf{D}\mathbf{x})_{ij} &= \sum_{i=0}^N \sum_{m=0}^N \mathbf{Q}_{im} \mathbf{v}_{ij}^T \tilde{\mathbf{f}}^\#(\mathbf{u}_{ij}, \mathbf{u}_{mj}) + \sum_{i=0}^N \sum_{m=0}^N \mathbf{Q}_{im} \mathbf{v}_{ij}^T \tilde{\mathbf{f}}^\#(\mathbf{u}_{mj}, \mathbf{u}_{ij}) \\ &\quad - \sum_{i=0}^N \left(\delta_{iN} [\mathbf{v}^T \tilde{\mathbf{f}} - \mathbf{v}^T \tilde{\mathbf{f}}^*]_{Nj} - \delta_{i0} [\mathbf{v}^T \tilde{\mathbf{f}} - \mathbf{v}^T \tilde{\mathbf{f}}^*]_{0j} \right). \end{aligned} \quad (\text{A.64})$$

A. Appendix

Using the SBP property $\mathbf{Q} = \mathbf{B} - \mathbf{Q}^T$ we obtain

$$\begin{aligned} \sum_{i=0}^N \mathbf{v}_{ij}^T (\mathbf{D}\mathbf{x})_{ij} &= \sum_{i=0}^N \sum_{m=0}^N \mathbf{Q}_{im} \mathbf{v}_{ij}^T \tilde{\mathbf{f}}^\#(\mathbf{u}_{ij}, \mathbf{u}_{mj}) - \sum_{i=0}^N \sum_{m=0}^N \mathbf{Q}_{mi} \mathbf{v}_{ij}^T \tilde{\mathbf{f}}^\#(\mathbf{u}_{mj}, \mathbf{u}_{ij}) \\ &\quad + \sum_{i=0}^N \sum_{m=0}^N \mathbf{B}_{im} \mathbf{v}_{ij}^T \tilde{\mathbf{f}}^\#(\mathbf{u}_{mj}, \mathbf{u}_{ij}) \\ &\quad - [\mathbf{v}^T \tilde{\mathbf{f}} - \mathbf{v}^T \tilde{\mathbf{f}}^*]_{Nj} + [\mathbf{v}^T \tilde{\mathbf{f}} - \mathbf{v}^T \tilde{\mathbf{f}}^*]_{0j}. \end{aligned} \quad (\text{A.65})$$

Rearranging terms, using the structure of \mathbf{B} and due to consistency of $\tilde{\mathbf{f}}^\#$ we get

$$\sum_{i=0}^N \mathbf{v}_{ij}^T (\mathbf{D}\mathbf{x})_{ij} = \sum_{i=0}^N \sum_{m=0}^N \mathbf{Q}_{im} (\mathbf{v}_{ij} - \mathbf{v}_{mj})^T \tilde{\mathbf{f}}^\#(\mathbf{u}_{ij}, \mathbf{u}_{mj}) + \mathbf{v}_{Nj}^T \tilde{\mathbf{f}}_{Nj}^* - \mathbf{v}_{0j}^T \tilde{\mathbf{f}}_{0j}^*. \quad (\text{A.66})$$

Next, we focus on the volume part of (A.66)

$$\begin{aligned} &\sum_{i=0}^N \sum_{m=0}^N \mathbf{Q}_{im} (\mathbf{v}_{ij} - \mathbf{v}_{mj})^T \tilde{\mathbf{f}}^\#(\mathbf{u}_{ij}, \mathbf{u}_{mj}), \\ &= \sum_{i=0}^N \sum_{m=0}^N \mathbf{Q}_{im} (y_\eta)_{ij} (\mathbf{v}_{ij} - \mathbf{v}_{mj})^T \mathbf{f}^\#(\mathbf{u}_{ij}, \mathbf{u}_{mj}) \\ &\quad - \sum_{i=0}^N \sum_{m=0}^N \mathbf{Q}_{im} (x_\eta)_{ij} (\mathbf{v}_{ij} - \mathbf{v}_{mj})^T \mathbf{g}^\#(\mathbf{u}_{ij}, \mathbf{u}_{mj}). \end{aligned} \quad (\text{A.67})$$

Using the Tadmor shuffle condition (A.49) we can rewrite (A.67) with the entropy flux potentials

$$\begin{aligned} &\sum_{i=0}^N \sum_{m=0}^N \mathbf{Q}_{im} (\mathbf{v}_{ij} - \mathbf{v}_{mj})^T \tilde{\mathbf{f}}^\#(\mathbf{u}_{ij}, \mathbf{u}_{mj}), \\ &= \sum_{i=0}^N \sum_{m=0}^N \mathbf{Q}_{im} \left((y_\eta)_{ij} (\Psi_{ij}^f - \Psi_{mj}^f) - (x_\eta)_{ij} (\Psi_{ij}^g - \Psi_{mj}^g) \right), \\ &= \sum_{i=0}^N (y_\eta)_{ij} \left(\Psi_{ij}^f \sum_{m=0}^N \mathbf{Q}_{im} - \sum_{m=0}^N \mathbf{Q}_{im} \Psi_{mj}^f \right) \\ &\quad - \sum_{i=0}^N (x_\eta)_{ij} \left(\Psi_{ij}^g \sum_{m=0}^N \mathbf{Q}_{im} - \sum_{m=0}^N \mathbf{Q}_{im} \Psi_{mj}^g \right). \end{aligned} \quad (\text{A.68})$$

A. Appendix

Assuming that $\sum_{m=0}^N \mathbf{Q}_{im} = 0$ (SBP operator of degree $p > 0$) we get

$$\begin{aligned} & \sum_{i=0}^N \sum_{m=0}^N \mathbf{Q}_{im} (\mathbf{v}_{ij} - \mathbf{v}_{mj})^T \tilde{\mathbf{f}}^\#(\mathbf{u}_{ij}, \mathbf{u}_{mj}), \\ &= \sum_{i=0}^N \left(-(y_\eta)_{ij} \sum_{m=0}^N \mathbf{Q}_{im} \Psi_{mj}^f + (x_\eta)_{ij} \sum_{m=0}^N \mathbf{Q}_{im} \Psi_{mj}^g \right). \end{aligned} \quad (\text{A.69})$$

