

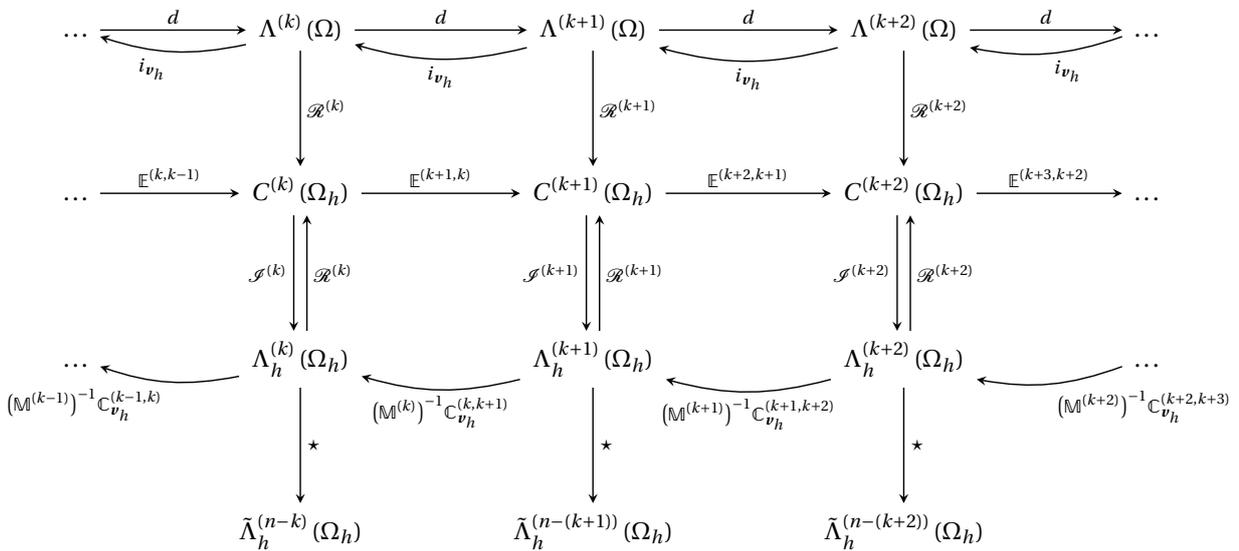
Mimetic isogeometric modeling & discretization of compressible Euler flows

M.Sc. thesis

by

Sebastiaan Pouwel Carolus van Schie

to obtain the degrees of
 Master of Science in Aerospace Engineering and
 Master of Science in Applied Mathematics
 at Delft University of Technology,
 to be defended publicly on Monday September 6, 2021 at 14:00.



Thesis committees:

Aerospace Engineering:

Dr. ir. M.I. Gerritsma	TU Delft, supervisor
Dr. S.J. Hulshoff	TU Delft
Dr. ir. D. Toshniwal	TU Delft
Dr. M. Möller	TU Delft

Applied Mathematics:

Dr. M. Möller	TU Delft, supervisor
Dr. ir. D. Toshniwal	TU Delft, supervisor
Prof. dr. H. M. Schuttelaars	TU Delft
Dr. ir. M.I. Gerritsma	TU Delft

An electronic version of this thesis is available at <http://repository.tudelft.nl/>.

Summary

This thesis poses a new geometric formulation for compressible Euler flows. A partial decomposition of this model into Roe variables is applied; this turns mass density, momentum and kinetic energy into product quantities of the Roe variables. Lie derivative advection operators of weak forms constructed with this decomposed model naturally follow to be self-adjoint, which results in skew-symmetric discrete advection operators in any number of dimensions. Under certain conditions these conserve products of the Roe variables, leading to a discrete model formulation with advection operators that simultaneously conserve mass, momentum, kinetic energy, internal energy and total energy in compressible Euler flows. While this idea is not new the novelty of this work lies in its extension to mimetic finite element methods and its application to discontinuous compressible Euler flows.

The regular geometric Euler model and its Roe variable decomposition have been discretized through mimetic isogeometric analysis. At the core of mimetic discretization methods lies the idea of retaining the De Rham sequence of differential form spaces when projecting these to finite-dimensional approximations and when constructing discrete operators. B-spline differential form spaces have been defined such that the exterior derivative maps in a topologically exact and metric-free way, while the interior product has been discretized in a weak way in order to retain its map between appropriate spaces in the De Rham sequence. Only primal grids are used; the Hodge \star operator is discretized through the definition of an L^2 inner product to resolve weak forms. Cartan's homotopy formula allows for a consistent discretization of the Lie derivative through compositions of the interior product and exterior derivative.

Several tests were carried out to determine the efficacy and behavior of this regular geometric Euler model, its Roe variable decomposition and the resulting discrete advection operators. Testing one-dimensional linear advection and Burgers' equation on periodic domains shows that discretizations of the self-adjoint advection operators are consistent with conservative formulations. Sod's shock tube is used as one-dimensional discontinuous compressible flow test. Both the regular Euler model and its Roe variable decomposition display strong oscillatory tendencies, yet solution convergence is obtained without any issues. While application of a simple moving average-filter removes the worst oscillations more sophisticated methods are necessary for obtaining solutions that are free of unphysical oscillations. The Roe variable decomposition negatively affects the accuracy of shock speed predictions. The presence of strong oscillations likely affects the numerical conservation errors of both methods. Compared to finite volume package Clawpack and the nodal Discontinuous Galerkin (DG) method of Hesthaven & Warburton both models have less numerical diffusion on coarse meshes with the regular model outperforming the Roe variable decomposition. Convergence of momentum conservation error is slow and the errors are large compared to the reference methods.

For two-dimensional periodic vortices both the regular geometric Euler model and its Roe variable decomposition outperform both reference methods for stationary and moving vortices. Clawpack displays diffusive behavior, resulting in large L^2 errors and high amounts of numerical diffusion. While the L^2 errors of the DG method are comparable to those of the two models developed in this work for all basis function orders considered, the resulting DG discretization requires significantly more degrees of freedom to attain this. To obtain similar levels of numerical diffusion the DG method needs up to ten times as many degrees of freedom as the methods presented in the current research.

Preface

This thesis report concludes over two years of work, during which I've looked at many different things and a lot has happened. Starting in the spring of 2019 I knew right away that I wanted to work on mimetic iso-geometric methods. I got a lot of freedom from my supervisors to look at different topics within that field. Among other things I looked at (semi)-Lagrangian conservative discretization methods for advection, different discretization approaches for the Lie derivative, time-staggered method for resolving nonlinear systems and ways of limiting dispersion in high-gradient solutions. I ended up being able to leverage a bit of most of these topics while working on my eventual thesis topic: Using Roe variables to construct advection operators that simultaneously conserve various physical quantities.

All in all I've worked on this particular topic for nearly 18 months; at the start of that period Covid-19 hit Europe, which forced me to work from home mostly independent from other thesis students. This made me realize how much I value the face-to-face contact, cooperation and exchange of ideas with other students and researchers. I'm looking forward to experiencing that once again when I start my PhD in Mechanical and Aerospace Engineering at University of California San Diego in a couple of weeks.

I'd like to take this opportunity to extend my gratitude and thanks to a number of people, for various reasons. For their guidance and supervision along the way I'd like to thank dr. ir. Marc Gerritsma, dr. Matthias Möller and dr. ir. Deepesh Toshniwal. Prior to Covid-19 your door was always open to me when I had questions or was stuck on something. Secondly I'd like to thank dr. ir. Marco ten Eikelder and dr. Wybe Rozema for their time in answering any questions I had about their work; I might not have used all of it in this thesis, but I really enjoyed learning about it! Third, I'd like to thank my fellow former basement dwellers at aerodynamics; since the start of Covid-19 lockdowns the basement has been closed, the time since then has made me realize how valuable and enjoyable it can be to have such a working environment with fellow nerds! Fourth, I can't not thank my parents and Floris, for supporting me through all of this despite having no idea of what I was actually looking at. It turns out that accurately explaining computational mathematics to laypeople can be kind of difficult. Lastly I would like to thank all my friends, both ones I already knew at the start of this entire ordeal and the ones I made along the way, who helped keep me (somewhat) sane by providing outlets for my puns and odd jokes. Having a beer together really does have therapeutic benefits!

*Sebastiaan Pouwel Carolus van Schie
Delft, September 2021*

Contents

List of Figures	xi
Introduction	2
1 Differential Geometry	3
1.1 Objects on manifolds	3
1.1.1 Vectors & covectors	3
1.1.2 Differential forms	4
1.1.3 Vector- & covector-valued differential forms	5
1.2 Operators	7
1.2.1 Exterior derivative	7
1.2.2 Interior product	8
1.2.3 Lie derivative.	9
1.2.4 Musical isomorphisms.	10
1.2.5 Hodge \star operator	11
1.3 De Rham complex and relations between spaces	12
2 Structure-preserving isogeometric analysis	13
2.1 Discrete objects	13
2.1.1 Cell complexes.	13
2.1.2 Chains and cochains.	14
2.1.3 Linking continuous and discrete variables	16
2.2 Splines and isogeometric analysis.	18
2.2.1 B-splines.	19
2.2.2 M-splines	20
2.2.3 Non-Uniform Rational B-Splines.	20
2.2.4 Spline multiplication.	20
2.2.5 Open and periodic B-spline bases	22
2.2.6 Mimetic B-splines	23
2.3 Discrete operators	25
2.3.1 Weak forms and mass matrices	25
2.3.2 Exterior derivative	26
2.3.3 Hodge \star operator	28
2.3.4 Interior product	28
2.3.5 Lie derivative.	30
2.4 Combining discrete operators into a discrete complex	31
3 Momentum, kinetic energy and the compressible Euler equations	33
3.1 Variational geometric Euler equations	33
3.2 Non-variational geometric Euler equations	35
3.3 Momentum- and kinetic energy-conserving methods.	36
3.4 Roe variable decompositions	38
3.5 Obtaining a partial Roe variable formulation	39
3.6 Aspects of discretization	43
3.6.1 Discrete reconstruction of physical variables	43
3.6.2 Time stepping	44
3.6.3 Linearization and nonlinear convergence	45

4	Reference models	47
4.1	Clawpack	47
4.1.1	Finite volume discretization	48
4.1.2	Time stepping	49
4.1.3	Flux functions and limiting	49
4.1.4	Boundary condition application	50
4.2	Nodal Discontinuous Galerkin	50
4.2.1	Discontinuous Galerkin discretization	50
4.2.2	Flux functions	51
4.2.3	Time stepping	52
4.2.4	Filtering and limiting	53
4.2.5	Boundary condition application	53
4.3	Grids and degrees of freedom	54
5	One-dimensional linear advection	55
5.1	Continuous models	55
5.2	Discrete models.	56
5.3	Sine wave test case	57
5.3.1	Discretization error	57
5.3.2	Numerical conservation	57
6	Burgers' equation	61
6.1	Continuous models	61
6.2	Discrete models.	62
6.3	Sine wave test case	63
6.3.1	Discretization error	64
6.3.2	Numerical conservation	64
7	One-dimensional Euler equations	69
7.1	Continuous models	70
7.2	Discrete models.	71
7.2.1	Regular model	71
7.2.2	Skew-symmetric Roe variable model.	73
7.2.3	Application of boundary conditions	75
7.2.4	Kinetic energy advection modeling	75
7.3	Application to Sod's shock tube	77
7.3.1	Kinetic energy modeling comparison	78
7.3.2	Regular & skew-symmetric model comparison.	80
7.3.3	Comparison with established methods	81
8	Two-dimensional Euler equations	91
8.1	Continuous models	91
8.2	Discrete models.	93
8.2.1	Regular model	94
8.2.2	Skew-symmetric Roe variable model.	96
8.2.3	Kinetic energy modeling	98
8.3	Two-dimensional isentropic vortex	98
8.4	Static isentropic vortex test	100
8.4.1	Kinetic energy modeling comparison	100
8.4.2	Regular & skew-symmetric model comparison.	101
8.4.3	Comparison with established methods	105
8.5	Moving isentropic vortex test	109
8.5.1	Kinetic energy modeling comparison	109
8.5.2	Regular & skew-symmetric model comparison.	112
8.5.3	Comparison with established methods	115

Conclusions	126
Recommendations	127
Bibliography	129

List of Figures

1	Combined computing performance of Top 500 supercomputers on Earth over time; shown as well are the performance of #1 and #500 on this list. Image courtesy of www.top500.org	1
1.1	Orientations of various spatial elements in \mathbb{R}^3 , taken from [55, p. 6]	12
2.1	Example positive global inner orientation directions of 2-, 1- and 0-cells	14
2.2	Example positive global inner orientation directions of 2-, 1- and 0-cells	16
2.3	B-spline basis functions $N_{1,p}$ for different p on knot vector $\Xi = [0, 1, 2, 3, 4, 5]$	19
2.4	B-spline basis functions $\{N_{i,2}\}$ on open knot vector $\Xi = [0, 0, 0, 1, 2, 3, 3, 3]$	22
2.5	Periodic B-spline basis functions $\{N_{i,2}\}$ on periodic knot vector $\Xi = [-\frac{2}{5}, -\frac{1}{5}, 0, \frac{1}{5}, \frac{2}{5}, \frac{3}{5}, \frac{4}{5}, 1]$; the resulting basis is periodic on interval $[0, 1]$	23
2.6	Example cochains with inner orientations indicated	27
4.1	Venn diagram showing the relation between the various types of numerical discretization approaches covered in this research work; Discontinuous Galerkin methods combine aspects and ideas of FV- and FE methods	47
4.2	Various meshes used for the nodal Discontinuous Galerkin method to simulate the two-dimensional Euler equations in chapter 8	54
5.1	Numerical solutions at $t = 10$ (obtained with $\Delta t = 10^{-2}$, $n = 25$ basis functions of degree $p = 2$) for both methods discussed in this chapter	58
5.2	Convergence of the relative L^2 error of the conservative and skew-symmetric models at $t = 1$ for $\Delta t = 10^{-4}$, varying amounts of degrees of freedom n and basis function degrees p	58
5.3	Convergence of the relative L^2 error of both considered numerical models at $t = 1$ for $n = 50$ and varying time step sizes Δt and basis function degrees p	59
5.4	Maximum momentum and kinetic energy conservation errors of the conservative and skew-symmetric models for various basis function degrees p , $\Delta t = 10^{-4}$ was used	60
6.1	Solutions obtained with conservative and skew-symmetric models given in (6.9) and (6.10) respectively compared to the exact solution, at various times for $\Delta t = 10^{-4}$, $n = 25$ basis functions of degree $p = 2$ with nonlinear convergence tolerance $\epsilon = 10^{-14}$	63
6.2	Convergence of the relative L^2 errors of the conservative (C) and skew-symmetric (SS) models at various times; $\Delta t = 10^{-4}$ and different basis function degrees p were used, together with $\epsilon = 10^{-6}$ and $\epsilon = 10^{-14}$	66
6.3	Convergence of the relative L^2 error of the conservative and skew-symmetric models at various times; $n = 50$ and different basis function degrees p were used, together with $\epsilon = 10^{-6}$ and $\epsilon = 10^{-14}$	67
6.4	Maximum momentum and kinetic energy conservation errors of the conservative (C) and skew-symmetric (SS) models for various basis function degrees p ; $\Delta t = 10^{-4}$ was used, together with $\epsilon = 10^{-6}$ and $\epsilon = 10^{-14}$	68
7.1	Initial condition of Sod's shock tube problem as given in [50], converted to primary variables	78
7.2	Mass density solutions at $t = 0.025$, $t = 0.05$ and $t = 0.1$ of the skew-symmetric model with exact kinetic energy modeling for $\Delta t = 10^{-3}$, $p = 2$ and $\epsilon = 10^{-12}$ compared to the exact solution	79
7.3	Solutions at $t = 0.21$ of primary physical variables of the skew-symmetric model with exact (Ex.) and approximate (Approx.) kinetic energy modeling for $\Delta t = 10^{-3}$, different solution orders p , $\epsilon = 10^{-6}$ and $\epsilon = 10^{-12}$	84
7.4	Maximum conservation errors for several physical variables of the skew-symmetric model with exact (Ex.) and approximate (Approx.) kinetic energy modeling for $\Delta t = 10^{-3}$, different solution orders p , $\epsilon = 10^{-6}$ and $\epsilon = 10^{-12}$	85

7.5	Solutions at $t = 0.21$ of primary physical variables of the regular (R) and skew-symmetric (SS) (with approximate kinetic energy advection) models for $\Delta t = 10^{-3}$, different solution orders p , $\epsilon = 10^{-6}$ and $\epsilon = 10^{-12}$	86
7.6	Maximum conservation errors for several physical variables of the regular (R) and skew-symmetric (SS) (with approximate kinetic energy advection) models for $\Delta t = 10^{-3}$, different solution orders p , $\epsilon = 10^{-6}$ and $\epsilon = 10^{-12}$	87
7.7	Solutions at $t = 0.21$ of primary physical variables of the regular (R) and skew-symmetric (SS) models with $\epsilon = 10^{-6}$, Clawpack and the DG method, all with $\Delta t = 10^{-3}$ and different solution orders p where possible	88
7.8	Solutions at $t = 0.21$ of primary physical variables of the regular (R) and skew-symmetric (SS) models with $\epsilon = 10^{-6}$, Clawpack and the DG method, all with $\Delta t = 10^{-3}$ and different solution orders p where possible	89
7.9	Maximum conservation errors for several physical variables of the regular (R) and skew-symmetric (SS) models with $\epsilon = 10^{-6}$, the DG method and Clawpack (CP), all with $\Delta t = 10^{-3}$ and different solution orders p where possible	90
8.1	Convergence of the relative mass density L^2 errors of the initial condition for different degrees p	100
8.5	Convergence of relative mass density L^2 errors of initial conditions of the regular (R) and skew-symmetric (SS) models for different degrees p	101
8.2	Convergence of the relative L^2 errors at $t = 10$ of the skew-symmetric model with exact (Ex.) and approximate (Approx.) kinetic energy modeling for $\Delta t = 5 \cdot 10^{-2}$, $(u_\infty, v_\infty) = (0, 0)$, different solution degrees p , $\epsilon = 10^{-6}$ and $\epsilon = 10^{-12}$	102
8.3	Maximum kinetic energy non-conservation errors of the skew-symmetric model with exact (Ex.) and approximate (Approx.) kinetic energy modeling for $\Delta t = 5 \cdot 10^{-2}$, $(u_\infty, v_\infty) = (0, 0)$, different solution degrees p , $\epsilon = 10^{-6}$ and $\epsilon = 10^{-12}$	103
8.4	Maximum non-conservation of primary variables of the skew-symmetric model with exact (Ex.) and approximate (Approx.) kinetic energy modeling for $\Delta t = 5 \cdot 10^{-2}$, $(u_\infty, v_\infty) = (0, 0)$, different solution degrees p , $\epsilon = 10^{-6}$ and $\epsilon = 10^{-12}$	104
8.6	Convergence of the relative L^2 errors at $t = 10$ of the regular (R) and skew-symmetric (SS) models for $\Delta t = 5 \cdot 10^{-2}$, $(u_\infty, v_\infty) = (0, 0)$, different solution degrees p , $\epsilon = 10^{-6}$ and $\epsilon = 10^{-12}$	105
8.7	Maximum kinetic energy non-conservation errors of the regular (R) and skew-symmetric (SS) models for $\Delta t = 5 \cdot 10^{-2}$, $(u_\infty, v_\infty) = (0, 0)$, different solution degrees p , $\epsilon = 10^{-6}$ and $\epsilon = 10^{-12}$	106
8.8	Conservation errors of primary variables of the regular (R) and skew-symmetric (SS) models for $\Delta t = 5 \cdot 10^{-2}$, $(u_\infty, v_\infty) = (0, 0)$, different solution degrees p , $\epsilon = 10^{-6}$ and $\epsilon = 10^{-12}$	107
8.9	Convergence of the relative L^2 errors at $t = 10$ of the regular (R) and skew-symmetric (SS) models for $\Delta t = 5 \cdot 10^{-2}$ and $\epsilon = 10^{-6}$, the DG model (DG) for $\Delta t = 1 \cdot 10^{-2}$ and Clawpack (CP) for $\Delta t = 5 \cdot 10^{-3}$; $(u_\infty, v_\infty) = (0, 0)$ and different solution degrees p were used	110
8.10	Maximum relative kinetic energy non-conservation versus the square root of the cell area for the regular (R) and skew-symmetric (SS) models with $\Delta t = 5 \cdot 10^{-2}$ and $\epsilon = 10^{-6}$, the DG model (DG) with $\Delta t = 1 \cdot 10^{-2}$ and Clawpack (CP) with $\Delta t = 5 \cdot 10^{-3}$; $(u_\infty, v_\infty) = (0, 0)$ and different solution degrees p were used	111
8.11	Maximum relative kinetic energy non-conservation versus the amount of degrees of freedom for the regular (R) and skew-symmetric (SS) models with $\Delta t = 5 \cdot 10^{-2}$ and $\epsilon = 10^{-6}$, the DG model (DG) with $\Delta t = 1 \cdot 10^{-2}$ and Clawpack (CP) with $\Delta t = 5 \cdot 10^{-3}$; $(u_\infty, v_\infty) = (0, 0)$ and different solution degrees p were used	112
8.12	Conservation errors of primary variables for the regular (R) and skew-symmetric (SS) models with $\Delta t = 5 \cdot 10^{-2}$ and $\epsilon = 10^{-6}$, the DG model (DG) with $\Delta t = 1 \cdot 10^{-2}$ and Clawpack (CP) with $\Delta t = 5 \cdot 10^{-3}$; $(u_\infty, v_\infty) = (0, 0)$ and different solution degrees p were used	113
8.13	Convergence of the relative L^2 errors at $t = 10$ of the skew-symmetric model with exact (Ex.) and approximate (Approx.) kinetic energy modeling for $\Delta t = 2.5 \cdot 10^{-2}$, $(u_\infty, v_\infty) = (1, 1)$, different solution degrees p , $\epsilon = 10^{-6}$ and $\epsilon = 10^{-12}$	114
8.14	Maximum kinetic energy non-conservation errors of the skew-symmetric model with exact (Ex.) and approximate (Approx.) kinetic energy modeling for $\Delta t = 2.5 \cdot 10^{-2}$, $(u_\infty, v_\infty) = (1, 1)$, different solution degrees p , $\epsilon = 10^{-6}$ and $\epsilon = 10^{-12}$	115

8.15	Maximum non-conservation of primary variables of the skew-symmetric model with exact (Ex.) and approximate (Approx.) kinetic energy modeling for $\Delta t = 2.5 \cdot 10^{-2}$, $(u_\infty, v_\infty) = (1, 1)$, different solution degrees p , $\epsilon = 10^{-6}$ and $\epsilon = 10^{-12}$	116
8.16	Convergence of the relative L^2 errors at $t = 10$ of the regular (R) and skew-symmetric (SS) models for $\Delta t = 2.5 \cdot 10^{-2}$, $(u_\infty, v_\infty) = (1, 1)$, different solution degrees p , $\epsilon = 10^{-6}$ and $\epsilon = 10^{-12}$	117
8.17	Maximum kinetic energy non-conservation errors of the regular (R) and skew-symmetric (SS) models for $\Delta t = 2.5 \cdot 10^{-2}$, $(u_\infty, v_\infty) = (1, 1)$, different solution degrees p , $\epsilon = 10^{-6}$ and $\epsilon = 10^{-12}$	118
8.18	Conservation errors of primary variables of the regular (R) and skew-symmetric (SS) models for $\Delta t = 2.5 \cdot 10^{-2}$, $(u_\infty, v_\infty) = (1, 1)$, different solution degrees p , $\epsilon = 10^{-6}$ and $\epsilon = 10^{-12}$	119
8.19	Convergence of the relative L^2 errors at $t = 10$ of the Roe variable model with $\epsilon = 10^{-6}$, $\Delta t = 2.5 \cdot 10^{-2}$ and the DG model with $\Delta t = 10^{-2}$, for $(u_\infty, v_\infty) = (1, 1)$ and different solution degrees p	121
8.20	Kinetic energy non-conservation errors of different models for $(u_\infty, v_\infty) = (1, 1)$ and different solution degrees p , graphed against the square-root of the cell area; $\Delta t = 2.5 \cdot 10^{-2}$ for the skew-symmetric model, $\Delta t = 1 \cdot 10^{-2}$ for the DG model	122
8.21	Kinetic energy non-conservation errors of different models for $(u_\infty, v_\infty) = (1, 1)$ and different solution degrees p , graphed against the number of degrees of freedom used; $\Delta t = 2.5 \cdot 10^{-2}$ for the skew-symmetric model, $\Delta t = 1 \cdot 10^{-2}$ for the DG model	122
8.22	Conservation errors of primary variables for the regular (R) and skew-symmetric (SS) models with $\Delta t = 2.5 \cdot 10^{-2}$ and $\epsilon = 10^{-6}$, the DG model (DG) with $\Delta t = 1 \cdot 10^{-2}$ and Clawpack (CP) with $\Delta t = 5 \cdot 10^{-3}$; $(u_\infty, v_\infty) = (1, 1)$ and different solution degrees p were used	123

Introduction

According to Moore's law the number of transistors on computer chips doubles every two years. Since he made this prediction in the 1960s the transistor count on a chip of a given size has increased by at least seven orders of magnitude¹. With this increase in transistor count the available computing power has also increased rapidly. One particular metric for this is the computing performance of the largest 500 supercomputers, which is shown in figure 1. As

can be seen the computing performance (expressed in floating point operations per second) of the lowest-ranked supercomputer on the Top 500 list has increased by a factor of one million (six orders of magnitude) since 1995. With this drastic increase in available computing power the analytic capabilities of physics simulations have grown rapidly as well. However, the development of improved computational methods has not kept the same speed. Commercial software for simulating continuum problems such as fluid dynamics, solid mechanics and electromagnetics still relies on the relatively simple and limited computational methods that were introduced in the sixties, seventies and eighties [32, 46]. These methods work

for relatively simply problems but require excessive amounts of computational power or outright fail for more complicated cases; moreover, specific discretization approaches work only for particular types of problems. Part of the academic world has turned to using a more extensive mathematical basis in an attempt to gain more understanding into the reasons why these kinds of issues happen, and to discover new discretization approaches that perform better.

One class of computational methods to come out of this push for a more rigorous use of mathematics are the so-called mimetic or structure-preserving methods. Tonti has pioneered this idea and has given an extensive overview for a variety of field theories in [53]. These methods aim to leverage the geometric structure of physical field theories, together with the various symmetries that such theories contain. This unlike more classical methods, where no such structure is taken into account - or it is used in often ad-hoc solutions that are hard to generalize and often contain certain approximations. At the heart of mimetic or structure-preserving methods lies the idea to construct discrete counterparts to the various geometrical objects that are contained in physical field theories, and to create discrete operators that map between the spaces of these objects in the same way their continuous counterparts do. These continuous operators are often defined independently of any metric notions such as distances, lengths or angles; in other words, they are purely topological. Retaining this topological structure for discrete operators leads to numerical methods that are constructed based on things that are universally true for a given domain - independent of how deformed a mesh is or how large its cells are. The theoretical framework underlying mimetic methods has been shown to encompass a large number of successful numerical methods, ranging from finite difference to finite volume and finite element

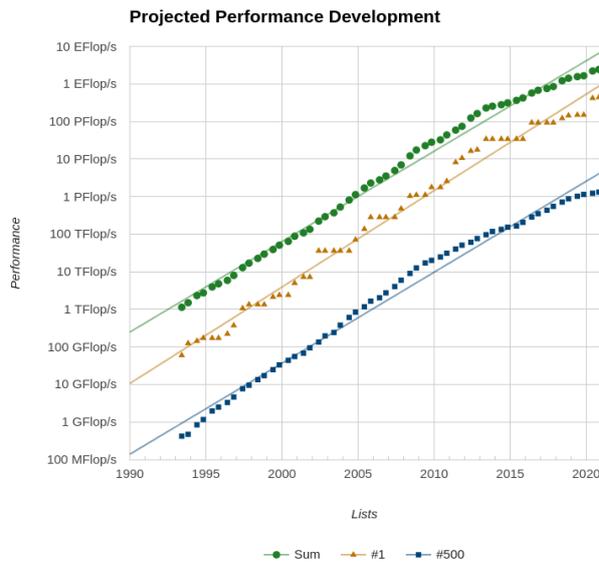


Figure 1: Combined computing performance of Top 500 supercomputers on Earth over time; shown as well are the performance of #1 and #500 on this list. Image courtesy of www.top500.org

¹https://en.wikipedia.org/wiki/Transistor_count

methods. Chapters 1 and 2 cover the relevant underlying theory in a continuous and discretized setting respectively.

The focus of this work lies in fluid flows; more specifically, Euler fluid flows. Mimetic or structure-preserving methods have already been successfully applied to the Euler and Navier-Stokes equations. Examples for incompressible flows include the work of Evans and Hughes on mimetic (isogeometric) finite element methods [14], the mimetic finite difference method of Abbà and Bonaventura [2] and the finite volume-like method of Pavlov et al. [42]. A large focus in these methods is the simultaneous conservation of momentum and kinetic energy; these are two conserved parameters for the Euler equations in any number of dimensions. As is covered in more detail in chapter 3, various numerical methods have been derived that can conserve both of these parameters in incompressible Euler flows. However, they manage to do so by exploiting the divergence-free nature of any solution; it turns out that for numerical solutions that are pointwise divergence-free the various forms of the nonlinear advection operator that is present in the Euler and Navier-Stokes equations are identical. Simultaneous conservation of mass, momentum and kinetic energy is contingent on this pointwise divergence-free character of any numerical solution. No such constraints exist for compressible flow models, and because of this simultaneous conservation of mass, momentum and kinetic energy in compressible Euler flows has not been achieved yet.

The aims of this thesis are threefold. First a geometric model is proposed for the compressible Euler equations. Secondly, a partial decomposition into Roe variables is applied to this model. This decomposition will lead to discrete advection operators that simultaneously conserve mass, momentum and kinetic energy. Lastly both of these models are discretized. This will be done with a mimetic finite element discretization approach that employs ideas from isogeometric analysis [13]; in this approach B-splines are used to discretize numerical solutions and domains. Since Computer-Aided Design (CAD) software uses B-splines to describe geometrical objects, any such object can be represented exactly within an isogeometric discretization. Isogeometric mimetic methods have been applied to a variety of field theories, among which are the works of Buffa et al. [11] and Back & Sonnendrücker [1] on electromagnetics, and the aforementioned work of Evans and Hughes on incompressible Navier-Stokes flows [14] and Janssen [30] on incompressible Euler flows.

The aforementioned discrete advection operators are constructed such that they are skew-symmetric; with an appropriate decomposition of variables this leads to advection operators that simultaneously conserve all the physical variables mentioned above. After having derived and constructed these operators their effectiveness will be demonstrated in a variety of tests. Starting with linear advection in chapter 5, the first nonlinear test is Burgers' equation in chapter 6. Both of these tests serve to show that the skew-symmetric advection operators are consistent with non-skew-symmetric ones. Following this the one-dimensional Euler equations are discretized in chapter 7, where Sod's shock tube is simulated. This presents the first time the conservation of all physical variables can be tested. Lastly the two-dimensional Euler equations are discretized and tested with static and moving isentropic vortices. Comparisons are made with Clawpack (a finite volume code) and a nodal Discontinuous Galerkin method for all simulations of the Euler equations; these are made in terms of L^2 errors, predicted shock locations (for Sod's shock tube) and conservation errors. Exact solutions exist for all test cases.



Differential Geometry

Since the discretization approach that is central to this thesis aims to mimic the structure of continuous Euler flow models, an understanding of this structure needs to be created first. The theory presented in this chapter is not given in a thorough way and only serves to create a topical understanding of some of the relevant concepts and aspects from differential geometry. For a more thorough overview of the relevant theory the reader is referred to the books by Frankel [15], Nakahara [37] and Nash & Sen [38], as well as the dissertation by Kreeft [31]; these works have also served as primary sources for this chapter.

Definition 1.1 (Frankel [15, Sec. 1.2]). *An n -dimensional **manifold** M is a topological space that is homeomorphic to \mathbb{R}^n and is covered by a collection of **coordinate patches** $\{U_i, \phi_{U_i}\}$, $i \in \mathbb{N}$ such that $M = U_1 \cup U_2 \cup \dots$. For each U_i there exists a one-to-one coordinate map $\phi_{U_i} : U_i \rightarrow \mathbb{R}^n$ such that $\phi_{U_i}(U_i) \subset \mathbb{R}^n$ is open. The image $\phi_{U_i}(p)$ for any point $p \in U_i$ defines an n -tuple of coordinates $(x_{U_i}^1(p), \dots, x_{U_i}^n(p))$. For any point p that lies within two overlapping coordinate patches ($p \in U_i \cap U_j$) its coordinates in both patches are related by the differentiable function f_{U_j, U_i} :*

$$x_{U_j}^k(p) = f_{U_j, U_i}^k(x_{U_i}^1, \dots, x_{U_i}^n)$$

For $k = 1, \dots, n$.

In other words, coordinate systems are not unique and can be converted between in a continuously differentiable way. Note that a manifold is a topological space, meaning that there is no natural measure of distance between points, nor any notion of angle or length on said manifold. While the coordinate maps ϕ_{U_i} exist these give out an n -tuple of coordinates in \mathbb{R}^n for each point on a patch independently; for a notion of distance, angle or length to exist a function has to be introduced that compares points to one another. In order to generalize the constructions in the following sections as much as possible, this function (called a metric tensor) will be introduced only when necessary.

1.1. Objects on manifolds

Having defined the necessary spatial structure contained in a manifold, several objects that live on manifolds are introduced.

1.1.1. Vectors & covectors

Definition 1.2 (Frankel [15, Sec. 1.3]). *Let M be a manifold in \mathbb{R}^n . A **tangent** or **(contravariant) vector** at a point $p_0 \in U \subset M$ assigns to coordinate patch (U, ϕ_U) an n -tuple of real numbers $X_U^i = (X_U^1, \dots, X_U^n)$ such that for $p_0 \in U \cap V$:*

$$X_V^i = \sum_{j=1}^n X_U^j \frac{\partial x_V^i}{\partial x_U^j}(p_0)$$

In other words, the expressions of said vector X_U and X_V in the coordinates of patches U and V respectively are related to each other in a continuously differentiable way.

*Similarly, differentially assigning a vector to every point $p \in U$ defines a **vector field** on U .*

It turns out that the set of all possible vectors at a point p define their own space:

Definition 1.3 (Frankel [15, Sec. 1.3a]). *Let M be a manifold in \mathbb{R}^n . The **tangent space** $T_p(M)$ to M at point $p \in M$ is the real vector space consisting of all tangent vectors to M at p . Suppose that (x^1, \dots, x^n) is a local coordinate system on the patch that contains p , then the basis of $T_p(M)$ is formed by the set $\left\{ \frac{\partial}{\partial x^i} \Big|_p \right\}_{i=1}^{i=n}$. Any vector in $T_p(M)$ can be expressed as a linear combination of the elements of this basis. The **tangent bundle** $T(M)$ is defined as the collection of all tangent vectors at all points of M :*

$$T(M) = \bigcup_{p \in M} T_p(M)$$

Note that the tangent space $T_p(M)$ has the same number of dimensions as the manifold it is locally tangent to. Its basis will vary depending on the coordinate patch and point $p \in M$. For example, suppose that manifold M is a hollow sphere in \mathbb{R}^3 ; $T_p(M)$ then consists of a (two-dimensional) plane that is tangent to M in p .

Another space can be defined, based on the tangent space and the manifold M :

Definition 1.4 (Frankel [15, Sec. 2.1]). *Let M be a manifold in \mathbb{R}^n and $T_p(M)$ the tangent space in $p \in M$. The **cotangent space** $T_p^*(M)$ is the space of linear functionals acting on elements of $T_p(M)$. That is, for $\mathbf{v} \in T_p(M)$ there exist **covectors** $\alpha \in T_p^*(M)$ such that $\alpha : T_p(M) \rightarrow \mathbb{R}$. Similarly, differentiably assigning a covector to all points $p \in U \subseteq M$ open defines a **covector field** on U . The **cotangent bundle** $T^*(M)$ is defined as the collection of all covectors at all points of M :*

$$T^*(M) = \bigcup_{p \in M} T_p^*(M)$$

Since cotangent space $T_p^*(M)$ consists of linear functionals (covectors) it follows that $\alpha \in T_p^*(M)$ acts directly on the basis $\left\{ \frac{\partial}{\partial x^i} \Big|_p \right\}_{i=1}^{i=n}$ of $T_p(M)$. The bases of cotangent space $T_p^*(M)$ and tangent space $T_p(M)$ thus have a certain correspondence to them. A basis for $T_p^*(M)$ can be defined that is exactly dual to the basis of the tangent space:

Definition 1.5 (Frankel [15, Sec. 2.1]). *Let M be a manifold in \mathbb{R}^n and $T_p(M)$ the tangent space in $p \in M$. Suppose $T_p(M)$ admits basis $\left\{ \frac{\partial}{\partial x^i} \Big|_p \right\}_{i=1}^{i=n}$. We define the **dual basis** $\left\{ dx_p^i \right\}_{i=1}^{i=n}$ of cotangent space $T_p^*(M)$ such that:*

$$dx_p^i \left(\frac{\partial}{\partial x^j} \Big|_p \right) = \delta_j^i = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}$$

For a vector field $\mathbf{v}(x) = v^1(x) \frac{\partial}{\partial x^1} + v^2(x) \frac{\partial}{\partial x^2} + \dots$ expressed in the primal basis and covector field $\alpha(x) = \alpha_1(x) dx^1 + \alpha_2(x) dx^2 + \dots$ expressed in the dual basis it follows that:

$$\alpha(\mathbf{v}) = \sum_{i,j} \alpha_i dx^i v^j \frac{\partial}{\partial x^j} = \sum_{i,j} \alpha_i v^j \delta_j^i = \sum_i \alpha_i v^i$$

1.1.2. Differential forms

Covector field $\alpha = \alpha_1 dx^1 + \alpha_2 dx^2 + \dots$ consists of a linear combination of the dual basis vectors dx^i , each weighed with α_i respectively. We would like to be able to associate multiple dual basis vectors to each α_i , thereby generalizing the underlying covector structure:

Definition 1.6 (Frankel [15, Sec. 2.5b]). *An (**exterior**) (**differential**) p -**form** is a covariant p -tensor $\alpha^{(p)} \in \otimes^p T^*(M) = \underbrace{T^*(M) \otimes T^*(M) \otimes \dots \otimes T^*(M)}_{p \text{ times}}$ that is antisymmetric:*

$$\alpha^{(p)}(\mathbf{v}_1, \dots, \mathbf{v}_r, \dots, \mathbf{v}_s, \dots, \mathbf{v}_p) = -\alpha^{(p)}(\mathbf{v}_1, \dots, \mathbf{v}_s, \dots, \mathbf{v}_r, \dots, \mathbf{v}_p)$$

The space of p -forms on M is denoted by $\Lambda^{(p)}(M)$.

In other words, swapping any two vectors in the order will change the sign of $\alpha^{(p)}$. Note that if $\mathbf{v}_r = \mathbf{v}_s$ it follows that $\alpha^{(p)} = -\alpha^{(p)}$, and thus $\alpha^{(p)} = 0$. This holds whenever the input vectors are not linearly independent, as this implies at least one of the input vectors can be written as linear combination of the others. Since the dimension of the tangent and cotangent spaces is equal to that of manifold M , the dimension of the manifold limits the maximal degree of the differential forms that can be supported by M .

Not every element of $\otimes^p T^*(M)$ is a differential p -form, since not all elements are antisymmetric. Taking the tensor product of p -form $\alpha^{(p)} \in \otimes^p T^*(M)$ and q -form $\beta^{(q)} \in \otimes^q T^*(M)$ does not automatically lead to a $p+q$ -form $\alpha^{(p)} \otimes \beta^{(q)}$; interchanging a vector of $\alpha^{(p)}$ with a vector of $\beta^{(q)}$ is not guaranteed to change the sign of $\alpha^{(p)} \otimes \beta^{(q)}$. A notable example of this is $\alpha^{(1)} \otimes \beta^{(1)}$; whereas both 1-forms have only one input vector their tensor product is not a 2-form, as interchanging both vectors does not change its sign. To remedy this a different product operator is defined:

Definition 1.7 (Frankel [15, Sec. 2.5b]). *Let M be a manifold in \mathbb{R}^n . The **exterior product** (also known as **wedge product** or **Grassmann product**) is the mapping $\wedge : \Lambda^{(p)}(M) \times \Lambda^{(q)}(M) \rightarrow \Lambda^{(p+q)}(M)$ that maps two lower-degree differential forms to a higher-degree form, such that for $\alpha^{(p)} \in \Lambda^{(p)}(M)$, $\beta^{(q)} \in \Lambda^{(q)}(M)$, $\gamma^{(l)} \in \Lambda^{(l)}(M)$ and $f \in \Lambda^{(0)}(M)$:*

$$\begin{aligned}\alpha^{(p)} \wedge \beta^{(q)} &= (-1)^{pq} \beta^{(q)} \wedge \alpha^{(p)} \\ (\alpha^{(p)} + \beta^{(q)}) \wedge \gamma^{(l)} &= \alpha^{(p)} \wedge \gamma^{(l)} + \beta^{(q)} \wedge \gamma^{(l)} \\ (\alpha^{(p)} \wedge \beta^{(q)}) \wedge \gamma^{(l)} &= \alpha^{(p)} \wedge (\beta^{(q)} \wedge \gamma^{(l)}) \\ f(\alpha^{(p)} \wedge \beta^{(q)}) &= f\alpha^{(p)} \wedge \beta^{(q)} = \alpha^{(p)} \wedge f\beta^{(q)}\end{aligned}$$

The wedge product thus provides a way of combining differential forms of various degrees to obtain higher-degree forms.

Example 1. *Let $\alpha^{(1)}$ and $\beta^{(1)}$ be 1-forms on $M \subset \mathbb{R}^3$, such that:*

$$\begin{aligned}\alpha^{(1)} &= \alpha_1 dx^1 + \alpha_2 dx^2 + \alpha_3 dx^3 \\ \beta^{(1)} &= \beta_2 dx^2 + \beta_3 dx^3\end{aligned}$$

Then:

$$\begin{aligned}\alpha^{(1)} \wedge \beta^{(1)} &= (\alpha_1 dx^1 + \alpha_2 dx^2 + \alpha_3 dx^3) \wedge (\beta_2 dx^2 + \beta_3 dx^3) \\ &= \alpha_1 \beta_2 dx^1 \wedge dx^2 + \alpha_2 \beta_2 \underbrace{dx^2 \wedge dx^2}_{=0} + \alpha_3 \beta_2 \underbrace{dx^3 \wedge dx^2}_{=-dx^2 \wedge dx^3} \\ &\quad + \alpha_1 \beta_3 \underbrace{dx^1 \wedge dx^3}_{=-dx^1 \wedge dx^3} + \alpha_2 \beta_3 dx^2 \wedge dx^3 + \alpha_3 \beta_3 \underbrace{dx^3 \wedge dx^3}_{=0} \\ &= \alpha_1 \beta_2 dx^1 \wedge dx^2 + (\alpha_2 \beta_3 - \alpha_3 \beta_2) dx^2 \wedge dx^3 - \alpha_1 \beta_3 dx^3 \wedge dx^1\end{aligned}$$

Which is thus a 2-form on M .

One of the main advantages of differential forms over vectors is their natural integrability over appropriate oriented regions. Differential 1-forms are naturally integrated along curves, 2-forms over surfaces and 3-forms over volumes without having to make any use of dot products, cross products or similar constructions. The integrals can be computed independently of the (local) coordinate system and without having to make use of so-called 'metric concepts' such as arc lengths or angles. Being able to compute these requires one to introduce additional structure to manifold M : A metric. At the same time integrating a vector field over such geometrical objects presents some difficulties: In order to integrate a vector field along a given curve on M we need to know at every point along this curve how well the curve's tangent vectors align with said vector field, necessitating the use of a dot product. Whereas this is no issue in a continuous setting, it will lead to approximation errors when projecting continuous relations to finite-dimensional discrete relations.

1.1.3. Vector- & covector-valued differential forms

In addition to vectors and covectors & other types of differential k -forms, some quantities are modeled as covector-valued differential form:

Definition 1.8 (Frankel [15, Sec. 16.3a], Kreeft [31, Sec. 11.2.1]). *Let M be a manifold in \mathbb{R}^n . A **covector-valued differential k -form** on M is a map $\alpha^{(1,k)} : \underbrace{T(M) \times T(M) \times \dots \times T(M)}_{k \text{ times}} \rightarrow T^*(M) = \Lambda^{(1)}(M)$. In other words, $\alpha^{(1,k)}$ takes in k vectors to produce a covector (a 1-form). A general formulation is:*

$$\alpha^{(1,k)} = \sum_j dx^j \otimes \sum_i \alpha_{j,i} dx^{i_1} \wedge dx^{i_2} \wedge \dots \wedge dx^{i_k}$$

The space of covector-valued k -forms will be denoted with $\Lambda^{(k)}(M, T^*(M))$.