Again, applying the SBP property we get

$$\begin{aligned} & \sum_{i=0}^N \sum_{m=0}^N \mathbf{Q}_{im} (\mathbf{v}_{ij} - \mathbf{v}_{mj})^T \tilde{\mathbf{f}}^\#(\mathbf{u}_{ij}, \mathbf{u}_{mj}) \\ &= \sum_{i=0}^N \left((y_\eta)_{ij} \sum_{m=0}^N \mathbf{Q}_{mi} \Psi_{mj}^f - (y_\eta)_{ij} \sum_{m=0}^N \mathbf{B}_{im} \Psi_{mj}^f \right) \\ & \quad - \sum_{i=0}^N \left((x_\eta)_{ij} \sum_{m=0}^N \mathbf{Q}_{mi} \Psi_{mj}^g - (x_\eta)_{ij} \sum_{m=0}^N \mathbf{B}_{im} \Psi_{mj}^g \right). \end{aligned} \quad (\text{A.70})$$

By rearranging terms we obtain

$$\begin{aligned} & \sum_{i=0}^N \sum_{m=0}^N \mathbf{Q}_{im} (\mathbf{v}_{ij} - \mathbf{v}_{mj})^T \tilde{\mathbf{f}}^\#(\mathbf{u}_{ij}, \mathbf{u}_{mj}) \\ &= \sum_{m=0}^N \Psi_{mj}^f \sum_{i=0}^N \mathbf{Q}_{mi} (y_\eta)_{ij} - (y_\eta)_{Nj} \Psi_{Nj}^f + (y_\eta)_{0j} \Psi_{0j}^f \\ & \quad - \sum_{m=0}^N \Psi_{mj}^g \sum_{i=0}^N \mathbf{Q}_{mi} (x_\eta)_{ij} + (x_\eta)_{Nj} \Psi_{Nj}^g - (x_\eta)_{0j} \Psi_{0j}^g. \end{aligned} \quad (\text{A.71})$$

By rewriting the potentials in terms of a contravariant potential defined by

$$\tilde{\Psi}^f := y_\eta \Psi^f - x_\eta \Psi^g, \quad (\text{A.72})$$

we get

$$\begin{aligned} & \sum_{i=0}^N \sum_{m=0}^N \mathbf{Q}_{im} (\mathbf{v}_{ij} - \mathbf{v}_{mj})^T \tilde{\mathbf{f}}^\#(\mathbf{u}_{ij}, \mathbf{u}_{mj}) \\ &= \sum_{m=0}^N \Psi_{mj}^f \sum_{i=0}^N \mathbf{Q}_{mi} (y_\eta)_{ij} - \sum_{m=0}^N \Psi_{mj}^g \sum_{i=0}^N \mathbf{Q}_{mi} (x_\eta)_{ij} - \tilde{\Psi}_{Nj}^f + \tilde{\Psi}_{0j}^f. \end{aligned} \quad (\text{A.73})$$

Including this result into (A.66) yields

$$\begin{aligned} \sum_{i=0}^N \mathbf{v}_{ij}^T (\mathbf{D}\mathbf{x})_{ij} &= \sum_{m=0}^N \Psi_{mj}^f \sum_{i=0}^N \mathbf{Q}_{mi} (y_\eta)_{ij} - \sum_{m=0}^N \Psi_{mj}^g \sum_{i=0}^N \mathbf{Q}_{mi} (x_\eta)_{ij} \\ & \quad + \mathbf{v}_{Nj}^T \tilde{\mathbf{f}}_{Nj}^* - \tilde{\Psi}_{Nj}^f - \mathbf{v}_{0j}^T \tilde{\mathbf{f}}_{0j}^* + \tilde{\Psi}_{0j}^f. \end{aligned} \quad (\text{A.74})$$

A. Appendix

Doing the same analysis for $\sum_{j=0}^N \mathbf{v}_{ij}^T (\mathbf{D}\mathbf{y})_{ij}$ leads to

$$\begin{aligned} \sum_{j=0}^N \mathbf{v}_{ij}^T (\mathbf{D}\mathbf{y})_{ij} &= \sum_{m=0}^N \Psi_{im}^g \sum_{j=0}^N \mathbf{Q}_{mj}(x_\xi)_{ij} - \sum_{m=0}^N \Psi_{im}^f \sum_{j=0}^N \mathbf{Q}_{mj}(y_\xi)_{ij} \\ &\quad + \mathbf{v}_{iN}^T \tilde{\mathbf{g}}_{iN}^* - \tilde{\Psi}_{iN}^g - \mathbf{v}_{i0}^T \tilde{\mathbf{g}}_{i0}^* + \tilde{\Psi}_{i0}^g. \end{aligned} \quad (\text{A.75})$$

With (A.74) and (A.75) we can evaluate the total discrete entropy growth (A.63)

$$\begin{aligned} &\sum_{i,j=0}^N J_{ij} \omega_i \omega_j \frac{\partial S_{ij}}{\partial t} \\ &= - \sum_{j=0}^N \omega_j \left(\sum_{m=0}^N \Psi_{mj}^f \sum_{i=0}^N \mathbf{Q}_{mi}(y_\eta)_{ij} - \sum_{m=0}^N \Psi_{mj}^g \sum_{i=0}^N \mathbf{Q}_{mi}(x_\eta)_{ij} \right) \\ &\quad - \sum_{j=0}^N \omega_j \left(\mathbf{v}_{Nj}^T \tilde{\mathbf{f}}_{Nj}^* - \tilde{\Psi}_{Nj}^f - \mathbf{v}_{0j}^T \tilde{\mathbf{f}}_{0j}^* + \tilde{\Psi}_{0j}^f \right) \\ &\quad - \sum_{i=0}^N \omega_i \left(\sum_{m=0}^N \Psi_{im}^g \sum_{j=0}^N \mathbf{Q}_{mj}(x_\xi)_{ij} - \sum_{m=0}^N \Psi_{im}^f \sum_{j=0}^N \mathbf{Q}_{mj}(y_\xi)_{ij} \right) \\ &\quad - \sum_{i=0}^N \omega_i \left(\mathbf{v}_{iN}^T \tilde{\mathbf{g}}_{iN}^* - \tilde{\Psi}_{iN}^g - \mathbf{v}_{i0}^T \tilde{\mathbf{g}}_{i0}^* + \tilde{\Psi}_{i0}^g \right). \end{aligned} \quad (\text{A.76})$$