Similarly to how vectors and covectors are dual to one another, there exists an object dual to covector-valued differential forms as well:

Definition 1.9 (Frankel [15, Sec. 16.3a], Kreeft [31, Sec. 11.2.1]). *Let M be a manifold in \mathbb{R}^n . A **vector-valued differential k -form** on M is a map $\alpha^{1,(k)} : \underbrace{T(M) \times T(M) \times \dots \times T(M)}_{k \text{ times}} \rightarrow T(M)$. In other words, $\alpha^{1,(k)}$ takes in k vectors to produce a vector. A general formulation is:*

$$\alpha^{1,(k)} = \sum_j \frac{\partial}{\partial x^j} \otimes \sum_i \alpha_{j,i} dx^{i_1} \wedge dx^{i_2} \wedge \dots \wedge dx^{i_k}$$

The space of vector-valued k -forms will be denoted with $\Lambda^{(k)}(M, T(M))$.

Superscript notation $(1, k)$ is used for covector-valued differential forms, while $1, (k)$ is used for vector-valued forms. Note that trivially a vector-valued 0-form is a vector field, since:

$$\alpha^{1,(0)} = \sum_j \frac{\partial}{\partial x^j} \otimes \alpha_j = \sum_j \alpha_j \frac{\partial}{\partial x^j}$$

While a covector-valued 0-form is simply a covector field:

$$\alpha^{(1,0)} = \sum_j dx^j \otimes \alpha_j = \sum_j \alpha_j dx^j$$

The duality of vector- and covector-valued differential forms can be used as follows:

Definition 1.10 (Kreeft [31, Sec. 11.2.2]). *The duality pairing of a vector-valued k -form and covector-valued l -form is defined as the mapping $\langle \cdot, \cdot \rangle : \Lambda^{(k)}(M, T^*(M)) \times \Lambda^{(l)}(M, T(M)) \rightarrow \Lambda^{(k+l)}(M)$ such that for $\alpha^{(1,k)} \in \Lambda^{(k)}(M, T^*(M))$ and $\beta^{1,(l)} \in \Lambda^{(l)}(M, T(M))$:*

$$\begin{aligned} \langle \alpha^{(1,k)}, \beta^{1,(l)} \rangle &= \left\langle \sum_j dx^j \otimes \sum_i \alpha_{j,i} dx^{i_1} \wedge dx^{i_2} \wedge \dots \wedge dx^{i_k}, \sum_m \frac{\partial}{\partial x^m} \otimes \sum_n \beta_{m,n} dx^{n_1} \wedge dx^{n_2} \wedge \dots \wedge dx^{n_l} \right\rangle \\ &= \sum_j \sum_m \left\langle dx^j, \frac{\partial}{\partial x^m} \right\rangle \otimes \sum_i \sum_n \alpha_{j,i} \beta_{m,n} dx^{i_1} \wedge dx^{i_2} \wedge \dots \wedge dx^{i_k} dx^{n_1} \wedge dx^{n_2} \wedge \dots \wedge dx^{n_l} \\ &= \sum_j \sum_m \delta_m^j \sum_i \sum_n \alpha_{j,i} \beta_{m,n} dx^{i_1} \wedge dx^{i_2} \wedge \dots \wedge dx^{i_k} dx^{n_1} \wedge dx^{n_2} \wedge \dots \wedge dx^{n_l} \end{aligned}$$

Specifically, linear momentum will be modeled as covector-valued differential form. This reflects the vector-valued structure of momentum in physics: A momentum component can be defined in every spatial direction; conservation of momentum implies that every component is separately conserved. Examples of this are the Euler or Navier-Stokes equations for fluid flows: For an incompressible (and thus divergence-free) flow in \mathbb{R}^n n coupled conservation equations have to be solved, usually one for each momentum component.

Example 2. *Let M be a manifold in \mathbb{R}^2 , with $m^{(1,2)} \in \Lambda^{(2)}(M, T^*(M))$ on basis $\{dx, dy\}$, such that:*

$$m^{(1,2)} = dx \otimes m_x dx \wedge dy + dy \otimes m_y dx \wedge dy$$

Recall that vector fields are simply vector-valued 0-form. The duality pairing of $m^{(1,2)}$ with vector fields $\mathbf{v} = v_x \frac{\partial}{\partial x} + v_y \frac{\partial}{\partial y}$ and $\mathbf{w} = w_x \frac{\partial}{\partial x} + w_y \frac{\partial}{\partial y}$ then is:

$$\begin{aligned}\langle m^{(1,2)}, \mathbf{v} \rangle &= v_x \langle \frac{\partial}{\partial x}, dx \rangle \otimes m_x dx \wedge dy + v_y \langle \frac{\partial}{\partial y}, dy \rangle \otimes m_y dx \wedge dy \\ \langle m^{(1,2)}, \mathbf{w} \rangle &= w_x \langle \frac{\partial}{\partial x}, dx \rangle \otimes m_x dx \wedge dy + w_y \langle \frac{\partial}{\partial y}, dy \rangle \otimes m_y dx \wedge dy\end{aligned}$$

To model conservation of $m^{(1,2)}$ two equations would have to be posed on M . While these are usually taken as conservation of $\langle m^{(1,2)}, \frac{\partial}{\partial x} \rangle$ and $\langle m^{(1,2)}, \frac{\partial}{\partial y} \rangle$ this is not strictly required. As long as vector fields \mathbf{v}, \mathbf{w} are linearly independent in all $p \in M$ (i.e. they span $T_p(M) \forall p \in M$) the momentum components m_x, m_y can be computed from $\langle m^{(1,2)}, \mathbf{v} \rangle$ and $\langle m^{(1,2)}, \mathbf{w} \rangle$.

1.2. Operators

The objects defined in the previous section can be modified and acted upon in various ways. Relevant operators are covered below; these will be used to pose the mathematical models that will be discretized later in this thesis.

1.2.1. Exterior derivative

Within vector calculus the concept of a derivative or differential is often used. Operators such as the gradient, curl and divergence can be constructed using differentials in various arrangements. Within differential geometry these operators are all replaced by a single operation:

Definition 1.11 (Frankel [15, Sec. 2.6a]). *Let M be a manifold in \mathbb{R}^n . The **exterior derivative** $d : \Lambda^{(p)}(M) \rightarrow \Lambda^{(p+1)}(M)$ is the unique operator that maps differential p -forms to differential $p+1$ -forms such that:*

$$\begin{aligned}d(\alpha^{(p)} \wedge \beta^{(q)}) &= d\alpha^{(p)} + d\beta^{(q)} \\ d(\alpha^{(p)} \wedge \beta^{(q)}) &= d\alpha^{(p)} \wedge \beta^{(q)} + (-1)^p \alpha^{(p)} \wedge d\beta^{(q)} \\ d \circ d\alpha^{(p)} &= 0 \\ d\alpha^{(0)} &= \sum_i \frac{\partial \alpha}{\partial x^i} dx^i\end{aligned}$$

Note that for an n -dimensional manifold $d\alpha^{(n)} = 0$, since the resulting $n+1$ -form will contain wedge products with duplicate dual basis functions. As mentioned above, the exterior derivative acts as gradient, curl and divergence operator; this depends on the degree of the differential form it acts upon.

Example 3. *Suppose that M is a manifold in \mathbb{R}^3 with coordinate basis $\{x_1, x_2, x_3\}$, let $\alpha^{(0)} = \alpha$. Then:*

$$d\alpha^{(0)} = \frac{\partial \alpha}{\partial x_1} dx^1 + \frac{\partial \alpha}{\partial x_2} dx^2 + \frac{\partial \alpha}{\partial x_3} dx^3$$

For $\alpha^{(1)} = \alpha_1 dx^1 + \alpha_2 dx^2 + \alpha_3 dx^3$:

$$\begin{aligned}d\alpha^{(1)} &= \left(\frac{\partial \alpha_1}{\partial x^2} dx^2 + \frac{\partial \alpha_1}{\partial x^3} dx^3 \right) dx^1 + \left(\frac{\partial \alpha_2}{\partial x^1} dx^1 + \frac{\partial \alpha_2}{\partial x^3} dx^3 \right) dx^2 + \left(\frac{\partial \alpha_3}{\partial x^1} dx^1 + \frac{\partial \alpha_3}{\partial x^2} dx^2 \right) dx^3 \\ &= \left(\frac{\partial \alpha_2}{\partial x^1} - \frac{\partial \alpha_1}{\partial x^2} \right) dx^1 \wedge dx^2 + \left(\frac{\partial \alpha_3}{\partial x^2} - \frac{\partial \alpha_2}{\partial x^3} \right) dx^2 \wedge dx^3 + \left(\frac{\partial \alpha_1}{\partial x^3} - \frac{\partial \alpha_3}{\partial x^1} \right) dx^3 \wedge dx^1\end{aligned}$$

While for $\alpha^{(2)} = \alpha_{12} dx^1 \wedge dx^2 + \alpha_{23} dx^2 \wedge dx^3 + \alpha_{31} dx^3 \wedge dx^1$:

$$\begin{aligned}d\alpha^{(2)} &= \frac{\partial \alpha_{12}}{\partial x^3} dx^3 \wedge dx^1 \wedge dx^2 + \frac{\partial \alpha_{23}}{\partial x^1} dx^1 \wedge dx^2 \wedge dx^3 + \frac{\partial \alpha_{31}}{\partial x^2} dx^2 \wedge dx^3 \wedge dx^1 \\ &\quad + \left(\frac{\partial \alpha_{23}}{\partial x^1} + \frac{\partial \alpha_{31}}{\partial x^2} + \frac{\partial \alpha_{12}}{\partial x^3} \right) dx^1 \wedge dx^2 \wedge dx^3\end{aligned}$$

From these examples it can be seen that the exterior derivative applied to 1-forms resembles taking its gradient, while applying it to a 2- or 3-form aligns with the curl and divergence operators respectively.

The exterior derivative also defines a relation between spaces of differential orders of all degrees on a manifold, also known as a De Rham complex. For manifold $M \subset \mathbb{R}^n$:

$$\mathbb{R} \longrightarrow \Lambda^{(0)}(M) \xrightarrow{d} \Lambda^{(1)}(M) \xrightarrow{d} \dots \xrightarrow{d} \Lambda^{(n)}(M) \xrightarrow{d} 0$$

A classic theorem that relates the exterior derivative of a differential form and integration is the generalized Stokes' theorem:

Theorem 1.2.1 (Frankel [15, Sec. 5.1]). *Let M be a manifold in \mathbb{R}^n with $V \subset M$ a compact oriented p -dimensional submanifold. Then for $\alpha^{(p-1)} \in \Lambda^{(p)}(M)$:*

$$\int_V d\alpha^{(p-1)} = \int_{\partial V} \alpha^{(p-1)}$$

Where ∂V is the boundary of V . Interpreting integration as a duality pairing of a domain and a differential form, it can be seen that boundary operator ∂ and exterior derivative d are each other's adjoint:

$$(d\alpha^{(p-1)}, V) = (\alpha^{(p-1)}, \partial V)$$

Within differential geometry the generalized Stokes' theorem is the equivalent of the theorems of Green & Stokes as well as (Gauss') divergence theorem.

1.2.2. Interior product

Similarly, there is an operator that decreases the degree of any differential form it acts upon. However, unlike the exterior derivative this operator does not act in isolation. It requires the application of a vector (field) in order to contract said differential form:

Definition 1.12 (Frankel [15, Sec. 2.9a]). *Let M be a manifold in \mathbb{R}^n . Suppose that $\mathbf{v} \in T(M)$ is a vector field. The **interior product** is a mapping $i_{\mathbf{v}} : \Lambda^{(p)}(M) \rightarrow \Lambda^{(p-1)}(M)$, such that for $\alpha^{(0)} \in \Lambda^{(0)}(M)$, $\beta^{(1)} \in \Lambda^{(1)}(M)$ and $\gamma^{(p)} \in \Lambda^{(p)}(M)$:*

$$\begin{aligned} i_{\mathbf{v}}\alpha^{(0)} &= 0 \\ i_{\mathbf{v}}\beta^{(1)} &= \beta^{(1)}(\mathbf{v}) \\ i_{\mathbf{v}}\gamma^{(p)} &= \gamma^{(p)}(\mathbf{v}, \mathbf{w}_2, \dots, \mathbf{w}_p) \end{aligned}$$

With $\mathbf{w}_2, \dots, \mathbf{w}_p \in T(M)$ arbitrary vector fields. The interior product has the following properties:

$$\begin{aligned} i_{a\mathbf{v}+b\mathbf{w}}\alpha^{(p)} &= (ai_{\mathbf{v}} + bi_{\mathbf{w}})\alpha^{(p)} \\ i_{\mathbf{v}} \circ i_{\mathbf{w}}\alpha^{(p)} &= -i_{\mathbf{w}} \circ i_{\mathbf{v}}\alpha^{(p)} \\ i_{\mathbf{v}}(\alpha^{(p)} \wedge \beta^{(q)}) &= i_{\mathbf{v}}\alpha^{(p)} \wedge \beta^{(q)} + (-1)^p \alpha^{(p)} \wedge i_{\mathbf{v}}\beta^{(q)} \end{aligned}$$

Where $\mathbf{v}, \mathbf{w} \in T(M)$ and $\alpha^{(p)}, \beta^{(q)}$ differential p - and q -forms respectively.

The interior product is thus linear: Any vector field can be broken up into its individual components, applied separately to a differential form and summed together afterwards.

Example 4. *Suppose that M is a manifold in \mathbb{R}^3 and let $\mathbf{v} = v^1 \frac{\partial}{\partial x_1} + v^2 \frac{\partial}{\partial x_2} + v^3 \frac{\partial}{\partial x_3}$ be a vector field. For a 1-form $\alpha^{(1)} = \alpha_1 dx^1 + \alpha_2 dx^2 + \alpha_3 dx^3$:*

$$i_{\mathbf{v}}\alpha^{(1)} = \alpha_1 v^1 + \alpha_2 v^2 + \alpha_3 v^3$$

Whereas for a 2-form $\alpha^{(2)} = \alpha_{12} dx^1 \wedge dx^2 + \alpha_{23} dx^2 \wedge dx^3 + \alpha_{31} dx^3 \wedge dx^1$:

$$\begin{aligned} i_{\mathbf{v}}\alpha^{(2)} &= \alpha_{12}(v^1 dx^2 - v^2 dx^1) + \alpha_{23}(v^2 dx^3 - v^3 dx^2) + \alpha_{31}(v^3 dx^1 - v^1 dx^3) \\ &= (\alpha_{31}v^3 - \alpha_{12}v^2) dx^1 + (\alpha_{12}v^1 - \alpha_{23}v^3) dx^2 + (\alpha_{23}v^2 - \alpha_{31}v^1) dx^3 \end{aligned}$$

While for 3-form $\alpha^{(3)} = \alpha_{123} dx^1 \wedge dx^2 \wedge dx^3$:

$$\begin{aligned} i_{\mathbf{v}}\alpha^{(3)} &= \alpha_{123}(v^1 dx^2 \wedge dx^3 - v^2 dx^1 \wedge dx^3 + v^3 dx^1 \wedge dx^2) \\ &= \alpha_{123}v^3 dx^1 \wedge dx^2 + \alpha_{123}v^1 dx^2 \wedge dx^3 + \alpha_{123}v^2 dx^3 \wedge dx^1 \end{aligned}$$

Similar to the exterior derivative, the interior product thus defines a sequence similar to a De Rham complex on manifold M :

$$0 \xleftarrow{i_\nu} \Lambda^{(0)}(M) \xleftarrow{i_\nu} \Lambda^{(1)}(M) \xleftarrow{i_\nu} \dots \xleftarrow{i_\nu} \Lambda^{(n)}(M) \xleftarrow{\quad} \mathbb{R}$$

Where ν is an arbitrary vector field. Note that the exterior derivative and interior product are not each other's inverses; although they map between spaces in opposite directions they describe fundamentally different things.

1.2.3. Lie derivative

Advection is a core part of the fluid flow models described in this thesis and a topic that has been widely described in research into mathematical fluid flows. A departure point for much of this research has been the definition of a flow:

Definition 1.13 (Frankel [15, Sec. 1.4a]). *Let M be a manifold in \mathbb{R}^n and let $\nu \in T(M)$. This vector field defines a one-parameter family of maps (parameterized with t) $\phi_t : M \rightarrow M$ that describe the transport of $x \in M$ under ν , such that:*

$$\nu_x := \frac{d\phi_t(x)}{dt}$$

Then ϕ_t is called a **flow**. The collection of points $\phi_t(x)$ at various t is called an **integral curve**.

As the name implies a flow models the transport of particles (or alternatively the deformation of a manifold) under a given vector field. Whereas in this definition vector field ν was kept independent of t (i.e. steady), time-varying vector fields $\nu(t)$ can also be used. The transport and deformation of a given flow can be measured and computed through the Lie derivative:

Definition 1.14 (Frankel [15, Sec. 4.2a]). *Let M be a manifold in \mathbb{R}^n and let $\nu \in T(M)$ with flow ϕ_t . The **Lie derivative** $\mathcal{L}_\nu : \Lambda^{(p)}(M) \rightarrow \Lambda^{(p)}(M)$ of a differential p -form $\alpha^{(p)}$ under ν is defined as:*

$$\begin{aligned} \mathcal{L}_\nu \alpha^{(p)}(x) &:= \frac{d}{dt} [\phi_t^* \alpha^{(p)}(x)]_{t=0} \\ &= \lim_{t \rightarrow 0} \frac{\phi_t^* \alpha^{(p)}(\phi_t(x)) - \alpha^{(p)}(x)}{t} \end{aligned}$$

With $x \in M$. In words: The Lie derivative of $\alpha^{(p)}$ under ν is equal to the difference between $\alpha^{(p)}$ evaluated at x and $\alpha^{(p)}$ evaluated at $\phi_t(x)$ pulled back to x under flow pullback operator ϕ_t^* , in the limit of $t \rightarrow 0$. The Lie derivative has the following properties:

$$\begin{aligned} \mathcal{L}_\nu \circ d\alpha^{(p)}(x) &= d \circ \mathcal{L}_\nu \alpha^{(p)}(x) \\ \mathcal{L}_\nu \circ i_\nu \alpha^{(p)}(x) &= i_\nu \circ \mathcal{L}_\nu \alpha^{(p)}(x) \\ \mathcal{L}_\nu (\alpha^{(p)} \wedge \beta^{(q)}) &= \mathcal{L}_\nu \alpha^{(p)} \wedge \beta^{(q)} + \alpha^{(p)} \wedge \mathcal{L}_\nu \beta^{(q)} \end{aligned}$$

This last property is often referred to as the Lie derivative's Leibniz rule.

One can see how this definition measures the instantaneous effect of a flow upon p -form $\alpha^{(p)}$. Within this work a different formulation of the Lie derivative is used. This formulation is entirely equivalent to the definition above, as was proven by Cartan. It is also known as Cartan's homotopy or magic formula:

Theorem 1.2.2 (Frankel [15, Sec. 4.2b]). *Let M be a manifold in \mathbb{R}^n , $\nu \in T(M)$ and $\alpha^{(p)} \in \Lambda^{(p)}(M)$. Then:*

$$\mathcal{L}_\nu \alpha^{(p)} = (d \circ i_\nu + i_\nu \circ d) \alpha^{(p)}$$

The proof is short and can be found in [15, Sec. 4.2b]. Note that the Lie derivative acting on differential forms maps p -forms into other p -forms; this is consistent with the interior product mapping p -forms into $p-1$ -forms and the exterior derivative mapping p -forms into $p+1$ -forms.

Example 5. Suppose that M is a manifold in \mathbb{R}^2 and let $\mathbf{v} = v^1 \frac{\partial}{\partial x^1} + v^2 \frac{\partial}{\partial x^2}$. For a 0-form $\alpha^{(0)} = \alpha$:

$$\begin{aligned}\mathcal{L}_{\mathbf{v}}\alpha^{(0)} &= (di_{\mathbf{v}} + i_{\mathbf{v}}d)\alpha^{(0)} \\ &= \frac{\partial \alpha}{\partial x^1} v^1 + \frac{\partial \alpha}{\partial x^2} v^2\end{aligned}$$

Whereas for 1-form $\alpha^{(1)} = \alpha_1 dx^1 + \alpha_2 dx^2$:

$$\begin{aligned}\mathcal{L}_{\mathbf{v}}\alpha^{(1)} &= (di_{\mathbf{v}} + i_{\mathbf{v}}d)\alpha^{(1)} = (di_{\mathbf{v}}\alpha^{(1)} + i_{\mathbf{v}}d\alpha^{(1)}) \\ &= \frac{\partial \alpha_1}{\partial x^1} v^1 dx^1 + \frac{\partial \alpha_2}{\partial x^2} v^2 dx^2 + v^2 \left(\frac{\partial \alpha_1}{\partial x^2} - \frac{\partial \alpha_2}{\partial x^1} \right) dx^1 + v^1 \left(\frac{\partial \alpha_2}{\partial x^1} - \frac{\partial \alpha_1}{\partial x^2} \right) dx^2 \\ &= \left(\frac{\partial \alpha_1}{\partial x^1} v^1 + v^2 \frac{\partial \alpha_1}{\partial x^2} - v^2 \frac{\partial \alpha_2}{\partial x^1} \right) dx^1 + \left(\frac{\partial \alpha_2}{\partial x^2} v^2 + v^1 \frac{\partial \alpha_2}{\partial x^1} - v^1 \frac{\partial \alpha_1}{\partial x^2} \right) dx^2\end{aligned}$$

While for 2-form $\alpha^{(2)} = \alpha_{12} dx^1 \wedge dx^2$:

$$\begin{aligned}\mathcal{L}_{\mathbf{v}}\alpha^{(2)} &= (di_{\mathbf{v}} + i_{\mathbf{v}}d)\alpha^{(2)} \\ &= \left(\frac{\partial v^1}{\partial x^1} \alpha_{12} + \frac{\partial v^2}{\partial x^2} \alpha_{12} \right) dx^1 \wedge dx^2\end{aligned}$$

Note that the expressions for the Lie derivative of 0- and n -forms (in \mathbb{R}^n) are the shortest; this is due to $di_{\mathbf{v}}$ and $i_{\mathbf{v}}d$ being zero for 0- and n -forms respectively. As is covered in more detail in chapter 3, the Lie derivative of a 0-form coincides with the convective formulation of the advection operator often found in fluid flow models. Similarly the Lie derivative of a volume form (an n -form on a manifold in \mathbb{R}^n) coincides with the conservative formulation of such an advection operator.