We rearrange the volume parts and by using $\mathbf{Q} = \mathbf{MD}$ we get

$$\begin{aligned} &\sum_{i,j=0}^N J_{ij} \omega_i \omega_j \frac{\partial S_{ij}}{\partial t} \\ &= \sum_{i,j=0}^N \omega_i \omega_j \Psi_{ij}^f \left(\sum_{m=0}^N \mathbf{D}_{jm}(y_\xi)_{im} - \sum_{m=0}^N \mathbf{D}_{im}(y_\eta)_{mj} \right) \\ &\quad + \sum_{i,j=0}^N \omega_i \omega_j \Psi_{ij}^g \left(\sum_{m=0}^N \mathbf{D}_{im}(x_\eta)_{mj} - \sum_{m=0}^N \mathbf{D}_{jm}(x_\xi)_{im} \right) \\ &\quad - \sum_{j=0}^N \omega_j \left(\mathbf{v}_{Nj}^T \tilde{\mathbf{f}}_{Nj}^* - \tilde{\Psi}_{Nj}^f - \mathbf{v}_{0j}^T \tilde{\mathbf{f}}_{0j}^* + \tilde{\Psi}_{0j}^f \right) \\ &\quad - \sum_{i=0}^N \omega_i \left(\mathbf{v}_{iN}^T \tilde{\mathbf{g}}_{iN}^* - \tilde{\Psi}_{iN}^g - \mathbf{v}_{i0}^T \tilde{\mathbf{g}}_{i0}^* + \tilde{\Psi}_{i0}^g \right). \end{aligned} \quad (\text{A.77})$$

A. Appendix

Due to the metric identities (A.46) and (A.47) we finally obtain

$$\begin{aligned}
& \sum_{i,j=0}^N J_{ij} \omega_i \omega_j \frac{\partial S_{ij}}{\partial t} \\
&= - \sum_{j=0}^N \omega_j \left(\left[\mathbf{v}^T \tilde{\mathbf{f}}^* - \tilde{\Psi}^f \right]_{Nj} - \left[\mathbf{v}^T \tilde{\mathbf{f}}^* - \tilde{\Psi}^f \right]_{0j} \right) \\
& \quad - \sum_{i=0}^N \omega_i \left(\left[\mathbf{v}^T \tilde{\mathbf{g}}^* - \tilde{\Psi}^g \right]_{iN} - \left[\mathbf{v}^T \tilde{\mathbf{g}}^* - \tilde{\Psi}^g \right]_{i0} \right).
\end{aligned} \tag{A.78}$$

We see that all the volume parts vanish and therefore the total growth in entropy reduces to evaluations at the element surface. As we focus a conforming mesh we consider a single interior interface and get

$$IS_t = \sum_{j=0}^N \omega_j \left(\left[\mathbf{v}^{R,T} \tilde{\mathbf{f}}^{*,R} - \tilde{\Psi}^{f,R} \right]_{0j} - \left[\mathbf{v}^{L,T} \tilde{\mathbf{f}}^{*,L} - \tilde{\Psi}^{f,L} \right]_{Nj} \right). \tag{A.79}$$

Note, that as we are on a conforming mesh the weights ω_j on the left and right element are the same and thus, they are not reliant on a superscript L or R . Setting $\tilde{\mathbf{f}}_j^* := \tilde{\mathbf{f}}_{Nj}^{*,L} = \tilde{\mathbf{f}}_{0j}^{*,R}$ and ignoring the first index within $\mathbf{v}^L, \tilde{\Psi}^{f,L}$ and $\mathbf{v}^R, \tilde{\Psi}^{f,R}$ (as these are always constant) we arrive at

$$IS_t = \sum_{j=0}^N \omega_j \left((\mathbf{v}^R - \mathbf{v}^L)^T \tilde{\mathbf{f}}^* - (\tilde{\Psi}^{f,R} - \tilde{\Psi}^{f,L}) \right)_j. \tag{A.80}$$

Next, we rewrite the contravariant fluxes in terms of

$$\tilde{\mathbf{f}}^* = y_\eta \mathbf{f}^* - x_\eta \mathbf{g}^*, \tag{A.81}$$

and (A.80) is equivalent to

$$\begin{aligned}
IS_t &= \sum_{j=0}^N \omega_j \left(y_\eta \left((\mathbf{v}^R - \mathbf{v}^L)^T \mathbf{f}^* - (\Psi^{f,R} - \Psi^{f,L}) \right) \right)_j \\
& \quad - \sum_{j=0}^N \omega_j \left(x_\eta \left((\mathbf{v}^R - \mathbf{v}^L)^T \mathbf{g}^* - (\Psi^{g,R} - \Psi^{g,L}) \right) \right)_j.
\end{aligned} \tag{A.82}$$

To finalize the proof of entropy stability we need to specify the numerical surface flux and set

$$\begin{aligned}
\mathbf{f}_{ES}^*(\mathbf{u}^R, \mathbf{u}^L) &:= \mathbf{f}_{EC}^*(\mathbf{u}^R, \mathbf{u}^L) - \left(\frac{y_\eta}{\sqrt{y_\eta^2 + x_\eta^2}} \right) \frac{\lambda_F}{2} \mathbf{K}(\mathbf{v}^R - \mathbf{v}^L), \\
\mathbf{g}_{ES}^*(\mathbf{u}^R, \mathbf{u}^L) &:= \mathbf{g}_{EC}^*(\mathbf{u}^R, \mathbf{u}^L) - \left(\frac{-x_\eta}{\sqrt{y_\eta^2 + x_\eta^2}} \right) \frac{\lambda_G}{2} \mathbf{K}(\mathbf{v}^R - \mathbf{v}^L),
\end{aligned} \tag{A.83}$$

A. Appendix

where λ_F, λ_G denote the absolute maximum eigenvalue of the Jacobian of \mathcal{F} and \mathcal{G} . Here, \mathbf{K} denotes a symmetric positive definite dissipation matrix with $\mathbf{K}(\mathbf{v}^R - \mathbf{v}^L) \approx (\mathbf{u}^R - \mathbf{u}^L)$, for more information see Chapter 4. Including (A.83) in (A.80) we get

$$\begin{aligned}
 IS_t = & \sum_{j=0}^N \omega_j \left(y_\eta \left((\mathbf{v}^R - \mathbf{v}^L)^T \mathbf{f}_{EC}^* - (\Psi^{f,R} - \Psi^{f,L}) \right) \right)_j \\
 & - \frac{\lambda_F}{2} \sum_{j=0}^N \omega_j \left(\left(\frac{y_\eta^2}{\sqrt{y_\eta^2 + x_\eta^2}} \right) (\mathbf{v}^R - \mathbf{v}^L)^T \mathbf{K}(\mathbf{v}^R - \mathbf{v}^L) \right)_j \\
 & + \sum_{j=0}^N \omega_j \left(x_\eta \left((\mathbf{v}^R - \mathbf{v}^L)^T \mathbf{g}_{EC}^* - (\Psi^{g,R} - \Psi^{g,L}) \right) \right)_j \\
 & - \frac{\lambda_G}{2} \sum_{j=0}^N \omega_j \left(\left(\frac{x_\eta^2}{\sqrt{y_\eta^2 + x_\eta^2}} \right) (\mathbf{v}^R - \mathbf{v}^L)^T \mathbf{K}(\mathbf{v}^R - \mathbf{v}^L) \right)_j.
 \end{aligned} \tag{A.84}$$

As $\mathbf{f}^\#$ and $\mathbf{g}^\#$ satisfy the Tadmor Shuffle condition (A.49) we obtain

$$\begin{aligned}
 IS_t = & - \frac{\lambda_F}{2} \sum_{j=0}^N \omega_j \left(\left(\frac{y_\eta^2}{\sqrt{y_\eta^2 + x_\eta^2}} \right) (\mathbf{v}^R - \mathbf{v}^L)^T \mathbf{K}(\mathbf{v}^R - \mathbf{v}^L) \right)_j \\
 & - \frac{\lambda_G}{2} \sum_{j=0}^N \omega_j \left(\left(\frac{x_\eta^2}{\sqrt{y_\eta^2 + x_\eta^2}} \right) (\mathbf{v}^R - \mathbf{v}^L)^T \mathbf{K}(\mathbf{v}^R - \mathbf{v}^L) \right)_j, \\
 \leq & - \frac{\min\{\lambda_F, \lambda_G\}}{2} \sum_{j=0}^N \omega_j \left(\left(\frac{y_\eta^2 + x_\eta^2}{\sqrt{y_\eta^2 + x_\eta^2}} \right) (\mathbf{v}^R - \mathbf{v}^L)^T \mathbf{K}(\mathbf{v}^R - \mathbf{v}^L) \right)_j.
 \end{aligned} \tag{A.85}$$

As \mathbf{K} is positive semi definite $(\mathbf{v}^R - \mathbf{v}^L)^T \mathbf{K}(\mathbf{v}^R - \mathbf{v}^L) > 0$ and therefore

$$\begin{aligned}
 IS_t \leq & - \frac{\min\{\lambda_F, \lambda_G\}}{2} \underbrace{\sum_{j=0}^N \omega_j (\sqrt{y_\eta^2 + x_\eta^2})_j \left((\mathbf{v}^R - \mathbf{v}^L)^T \mathbf{K}(\mathbf{v}^R - \mathbf{v}^L) \right)_j}_{\geq 0}, \\
 \leq & 0.
 \end{aligned} \tag{A.86}$$

This inequality holds for all interior interfaces and therefore the curvilinear discretization (A.43) is provably primary conservative and entropy stable for an arbitrary conservation law. Here, it is necessary that a numerical flux exists which satisfies the Tadmor Shuffle condition (A.49) and that the metric terms satisfy the metric identities (A.46),(A.47).

Bibliography

- [1] M.J. Aftosmis, M.J. Berger, and J.E. Melton. Adaptive Cartesian mesh generation. In *CRC Handbook of Mesh Generation (Contributed Chapter)*, 1998.
- [2] Mark Ainsworth. Dispersive and dissipative behaviour of high order discontinuous Galerkin finite element methods. *SIAM Journal on Scientific Computing*, 198(1):106–130, 2004.
- [3] Nathan Albin and Joshua Klarmann. An algorithmic exploration of the existence of high-order summation by parts operators with diagonal norm. *Journal of Scientific Computing*, 69(2):633–650, 2016.
- [4] Bradley K. Alpert. Hybrid Gauss-trapezoidal quadrature rules. *SIAM Journal on Scientific Computing*, 5:1551–1584, 1999.
- [5] M.J. Berger, M.J. Aftosmis, and S.R. Allmaras. Progress towards a Cartesian cut-cell method for viscous compressible flow. *50th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition Jan. 9-12, 2012. Nashville, TN*, 2012.
- [6] Philipp Birken, Gregor J. Gassner, Mark Haas, and Claus-Dieter Munz. Preconditioning for modal discontinuous Galerkin methods for unsteady 3D Navier–Stokes equations. *Journal of Computational Physics*, 240:20–35, 2013.
- [7] Marvin Bohm, Andrew R. Winters, Dominik Derigs, Gregor J. Gassner, Stefanie Walch, and Joachim Saur. An entropy stable nodal discontinuous Galerkin method for the resistive MHD equations. Part I: Theory and numerical verification. *Journal of Computational Physics (submitted)*, ArXiv e-prints: arXiv:1802.07341, 2018.
- [8] Pieter D. Boom and David W. Zingg. High-order implicit time-marching methods based on generalized summation-by-parts operators. *SIAM Journal on Scientific Computing*, 37(6):A2682–A2709, 2015.

- [9] Tan Bui-Thanh and Omar Ghattas. Analysis of an hp -nonconforming discontinuous Galerkin spectral element method for wave propagation. *SIAM Journal on Numerical Analysis*, 50(3):1801–1826, 2012.
- [10] Mark H. Carpenter, Travis C. Fisher, Eric J. Nielsen, and Steven H. Frankel. Entropy stable spectral collocation schemes for the Navier–Stokes equations: Discontinuous interfaces. *SIAM Journal on Scientific Computing*, 36(5):B835–B867, 2014.
- [11] Mark H. Carpenter and David Gottlieb. Spectral methods on arbitrary grids. *Journal of Computational Physics*, 129(1):74–86, 1996.
- [12] Mark H. Carpenter, David Gottlieb, and Saul Abarbanel. The stability of numerical boundary treatments for compact high-order finite-difference schemes. *Journal of Computational Physics*, 108(2):272–295, 1993.
- [13] Mark H. Carpenter, David Gottlieb, and Saul Abarbanel. Time-stable boundary conditions for finite-difference schemes solving hyperbolic systems: Methodology and application to high-order compact schemes. *Journal of Computational Physics*, 111(2):220–236, 1994.
- [14] Mark H. Carpenter and Christopher A. Kennedy. Fourth-order $2N$ -storage Runge-Kutta schemes. Technical report, NASA Langley Research Center, 1994.
- [15] Mark H. Carpenter, Jan Nordström, and David Gottlieb. A stable and conservative interface treatment of arbitrary spatial accuracy. *Journal of Computational Physics*, 148(2):341–365, 1999.
- [16] Mark H. Carpenter, Matteo Parsani, Eric J. Nielsen, and Travis C. Fisher. Towards an entropy stable spectral element framework for computational fluid dynamics. In *54th AIAA Aerospace Sciences Meeting, AIAA*, volume 1058, 2016.
- [17] Jesse Chan. On discretely entropy conservative and entropy stable discontinuous Galerkin methods. *Journal of Computational Physics*, 362:346–374, 2018.
- [18] Praveen Chandrashekar. Kinetic energy preserving and entropy stable finite volume schemes for compressible Euler and Navier-Stokes equations. *Communications in Computational Physics*, 14(5):1252–1286, 2013.