1.2.4. Musical isomorphisms

The previous operators could all be defined independently from any metric; no notions of distances, angles or lengths on manifold M were necessary to define and apply them to any differential p -form. There are however some operations that do require such a metric definition. For this a function will first be defined that allows one to work with metric structures:

Definition 1.15 (Frankel [15, Sec. 2.1d]). Let M be a manifold in \mathbb{R}^n with $\left\{ \frac{\partial}{\partial x^i} \right\}$ a basis for tangent space $T(M)$. We define matrix g as the **metric tensor** with the following entries:

$$g_{ij}(x) = \left\langle \frac{\partial}{\partial x^i}, \frac{\partial}{\partial x^j} \right\rangle$$

For every point $x \in M$. Here $\langle \cdot, \cdot \rangle : T(M) \times T(M) \rightarrow \mathbb{R}$ is an **inner product**. If the inner product is positive definite for all $x \in M$, M is called a **Riemannian manifold**. The inverse of a positive definite metric tensor is denoted by $(g)^{-1}$, and its entries by g^{ij} . If $\left\{ \frac{\partial}{\partial x^i} \right\}$ is an orthonormal basis $g_{ij} = \delta_i^j$ and the metric tensor is thus equal to the identity matrix.

The metric tensor allows us to define operators that convert vectors into covectors and vice versa. In fact, for a given vector \mathbf{v} we can interpret the inner product as a covector $v^{(1)}(\mathbf{w}) = \langle \mathbf{v}, \mathbf{w} \rangle$, as it takes in vector \mathbf{w} and maps from tangent space $T(M)$ to \mathbb{R} . Expanding $v^{(1)}$ in the covector basis $\{dx^j\}$:

$$\begin{aligned}v^{(1)} &= \sum_j v_j dx^j = \sum_j v \left(\frac{\partial}{\partial x^j} \right) dx^j \\ &= \sum_j \left\langle \mathbf{v}, \frac{\partial}{\partial x^j} \right\rangle dx^j \\ &= \sum_j \left\langle \sum_i v^i \frac{\partial}{\partial x^i}, \frac{\partial}{\partial x^j} \right\rangle dx^j \\ &= \sum_j \left(\underbrace{\sum_i v^i g_{ij}}_{=v_j} \right) dx^j\end{aligned}$$

From which follow two operators:

Definition 1.16 (Kreeft [31, Sec. 10.2.1]). *Let M be a Riemannian manifold in \mathbb{R}^n equipped with metric tensor g . The 'musical' isomorphisms (named after their notation) map between $T(M)$ and $T^*(M) = \Lambda^{(1)}(M)$. The **flat** $\flat : T(M) \rightarrow \Lambda^{(1)}(M)$ and **sharp** $\sharp : \Lambda^{(1)}(M) \rightarrow T(M)$ operators acting on 1-form $\alpha^{(1)}$ and vector field \mathbf{v} are defined as:*

$$\begin{aligned} (\mathbf{v})^\flat &= \left(\sum_i v^i \frac{\partial}{\partial x^i} \right)^\flat = \sum_i \underbrace{\left(\sum_j v^j g_{ij} \right)}_{=v_i} dx^i = v^{(1)} \\ (\alpha^{(1)})^\sharp &= \left(\sum_i \alpha_i dx^i \right)^\sharp = \sum_i \underbrace{\left(\sum_j v_j g^{ij} \right)}_{=\alpha^i} \frac{\partial}{\partial x^i} = \boldsymbol{\alpha} \end{aligned}$$

At this point it should be noted that the domains used in this research are equipped with orthogonal coordinate systems, such that $\langle \frac{\partial}{\partial x^i}, \frac{\partial}{\partial x^j} \rangle = \delta_i^j$. Hence metric tensor g is just the identity matrix.

1.2.5. Hodge \star operator

Similar to how vectors and covectors (1-forms) are related to one another, $n - 1$ -forms are associated to a vector (field) through the standard volume form:

Definition 1.17 (Kreeft [31, Sec. 4.3]). *Let M be a Riemannian manifold in \mathbb{R}^n . The **standard or canonical volume form** $\sigma^{(n)}$ is defined as:*

$$\sigma^{(n)} := \sqrt{g} dx^1 \wedge dx^2 \wedge \dots \wedge dx^n$$

With some abuse of notation, g is the determinant of the metric tensor.

For a given vector \mathbf{v} the aforementioned associated $n - 1$ -form can be computed with the interior product: $i_{\mathbf{v}}\sigma^{(n)}$. Vectors are thus associated to both 1-forms and $n - 1$ -forms, and these differential 1- and $n - 1$ -forms are thus transitively dual to one another as well. This duality of differential forms of different degrees is generalized by the Hodge \star operator:

Definition 1.18 (Frankel [15, Sec. 14.1]). *Let M be a Riemannian manifold in \mathbb{R}^n . The Hodge \star operator is a mapping $\star : \Lambda^{(p)}(M) \rightarrow \Lambda^{(n-p)}(M)$ such that, for a p -form $\alpha^{(p)} \in \Lambda^{(p)}(M)$:*

$$\star \alpha(x_1, \dots, x_n) dx^{i_1} \wedge dx^{i_2} \wedge \dots \wedge dx^{i_p} = \sqrt{g} \text{sgn} \alpha(x_1, \dots, x_n) dx^{i_{p+1}} \wedge dx^{i_{p+2}} \wedge \dots \wedge dx^{i_n}$$

With $\{i_1, i_2, \dots, i_p\} \subseteq \{1, 2, \dots, n\}$, $\{i_{p+1}, i_{p+2}, \dots, i_n\} = \{1, 2, \dots, n\} \setminus \{i_1, i_2, \dots, i_p\}$ and where:

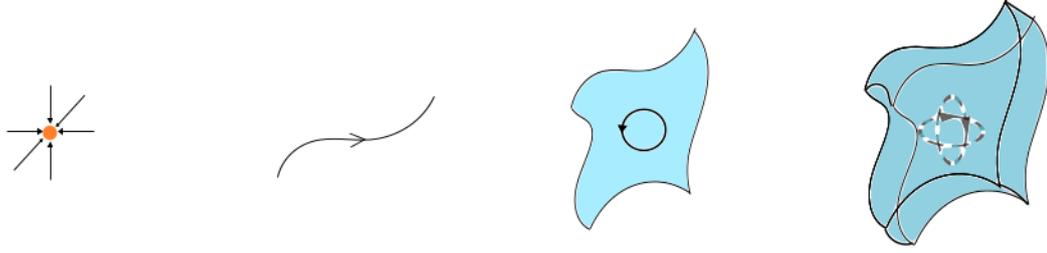
$$\text{sgn} = \begin{cases} -1 & \text{if } \{i_1, i_2, \dots, i_p, i_{p+1}, \dots, i_n\} \text{ is an even permutation of } \{1, 2, \dots, n\} \\ 1 & \text{otherwise} \end{cases}$$

Applying the Hodge \star twice gives:

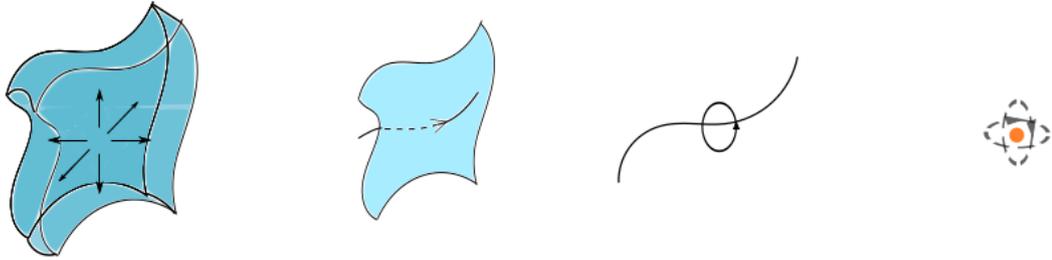
$$\star \star \alpha^{(p)} = (-1)^{p(n-p)} \alpha^{(p)}$$

For the inverse mapping $\Lambda^{(n-p)}(M) \rightarrow \Lambda^{(p)}(M)$ the factor $\sqrt{g^{-1}}$ appears, causing the composition $\star \star$ to be equal to plus or minus the identity map.

The Hodge \star maps any k -form basis to its dual elements and does not affect the value of any differential form; the factor \sqrt{g} merely accounts for the curvature of manifold M . There is a geometrical interpretation for the action of the Hodge operator. To see this we endow every geometric element in \mathbb{R}^3 with an inner orientation, which defines a positive and negative direction internal to each object. These are shown in figure 1.1a. The Hodge \star maps an object with a given inner orientation to its dual object with an outer orientation. These outer orientations are shown in figure 1.1b. Note how the inner orientations of k -dimensional objects match with the outer orientations of $3 - k$ -dimensional objects. This is not coincidental: While the orientation of a differential form is conserved by the Hodge \star its association with a geometric object is modified under the same operation. While this property of the Hodge \star is not used in this research, numerical methods can be constructed that use this structure to discretize the Hodge \star operator. The finite volume method of Toshniwal [55, 56] is an example of this.



(a) Inner orientations of geometric objects (from left to right: points, curves, surfaces and volumes) in \mathbb{R}^3



(b) Outer orientations of geometric objects (from left to right: volume, surfaces, curves and points) in \mathbb{R}^3

Figure 1.1: Orientations of various spatial elements in \mathbb{R}^3 , taken from [55, p. 6]

Example 6. For manifold M in \mathbb{R}^2 the Hodge \star acts as follows:

$$\begin{aligned}\star 1 &= \sigma^{(2)} = \sqrt{g} dx^1 \wedge dx^2 \\ \star dx^1 &= \sqrt{g} dx^2 \\ \star dx^2 &= -\sqrt{g} dx^1 \\ \star dx^1 \wedge dx^2 &= \sqrt{g}\end{aligned}$$

With the Hodge \star wedge products of two differential p -forms can be written as an n -form, for which integration over n -dimensional manifold M is possible:

Definition 1.19 (Kreeft [31, Sec. 4.7]). Let M be a Riemannian manifold in \mathbb{R}^n . The Hodge \star operator gives rise to an L^2 inner product on M , $(\cdot, \cdot)_M : \Lambda^{(p)}(M) \times \Lambda^{(p)}(M) \rightarrow \mathbb{R}$. For $\alpha^{(p)}, \beta^{(p)} \in \Lambda^{(p)}(M)$:

$$(\alpha^{(p)}, \beta^{(p)})_M := \int_M \alpha^{(p)} \wedge \star \beta^{(p)}$$

This L^2 inner product allows us to restrict the spaces $\Lambda^{(p)}(M)$ of differential p -forms on Riemannian manifold M . The space of L^2 -integrable p -forms on M will be denoted by $L^2 \Lambda^{(p)}(M)$.

1.3. De Rham complex and relations between spaces

As was already alluded to before, the operators covered here can be used to relate the various spaces of differential forms to one another in a De Rham complex. With the Hodge \star operator this can be extended to a double De Rham complex:

$$\begin{array}{ccccccc} \mathbb{R} & \longrightarrow & \Lambda^{(0)}(M) & \xrightarrow{d} & \Lambda^{(1)}(M) & \xrightarrow{d} & \dots & \xrightarrow{d} & \Lambda^{(n)}(M) & \xrightarrow{d} & 0 \\ & & \downarrow \star & & \downarrow \star & & \downarrow \star & & \downarrow \star & & \\ 0 & \xleftarrow{d} & \Lambda^{(n)}(M) & \xleftarrow{d} & \Lambda^{(n-1)}(M) & \xleftarrow{d} & \dots & \xleftarrow{d} & \Lambda^{(0)}(M) & \xleftarrow{d} & \mathbb{R} \end{array}$$

Here the horizontal mappings are purely topological and do not incorporate any notion of metric concepts such as distance, length or angles. Hence the exact same mappings are valid for any member of a given equivalence class of domains. It is the vertical mappings, performed by the Hodge \star , where metric notions come into play through the metric tensor g .

2

Structure-preserving isogeometric analysis

Now that the relevant concepts from differential geometry have been introduced the topic of structure-preserving isogeometric discretization methods can be covered. This chapter will treat the ways in which discrete operators are constructed that reflect the structure of the continuous operators which were covered in chapter 1. Subspaces of the k -form spaces are constructed on which 'discrete differential forms' are represented with a finite number of continuous degrees of freedom. The underlying thought behind the construction of discrete operators is to preserve the mappings of the continuous operators between different k -form spaces. Since the discrete k -forms are a subspace of all k -forms, it is thus paramount that any discrete operator maps between the spaces of discrete k -forms. In this way the De Rham complex presented in the previous chapter is reflected in structure-preserving methods.

To kick this off section 2.1 goes over the relevant discrete objects that are used as counterparts of the continuous objects presented in chapter 1. Section 2.2 introduces the spline basis functions from isogeometric analysis that are used in this research, in addition to several operations that can be performed on splines. These functions are then used in section 2.3 to construct discrete versions of the exterior derivative, interior product and Lie derivative; each of these discrete operators maps between the discretized spaces that correspond to the spaces between which their continuous counterparts map. Finally section 2.4 combines all of the preceding information into a large commutative diagram that shows the relations between the various spaces introduced in this & the previous chapter and how the continuous and discrete operators map between them.

2.1. Discrete objects

A central theme in scientific computing is the approximation of infinite-dimensional problems by a finite number of variables, such that computers can determine approximate solutions in some finite time. The ways in which numerical methods construct these approximations are what set them apart from one another. Numerical solutions, their governing problems and the domain on which these problems are posed all have to be represented by some finite-dimensional set of variables.

2.1.1. Cell complexes

Many numerical methods start by decomposing the computational domain into a finite number of subdomains called cells:

Definition 2.1 (Kreeft [31, Sec. 5.1.1]). *Let M be a manifold in \mathbb{R}^n . A **k -cell** $\tau_{(k)}$ of dimension $k \leq n$ is a set $\tau_{(k)} = \{x \in M\}$ that is homeomorphic to the closed k -ball $B_k = \{x \in \mathbb{R}^k : \|x\| \leq 1\}$. Its boundary $\partial\tau_{(k)}$ is homeomorphic to $\partial B_k = \{x \in \mathbb{R}^k : \|x\| = 1\}$.*

The boundary of a k -cell ($\partial\tau_{(k)}$) consists of $k - 1$ -cells. Starting with 0-cells (points) on a manifold in \mathbb{R}^3 , 1-cells (lines or curves) can be drawn to connect the points; these 1-cells define the borders of 2-cells (surfaces), which in turn are the boundaries of 3-cells (volumes). Within this work cuboid cells are used and the test cases are limited to \mathbb{R}^2 . Thus only points, lines and surfaces are used to cover any computational domain. In \mathbb{R}^2 all surfaces are bordered by four lines, which in turn have two points on their boundary (one point at each end).

Similarly a one-dimensional manifold can be covered only with lines, each of which also has two points on its boundary. A collection of k -cells that covers a manifold is also called a cell complex:

Definition 2.2 (Kreeft [31, Sec. 5.1.1], Hatcher [23, p. 5]). *Let M be a manifold in \mathbb{R}^n . A **cell complex** D on M is a finite collection of k -cells with $k = \{n, n-1, \dots, 0\}$:*

$$D = \{\tau_{(n)}\} \cup \{\tau_{(n-1)}\} \cup \dots \cup \{\tau_{(0)}\}$$

Such that $\{\tau_{(n)}\}$ covers M , every face of a k -cell is contained in $\{\tau_{(k-1)}\}$ and for any pair of k -cells $\tau_{(k)}, \sigma_{(k)} \in D$ for which $\tau_{(k)} \neq \sigma_{(k)}$:

$$\tau_{(k)} \cap \sigma_{(k)} = \begin{cases} s_{(k-1)} & \text{if } \tau_{(k)}, \sigma_{(k)} \text{ share boundary } s_{(k-1)} \\ \emptyset & \text{otherwise} \end{cases}$$

For finite-dimensional cell complexes to exist, manifold M must be compact.

Cell complexes are commonly referred to as meshes in computational science. These thus form a finite-dimensional representation of a compact manifold, in the sense that this manifold is covered by a finite number of n -cells. From these n -cells follow the $n-1$ -cells that define the boundary of each n -cell and so forth down to 0-cells as is covered above. Each cell is given an inner orientation; that is, a positive orientation that is internal to each cell. An example of these orientations for a 2-cell (surface) and its neighboring 1- (lines) and 0-cells (points) is shown in figure 2.1. As can be seen the 2-cell orientation consists of a positive rotational direction, whereas the 1-cell orientation defines a positive direction along the cell and the 0-cell orientation defines a positive incidence direction. Also note how the different inner orientations of k - and $k-1$ -cells align: They either agree or oppose one another. Which direction is considered positive for a k -cell depends on the global orientation; the exact choice of this global orientation does not matter, as long as it is applied consistently to all elements of cell complex D . For example, in figure 2.1 the positive orientation of 1-cells has been defined to be upward or to the right.

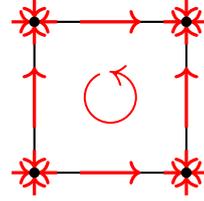


Figure 2.1: Example positive global inner orientation directions of 2-, 1- and 0-cells

2.1.2. Chains and cochains

With these cell orientations linear combinations of k -cells can be made, which is formalized in the following definition:

Definition 2.3 (Kreeft [31, Sec. 5.1.2], Hatcher [23, p. 105]). *Let D be a cell complex D on compact manifold M . $C_k(D)$, the space of **k -chains** of D , consists of linear combinations of k -cells, such that:*

$$C_k(D) = \left\{ c_{(k)} \in C_k(D) : c_{(k)} = \sum_i c^i \tau_{(k),i}, c^i = \{-1, 0, 1\}, \tau_{(k)} \in D \right\}$$

The coefficients c^i depend on the orientation of each cell in chain $c_{(k)}$:

$$c^i = \begin{cases} 1 & \text{if the orientation of } \tau_{(k),i} \text{ agrees with its orientation in } D \\ -1 & \text{if the orientation of } \tau_{(k),i} \text{ opposes its orientation in } D \\ 0 & \text{if } \tau_{(k),i} \text{ is not in chain } c_{(k)} \end{cases}$$

Chains are thus simply combinations of k -cells that take into account the alignment of orientation of each k -cell relative to its global orientation. Chains of k - and $k-1$ cells are related to one another through the boundary operator:

Definition 2.4 (Kreeft [31, Sec. 5.1.2], Hatcher [23, p. 105]). *Let D be a cell complex on compact manifold M . The **boundary operator** ∂ is a map $\partial : C_k(D) \rightarrow C_{k-1}(D)$, such that for chain $c_k \in C_k(D)$:*

$$\begin{aligned} \partial c_k &= \partial \sum_i c^i \tau_{(k),i} = \sum_i c^i \partial \tau_{(k),i} \\ &= \sum_i c^i \left(\sum_j e_j^i \tau_{(k-1),j} \right) \end{aligned}$$

Where the coefficients e_j^i are defined as:

$$e_j^i = \begin{cases} 1 & \text{if the orientation of } \tau_{(k-1),j} \text{ aligns with the orientation of } \partial\tau_{(k),i} \\ -1 & \text{if the orientation of } \tau_{(k-1),j} \text{ opposes the orientation of } \partial\tau_{(k),i} \\ 0 & \text{if } \tau_{(k-1),j} \cap \partial\tau_{(k),i} = \emptyset \end{cases}$$

For ∂ it holds that:

$$\begin{aligned} \partial c_{(0)} &= \emptyset \\ \partial\partial c_{(k)} &= \emptyset \end{aligned}$$

In other words, the boundary of a 0-chain is empty and the boundary's boundary is empty as well. The latter holds because the boundary of a k -chain consists of closed cycles, which do not have any end-nodes.

The boundary operator thus maps between the various spaces of chains $C_k(D)$ on cell complex D . All chain spaces are thus related through a series of maps:

$$0 \xleftarrow{\partial} C_0(D) \xleftarrow{\partial} C_1(D) \xleftarrow{\partial} \dots \xleftarrow{\partial} C_n(D) \xleftarrow{\quad} \mathbb{R}$$

If one introduces a global ordering of chain spaces $C_k(D)$ and $C_{k-1}(D)$, the coefficients of chains $c_{(k)}$, $c_{(k-1)}$ can be placed in global coefficient vectors $\mathbf{c}_{(k)}$, $\mathbf{c}_{(k)}$. The boundary mapping $\partial : C_k(D) \rightarrow C_{k-1}(D)$ can then be written as a matrix $\mathbb{E}_{k-1,k}$.

Example 7. Consider the cell complex shown in figure 2.1 with their global orientations; these are also shown (and have been labeled) in figure 2.2a. Also shown are a 2-chain in figure 2.2b and a 1-chain in 2.2c; both with the orientation of each cell in their respective chains. Then:

$$\partial c_{(2)} = \partial(s_1) = -l_1 + l_2 + l_3 - l_4$$

Since $\partial c_{(2)}$ is a cycle consisting of the four lines bordering s_1 , $\partial\partial c_{(2)} = \emptyset$. On the other hand, for the 1-chain $c_{(1)} = -l_1 + l_2$ shown in figure 2.2c:

$$\partial c_{(1)} = \partial(-l_1 + l_2) = -(p_4 - p_1) + (-p_1 + p_2) = p_2 - p_4$$

Thus the boundary operator acting on $c_{(2)}$ can be represented with the following vector:

$$\mathbb{E}_{1,2} = \begin{bmatrix} -1 \\ 1 \\ 1 \\ -1 \end{bmatrix}$$

Which maps the coefficient vector $[s^1]$ (corresponding to s_1) to coefficient vector $[l_1 \ l_2 \ l_3 \ l_4]$. Similarly the boundary operator acting on $c_{(1)}$ can be formulated as:

$$\mathbb{E}_{0,1} = \begin{bmatrix} -1 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & -1 \end{bmatrix}$$

From which it follows that:

$$\mathbb{E}_{0,1}\mathbb{E}_{1,2} = \begin{bmatrix} -1 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} -1 \\ 1 \\ 1 \\ -1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Which reflects the nilpotency of the boundary operator: Applying ∂ twice in succession results in an empty set.