- [19] Tianheng Chen and Chi-Wang Shu. Entropy stable wall boundary conditions for the three-dimensional compressible Navier–Stokes equations. *Journal of Computational Physics*, 345:427–461, 2016.
- [20] Tianheng Chen and Chi-Wang Shu. Entropy stable high order discontinuous Galerkin methods with suitable quadrature rules for hyperbolic conservation laws. *Journal of Computational Physics*, 345:427–461, 2017.
- [21] Jared Crean, Jason E. Hicken, David c. Del Rey Fernández, David W. Zingg, and Mark H. Carpenter. Entropy-stable summation-by-parts discretization of the Euler equations on general curved elements. *Journal of Computational Physics*, 356:410–438, 2018.
- [22] Constantine M. Dafermos. *Hyperbolic conservation laws in continuum physics*. Springer, 4 edition, 2016.
- [23] David C. Del Rey Fernández, Pieter D. Boom, and David W. Zingg. A generalized framework for nodal first derivative summation-by-parts operators. *Journal of Computational Physics*, 266(1):214–239, 2014.
- [24] David C. Del Rey Fernández, Jason E. Hicken, and David W. Zingg. Review of summation-by-parts operators with simultaneous approximation terms for the numerical solution of partial differential equations. *Computers & Fluids*, 95(22):171–196, 2014.
- [25] David C. Del Rey Fernández, Jason E. Hicken, and David W. Zingg. Simultaneous approximation terms for multidimensional summation-by-parts operators. *Journal of Scientific Computing*, 75(1):83–110, 2018.
- [26] David C. Del Rey Fernández and David W. Zingg. New diagonal-norm summation-by-parts operators for the first derivative with increased order of accuracy. *AIAA Aviation 2015, Paper 2015-2914*, 2014.
- [27] David C. Del Rey Fernández and David W. Zingg. Generalized summation-by-parts operators for the second derivative with a variable coefficient. *SIAM Journal on Scientific Computing*, 37(6):A2840–A2864, 2015.
- [28] David C. Del Rey Fernández and David W. Zingg. Corner-corrected diagonal-norm summation-by-parts operators for the first derivative with increased order of accuracy. *Journal of Computational Physics*, 330:902–923, 2017.

- [29] Laslo T. Diosady and Scott M. Murman. Higher-order methods for compressible turbulent flows using entropy variables. In *53rd AIAA Aerospace Science Meeting*, page 0294, 2015.
- [30] Parvir Dutt. Stable boundary conditions and difference schemes for Navier-Stokes equations. *SIAM Journal of Numerical Analysis*, 25(2):245–267, 1988.
- [31] Laurence C. Evans. *Partial Differential Equations*. American Mathematical Society, 2012.
- [32] Krzysztof Jakub Fidkowski. *A simplex cut-cell adaptive method for high-order discretizations of the compressible Navier-Stokes equations*. PhD thesis, Massachusetts Institute of Technology, 2007.
- [33] Travis C. Fisher and Mark H. Carpenter. High-order entropy stable finite difference schemes for nonlinear conservation laws: Finite domains. *Journal of Computational Physics*, 252(1):518–557, 2013.
- [34] Travis C. Fisher, Mark H. Carpenter, Jan Nordström, and Nail K. Yamaleev. Discretely conservative finite-difference formulations for nonlinear conservation laws in split form: Theory and boundary conditions. *Journal of Computational Physics*, 234(1):353–375, 2013.
- [35] Ulrik S. Fjordholm. *High-order accurate entropy stable numerical schemes for hyperbolic conservation laws*. PhD thesis, ETH Zürich, 2013.
- [36] Ulrik S. Fjordholm, S. Mishra, and E. Tadmor. Well-balanced and energy stable schemes for the shallow water equations with discontinuous topography. *Journal of Computational Physics*, 230(14):5587–5609, 2011.
- [37] Ulrik S. Fjordholm, Siddhartha Mishra, and Eitan Tadmor. Arbitrarily high-order accurate entropy stable essentially nonoscillatory schemes for systems of conservation laws. *SIAM Journal on Numerical Analysis*, 50(2):544–573, 2012.
- [38] Lucas Friedrich, David C. Del Rey Fernández, Andrew R. Winters, Gregor J. Gassner, Matteo Parsani, and Mark H. Carpenter. An entropy stable h/p non-conforming discontinuous Galerkin method with the summation-by-parts property. *Journal on Scientific Computing*, 2018.
- [39] Lucas Friedrich, David C. Del Rey Fernández, Andrew R. Winters, Gregor J. Gassner, David W. Zingg, and Jason Hicken. Conservative and stable degree preserving SBP finite difference operators for non-conforming meshes. *Journal on Scientific Computing* (doi:10.1007/s10915-017-0563-z), 2017.

- [40] Lucas Friedrich, Gero Schnücke, Andrew R. Winters, David C. Del Rey Fernández, Gregor J. Gassner, and Mark H. Carpenter. Entropy stable space-time discontinuous Galerkin approximation for non-linear hyperbolic conservation laws with the summation-by-parts property. *Submitted to Journal on Scientific Computing (see arXiv:1808.08218)*, 2018.
- [41] Maximilian Froitzheim. Ein konservatives und stabiles, nicht konformes SBP-spacetime-verfahren. *Master Thesis; Universität zu Köln*, 2018.
- [42] Gregor J. Gassner. A skew-symmetric discontinuous Galerkin spectral element discretization and its relation to SBP-SAT finite difference methods. *SIAM Journal on Scientific Computing*, 35(3):A1233–A1253, 2013.
- [43] Gregor J. Gassner. A kinetic energy preserving nodal discontinuous Galerkin spectral element method. *International Journal for Numerical Methods in Fluids*, 76:28–50, 2014.
- [44] Gregor J. Gassner and David A. Kopriva. A comparison of the dispersion and dissipation errors of Gauss and Gauss-Lobatto discontinuous Galerkin spectral element methods. *SIAM Journal on Scientific Computing*, 33(5):2560–2579, 2011.
- [45] Gregor J Gassner, Andrew R Winters, Florian J. Hindenlang, and David A. Kopriva. The BR1 scheme is stable for the compressible Navier-Stokes equations. *Journal of Scientific Computing (doi:10.1007/s10915-018-0702-1)*, 2017.
- [46] Gregor J. Gassner, Andrew R. Winters, and David A. Kopriva. Split form nodal discontinuous Galerkin schemes with summation-by-parts property for the compressible Euler equations. *Journal of Computational Physics*, 327:39–66, 2016.
- [47] Gregor J. Gassner, Andrew R. Winters, and David A. Kopriva. A well balanced and entropy conservative discontinuous Galerkin spectral element method for the shallow water equations. *Journal of computational physics*, 272:291–308, 2016.
- [48] Christophe Geuzaine and Jean-François Remacle. Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering*, 79:1309 – 1331, 2009.
- [49] Ayoub Gouasmi, Karthik Duraisamy, and Scott Murman. Short note: On entropy stable temporal fluxes. *arXiv:1807.03483*, 2018.

- [50] Bertil Gustafsson, Heinz-Otto Kreiss, and Joseph Oliger. *Time-Dependent Problems and Difference Methods*. Pure and Applied Mathematics. Wiley, second edition, 2013.
- [51] Ami Harten, Peter D. Lax, C. D. Levermore, and W. J. Morokoff. Convex Entropy and hyperbolicity for general Euler equations. *SIAM Journal on Numerical Analysis*, 35(6):2217–2127, 1998.
- [52] Amiram Harten. On the symmetric form of systems of conservation laws with entropy. *Journal of Computational Physics*, 49:151–164, 1983.
- [53] Jan S. Hesthaven. A stable penalty method for the compressible Navier-Stokes equations: II. One-dimensional domain decomposition schemes. *SIAM Journal on Scientific Computing*, 18(3):658–685, 1997.
- [54] Jan S. Hesthaven and Tim Warburton. *Nodal discontinuous Galerkin methods algorithms, analysis, and applications*. Springer, 2008.
- [55] Jason E. Hicken, David C. Del Rey Fernández, and David W. Zingg. Opportunities for efficient high-order methods based on the summation-by-parts property. *22nd AIAA Computational Fluid Dynamics Conference, AIAA Paper 2015-3198, Dallas, Texas, USA, June 2015*, 2015.
- [56] Jason E. Hicken, David C. Del Rey Fernández, and David W. Zingg. Multi-dimensional summation-by-part operators: General theory and application to simplex elements. *SIAM Journal on Scientific Computing*, 38(4):A1935–A1958, 2016.
- [57] Jason E. Hicken, David C. Del Rey Fernández, and David W. Zingg. Simultaneous approximation terms for multidimensional summation-by-parts operators. *46th AIAA Fluid Dynamics Conference, Washington D.C., USA, June 2016 (accepted)*, 2016.
- [58] Jason E. Hicken and David W. Zingg. Summation-by-parts operators and high-order quadrature. *Journal of Computational and Applied Mathematics*, 237(1):111–125, 2013.
- [59] F. Hindenlang, T. Bolemann, and C-D. Munz. Mesh curving techniques for high order discontinuous Galerkin simulations. In *IDIHOM: Industrialization of High-Order Methods-A Top-Down Approach*, pages 133–152. Springer, 2015.

- [60] Florian Hindenlang, Gregor J. Gassner, Christoph Altmann, Andrea Beck, Marc Staudenmaier, and Claus-Dieter Munz. Explicit discontinuous Galerkin methods for unsteady problems. *Computers & Fluids*, 61(0):86–93, 2012.
- [61] D.M. Ingram, D.M. Causon, and C.G. Mingham. Developments in Cartesian cut cell methods. *Mathematics and Computers in Simulation*, 61:561–572, 2003.
- [62] Farzad Ismail and Philip L. Roe. Affordable, entropy-consistent Euler flux functions II: Entropy production at shocks. *Journal of Computational Physics*, 228:5410–5436, 2009.
- [63] Mark W. Johnson. A novel Cartesian CFD cut cell approach. *Computer & Fluids*, 79:105–119, 2013.
- [64] G Karniadakis and S Sherwin. *Spectral/hp element methods for computational fluid dynamics*. Oxford University Press, 2nd edition edition, 2005.
- [65] D. A. Kopriva. Metric identities and the discontinuous spectral element method on curvilinear meshes. *The Journal of Scientific Computing*, 26(3):301–327, March 2006.
- [66] David A. Kopriva. A conservative staggered-grid Chebyshev multidomain method for compressible flows. II. A semi-structured method. *Journal of Computational Physics*, 128(2):475–488, 1996.
- [67] David A. Kopriva. *Implementing Spectral Methods for Partial Differential Equations*. Scientific Computation. Springer, May 2009.
- [68] David A. Kopriva and Gregor J. Gassner. An energy stable discontinuous Galerkin spectral element discretization for variable coefficient advection problems. *SIAM Journal on Scientific Computing*, 4(36):A2076–A2099, 2014.
- [69] David A. Kopriva, Stephen L. Woodruff, and Mohammed Y. Hussaini. Computation of electromagnetic scattering with a non-conforming discontinuous spectral element method. *International Journal for Numerical Methods in Engineering*, 53(1):105–122, 2002.
- [70] Jeremy E. Kozdon and Lucas C. Wilcox. Stable coupling of nonconforming, high-order finite difference methods. *SIAM Journal on Scientific Computing*, 3(38):A923–A952, 2016.