Dual objects can be created for chains just as was done for vectors:

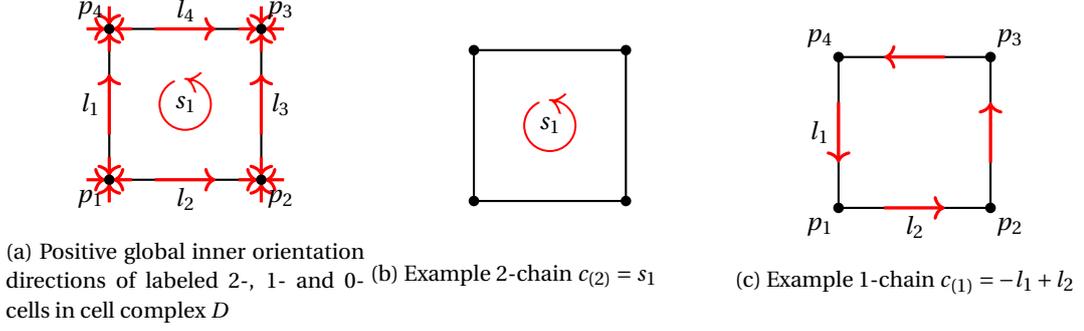


Figure 2.2: Example positive global inner orientation directions of 2-, 1- and 0-cells

Definition 2.5 (Kreeft [31, Sec. 5.2.1], Frankel [15, App. B.b]). Let $C_k(D)$ be the space of k -chains on cell complex D . Its dual space is the **cochain space** $C^k(D)$. **Cochains** are maps $c^{(k)} \in C^k(D)$ such that $c^{(k)} : C_k(D) \rightarrow \mathbb{R}$, and:

$$c^{(k)} = \sum_i c_i \tau^{(k),i}$$

Where $\tau^{(k),i}$ is dual to k -chain $\tau_{(k),i}$, such that for their duality pairing $\langle \tau^{(k),i}, \tau_{(k),j} \rangle = \delta_i^j$. Then:

$$\langle c^{(k)}, c_{(k)} \rangle = \sum_i \sum_j c_i \underbrace{\langle \tau^{(k),i}, \tau_{(k),j} \rangle}_{=\delta_i^j} c^j = \sum_i c_i c^i$$

Cochains play the part of discrete counterparts to differential forms in structure-preserving methods. The duality pairing of cochains and chains can be used to define an operator that is adjoint to boundary operator ∂ :

Definition 2.6 (Kreeft [31, Sec. 5.2.1], Frankel [15, App. B.b]). Let D be a cell complex. The **coboundary operator** δ is the mapping $\delta : C^k(D) \rightarrow C^{(k+1)}(D)$ that is adjoint to boundary operator ∂ . For $k+1$ -chain $c_{(k+1)}$ and k -cochain $c^{(k)}$:

$$\langle \delta c^{(k)}, c_{(k+1)} \rangle = \langle c^{(k)}, \partial c_{(k+1)} \rangle$$

Then, similarly to the boundary operator, δ is nilpotent: $\delta \delta c^{(k)}$.

Just like how the boundary operator defined a series of mappings that relate all spaces of k -chains to one another, the coboundary operator links all spaces of k -cochains together:

$$\mathbb{R} \longrightarrow C^{(0)}(D) \xrightarrow{\delta} C^{(1)}(D) \xrightarrow{\delta} \dots \xrightarrow{\delta} C^{(n)}(D) \xrightarrow{\delta} 0$$

As it is the adjoint of ∂ , the coboundary operator is equal to the transpose of the incidence matrix $\mathbb{E}_{k-1,k}$, since for k -cochain $a^{(k)} = \mathbf{a} \boldsymbol{\tau}^{(k)}$ and $k+1$ -chain $b_{(k+1)} = \mathbf{b} \boldsymbol{\tau}^{(k+1)}$:

$$\begin{aligned} \langle \delta a^{(k)}, b_{(k+1)} \rangle &= \langle \delta (\mathbf{a} \boldsymbol{\tau}^{(k)}), \mathbf{b} \boldsymbol{\tau}^{(k+1)} \rangle = \delta (\mathbf{a}) \underbrace{\boldsymbol{\tau}^{(k+1)} (\boldsymbol{\tau}^{(k+1)})^T}_{=I} \mathbf{b}^T = \delta (\mathbf{a}) \mathbf{b}^T \\ \langle a^{(k)}, \partial b_{(k+1)} \rangle &= \langle \mathbf{a} \boldsymbol{\tau}^{(k)}, \partial (\mathbf{b} \boldsymbol{\tau}^{(k+1)}) \rangle = \mathbf{a} \boldsymbol{\tau}^{(k)} \underbrace{(\boldsymbol{\tau}^{(k)})^T}_{=I} (\mathbb{E}_{k,k+1} \mathbf{b})^T = \mathbf{a} \mathbf{b}^T \mathbb{E}_{k,k+1}^T \end{aligned}$$

From which follows that $\delta = \mathbb{E}^{k+1,k} = \mathbb{E}_{k,k+1}^T$.

2.1.3. Linking continuous and discrete variables

Having defined cochains, a connection needs to be defined that links the infinite-dimensional differential forms to the finite-dimensional cochains that are dual to the collection of chain spaces on manifold Ω . This

raises a bigger question: How can (continuous) models posed in terms of differential forms in $\Lambda^{(k)}(\Omega)$ be projected to spaces of discrete differential forms $\Lambda_h^{(k)}(\Omega_h) \subset \Lambda^{(k)}(\Omega)$ on cell complex Ω_h such that the discretized model retains the structure of the continuous model? Ideally any map $T : \Lambda^{(k)}(\Omega) \rightarrow \Lambda^{(l)}(\Omega)$ would commute with the projection operator $\pi^{(q)} : \Lambda^{(q)}(\Omega) \rightarrow \Lambda_h^{(q)}(\Omega_h)$, such that:

$$\begin{array}{ccc} \Lambda^{(k)}(\Omega) & \xrightarrow{T} & \Lambda^{(l)}(\Omega) \\ \downarrow \pi^{(k)} & & \downarrow \pi^{(l)} \\ \Lambda_h^{(k)}(\Omega_h) & \xrightarrow{T} & \Lambda_h^{(l)}(\Omega_h) \end{array}$$

In this way it does not matter whether one applies a given operator to an already-discretized problem or whether said operator is applied to the continuous problem followed by a projection to the finite-dimensional (discrete) basis: $\pi^{(l)} \circ T = T \circ \pi^{(k)}$. This idea has been formalized in greater detail by (among others) Bochev & Hyman in [7] and Hyman & Scovel in [28]. Projection $\pi^{(k)}$ is defined as the composition of a reduction step followed by a reconstruction step.

Definition 2.7 (Kreft [31, Sec. 6.1]). *Let Ω be a manifold in \mathbb{R}^n covered by cell complex Ω_h . The **reduction operator** $\mathcal{R}^{(k)}$ is a mapping $\mathcal{R}^{(k)} : \Lambda^{(k)}(\Omega) \rightarrow C^{(k)}(\Omega_h)$ that maps differential k -forms to k -cochains, such that:*

$$\begin{array}{ccc} \Lambda^{(k)}(\Omega) & \xrightarrow{d} & \Lambda^{(k+1)}(\Omega) \\ \downarrow \mathcal{R}^{(k)} & & \downarrow \mathcal{R}^{(k+1)} \\ C^{(k)}(\Omega_h) & \xrightarrow{\delta} & C^{(k+1)}(\Omega_h) \end{array}$$

In other words, the reduction operator preserves the commutative relation between the coboundary operator δ and exterior derivative d , since $\delta \mathcal{R}^{(k)} = \mathcal{R}^{(k+1)} d$. Note that the reduction operator is degree-specific ($\mathcal{R}^{(k)} \neq \mathcal{R}^{(k+1)}$).

The reduction operator thus maps differential forms on manifold Ω to corresponding cochains on cell complex Ω_h . Note that the cochains under consideration in this work are of a finite size. This defines the first part of projection operator π . To finish the map $\Lambda^{(k)}(\Omega) \rightarrow \Lambda_h^{(k)}(\Omega_h)$ a reconstruction operator is to be defined as well:

Definition 2.8 (Kreft [31, Sec. 6.1]). *Let Ω be a manifold in \mathbb{R}^n covered by cell complex Ω_h . The **reconstruction operator** $\mathcal{S}^{(k)}$ is a mapping $\mathcal{S}^{(k)} : C^{(k)}(\Omega_h) \rightarrow \Lambda_h^{(k)}(\Omega_h)$ that maps k -cochains to discrete differential k -forms, such that:*

$$\begin{array}{ccc} \Lambda_h^{(k)}(\Omega_h) & \xrightarrow{d} & \Lambda_h^{(k+1)}(\Omega_h) \\ \mathcal{S}^{(k)} \uparrow & & \mathcal{S}^{(k+1)} \uparrow \\ C^{(k)}(\Omega_h) & \xrightarrow{\delta} & C^{(k+1)}(\Omega_h) \end{array}$$

The reconstruction operator preserves the commutative relation between δ and d , as did the reduction operator.

Reconstruction operator $\mathcal{S}^{(k)}$ thus maps k -cochains to discrete differential forms. These are differential forms that can be represented by a finite number of degrees of freedom (the cochains); the reconstruction operator merely defines the map that is used to go from a discrete representation in terms of cochains to a continuous representation that is contained in $\Lambda_h^{(k)}(\Omega_h) \subset \Lambda^{(k)}(\Omega)$. Different choices for the reduction and reconstruction operators distinguish many discretization methods. For example, classical finite difference methods represent differential forms using sets of point values, and reconstruct these into discrete differential forms through Taylor series expansions. On the other hand, finite volume- and finite element methods reduce differential forms into a set of integral values and either use interpolation methods (finite volume) or basis function expansions (finite element) to reconstruct continuous solutions.

Having defined the reduction and reconstruction operators, projection $\pi^{(k)} : \Lambda^{(k)}(\Omega) \rightarrow \Lambda_h^{(k)}(\Omega_h)$ can be defined as $\pi^{(k)} = \mathcal{I}^{(k)} \circ \mathcal{R}^{(k)}$. Differential k -forms are thus mapped to the space of 'discrete differential k -forms'. This space contains precisely those differential forms that can be represented on the k -cochains in cell complex Ω_h , and as such $\Lambda_h^{(k)}(\Omega_h) \subset \Lambda^{(k)}(\Omega)$. In order to have an effective discretization approach the following two properties are required:

- $\mathcal{R}^{(k)} \circ \mathcal{I}^{(k)} \alpha_h^{(k)} = I \forall \alpha_h^{(k)} \in \Lambda_h^{(k)}(\Omega_h)$
- $\|\alpha^{(k)} - \mathcal{I}^{(k)} \circ \mathcal{R}^{(k)} \alpha^{(k)}\| = \mathcal{O}(h^p) \forall \alpha^{(k)} \in \Lambda^{(k)}(\Omega)$

The first of these describes consistency: Any discrete differential k -form $\alpha_h^{(k)} \in \Lambda_h^{(k)}(\Omega_h)$ should be reduced in such a way that reconstructing it from its associated k -cochain description leads again to $\alpha_h^{(k)}$. Repeated projection of $\alpha_h^{(k)}$ should thus not alter it in any way. The latter is called the approximation property, and implies that successful discretization methods should approximate any differential form more closely (expressed in some error norm $\|\cdot\|$) as characteristic grid size h is reduced; the speed with which this error decreases depends on the polynomial degree p of the approximations made in the projection step.

Combining the commutative diagrams shown earlier then leads to the following relation between continuous differential forms, cochains and discrete differential forms:

$$\begin{array}{ccccccc}
 \dots & \xrightarrow{d} & \Lambda^{(k)}(\Omega) & \xrightarrow{d} & \Lambda^{(k+1)}(\Omega) & \xrightarrow{d} & \Lambda^{(k+2)}(\Omega) & \xrightarrow{d} & \dots \\
 & & \downarrow \mathcal{R}^{(k)} & & \downarrow \mathcal{R}^{(k+1)} & & \downarrow \mathcal{R}^{(k+2)} & & \\
 \dots & \xrightarrow{\delta} & C^{(k)}(\Omega_h) & \xrightarrow{\delta} & C^{(k+1)}(\Omega_h) & \xrightarrow{\delta} & C^{(k+2)}(\Omega_h) & \xrightarrow{\delta} & \dots \\
 & & \mathcal{I}^{(k)} \downarrow \uparrow \mathcal{R}^{(k)} & & \mathcal{I}^{(k+1)} \downarrow \uparrow \mathcal{R}^{(k+1)} & & \mathcal{I}^{(k+2)} \downarrow \uparrow \mathcal{R}^{(k+2)} & & \\
 \dots & \xrightarrow{d} & \Lambda_h^{(k)}(\Omega_h) & \xrightarrow{d} & \Lambda_h^{(k+1)}(\Omega_h) & \xrightarrow{d} & \Lambda_h^{(k+2)}(\Omega_h) & \xrightarrow{d} & \dots
 \end{array}$$

2.2. Splines and isogeometric analysis

Isogeometric analysis is used within this research as finite element discretization technique. This approach was introduced by Hughes et al. in [27] and covered more extensively by Cottrell et al. in [13]. Its premise is the following: Instead of using the normal basis consisting of local polynomial functions defined on each cell, a basis consisting of B-spline functions is used. One major advantage of this is the fact that geometries generated in Computer-Aided Design (CAD) software can be discretized exactly on (very) coarse meshes. CAD software uses B-splines and Non-Uniform Rational B-splines (NURBS) to describe continuous shapes with only a finite number of degrees of freedom; using the same functions in a finite element method allows one to describe numerical solutions and cell boundaries in terms of B-splines. The discrete representation of an object in CAD software can thus be taken as a direct input for generating both object-internal and -external meshes. An example of the former would be calculating stress distributions throughout an object subject to a load, while an example of the latter is calculating the fluid flow around said object. No local refinements are thus required for capturing the curvature of an object accurately.

A second advantage of isogeometric analysis over classical finite elements is the higher-order continuity found in numerical solutions and geometries. Whereas classical finite elements feature only C^0 inter-element continuity in both the numerical solutions and mesh boundaries, isogeometric analysis allows one to achieve higher-order continuity by simply using higher-order spline functions. Examples mentioned in [27] where this is especially advantageous are the structural analysis of thin-shell structures and compressible Euler flows; these both exhibit particular sensitivity to the smoothness of geometries.

Most spline functions $S(x)$ consists of a summation of compactly-supported basis functions:

$$S(x) = \sum_{i=1}^N s_i \psi_i(x) \quad (2.1)$$

Where $\psi_i(x)$ are the spline basis functions and s_i are the spline coefficients. Whereas the exact definition of each basis function type used within this research will vary, the support of each individual function depends directly on the definition of a finite set of discrete numbers called a knot vector.

Definition 2.9 (Piegl & Tiller [45, p. 50]). Let $\Xi = [\xi_0, \xi_1, \dots, \xi_m]$ be a non-decreasing sequence of numbers in \mathbb{R} ; then $\xi_i \leq \xi_{i+1}$, $i = 0, 1, \dots, m-1$. The numbers ξ_i are called **knots** while Ξ is the **knot vector**. The half-open intervals $[\xi_i, \xi_{i+1})$ are called **knot spans**. Note that knot repetition is allowed.

2.2.1. B-splines

Various ways exist that can be used to evaluate B-spline basis functions. Some of these are covered by Piegl & Tiller at the start of section 2.2 in [45].

Definition 2.10 (Piegl & Tiller [45, p. 50]). Let $\Xi = [\xi_0, \xi_1, \dots, \xi_m]$ be a knot vector. Then the i^{th} **B-spline basis function** of degree p at point $x \in [\xi_0, \xi_m]$ is denoted by $N_{i,p}(x)$ and defined as:

$$N_{i,0}(x) = \begin{cases} 1 & \text{if } x \in [\xi_i, \xi_{i+1}) \\ 0 & \text{otherwise} \end{cases}$$

$$N_{i,p}(x) = \frac{x - \xi_i}{\xi_{i+p} - \xi_i} N_{i,p-1}(x) + \frac{\xi_{i+p+1} - x}{\xi_{i+p+1} - \xi_{i+1}} N_{i+1,p-1}(x)$$

This algorithm for evaluating B-spline basis functions is called the **Cox-De Boor recursion method**. For the fractions used it is defined that $\frac{0}{0} = 0$.

B-spline basis function $N_{i,0}$ is thus piecewise-constant and nonzero only on knot span $[\xi_i, \xi_{i+1})$. Figure 2.3 shows basis functions of different degrees on the same uniform knot vector. As can be seen only $N_{0,0}$ is nonzero at $x = \xi_0 = 0$. The support of each basis function is equal to $p + 1$ knot spans; $N_{i,p}$ is nonzero on $[\xi_{i+1}, \xi_{i+p+2})$. Basis functions of degree p are calculated by taking linear combinations of two basis functions of degree $p - 1$, with the weights of both basis functions depending on the location x . Note that $N_{i,p}(x) \in [0, 1]$ and that the supports of successive basis functions overlap for $p > 0$. Moreover, the set of basis functions $\{N_{i,p}\}_{i=1}^{i=m-p}$ generated by knot vector $\Xi = [\xi_0, \xi_1, \dots, \xi_m]$ forms a partition of unity. That is, summing all basis functions at any point $x \in [\xi_0, \xi_m]$ gives a combined value of 1. Lastly it is noted that B-spline basis functions are C^∞ -continuous everywhere except at the knots; here C^{p-r} -continuity is achieved, with r the multiplicity of said knot.

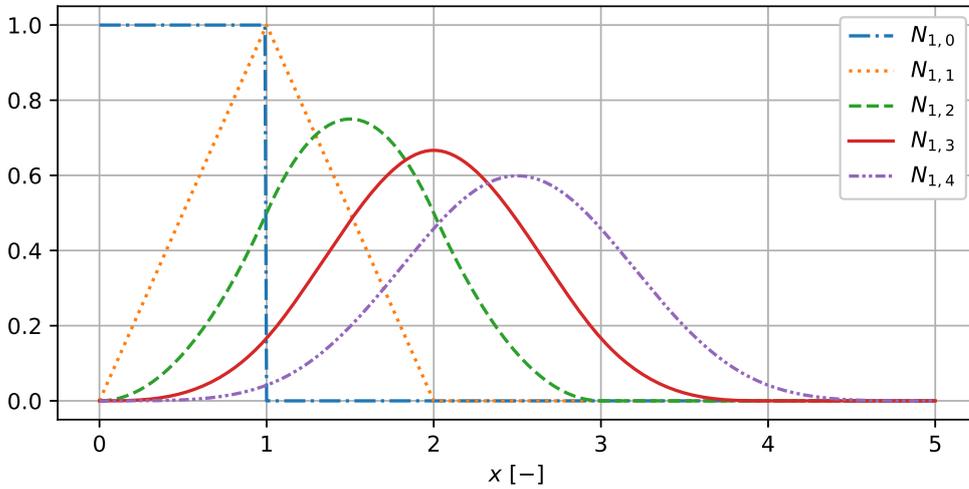


Figure 2.3: B-spline basis functions $N_{1,p}$ for different p on knot vector $\Xi = [0, 1, 2, 3, 4, 5]$

In a similar way to evaluating $N_{i,p}$ at location x its derivative can be determined:

Theorem 2.2.1 (Piegl & Tiller [45, p. 59]). Let $\Xi = [\xi_0, \xi_1, \dots, \xi_m]$ be a knot vector. Then the derivative of the i^{th} **B-spline basis function** of degree p at point $x \in [\xi_0, \xi_m]$ can be calculated with:

$$\frac{dN_{i,p}(x)}{dx} = \frac{p}{\xi_{i+p} - \xi_i} N_{i,p-1}(x) - \frac{p}{\xi_{i+p+1} - \xi_{i+1}} N_{i+1,p-1}(x)$$

The proof for this theorem can be found on page 59 of [45].

2.2.2. M-splines

A different kind of splines used as well in structure-preserving isogeometric analysis are the M-splines; these are based on B-splines as follows:

Definition 2.11 (Back & Sonnendrücker [1, p. 4]). *Let $\Xi = [\xi_0, \xi_1, \dots, \xi_m]$ be the knot vector that generates the set of B-spline basis functions $\{N_{i,p}\}_{i=1}^{i=m-p}$. The corresponding **M-spline basis functions** (also known as Curry-Schoenberg B-splines) are calculated as follows:*

$$M_{i,p}(x) = \frac{p+1}{\xi_{i+p+1} - \xi_i} N_{i,p}(x)$$

Note that M-splines are thus 'merely' scaled B-splines; this scaling results in their integral being equal to exactly 1. For knot vector $\Xi = [\xi_0, \xi_1, \dots, \xi_m]$:

$$\int_{\xi_i}^{\xi_{i+p+1}} M_{i,p}(x) dx = 1$$

Secondly, the derivative of a B-spline can be written exactly as linear combination of lower-order M-splines:

$$\frac{dN_{i,p}(x)}{dx} = \frac{p}{\xi_{i+p} - \xi_i} N_{i,p-1}(x) - \frac{p}{\xi_{i+p+1} - \xi_{i+1}} N_{i+1,p-1}(x) \quad (2.2)$$

$$= M_{i,p-1}(x) - M_{i+1,p-1}(x) \quad (2.3)$$

This property will come in useful when constructing discrete operators in section 2.3.

2.2.3. Non-Uniform Rational B-Splines

The third kind of spline used in this work is also related to the B-splines covered above. Whereas M-splines are simply a scaled version of B-splines, the Non-Uniform Rational B-Splines (NURBS) form a generalization. Recall from (2.1) that B-splines are a direct summation of products of expansion coefficients and (B-spline) basis functions. NURBS are formulated in a similar albeit slightly different way:

Definition 2.12 (Piegl & Tiller [45, p. 59]). *Let $\Xi = [\xi_0, \xi_1, \dots, \xi_m]$ be the knot vector that generates B-spline basis functions $\{N_{i,p}\}_{i=1}^{i=m-p}$. Then the Non-Uniform Rational B-Spline (NURBS) function $S(x)$ based on Ξ is:*

$$S(x) = \frac{\sum_{i=1}^N \frac{N_{i,p} s_i}{\sum_{j=1}^N N_{j,p} s_j}}$$

Where s_i are the spline coefficients. While this definition of NURBS specifically uses B-spline basis functions and the same spline functions are used in the numerator and denominator, further generalizations and variations of this structure (using M-splines or using different splines in the numerator and denominator) are used throughout this thesis; these are all referred to as NURBS.

2.2.4. Spline multiplication

A specific operation carried out on B- and M-splines in later chapters of this thesis is spline multiplication. From the recursive definition of B-spline basis functions it can already be seen that higher-degree functions are built up from lower-degree functions, and this is the case with products of B-splines as well. Since B-splines are defined through knot vectors, a product knot vector needs to be constructed:

Theorem 2.2.2 (Mørken [36, Eq. 3.1]). *Let $\Xi = [\xi_0, \xi_1, \dots, \xi_n]$, $\Theta = [\theta_0, \theta_1, \dots, \theta_m]$ be knot vectors defining the basis function sets $\{N_{i,p}\}_{i=1}^{i=n-p}$, $\{N_{j,q}\}_{j=1}^{j=m-q}$ respectively, such that:*

$$f(x) = \sum_{i=1}^{n-p} f_i N_{i,p}(x)$$

$$g(x) = \sum_{j=1}^{m-q} g_j N_{j,q}(x)$$

Let $\Xi^\circ = [\xi_0^\circ, \xi_1^\circ, \dots]$, $\Theta^\circ = [\theta_0^\circ, \theta_1^\circ, \dots]$ denote the knot vectors containing only unique knots (i.e. with the multiplicities reduced to one) and let $\Xi_c = [\xi_{c,0}, \xi_{c,1}, \dots]$, $\Theta_c = [\theta_{c,0}, \theta_{c,1}, \dots]$ denote the multiplicity vectors of Ξ , Θ ;

that is, Ξ and Θ can be constructed from repeating each knot in Ξ° , Θ° by the number of times indicated by the corresponding entry in Ξ_c , Θ_c .