- [71] Heinz-Otto Kreiss and Joseph Oliger. Comparison of accurate methods for the integration of hyperbolic equations. *Tellus*, 24(3):199–215, 1972.
- [72] Heinz-Otto Kreiss and Godela Scherer. Finite element and finite difference methods for hyperbolic partial differential equations. In *Mathematical aspects of finite elements in partial differential equations*, pages 195–212. Academic Press, New York/London, 1974.
- [73] Philippe G. LeFloch and Christian Rohde. High-order schemes, entropy inequalities, and nonclassical shocks. *SIAM Journal on Numerical Analysis*, 37(6):2023–2060, 2000.
- [74] Randall J. LeVeque. *Finite volume methods for hyperbolic problems*, volume 31. Cambridge University Press, 2002.
- [75] Xinghua Liang, Mohamed S. Ebeida, and Yongjie Zhang. Guaranteed-quality all-quadrilateral mesh generation with feature preservation. *Computer Methods in Applied Mechanics and Engineering*, 199:2072–2083, 2010.
- [76] Jianming Liu, Jianxian Qiu, Ou Hu, Ning Zhao, Mikhail Goman, and Xinkai Li. Adaptive Runge–Kutta discontinuous Galerkin method for complex geometry problems on Cartesian grid. *International Journal for Numerical Methods in Fluids*, 73:847–868, 2013.
- [77] Jean Hsiang-Chun Lu, William Roshan Quadros, and Kenji Shimada. Evaluation of user-guided semi-automatic decomposition tool for hexahedral mesh generation. *Journal of Computational Design and Engineering*, 4:330–338, 2017.
- [78] Yong Lu, Rajit Gadh, and Timothy J. Tautges. Volume decomposition and feature recognition for hexahedral mesh generation. In *8th International Meshing Roundtable, SAND99-2288, Sandia National Laboratories*, pages 269–280, 1999.
- [79] Tomas Lundquist and Jan Nordström. The SBP-SAT technique for initial value problems. *Journal of Computational Physics*, 270:86–104, 2014.
- [80] Tomas Lundquist and Jan Nordström. On the suboptimal accuracy of summation-by-parts schemes with non-conforming block interfaces. Technical report, Linköpings Universitet, 2015.
- [81] Ken Mattsson. Summation by parts operators for finite difference approximations of second-derivatives with variable coefficients. *Journal of Scientific Computing*, 51(3):650–682, 2012.

- [82] Ken Mattsson, Martin Almquist, and Mark H. Carpenter. Optimal diagonal-norm SBP operators. *Journal of Computational Physics*, 264(1):91–111, 2014.
- [83] Ken Mattsson and Mark H. Carpenter. Stable and accurate interpolation operators for high-order multiblock finite difference methods. *SIAM Journal on Scientific Computing*, 32(4):2298–2320, 2010.
- [84] Ken Mattsson and Jan Nordström. Summation by parts operators for finite difference approximations of second derivatives. *Journal of Computational Physics*, 199:503–540, 2004.
- [85] Marshal L. Merriam. An entropy-based approach to nonlinear stability. Technical report, Ames Research Center, California, 1989.
- [86] Michael S. Mock. Systems of conservation laws of mixed type. *Journal of Differential Equations*, 37(1):70–88, 1980.
- [87] James M. Modisette. *An automated reliable method for two-dimensional Reynolds-averaged Navier-Stokes simulations*. PhD thesis, Massachusetts Institute of Technology, 2011.
- [88] Rodrigo C. Moura, Gianmarco Mengaldo, Joaquim Peiro, and Spencer J. Sherwin. An LES setting for DG-based implicit LES with insights on dissipation and robustness. In *Spectral and High Order Methods for Partial Differential Equations ICOSAHOM 2016*, pages 161–173. Springer, 2017.
- [89] Jan Nordström and Mark H. Carpenter. Boundary and interface conditions for high-order finite-difference methods applied to the Euler and Navier-Stokes equations. *Journal of Computational Physics*, 148(2):621–645, 1999.
- [90] Jan Nordström and Mark H. Carpenter. High-order finite-difference methods, multidimensional linear problems, and curvilinear coordinates. *Journal of Computational Physics*, 173(1):149–174, 2001.
- [91] Jan Nordström and Tomas Lundquist. Summation-by-parts in time. *Journal of Computational Physics*, 251:487–499, 2013.
- [92] Jan Nordström and Magnus Svärd. Well-posed boundary conditions for the Navier-Stokes equations. *SIAM Journal on Numerical Analysis*, 43(3):1231–1255, 2005.
- [93] Pelle Olsson. Summation by parts, projections, and stability. I. *Mathematics of Computation*, 64(211):1035–1065, 1995.

- [94] Pelle Olsson. Summation by parts, projections, and stability. II. *Mathematics of Computation*, 64(212):1473–1493, 1995.
- [95] M. Parsani, Mark H. Carpenter, and Eric J. Nielsen. Entropy stable wall boundary conditions for the three-dimensional compressible Navier–Stokes equations. *Journal of Computational Physics*, 292:88–113, 2015.
- [96] Matteo Parsani, Mark H. Carpenter, Travis C. Fisher, and Eric J. Nielsen. Entropy stable staggered grid discontinuous spectral collocation methods of any order for the compressible Navier–Stokes equations. *SIAM Journal on Scientific Computing*, 38(5):A3129–A3162, 2016.
- [97] Matteo Parsani, Mark H. Carpenter, and Eric J. Nielsen. Entropy stable discontinuous interfaces coupling for the three-dimensional compressible Navier–Stokes equations. *Journal of Computational Physics*, 290(C):132–138, 2015.
- [98] Maharavo Randrianarivony. Quadrilateral decomposition by two-ear property resulting in cad segmentation. *Word Academy of Science, Engineering and Technology*, 44, 2008.
- [99] Hendrik Ranocha. *Generalised summation-by-parts operators and entropy stability of numerical methods for hyperbolic balance laws*. PhD thesis, TU Braunschweig, 2018.
- [100] Hendrik Ranocha, Philipp Öffner, and Thomas Sonar. Summation-by-parts operators for correction procedure via reconstruction. *Journal of Scientific Computing*, 311:299–328, 2016.
- [101] Hendrik Ranocha, Philipp Öffner, and Thomas Sonar. Extended skew-symmetric form for summation-by-parts operators and varying jacobians. *Journal of Scientific Computing*, 342:13–28, 2017.
- [102] Deep Ray and Praveen Chandrashekar. An entropy stable finite volume scheme for the two dimensional Navier–Stokes equations on triangular grids. *Applied Mathematics and Computation*, 314:257–286, 2017.
- [103] W.H. Reed and T.R. Hill. Triangular mesh methods for the neutron transport equation. *Proceedings of the American Nuclear Society*, 1973.
- [104] Robert Schneiders. Algorithms for quadrilateral and hexahedral mesh generation, 2000.
- [105] Jason F. Shepherd and Chris R. Johnson. Hexahedral mesh generation constraints. *Engineering with Computers*, 24:195–213, 2008.