Let $\Pi^\circ = \Xi^\circ \cup \Theta^\circ$ be the knot vector that contains all unique entries of Ξ° , Θ° . Then the entries $\pi_{c,k}$ of multiplicity vector Π_c are defined as follows:

$$\pi_{c,k} = \begin{cases} \max(p + \theta_{c,k}, q + \xi_{c,k}) & \text{if } \theta_{c,k} > 0 \text{ and } \xi_{c,k} > 0 \\ p + \theta_{c,k} & \text{if } \theta_{c,k} > 0 \text{ and } \xi_{c,k} = 0 \\ q + \xi_{c,k} & \text{if } \theta_{c,k} = 0 \text{ and } \xi_{c,k} > 0 \\ 0 & \text{if } \theta_{c,k} = 0 \text{ and } \xi_{c,k} = 0 \end{cases}$$

In other words, the multiplicity of each knot is increased by the amount necessary to maintain the same local degree of continuity in the product spline $f(x)g(x)$ as is found in the separate splines $f(x)$ and $g(x)$. Knot vector Π can be constructed from Π° and Π_c .

Generating knot vector Π from Ξ and Θ like this allows us to compute the product $f(x)g(x)$ pointwise-exact as a higher-degree B-spline.

Theorem 2.2.3 (Mørken [36, Sec. 5]). *Let $f(x)$, $g(x)$ be B-splines functions with knot vectors Ξ , Θ and bases $\{N_{i,p}\}_{i=1}^{i=n-p}$, $\{N_{j,q}\}_{j=1}^{j=m-q}$ respectively. Also let $h(x) = f(x)g(x)$ be the product of these two functions. Then the knot vector of $h(x)$, $\Pi = [\pi_0, \pi_1, \dots, \pi_l]$, can be found through theorem 2.2.2. This knot vector generates basis $\{N_{k,p+q}\}_{k=1}^{k=l-(p+q)}$ such that:*

$$\begin{aligned} f(x) &= \sum_{i=1}^{n-p} f_i N_{i,p}(x) \\ g(x) &= \sum_{j=1}^{m-q} g_j N_{j,q}(x) \\ h(x) &= \sum_{k=1}^{l-(p+q)} h_k N_{k,p+q}(x) \end{aligned}$$

The coefficients h_k are defined as:

$$h_k = \sum_{i,j} f_i g_j \Gamma_{i,j,p,q}(k)$$

With Γ computed from the following recurrence relations:

$$\begin{aligned} \Gamma_{i,j,p,q}(k) &= \frac{p-1}{p+q-1} [\omega_{i,p}(\pi_{k+p+q-1}) \Gamma_{i,j,p-1,q}(k) + (1 - \omega_{i+1,p}(\pi_{k+p+q-1})) \Gamma_{i+1,j,p-1,q}(k)] \\ &\quad + \frac{q-1}{p+q-1} [\omega_{j,q}(\pi_{k+p+q-1}) \Gamma_{i,j,p,q-1}(k) + (1 - \omega_{j+1,q}(\pi_{k+p+q-1})) \Gamma_{i+1,j,p,q-1}(k)] \end{aligned}$$

Where:

$$\Gamma_{i,j,1,q}(k) = \alpha_{i,1}(k) \alpha_{j,q}(k)$$

$$\Gamma_{i,j,p,1}(k) = \alpha_{i,p}(k) \alpha_{j,1}(k)$$

$$\alpha_{i,p}(k) = \omega_{i,p}(\pi_{k+p+q-1}) \alpha_{i,p-1}(k) + (1 - \omega_{i+1,p}(\pi_{k+p+q-1})) \alpha_{i+1,p-1}(k)$$

And lastly:

$$\begin{aligned} \alpha_{i,0}(k) &= N_{i,0}(\pi_k) \\ \omega_{i,p}(x) &= \begin{cases} \frac{x - \pi_i}{\pi_{i+p-1} - \pi_i} & \text{if } \pi_i < \pi_{i+p-1} \\ 0 & \text{otherwise} \end{cases} \end{aligned}$$

The resulting function $h(x)$ is pointwise-equal to the product of B-splines $f(x)$ and $g(x)$ and has a B-spline description with its own knot vector and function basis of degree $p+q$.

2.2.5. Open and periodic B-spline bases

As mentioned before, knot vector Ξ and degree p define a function basis $\{N_{i,p}\}$. Within this research two kinds of knot vectors are used, leading to two function bases with different properties. The first of these is the open knot vector:

Definition 2.13. Let Ξ be a knot vector and $p \in \mathbb{N}$, such that:

$$\Xi = [\underbrace{\xi_0, \xi_1 = \xi_0, \dots, \xi_p = \xi_0}_{\xi_0 \text{ repeated } p+1 \text{ times}}, \xi_{p+1}, \dots, \xi_{m-p-1}, \underbrace{\xi_{m-p} = \xi_m, \dots, \xi_{m-1} = \xi_m, \xi_m}_{\xi_m \text{ repeated } p+1 \text{ times}}]$$

If this happens, i.e. if the first $p+1$ knots of Ξ are equal to one another and the same holds for the last $p+1$ knots, then Ξ is an **open knot vector**.

Note that knot vectors are thus open only for a specific basis function degree p . Recall from figure 2.3 that only $p=0$ leads to basis functions that are nonzero on the left-side boundary of knot vector Ξ . With an open knot vector the associated B-spline function basis $\{N_{i,p}\}$ has nonzero values on $\partial\Xi = [\xi_0, \xi_m]$. Figure 2.4 shows such a function basis for $p=2$; as can be seen the first basis function $N_{1,2}$ is nonzero at $x = \xi_0 = 0$, whereas the last basis function $N_{5,2}$ is nonzero at $x = \xi_m = 3$.

Using a spline basis with an open knot vector thus allows one to have nonzero spline values at both ends

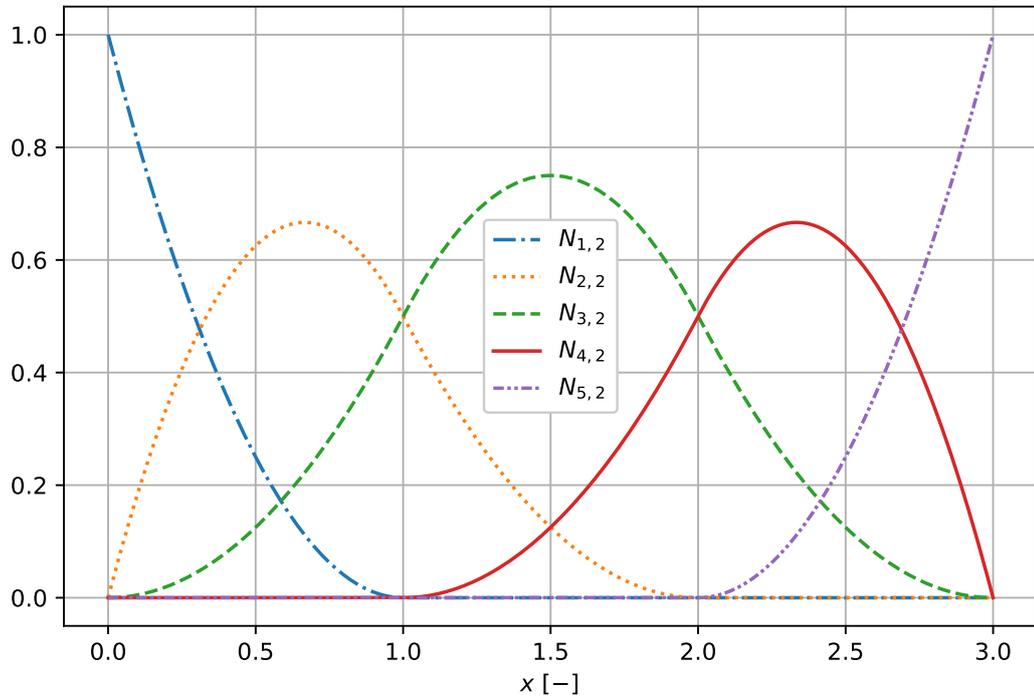


Figure 2.4: B-spline basis functions $\{N_{i,2}\}$ on open knot vector $\Xi = [0, 0, 0, 1, 2, 3, 3, 3]$

of a one-dimensional domain. The exact spline multiplication algorithm covered in the previous section is valid for open knot vectors as well. This spline basis is used in chapter 7 to generate numerical solutions to Sod's shock tube problem, for which nonzero mass and energy densities are used on both ends of a one-dimensional domain.

In addition to bases generated from open knot vectors, a periodic basis will be used for simulations on periodic domains. The knot vector that generates this basis is defined as follows:

Definition 2.14. Let Ξ be a knot vector and let $\Omega = [x_l, x_r]$ be a one-dimensional periodic domain. Ξ is a **periodic knot vector** if:

$$\Xi = [\xi_0 = -\frac{(x_r - x_l)p}{n} + x_l, \xi_1 = \xi_0 + \Delta\xi, \xi_2 = \xi_1 + \Delta\xi, \dots, \xi_{m-1} = \xi_m - \Delta\xi, \xi_m = x_r]$$

Where n is the number of elements in the resulting basis, p the degree of the basis functions and:

$$\Delta\xi = \frac{x_r + \frac{(x_r - x_l)p}{n}}{n - 1}$$

The resulting spline basis of degree p generated by Ξ is periodic across $\partial\Omega$.

This periodic knot vector is not the only ingredient necessary for obtaining a periodic basis: The basis functions themselves have to be modified as well. This is a fairly straightforward operation:

Definition 2.15. Let Ξ be a periodic knot vector with periodic domain $\Omega = [x_l, x_r]$ and let $\{N_{i,p}\}$ be the (regular) B-spline basis generated by Ξ . Then the **periodic B-spline basis** can be generated by the following basis function map:

$$N_{i,p}(x) \rightarrow N_{i,p}(x) + N_{i,p}(x - x_r)$$

The same map can be implemented for M-splines as well, replacing $N_{i,p}$ with $M_{i,p}$.

The result of using a periodic knot vector with the periodic basis map is a set of basis functions that are periodic across a certain domain. Note that the knot vector extends outside of this domain on one side. Figure 2.5 shows such a periodic basis. Note that with this basis the spline values on the edges of the periodic domain $([0, 1])$ in figure 2.5) are again nonzero for all degrees p .

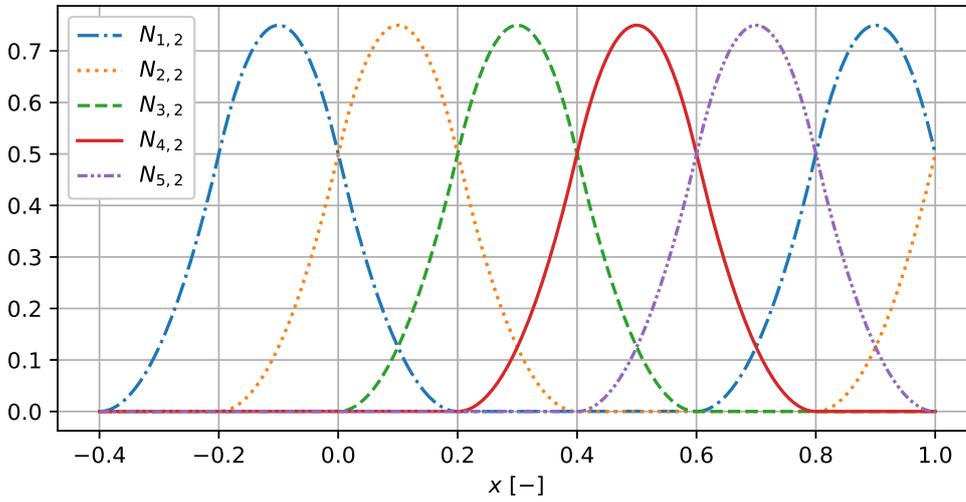


Figure 2.5: Periodic B-spline basis functions $\{N_{i,2}\}$ on periodic knot vector $\Xi = [-\frac{2}{5}, -\frac{1}{5}, 0, \frac{1}{5}, \frac{2}{5}, \frac{3}{5}, \frac{4}{5}, 1]$; the resulting basis is periodic on interval $[0, 1]$

2.2.6. Mimetic B-splines

Mimetic isogeometric analysis is not an entirely new concept: Among others Back & Sonnendrücker [1, 5], Buffa et al. [10, 11], Evans & Hughes [14], Hiemstra [25] and Janssen [30] already looked into mimetic isogeometric methods for a range of continuum physics applications. While the various discrete operators that are used in this research are introduced in section 2.3, this section focuses on the spline functions used to discretize the various differential forms that occur. Let Ω be a manifold in \mathbb{R} ; it thus supports differential 0- and 1-form. Let Ξ be a knot vector of length $m + p + 1$ that generates B-spline basis $\{N_{i,p}\}_{i=1}^{i=m+1}$ of degree p . The space of discrete 0-forms $\Lambda_{h,p}^{(0)}(\Omega_h) \subset \Lambda^{(0)}(\Omega)$ is then defined as:

$$\Lambda_{h,p}^{(0)}(\Omega_h) := \text{Span}\{N_{i,p} : i = 1, 2, \dots, m + 1\}$$

Although not directly reflected in this notation, $\Lambda_{h,p}^{(0)}(\Omega_h)$ also depends on knot vector Ξ . This knot vector also generates the space of discrete 1-forms $\Lambda_{h,p-1}^{(1)}(\Omega_h) \subset \Lambda^{(1)}(\Omega_h)$:

$$\Lambda_{h,p-1}^{(1)}(\Omega_h) := \text{Span}\{M_{i,p-1} : i = 2, \dots, m + 1\}$$

The choice for modeling $\Lambda_{h,p-1}^{(1)}(\Omega_h)$ in this way with M-splines of degree $p-1$ is not expanded upon in this section; instead it will come up naturally when the discrete exterior derivative operator is introduced in section 2.3.2.

Discrete 0- and 1-forms $\alpha_h^{(0)} \in \Lambda_{h,p}^{(0)}(\Omega_h)$ and $\beta_h^{(1)} \in \Lambda_{h,p-1}^{(1)}(\Omega_h)$ are generated by defining vectors of expansion coefficients α and β of appropriate lengths and associating each expansion coefficient to one of the basis functions of $\Lambda_{h,p}^{(0)}(\Omega_h)$, $\Lambda_{h,p-1}^{(1)}(\Omega_h)$ respectively.

Now let Ω be a manifold in \mathbb{R}^2 , with knot vectors $\Xi(x)$ (of length $m+p+1$) and $\Theta(y)$ (of length $l+q+1$); now Ω supports 0-, 1- and 2-forms. The one-dimensional basis function definitions can simply be extended to two dimensions by using a tensor product structure:

$$\begin{aligned}\alpha_h^{(0)}(x, y) &= \sum_i \sum_j \alpha_{ij} N_{i,p}(x) N_{j,q}(y) \\ \beta_h^{(1)}(x, y) &= \beta_h^{(1,x)} + \beta_h^{(1,y)} \\ &= \sum_i \sum_j \beta_{ij}^x M_{i,p-1}(x) N_{j,q}(y) dx \\ &\quad + \sum_i \sum_j \beta_{ij}^y N_{i,p}(x) M_{j,q-1}(y) dy \\ \gamma_h^{(2)}(x, y) &= \sum_i \sum_j \gamma_{ij} M_{i,p-1}(x) M_{j,q-1}(y) dx \wedge dy\end{aligned}$$

And thus the spaces of discrete differential forms are defined by:

$$\begin{aligned}\Lambda_{h,p,q}^{(0)}(\Omega_h) &= \text{Span}\{N_{i,p} : i = 1, 2, \dots, m+1\} \times \text{Span}\{N_{j,q} : j = 1, 2, \dots, l+1\} \\ \Lambda_{h,p-1,q}^{(1,x)}(\Omega_h) &= \text{Span}\{M_{i,p-1} : i = 2, \dots, m+1\} \times \text{Span}\{N_{j,q} : j = 1, 2, \dots, l+1\} \\ \Lambda_{h,p,q-1}^{(1,y)}(\Omega_h) &= \text{Span}\{N_{i,p} : i = 1, 2, \dots, m+1\} \times \text{Span}\{M_{j,q-1} : j = 2, \dots, l+1\} \\ \Lambda_{h,p,q}^{(2)}(\Omega_h) &= \text{Span}\{M_{i,p-1} : i = 2, \dots, m+1\} \times \text{Span}\{M_{j,q-1} : j = 1, 2, \dots, l+1\}\end{aligned}$$

This structure generalizes for manifolds in \mathbb{R}^n , but this is outside of the cases that are presented in this research. Note that the discrete 1-form space is a combination $\Lambda_{h,p-1,q}^{(1,x)}(\Omega_h) \oplus \Lambda_{h,p,q-1}^{(1,y)}(\Omega_h)$ of the 1-form space generated by each dual basis element.

To simplify the notation of basis functions a unified notation is introduced. All basis functions are denoted by $\psi_i^{(k)}$, where i is the index in the corresponding basis; in case of multidimensional problems lexicographic ordering is used (keeping the x -index constant while looping over the y -index). The dimensionality of any problem is apparent from context, and hence no notational differences are made between basis functions in one or two dimensions unless required for putting emphasis on certain things. This single-index convention lends itself to vector notation as well. For Ω in \mathbb{R}^2 :

$$\begin{aligned}\alpha_h^{(0)}(x, y) &= \sum_i \alpha_i \psi_i^{(0)} = \alpha^T \boldsymbol{\psi}^{(0)} \\ \beta_h^{(1)}(x, y) &= \beta_h^{(1,x)} + \beta_h^{(1,y)} = \sum_i \beta_i^x \psi_i^{(1,x)} dx + \sum_i \beta_i^y \psi_i^{(1,y)} dy \\ &= (\boldsymbol{\beta}^x)^T \boldsymbol{\psi}^{(1,x)} + (\boldsymbol{\beta}^y)^T \boldsymbol{\psi}^{(1,y)} \\ \gamma_h^{(2)}(x, y) &= \sum_i \gamma_i \psi_i^{(2)} dx \wedge dy = \boldsymbol{\gamma}^T \boldsymbol{\psi}^{(2)} dx \wedge dy\end{aligned}$$

While here the dual basis functions dx^i are explicitly shown they will often be left out in order to simplify and shorten the notation used in this thesis. They will be included only when relevant to the discussion or when to clarify a certain construction. An example of this is section 2.3.3, where the discrete Hodge \star operator is covered.

No explicit connection has been made up to now between the cochain complexes covered in section 2.1.3 and the mimetic B-splines covered here. As covered in more detail by Janssen in [30, p. 43], a dual B-spline basis

can be determined such that multiplying any B-spline basis function with its dual results in the Kronecker delta δ_i^j . This allows one to define reduction and reconstruction operators \mathcal{I} and \mathcal{R} that map between the cochain spaces $C^{(k)}(\Omega_h)$ and B-spline spaces $\Lambda_h^{(k)}(\Omega_h)$.

2.3. Discrete operators

Now that the relevant theory has been covered and isogeometric analysis has been introduced, the discrete operators that have been used in this research can be covered. Only the operators that are present in the test cases presented in chapters 5-8 are covered here. A couple of examples are given for each operator; most of these correspond to how they are used in later chapters.

Section 2.3.1 first goes over how the overall finite element discretization is carried out and how this leads to the definition of so-called mass matrices. Independent from this the discrete exterior derivative is defined in section 2.3.2 such that it is exact and topological, by exploiting the relation of discrete 0- and 1-form spaces. At this point metric concepts have to be introduced, starting with the Hodge \star operator. This discrete metric operator is defined in accordance with the aforementioned mass matrices. Lastly the discrete interior product is covered in section 2.3.4, based on which the discrete Lie derivative is introduced in section 2.3.5; through Cartan's homotopy formula the Lie derivative is defined as a composition of the discrete interior product and exterior derivative.

The numerical entries of the various matrices used throughout this work to represent differential operators can be computed with appropriate numerical integration rules. In this research Gaussian quadrature rules were used.

2.3.1. Weak forms and mass matrices

In this research a finite element (isogeometric) discretization approach is used. A number of works that use the same approach have been given in section 2.2.6. Some works that cover the finite element method in general are [8, 57]. In the finite element approach governing models defined on domain Ω are multiplied with a test function and integrated over Ω . Let \mathcal{P} denote such a (linear) model, acting on differential form $\alpha^{(k)}$ such that:

$$\mathcal{P}\alpha^{(k)} = 0$$

Let $\beta^{(k)}$ be an arbitrary discrete k -form, referred to as a test function. Multiplying after applying the Hodge \star to $\beta^{(k)}$ gives, after integration over Ω :

$$\int_{\Omega} \mathcal{P}\alpha^{(k)} \wedge \star\beta^{(k)} = 0 \quad (2.4)$$

This is the weak form of the governing model. Solutions that satisfy the weak form are called weak solutions whereas solutions to the original model are called strong solutions, since they satisfy said model in a point-wise fashion. Note that strong solutions are necessarily also weak solutions, but not every weak solution is a strong solution. Let Ω_h denote a cell complex on Ω . A basis function expansion is used to project $\alpha^{(k)}$ to a finite-dimensional space:

$$\alpha_h^{(k)} = \sum_{i=1}^n \alpha_i \psi_i^{(k)}$$

Where the subscript h denotes a discretized version of $\alpha^{(k)}$. In isogeometric analysis B-splines are used for this, whereas classic finite elements use polynomial functions with compact, local supports and spectral element methods use functions with a global support (for example, trigonometric functions). Applying this decomposition to (2.4) gives:

$$\int_{\Omega_h} \mathcal{P}\alpha_h^{(k)} \wedge \star\beta^{(k)} = \sum_{i=1}^n \alpha_i \int_{\Omega_h} \mathcal{P}\psi_i^{(k)} \wedge \star\beta^{(k)} = 0$$

In the Bubnov-Galerkin finite element method the test function is expanded in the same basis that was used for the decomposition of $\alpha^{(k)}$. Expanding $\beta^{(k)}$ in the same way gives:

$$\sum_{i=1}^n \alpha_i \int_{\Omega_h} \mathcal{P}\psi_i^{(k)} \wedge \star\beta_h^{(k)} = \sum_{j=1}^n \beta_j \sum_{i=1}^n \alpha_i \int_{\Omega_h} \mathcal{P}\psi_i^{(k)} \wedge \star\psi_j^{(k)} = 0$$

A unique solution to this problem with n unknowns can be found by using n linearly independent test functions; letting the test function coefficient $\beta_j = 1$ for a single $j \in \{1, 2, \dots, n\}$ with $\beta_l = 0$ for $l \neq j$ allows one to consider each individual basis element $\psi_j^{(k)}$ as separate test function. Doing this results in a system of equations with one equation for each $\psi_j^{(k)}$. This results in the following linear system:

$$\begin{bmatrix} \int_{\Omega_h} \mathcal{P}\psi_1^{(k)} \wedge \star\psi_1^{(k)} & \int_{\Omega_h} \mathcal{P}\psi_2^{(k)} \wedge \star\psi_1^{(k)} & \dots & \int_{\Omega_h} \mathcal{P}\psi_n^{(k)} \wedge \star\psi_1^{(k)} \\ \int_{\Omega_h} \mathcal{P}\psi_1^{(k)} \wedge \star\psi_2^{(k)} & \int_{\Omega_h} \mathcal{P}\psi_2^{(k)} \wedge \star\psi_2^{(k)} & \dots & \int_{\Omega_h} \mathcal{P}\psi_n^{(k)} \wedge \star\psi_2^{(k)} \\ \vdots & \vdots & \ddots & \vdots \\ \int_{\Omega_h} \mathcal{P}\psi_1^{(k)} \wedge \star\psi_n^{(k)} & \int_{\Omega_h} \mathcal{P}\psi_2^{(k)} \wedge \star\psi_n^{(k)} & \dots & \int_{\Omega_h} \mathcal{P}\psi_n^{(k)} \wedge \star\psi_n^{(k)} \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_n \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Which can be solved for coefficient vector α , thereby defining numerical solution $\alpha_h^{(k)}$ that weakly satisfies $\mathcal{P}\alpha_h^{(k)} = 0$.

Governing model \mathcal{P} often contains multiple terms. In the process of setting up a finite element discretization one can expect to find matrices whose entries consist of an integral of products $\psi_i^{(k)} \wedge \star\psi_i^{(k)}$:

$$\mathbb{M}^{(k)} = \begin{bmatrix} \int_{\Omega_h} \psi_1^{(k)} \wedge \star\psi_1^{(k)} & \int_{\Omega_h} \psi_2^{(k)} \wedge \star\psi_1^{(k)} & \dots & \int_{\Omega_h} \psi_n^{(k)} \wedge \star\psi_1^{(k)} \\ \int_{\Omega_h} \psi_1^{(k)} \wedge \star\psi_2^{(k)} & \int_{\Omega_h} \psi_2^{(k)} \wedge \star\psi_2^{(k)} & \dots & \int_{\Omega_h} \psi_n^{(k)} \wedge \star\psi_2^{(k)} \\ \vdots & \vdots & \ddots & \vdots \\ \int_{\Omega_h} \psi_1^{(k)} \wedge \star\psi_n^{(k)} & \int_{\Omega_h} \psi_2^{(k)} \wedge \star\psi_n^{(k)} & \dots & \int_{\Omega_h} \psi_n^{(k)} \wedge \star\psi_n^{(k)} \end{bmatrix}$$

$\mathbb{M}^{(k)}$ is called a mass matrix; these are used throughout this research. The Hodge \star appearing here is covered in section 2.3.3. Discrete operators can map $\alpha_h^{(k)}$ to the space of $k-1$ -, k - or $k+1$ -forms, but they do not interact in any way with the test functions nor with the integral over Ω_h . Hence mass matrices are can be encountered in finite element formulations after discrete operators have been applied. Note that multiple mass matrices are defined on cell complex Ω_h , and thus a unique mass matrix exists for every type of differential k -form that is supported by Ω_h .

2.3.2. Exterior derivative

As was alluded to in section 2.2.6 the spline spaces of discrete differential forms can be defined such that the discrete exterior derivative is exact. Let $\alpha_h^{(0)} \in \Lambda_{h,p}^{(0)}(\Omega_h)$ with Ω_h a covering of $\Omega \subset \mathbb{R}$, such that:

$$\alpha_h^{(0)}(x) = \sum_i \alpha_i N_{i,p}(x)$$

Then:

$$d\alpha_h^{(0)}(x) = \sum_i \alpha_i \frac{\partial N_{i,p}(x)}{\partial x} dx$$

And as can be seen in (2.2), this derivative of B-spline basis functions is exactly equal to:

$$d\alpha_h^{(0)}(x) = \sum_i \alpha_i (M_{i,p-1}(x) - M_{i+1,p-1}(x)) dx$$

The 1-form spline space that is thus necessary for having an exact relation $\beta_h^{(1)} = d\alpha_h^{(0)}$ is $\Lambda_{h,p-1}^{(1)}(\Omega_h)$. Then:

$$\beta_h^{(1)} = \sum_{j=2}^n \beta_j M_{j,p-1}(x) dx = d\alpha^{(0)} = \sum_{i=1}^{n-1} \alpha_i (M_{i,p-1}(x) - M_{i+1,p-1}(x)) dx = \sum_{i=2}^n (\alpha_i - \alpha_{i-1}) M_{i,p-1}(x) dx$$

Thus $\beta_j = \alpha_j - \alpha_{j-1}$ ensures that $\beta_h^{(1)} = d\alpha_h^{(0)}$. This relation is encoded in incidence matrix $\mathbb{E}^{(1,0)}$ in the exact same way as is the case for the coboundary operator acting on 0-cochains. With this specific choice of discrete spline spaces the exterior derivative thus maps exactly between the discrete spline spaces of 0- and 1-forms; the expansion coefficients of $\alpha_h^{(0)}$ correspond to a cochain and are mapped into $C^{(1)}(\Omega_h)$ by the coboundary operator. Recall that the coboundary operator δ is the adjoint of the boundary operator ∂ , which is a purely topological operator that only takes into account adjacency of the different chains present in cell complex

Ω_h ; it is exact even for highly deformed meshes. Due to the tensor product structure of the multidimensional discrete spline spaces the exterior derivative can be made to map in this way between the discrete k - and $k+1$ -form spline spaces exactly, with an incidence matrix $\mathbb{E}^{(k+1,k)}$ that is equal to the transpose of the boundary operator ∂ mapping between chain spaces $C_{k+1}(\Omega_h) \rightarrow C_k(\Omega_h)$. The connection between k -cochains and discrete k -forms that was covered in section 2.2.6 shows that the incidence matrices conserve the relations between the spaces of discrete k - and $k-1$ -forms.

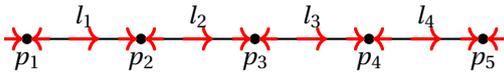
This can also be seen directly by applying some of the commutative relations between the various operators of section 2.1.2 to the adjoint relation between δ and ∂ :

$$\underbrace{\langle \mathcal{R}^{k+1} d\alpha_h^{(k)}, c_{(k+1)} \rangle}_{=\delta \mathcal{R}^k \alpha^{(k)}} = \langle \delta \mathcal{R}^k \alpha_h^{(k)}, c_{(k+1)} \rangle = \langle \underbrace{\mathcal{R}^k \alpha_h^{(k)}}_{=\alpha}, \partial c_{(k+1)} \rangle$$

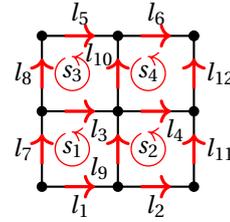
Example 8. As mentioned extensively the action of the exterior derivative on a discrete k -form $\alpha_h^{(k)}$ can be discretized by letting incidence matrix $\mathbb{E}^{(k+1,k)}$ act on its cochain vector and noting that the resulting spline is a discrete $k+1$ -form:

$$d\alpha_h^{(k)} = d(\boldsymbol{\psi}^{(k)})^T \boldsymbol{\alpha} = (\boldsymbol{\psi}^{(k)})^T \mathbb{E}^{(k+1,k)} \boldsymbol{\alpha}$$

Consider the orientations of 0- and 1-cochains in \mathbb{R}^1 shown in figure 2.6a and 1- and 2-cochains in \mathbb{R}^2 shown in figure 2.6b. The incidence matrix for the 0-cochain vector $[p_1 \ p_2 \ p_3 \ p_4 \ p_5]^T$ shown in figure 2.6a then



(a) Inner orientations of 0- and 1-cochains in \mathbb{R}^1



(b) Inner orientations of 1- and 2-cochains in \mathbb{R}^2

Figure 2.6: Example cochains with inner orientations indicated

is $\mathbb{E}^{(1,0)} \in \mathbb{R}^{4 \times 5}$:

$$\mathbb{E}^{(1,0)} = \begin{bmatrix} -1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 \end{bmatrix}$$

Which maps from the aforementioned 0-cochain to 1-cochain $[l_1 \ l_2 \ l_3 \ l_4]^T$. Note that each column corresponding to an interior node and every row (each corresponding to a line section) contain a single 1 and a single -1 . Similarly, the incidence matrix for the 1-cochain $[l_1 \ l_2 \ \dots \ l_{11} \ l_{12}]^T$ shown in figure 2.6b is $\mathbb{E}^{(2,1)} \in \mathbb{R}^{4 \times 12}$:

$$\mathbb{E}^{(2,1)} = \begin{bmatrix} 1 & 0 & -1 & 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & -1 & 0 & 0 & -1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & -1 & 0 & 1 \end{bmatrix}$$

Which thus maps the 1-cochain to 2-cochain $[s_1 \ s_2 \ s_3 \ s_4]^T$. Every column now corresponds to a line section and every row corresponds to a surface. Each column contains at most two nonzero entries (one positive, one negative) while every row contains precisely four nonzero entries, two of which are positive and two of which are negative.

It is worth emphasizing at this point that the discrete spline spaces for 0- and 1-forms have been explicitly defined such that the exterior derivative can be discretized exactly.

2.3.3. Hodge \star operator

While defining the mass matrices in section 2.3.1 the Hodge \star operator was encountered. For example, when constructing $\mathbb{M}^{(k)}$ on manifold $\Omega \subset \mathbb{R}^n$ with cell complex Ω_h :

$$\mathbb{M}^{(k)} = \int_{\Omega_h} \left[\left(\boldsymbol{\psi}^{(k)} \right)^T dx^{i_1} \wedge dx^{i_2} \wedge \dots \wedge dx^{i_k} \right] \wedge \star \left[\boldsymbol{\psi}^{(k)} dx^{i_1} \wedge dx^{i_2} \wedge \dots \wedge dx^{i_k} \right]$$

In these cases the discrete Hodge \star operator is taken to act like its continuous counterpart, in the sense that it maps k -forms to their Hodge-dual $n - k$ -forms and does not affect the values of the underlying description of a discrete k -form. For example, mapping $\alpha_h^{(1)} = \boldsymbol{\alpha}^T \boldsymbol{\psi}^{(1)} dx$ on $\Omega \subset \mathbb{R}$ to its dual 0-form gives:

$$\star \alpha_h^{(1)} = \star \boldsymbol{\alpha}^T \boldsymbol{\psi}^{(1)} dx = \tilde{\boldsymbol{\alpha}}^T \tilde{\boldsymbol{\psi}}^{(1)}$$

Where $\tilde{\boldsymbol{\alpha}}$ and $\tilde{\boldsymbol{\psi}}$ are the Hodge-dual 0-form coefficients and basis functions respectively. Pointwise it holds that $\tilde{\boldsymbol{\alpha}}^T \tilde{\boldsymbol{\psi}}^{(1)} = \boldsymbol{\alpha}^T \boldsymbol{\psi}^{(1)}$. Instead of computing the Hodge-dual coefficients and basis functions the original description $\boldsymbol{\alpha}^T \boldsymbol{\psi}^{(1)}$ is used here. Hence mass matrix $\mathbb{M}^{(k)}$ has the following entries:

$$\begin{aligned} \mathbb{M}^{(k)} &= \int_{\Omega_h} \left[\left(\boldsymbol{\psi}^{(k)} \right)^T dx^{i_1} \wedge dx^{i_2} \wedge \dots \wedge dx^{i_k} \right] \wedge \star \left[\boldsymbol{\psi}^{(k)} dx^{i_1} \wedge dx^{i_2} \wedge \dots \wedge dx^{i_k} \right] \\ &= \sqrt{g} \int_{\Omega_h} \left[\left(\boldsymbol{\psi}^{(k)} \right)^T dx^{i_1} \wedge dx^{i_2} \wedge \dots \wedge dx^{i_k} \right] \wedge \left[\boldsymbol{\psi}^{(k)} dx^{i_{k+1}} \wedge dx^{i_{k+2}} \wedge \dots \wedge dx^{i_n} \right] \\ &= \text{sgn} \sqrt{g} \int_{\Omega_h} \left(\boldsymbol{\psi}^{(k)} \right)^T \boldsymbol{\psi}^{(k)} dx^{i_1} \wedge dx^{i_2} \wedge \dots \wedge dx^{i_n} \end{aligned}$$

Where $\text{sgn} = \pm 1$ depending on the order of the dual basis elements dx^i , and \sqrt{g} is the square root of the determinant of metric tensor g . Note that in this research $\text{sgn} = 1$ and $\sqrt{g} = 1$.

Using this discrete description of the Hodge \star does come at a certain cost: It cannot be used to easily construct a double discrete De Rham complex since part of the commutativity of the Hodge \star with the exterior derivative is lost. For example, for $\Lambda_{h,p+1}^{(0)}(\Omega_h), \Lambda_{h,p}^{(1)}(\Omega_h)$ on cell complex Ω_h covering $\Omega \subset \mathbb{R}$:

$$\begin{array}{ccc} \Lambda_{h,p+1}^{(0)}(\Omega_h) & \xrightarrow{d} & \Lambda_{h,p}^{(1)}(\Omega_h) \\ \downarrow \star & & \downarrow \star \\ \tilde{\Lambda}_{h,p+1}^{(1)}(\Omega_h) & & \tilde{\Lambda}_{h,p}^{(0)}(\Omega_h) \end{array}$$

In other words, the discrete Hodge \star maps discrete 0-forms to Hodge-dual 1-forms and discrete 1-forms to Hodge-dual 0-forms. While the discrete exterior derivative d maps between the spaces of discrete 0- and 1-forms, this construction was made by selecting conforming bases for the spline spaces. It remains to be seen whether this arrangement extends to the Hodge-dual 0- and 1-form spaces in a consistent way. This is not an issue within the current research, since all discrete exterior derivatives are imposed on Hodge-primal k -forms. Thus this loss of structure when discretizing the Hodge \star operator in the current way does not impact the resulting numerical methods.

2.3.4. Interior product

For the interior product a different approach is used. As shown by Hirani [26, Lemma 8.2.1], for vector field \boldsymbol{v}_h and $\alpha^{(k)} \in \Lambda^{(k)}(\Omega)$:

$$i_{\boldsymbol{v}_h} \alpha_h^{(k)} = (-1)^{k(n-k)} \star \left(\star \alpha_h^{(k)} \wedge \boldsymbol{v}_h^b \right) \quad (2.5)$$

Based on this, Kreeft remarked that the interior product is adjoint to the wedge product. For $\beta^{(k-1)} \in \Lambda^{(k-1)}(\Omega)$:

$$\langle i_{\boldsymbol{v}_h} \alpha_h^{(k)}, \beta_h^{(k-1)} \rangle = \langle \alpha_h^{(k)}, \beta_h^{(k-1)} \wedge \boldsymbol{v}_h^b \rangle \quad (2.6)$$

Even though $\beta_h^{(k-1)} \wedge \boldsymbol{v}_h^b \in \Lambda^{(k)}(\Omega)$, using the mimetic spline spaces defined in section 2.2.6 does not give guarantees that $\beta_h^{(k-1)} \wedge \boldsymbol{v}_h^b \in \Lambda_{h,p}^{(k)}(\Omega)$ for any degree p . Instead a weak form based on (2.5) and (2.6) is used

to ensure that $i_{\mathbf{v}_h}$ maps between the spaces of discrete k - and $k-1$ -forms. For this mapping to behave in this way it is required that:

$$i_{\mathbf{v}_h} \alpha_h^{(k)} = i_{\mathbf{v}_h} \sum_i \alpha_i \psi_i^{(k)} = \gamma_h^{(k-1)} = \sum_j \gamma_j \psi_j^{(k-1)}$$

By integrating (2.6) over cell complex Ω_h it follows that:

$$\begin{aligned} \int_{\Omega_h} \langle i_{\mathbf{v}_h} \alpha_h^{(k)}, \beta_h^{(k-1)} \rangle &= \int_{\Omega_h} \langle \alpha_h^{(k)}, \beta_h^{(k-1)} \wedge \mathbf{v}_h^b \rangle \\ \int_{\Omega_h} \underbrace{i_{\mathbf{v}_h} \alpha_h^{(k)}}_{=\gamma_h^{(k-1)}} \wedge \star \beta_h^{(k-1)} &= \int_{\Omega_h} \alpha_h^{(k)} \wedge \star (\beta_h^{(k-1)} \wedge \mathbf{v}_h^b) \\ \int_{\Omega_h} \beta^T \psi^{(k-1)} (\psi^{(k-1)})^T \boldsymbol{\gamma} &= \int_{\Omega_h} \mathbf{v}_h^b \beta^T \psi^{(k-1)} (\psi^{(k-1)})^T \boldsymbol{\alpha} \\ \beta^T \underbrace{\int_{\Omega_h} \psi^{(k-1)} (\psi^{(k-1)})^T \boldsymbol{\gamma}}_{=\mathbb{M}^{(k-1)}} &= \beta^T \underbrace{\int_{\Omega_h} \mathbf{v}_h^b \psi^{(k-1)} (\psi^{(k-1)})^T \boldsymbol{\alpha}}_{=\mathbb{C}_{\mathbf{v}_h}^{(k-1,k)}} \\ \beta^T \mathbb{M}^{(k-1)} \boldsymbol{\gamma} &= \beta^T \mathbb{C}_{\mathbf{v}_h}^{(k-1,k)} \boldsymbol{\alpha} \end{aligned}$$

Since this equality has to hold for all $\beta \in \mathbb{R}^r$ (with r the total number of $k-1$ -form basis vectors) it follows that:

$$\boldsymbol{\gamma} = (\mathbb{M}^{(k-1)})^{-1} \mathbb{C}_{\mathbf{v}_h}^{(k-1,k)} \boldsymbol{\alpha}$$

Here $\boldsymbol{\alpha}$ is the coefficient vector of discrete k -form $\alpha_h^{(k)}$ and $\boldsymbol{\gamma}$ contains the coefficients of the discrete $k-1$ -form $\gamma_h^{(k-1)}$ such that $i_{\mathbf{v}_h} \alpha_h^{(k)} = \gamma_h^{(k-1)}$. $\mathbb{M}^{(k-1)}$ is the mass matrix corresponding to discrete $k-1$ -forms, as covered in section 2.3.1, while $\mathbb{C}_{\mathbf{v}_h}^{(k-1,k)}$ contains at index (i, j) the integral of the product of basis functions $\psi_i^{(k-1)} \psi_j^{(k)}$ multiplied with \mathbf{v}_h^b . Thus the discrete formulation of the interior product that maps between the spaces of discrete k - and $k-1$ -forms is $(\mathbb{M}^{(k-1)})^{-1} \mathbb{C}_{\mathbf{v}_h}^{(k-1,k)}$.

Example 9. Let Ω be a manifold in \mathbb{R} , with cell complex Ω_h . Let \mathbf{v}_h be a vector field and $\alpha_h^{(1)} = \sum_i \alpha_i \psi_i^{(1)} \in \Lambda_{h,p}^{(1)}(\Omega_h)$ a discrete 1-form. Then:

$$i_{\mathbf{v}_h} \alpha_h^{(1)} = i_{\mathbf{v}_h} \sum_i \alpha_i \psi_i^{(1)} = i_{\mathbf{v}_h} \boldsymbol{\alpha}^T \boldsymbol{\psi}^{(1)} = \left[(\mathbb{M}^{(0)})^{-1} \mathbb{C}_{\mathbf{v}_h}^{(0,1)} \boldsymbol{\alpha} \right]^T \boldsymbol{\psi}^{(0)}$$

And thus $i_{\mathbf{v}_h} \alpha_h^{(1)} \in \Lambda_{h,p+1}^{(0)}(\Omega_h)$.