- [106] Jonathan Richard Shewchuk. Triangle: Engineering a 2D quality mesh generator and delaunay triangulator. In Ming C. Lin and Dinesh Manocha, editors, *Applied Computational Geometry: Towards Geometric Engineering*, volume 1148 of *Lecture Notes in Computer Science*, pages 203–222. Springer-Verlag, May 1996. From the First ACM Workshop on Applied Computational Geometry.
- [107] Björn Sjögreen, Helen C. Yee, and Dmitry Kotov. Skew-symmetric splitting and stability of high order central schemes. In *Journal of Physics: Conference Series*, volume 837, page 012019, 2017.
- [108] C. Stimpson, C. D. Ernst, P. Knupp, P. P. Pébay, and D. C. Thompson. *The Verdict library reference manual*. Sandia National Laboratories, 2007.
- [109] Bo Strand. Summation by parts for finite difference approximations for d/dx . *Journal of Computational Physics*, 110(1):47–67, 1994.
- [110] Huafei Sun and David L. Darmofal. An adaptive simplex cut-cell method for high-order discontinuous Galerkin discretizations of elliptic interface problems and conjugate heat transfer problems. *Journal of Computational Physics*, 278:445–468, 2014.
- [111] Magnus Svärd and Jan Nordström. Review of summation-by-parts schemes for initial-boundary-value-problems. *Journal of Computational Physics*, 268(1):17–38, 2014.
- [112] Eitan Tadmor. Skew-selfadjoint form for systems of conservation laws. *Journal of Mathematical Analysis and Applications*, 103(2):428–442, 1984.
- [113] Eitan Tadmor. Entropy functions for symmetric systems of conservation laws. *Journal of Mathematical Analysis and Applications*, 122(2):355–359, 1987.
- [114] Eitan Tadmor. The numerical viscosity of entropy stable schemes for systems of conservation laws. *Mathematics of Computation*, 49(179):91–103, 1987.
- [115] Eitan Tadmor. Entropy stability theory for difference approximations of nonlinear conservation laws and related time-dependent problems. *Acta Numerica*, 12:451–512, 5 2003.
- [116] Elewterio F. Toro. *Riemann solvers and numerical methods for fluid dynamics*. Scientific Computation. Springer, 1997.

- [117] P.G. Tucker and Z. Pan. A Cartesian cut cell method for incompressible viscous flow. *Applied Mathematical Modelling*, 24:591–606, 2008.
- [118] Niklas Wintermeyer, Andrew R. Winters, Gregor J. Gassner, and David A. Kopriva. An entropy stable nodal discontinuous Galerkin method for the two dimensional shallow water equations on unstructured curvilinear meshes with discontinuous bathymetry. *Journal of Computational Physics*, 340:200–242, 2017.
- [119] Andrew R. Winters. *Discontinuous Galerkin spectral element approximations for the reflection and transmission of waves from moving material interface*. PhD thesis, Florida State University, 2014.
- [120] Andrew R. Winters, Dominik Derigs, Gregor J. Gassner, and Stefanie Walch. A uniquely defined entropy stable matrix dissipation operator for high mach number ideal mhd and euler simulations. *Journal of Computational Physics*, 332:274–289, 2017.
- [121] Andrew R. Winters and Gregor J. Gassner. An entropy stable finite volume scheme for the equations of shallow water magnetohydrodynamics. *Journal on Scientific Computing*, pages 1–26, 2015.
- [122] Andrew R. Winters and Gregor J. Gassner. Affordable, entropy conserving and entropy stable flux functions for the ideal MHD equations. *Journal of Computational Physics*, 304:72–108, 2016.
- [123] Andrew R. Winters, Rodrigo C. Moura, Gianmarco Mengaldo, Gregor J. Gassner, Stefanie Walch, Joachim Peiro, and Spencer J. Sherwin. A comparative study on polynomial dealiasing and split form discontinuous Galerkin schemes for under-resolved turbulence computations. *Journal of Computational Physics*, 372:1–21, 2018.
- [124] Bing Yuan and David A. Kopriva. A high order quad mesh generator for spectral methods. *26th International Meshing Roundtable*, 2017.
- [125] Xueqian Zhou, Serge Sutulo, and C. Guedes Soares. A paving algorithm for dynamic generation of quadrilateral meshes for online numerical simulations of ship manoeuvring in shallow water. *Ocean Engineering*, 122:10–21, 2016.

Erklärung

Ich versichere, dass ich die von mir vorgelegte Dissertation selbständig angefertigt, die benutzten Quellen und Hilfsmittel vollständig angegeben und die Stellen der Arbeit - einschließlich Tabellen, Karten und Abbildungen -, die anderen Werken im Wortlaut oder dem Sinn nach entnommen sind, in jedem Einzelfall als Entlehnung kenntlich gemacht habe; dass diese Dissertation noch keiner anderen Fakultät oder Universität zur Prüfung vorgelegen hat; dass sie - abgesehen von unten angegebenen Teilpublikationen - noch nicht veröffentlicht worden ist, sowie, dass ich eine solche Veröffentlichung vor Abschluss des Promotionsverfahrens nicht vornehmen werde. Die Bestimmungen der Promotionsordnung sind mir bekannt. Die von mir vorgelegte Dissertation ist von Prof. Dr. Gregor Gassner betreut worden.

Bonn, den 22.10.2018

(LUCAS FRIEDRICH)

Curriculum Vitae

Persönliche Daten:

Name	Lucas Friedrich
Geburtstag	23.10.1988
Geburtsort	Bonn, Deutschland
Nationalität	Deutsch/Britisch

Bildungsweg:

2015-2018	Promotionsstudium in Mathematik Universität zu Köln
2012-2014	Masterstudium in Wirtschaftsmathematik Universität zu Köln
2008-2012	Bachelorstudium in Wirtschaftsmathematik Universität zu Köln
1999-2008	Gymnasiales Abitur Helmholtz-Gymnasium, Bonn

Bonn, den 22.10.2018

(LUCAS FRIEDRICH)

Publikationsliste

Verzeichnis veröffentlichter Wissenschaftlicher Arbeiten

1. Entropy Stable Space-Time Discontinuous Galerkin Schemes with Summation-by-Parts Property for Hyperbolic Conservation Laws (submittiert an Journal of Scientific Computing) [40]
2. An Entropy Stable h/p Non-Conforming Discontinuous Galerkin Method with the Summation-by-Parts Property (publiziert im Journal of Scientific Computing) [38]
3. Conservative and Stable Degree Preserving SBP Finite Difference Operators for Non-Conforming Meshes (publiziert im Journal of Scientific Computing) [39]

Bonn, den 22.10.2018

(LUCAS FRIEDRICH)