Example 10. Let Ω be a manifold in \mathbb{R}^2 , with cell complex Ω_h . Let $\alpha_h^{(2)} \in \Lambda_{h,p,p}^{(2)}(\Omega_h)$ and $\beta_h^{(1)}, \gamma_h^{(1)} \in \Lambda_{h,p,p+1}^{(1,x)}(\Omega_h) \oplus \Lambda_{h,p+1,p}^{(1,y)}(\Omega_h)$ and $\mathbf{v}_h \in T(\Omega)$, such that:

$$\begin{aligned} \alpha_h^{(2)} &= \sum_i \alpha_i \psi_i^{(2)} dx \wedge dy = \boldsymbol{\alpha}^T \boldsymbol{\psi}^{(2)} \\ \beta_h^{(1)} &= \sum_i \beta_i^x \psi_i^{(1,x)} dx + \beta_i^y \psi_i^{(1,y)} dy = (\boldsymbol{\beta}^x)^T \boldsymbol{\psi}^{(1,x)} dx + (\boldsymbol{\beta}^y)^T \boldsymbol{\psi}^{(1,y)} dy \\ \gamma_h^{(1)} &= \sum_i \gamma_i^x \psi_i^{(1,x)} dx + \gamma_i^y \psi_i^{(1,y)} dy = (\boldsymbol{\gamma}^x)^T \boldsymbol{\psi}^{(1,x)} dx + (\boldsymbol{\gamma}^y)^T \boldsymbol{\psi}^{(1,y)} dy \\ \mathbf{v}_h^b &= v^x dx + v^y dy \end{aligned}$$

Then:

$$\begin{aligned}
\int_{\Omega_h} \underbrace{i_{v_h} \alpha_h^{(2)}}_{=\gamma_h^{(1)}} \wedge \star \beta_h^{(1)} &= \int_{\Omega_h} \left(\gamma^x (\psi^{(1,x)})^T dx + \gamma^y (\psi^{(1,y)})^T dy \right) \wedge \star \left((\beta^x)^T \psi^{(1,x)} dx + (\beta^y)^T \psi^{(1,y)} dy \right) \\
&= \int_{\Omega_h} \left(\gamma^x (\psi^{(1,x)})^T dx + \gamma^y (\psi^{(1,y)})^T dy \right) \wedge \left((\beta^x)^T \psi^{(1,x)} dy - (\beta^y)^T \psi^{(1,y)} dx \right) \\
&= \int_{\Omega_h} \left(\gamma^x (\psi^{(1,x)})^T \psi^{(1,x)} (\beta^x)^T + \gamma^y (\psi^{(1,y)})^T \psi^{(1,y)} (\beta^y)^T \right) dx \wedge dy \\
&= \int_{\Omega_h} \begin{bmatrix} \gamma^x (\psi^{(1,x)})^T \psi^{(1,x)} (\beta^x)^T \\ \gamma^y (\psi^{(1,y)})^T \psi^{(1,y)} (\beta^y)^T \end{bmatrix} dx \wedge dy \\
\int_{\Omega_h} \alpha_h^{(2)} \wedge \star (\beta_h^{(1)} \wedge v_h^b) &= \int_{\Omega_h} \alpha (\psi^{(2)})^T dx \wedge dy \wedge \star \left[\left((\beta^x)^T \psi^{(1,x)} dx + (\beta^y)^T \psi^{(1,y)} dy \right) \wedge (v^x dx + v^y dy) \right] \\
&= \int_{\Omega_h} \alpha (\psi^{(2)})^T dx \wedge dy \wedge \star \left[\left((\beta^x)^T \psi^{(1,x)} v^y - (\beta^y)^T \psi^{(1,y)} v^x \right) dx \wedge dy \right] \\
&= \int_{\Omega_h} \alpha (\psi^{(2)})^T dx \wedge dy \wedge \left((\beta^x)^T \psi^{(1,x)} v^y - (\beta^y)^T \psi^{(1,y)} v^x \right) \\
&= \int_{\Omega_h} \begin{bmatrix} \alpha (\psi^{(2)})^T \psi^{(1,x)} v^y (\beta^x)^T \\ -\alpha (\psi^{(2)})^T \psi^{(1,y)} v^x (\beta^y)^T \end{bmatrix} dx \wedge dy
\end{aligned}$$

And thus:

$$\begin{aligned}
\int_{\Omega_h} i_{v_h} \alpha_h^{(2)} \wedge \star \beta_h^{(1)} &= \int_{\Omega_h} \alpha_h^{(2)} \wedge \star (\beta_h^{(1)} \wedge v_h^b) \\
\int_{\Omega_h} \begin{bmatrix} \gamma^x (\psi^{(1,x)})^T \psi^{(1,x)} (\beta^x)^T \\ \gamma^y (\psi^{(1,y)})^T \psi^{(1,y)} (\beta^y)^T \end{bmatrix} dx \wedge dy &= \int_{\Omega_h} \begin{bmatrix} \alpha (\psi^{(2)})^T \psi^{(1,x)} v^y (\beta^x)^T \\ -\alpha (\psi^{(2)})^T \psi^{(1,y)} v^x (\beta^y)^T \end{bmatrix} dx \wedge dy
\end{aligned}$$

So to determine the 1-form coefficient vector $\gamma = [\gamma^x \quad \gamma^y]^T$ two separate linear systems have to be solved:

$$\begin{aligned}
\mathbb{M}^{(1,x)} \gamma^x &= \mathbb{C}_{v_h}^{(1,x,2)} \alpha \\
\mathbb{M}^{(1,y)} \gamma^y &= -\mathbb{C}_{v_h}^{(1,y,2)} \alpha
\end{aligned}$$

Where contraction matrix $\mathbb{C}_{v_h}^{(1,x,2)}$ uses the velocity component in y -direction and $\mathbb{C}_{v_h}^{(1,y,2)} \alpha$ the component in x -direction. The discrete interior product then is:

$$i_{v_h} \alpha_h^{(2)} = \left[(\mathbb{M}^{(1,x)})^{-1} \mathbb{C}_{v_h}^{(1,x,2)} \alpha \right]^T \psi^{(1,x)} - \left[(\mathbb{M}^{(1,y)})^{-1} \mathbb{C}_{v_h}^{(1,y,2)} \alpha \right]^T \psi^{(1,y)}$$

Which indeed maps $\alpha_h^{(2)}$ to $\Lambda_{h,p,p+1}^{(1,x)}(\Omega_h) \oplus \Lambda_{h,p+1,p}^{(1,y)}(\Omega_h)$.

2.3.5. Lie derivative

With the discrete exterior derivative and interior product in place, discretizing the Lie derivative is straightforward. Recalling Cartan's homotopy formula given in (1.2.2), the discrete Lie derivative simplifies to a composition of the aforementioned discrete operators. For vector field v_h and discrete k -form $\alpha_h^{(k)}$:

$$\begin{aligned}
\mathcal{L}_{v_h} \alpha_h^{(k)} &= (di_{v_h} + i_{v_h} d) \alpha_h^{(k)} \\
&= \left(\mathbb{E}^{(k,k-1)} \left(\mathbb{M}^{(k-1)} \right)^{-1} \mathbb{C}_{v_h}^{(k-1,k)} + \left(\mathbb{M}^{(k)} \right)^{-1} \mathbb{C}_{v_h}^{(k,k+1)} \mathbb{E}^{(k+1,k)} \right) \alpha_h^{(k)}
\end{aligned}$$

In the same way as is the case for the continuous Lie derivative, this discrete operator maps from the space of discrete k -forms back into the same space. Furthermore, the discrete Lie derivative of a 0-form contains only the first of the two terms, whereas a discrete n -form (in \mathbb{R}^n) is only affected by the latter term.

2.4. Combining discrete operators into a discrete complex

The projection to a finite-dimensional space $\Lambda_h^{(k)}(\Omega_h)$ can be combined with the discrete exterior derivative and interior product in a commutative diagram, in order to show the relations between continuous operations and their discrete counterparts:

$$\begin{array}{ccccccc}
 \dots & \xleftarrow{d} & \Lambda^{(k)}(\Omega) & \xrightarrow{d} & \Lambda^{(k+1)}(\Omega) & \xleftarrow{d} & \Lambda^{(k+2)}(\Omega) & \xrightarrow{d} & \dots \\
 & & \downarrow \mathcal{R}^{(k)} & & \downarrow \mathcal{R}^{(k+1)} & & \downarrow \mathcal{R}^{(k+2)} & & \\
 \dots & \xrightarrow{\mathbb{E}^{(k,k-1)}} & C^{(k)}(\Omega_h) & \xrightarrow{\mathbb{E}^{(k+1,k)}} & C^{(k+1)}(\Omega_h) & \xrightarrow{\mathbb{E}^{(k+2,k+1)}} & C^{(k+2)}(\Omega_h) & \xrightarrow{\mathbb{E}^{(k+3,k+2)}} & \dots \\
 & & \uparrow \mathcal{S}^{(k)} & & \uparrow \mathcal{S}^{(k+1)} & & \uparrow \mathcal{S}^{(k+2)} & & \\
 \dots & \xleftarrow{(\mathbb{M}^{(k-1)})^{-1} C_{v_h}^{(k-1,k)}} & \Lambda_h^{(k)}(\Omega_h) & \xleftarrow{(\mathbb{M}^{(k)})^{-1} C_{v_h}^{(k,k+1)}} & \Lambda_h^{(k+1)}(\Omega_h) & \xleftarrow{(\mathbb{M}^{(k+1)})^{-1} C_{v_h}^{(k+1,k+2)}} & \Lambda_h^{(k+2)}(\Omega_h) & \xleftarrow{(\mathbb{M}^{(k+2)})^{-1} C_{v_h}^{(k+2,k+3)}} & \dots \\
 & & \downarrow \star & & \downarrow \star & & \downarrow \star & & \\
 & & \tilde{\Lambda}_h^{(n-k)}(\Omega_h) & & \tilde{\Lambda}_h^{(n-(k+1))}(\Omega_h) & & \tilde{\Lambda}_h^{(n-(k+2))}(\Omega_h) & &
 \end{array}$$

In the upper row the k -form spaces $\Lambda^{(k)}(\Omega)$ on manifold Ω are shown, together with the exterior derivative and interior product with vector field v_h ; these relations are all metric-independent and topological, meaning that local deformations of Ω do not affect them in any way (due to deformations not affecting the topology of Ω). Each k -form can be reduced to a k -cochain in $C^{(k)}(\Omega_h)$, as indicated by the reduction mapping $\mathcal{R}^{(k)}$. The reconstruction mapping $\mathcal{S}^{(k)}$ can be used to convert the k -cochain to a discrete k -form. The spaces of discrete k -forms are subspaces of the k -form spaces: $\Lambda_h^{(k)}(\Omega_h) \subset \Lambda^{(k)}(\Omega)$.

The discrete exterior derivative acting on discrete k -forms can be computed in a purely topological (and thus metric-independent) way by reducing any discrete k -form back to its corresponding k -cochain and applying the coboundary operator; this operator is represented exactly by incidence matrix $\mathbb{E}^{(k+1,k)}$ acting on said k -cochain. Subsequently reconstructing a discrete $k+1$ -form from the resulting $k+1$ -cochain in a consistent way allows one to compute the exterior derivative mapping discrete k -forms to $k+1$ -forms in a topologically exact way.

On the other hand the interior product maps directly from discrete $k+1$ - to k -form spaces, using the Hodge \star operator to help define a weak form of the map $\Lambda_h^{(k+1)}(\Omega_h) \rightarrow \Lambda_h^{(k)}(\Omega_h)$. This construction does not use the cochains and depends on the metric tensor through the Hodge \star . The resulting discrete k -form satisfies the interior product only weakly and in an integral sense; this is necessary for restricting the image of the interior product map to $\Lambda_h^{(k)}(\Omega_h)$.

Not shown here is the Lie derivative. Through Cartan's homotopy formula a discrete Lie derivative is constructed by successively applying the discrete exterior derivative and interior product to any discrete k -form. Although the continuous Lie derivative is metric-free, requiring that the discrete exterior derivative and interior product map between spaces of discrete differential forms thus induces a metric dependence for the discrete Lie derivative as well.

3

Momentum, kinetic energy and the compressible Euler equations

This chapter contains most of the novel ideas presented in this thesis work. Section 3.1 will start by covering variational formulations of the Euler equations. As will be shown these variational formulations are limited to incompressible flows and isentropic compressible flows. This necessitates the use of a non-variational formulation to be able to define numerical models for the general compressible Euler equations; this non-variational formulation is posed in section 3.2. This thesis work focuses on improving the numerical results that are obtained by directly discretizing the model given in section 3.2 through the application of a partial decomposition of its primary variables into Roe variables. Before doing this section 3.3 discusses how the discretization approach for incompressible flows cannot be generalized to compressible flows and covers some alternative approaches that have been used in literature. Following this the concept of Roe variables is introduced in section 3.4; despite having first been described and used in 1981 by Philip Roe for the construction of his approximate Roe solver for finite volume methods, not a lot of work has been published that exploits their structure.

The aim of this partial decomposition into Roe variables is to minimize numerical diffusion by constructing discrete skew-symmetric advection operators that simultaneously conserve momentum and kinetic energy. Section 3.5 covers the way in which the (partial) Roe-decomposed formulation can be derived from the non-variational Euler equations, in an arbitrary number of dimensions. The time stepping methods that are used in this thesis are then covered in section 3.6.2; their application to each (semi-discrete) numerical model will be covered in later chapters, as is the case for the spatial discretization. Lastly, section 3.6.3 covers the approach that has been used to linearize the numerical models employed throughout this thesis.

3.1. Variational geometric Euler equations

Before the Roe variable discretization can be derived in a geometric setting, the corresponding Euler equations need to be defined. We start by considering the case of incompressible inviscid (perfect) flows followed by perfect compressible flows, which result in the isentropic (entropy-conserving) Euler equations.

The equations corresponding to perfect fluid flows (incompressible Euler) can be derived from a variational principle [4, 17, 20], leaning directly on the fact that kinetic energy is conserved. The Lagrangian density used in variational formulations with perfect fluids is the kinetic energy density; thus conservation of this Lagrangian coincides with kinetic energy conservation. As it turns out the incompressible Euler equations are recovered when minimizing an action principle over the Lie group of volume-preserving (divergence-free) diffeomorphic flow maps. This results in the following equations:

$$\begin{aligned}\frac{\partial v^{(1)}}{\partial t} + \mathcal{L}_{\mathbf{u}} v^{(1)} &= -d\pi^{(0)} \\ \mathcal{L}_{\mathbf{u}} \sigma^{(n)} &= \operatorname{div} \mathbf{u} = 0\end{aligned}\tag{3.1}$$

Where $v^{(1)} = g(\mathbf{u}, \cdot) = \mathbf{u}^b$ with g the metric tensor and $\sigma^{(n)}$ is the canonical volume form given in definition

1.17. $v^{(1)}$ is the 1-form that represents momentum, which is being transported by the vector field \mathbf{u} . $\pi^{(0)}$ is a Lagrange multiplier that enforces the divergence-free nature of $v^{(1)}$; since $v^{(1)}$ directly corresponds to \mathbf{u} the divergence-free requirement on \mathbf{u} implies that $v^{(1)}$ should be divergence-free as well. This coincides with derivations in vector calculus; usually an argument is made that the pressure solution on the right-hand side of the momentum equation is not unique, since any constant can be added to the pressure field without changing the momentum balance at any point in the domain. Hence not the absolute pressure values matter, but only their gradients. From the more extensive theoretical basis of variational analysis and differential geometry the pressure field $\pi^{(0)}$ turns out to be a Lagrange multiplier.

Since $v^{(1)} = \mathbf{u}^\flat$:

$$\frac{\partial \mathbf{u}^\flat}{\partial t} + \mathcal{L}_{\mathbf{u}} \mathbf{u}^\flat = -d\pi^{(0)} \quad (3.2)$$

Arnold and Khesin derive in [4, Thm. IV.1.17] for a vector field and its dual 1-form:

$$\mathcal{L}_{\mathbf{u}} v^{(1)} = (\nabla_{\mathbf{u}} \mathbf{u})^\flat + \frac{1}{2} dg(\mathbf{u}, \mathbf{u})$$

We use this identity and apply to (3.2) the sharp operator \sharp , as covered in section 1.2.4. This then gives:

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla_{\mathbf{u}} \mathbf{u} = -\nabla p \quad (3.3)$$

Where $p^{(0)} = \pi^{(0)} + \frac{1}{2} g(\mathbf{u}, \mathbf{u})$ and $\nabla p = (dp^{(0)})^\sharp$. As is known from vector calculus the pressure field depends on the square of the velocity. Note furthermore that ∇p is a vector; this is consistent with the notion of \sharp converting 1-forms (such as $dp^{(0)}$) to their vector field counterpart, in the same way as how $v^{(1)}$ and \mathbf{u} are related. $\nabla_{\mathbf{u}} \mathbf{u}$ is the covariant derivative of \mathbf{u} under influence of \mathbf{u} . Lastly we note that taking the exterior derivative of (3.1) results in a conservation equation for a 2-form $dv^{(1)}$ often denoted as vorticity:

$$\frac{\partial dv^{(1)}}{\partial t} + \mathcal{L}_{\mathbf{u}} dv^{(1)} = 0$$

Which follows from the geometric momentum equation since the exterior derivative is nilpotent ($d \circ d = 0$) and commutes with the time and Lie derivatives.

This approach can be extended to certain compressible flow models. Instead of requiring the diffeomorphic flow maps to be volume-preserving (incompressible) with respect to (constant) volume forms they are now required to be incompressible with respect to a coordinate system moving with the fluid density [4, p. 320] [16]. A similar derivation procedure is used as in the incompressible case [20] and leads to the following system of equations:

$$\begin{aligned} \frac{\partial \rho^{(0)}}{\partial t} + \mathcal{L}_{\mathbf{u}} \rho^{(0)} &= 0 \\ \rho^{(0)} \left[\frac{\partial v^{(1)}}{\partial t} + \mathcal{L}_{\mathbf{u}} v^{(1)} - \frac{1}{2} dg(\mathbf{u}, \mathbf{u}) \right] &= -dp^{(0)} \\ \frac{\partial \theta^{(0)}}{\partial t} + \mathcal{L}_{\mathbf{u}} \theta^{(0)} &= 0 \end{aligned} \quad (3.4)$$

Where (again) $v^{(1)}$ is the 1-form velocity corresponding to the velocity vector field \mathbf{u} , $\rho^{(0)}$ is the fluid mass density and $\theta^{(0)}$ is the entropy density. Note the similarity of the momentum equations of (3.1) and (3.4): The only apparent differences are the addition of the term $-\frac{1}{2} dg(\mathbf{u}, \mathbf{u})$ and the factor $\rho^{(0)}$ that appear on the left-hand side when going from incompressible to compressible flows. Another difference is that the pressure field is no longer a Lagrange multiplier, but has a physical interpretation right from the outset and is usually obtained through an equation of state. This constitutive relation links the pressure field $p^{(0)}$ to the other variables being solved for. Often $p^{(0)} = p^{(0)}(\rho^{(0)})$, $p^{(0)} = p^{(0)}(\rho^{(0)}, e^{(0)})$ (with $e^{(0)}$ the internal energy) or, as in this case, $p^{(0)} = p^{(0)}(\rho^{(0)}, \theta^{(0)})$. There is no physical quantity 'linear momentum' explicitly present in the momentum equation; the entire equation is formulated in terms of $v^{(1)}$, \mathbf{u} , $\rho^{(0)}$ and $p^{(0)}$.

These variational geometric formulations do not allow for reversible effects like heat diffusion, friction, viscosity and shock waves. This prevents them from being extended to more elaborate physical models (such as Navier-Stokes) and generally restricts them to simple, academic cases: For example, no-slip boundary conditions produce friction, which destroys the variational structure.

3.2. Non-variational geometric Euler equations

This limited applicability of variational formulations for the Euler equations has lead to a need for posing a non-variational model that is to be decomposed into Roe variables. Instead of using the 1-form $\nu^{(1)}$ linear momentum is modeled as covector-valued volume form. Recall that in vector calculus momentum is modeled as vector quantity $\rho \mathbf{u}$, with ρ the mass density and \mathbf{u} the velocity vector field. Each entry of $\rho \mathbf{u}$ is a scalar field ρu_i , corresponding to the linear momentum in the direction of velocity component u_i . In this research we denote (linear) momentum with $m^{(1,n)}$, such that:

$$m^{(1,n)} = \sum_{i=1}^n m_i dx^i \otimes \sigma^{(n)}$$

Instead of being treated as a (co)vector, $m^{(1,n)}$ is now a covector-valued differential form. To pick out a component of $m^{(1,n)}$ we take its duality pairing with vector field direction $\partial_i = \frac{\partial}{\partial x^i}$, as introduced in definition 1.10:

$$\langle m^{(1,n)}, \partial_i \rangle = \sum_j m_j \underbrace{\langle dx^j, \partial_i \rangle}_{=\delta_i^j} \sigma^{(n)} = m_i \sigma^{(n)} = m_i^{(n)}$$

This construction can be used to derive a set of geometric Euler equations for compressible flows that explicitly encode momentum conservation. Note that each momentum component $m_i^{(n)}$ is a volume form that thus naturally integrates over volumes. Toshniwal in [55, 56] follows a control volume approach motivated by Tonti in chapter 12 of [53]. In this approach the rate of change of momentum present in a volume is equal to the momentum flux across the volume's surface plus the impulse of volume and surface forces. By neglecting the effects of viscosity only the pressure force remains, acting on the surface of said volume. Through Stokes' theorem the following conservation law for linear momentum is then obtained:

$$\frac{\partial \langle m^{(1,n)}, \partial_i \rangle}{\partial t} + di_{\mathbf{u}} \langle m^{(1,n)}, \partial_i \rangle = -di_{\partial_i} p^{(n)} \quad (3.5)$$

Since $\langle m^{(1,n)}, \partial_i \rangle$ is a volume form we recognize that the convection term can be restated as Lie derivative through the use of Cartan's homotopy formula, leading to:

$$\frac{\partial \langle m^{(1,n)}, \partial_i \rangle}{\partial t} + \mathcal{L}_{\mathbf{u}} \langle m^{(1,n)}, \partial_i \rangle = -di_{\partial_i} p^{(n)} \quad (3.6)$$

In a similar way it can be shown that the rate of change of energy contained in a volume is equal to the energy flux across its surface plus the work done by volume and surface forces. Again volume forces are neglected (for now) leaving only the pressure force, acting on the volume's boundary. The work performed by the pressure force $p^{(n)}$ is equal to its contraction with the velocity field \mathbf{u} : $i_{\mathbf{u}} p^{(n)}$. Through Stokes' theorem the surface integral of $i_{\mathbf{u}} p^{(n)}$ is then equal to the volume integral of $di_{\mathbf{u}} p^{(n)}$, which is interpreted as the Lie derivative $\mathcal{L}_{\mathbf{u}} p^{(n)}$ in the same way as was done before. The resulting geometric energy conservation law becomes:

$$\frac{\partial E^{(n)}}{\partial t} + \mathcal{L}_{\mathbf{u}} E^{(n)} + \mathcal{L}_{\mathbf{u}} p^{(n)} = 0 \quad (3.7)$$

Where $E^{(n)}$ is the total energy density. A similar conservation law holds for the mass contained in a volume, the primary difference being that no mass is destroyed or created by any force acting upon said volume. Combining the mass, momentum and energy conservation laws mentioned here results in the following formulation for the compressible geometric Euler equations:

$$\begin{aligned} \frac{\partial \rho^{(n)}}{\partial t} + \mathcal{L}_{\mathbf{u}} \rho^{(n)} &= 0 \\ \frac{\partial \langle m^{(1,n)}, \partial_i \rangle}{\partial t} + \mathcal{L}_{\mathbf{u}} \langle m^{(1,n)}, \partial_i \rangle + di_{\partial_i} p^{(n)} &= 0 \\ \frac{\partial E^{(n)}}{\partial t} + \mathcal{L}_{\mathbf{u}} E^{(n)} + \mathcal{L}_{\mathbf{u}} p^{(n)} &= 0 \end{aligned} \quad (3.8)$$

Not covered until now is the velocity vector field \mathbf{u} and the equation of state that is necessary to close system (3.23). Note that although \mathbf{u} appears in (3.8), in its discretization only the corresponding 1-form \mathbf{u}^\flat is used.

We thus define the velocity field as follows:

$$\begin{aligned} \mathbf{u}^b &= \frac{\star m^{(1,n)}}{\star \rho^{(n)}} = \frac{\sum_i dx^i \otimes \star (m_i dx^1 \wedge dx^2 \wedge \dots \wedge dx^n)}{\star (\rho dx^1 \wedge dx^2 \wedge \dots \wedge dx^n)} \\ &= \frac{\sqrt{g} \sum_i dx^i \otimes m_i}{\sqrt{g} \rho} \\ &= \sum_i dx^i \otimes \frac{m_i}{\rho} \end{aligned}$$

The metric contributions of \star thus cancel. Kreeft [31, Sec. 11.2.3] introduces the flat and sharp Hodge \star operators to convert (co)vector-valued k -forms into (co)vector-valued $n - k$ -forms, based on the work of Yavari in [58]; while the same construction can be used here to compute \mathbf{u} , doing so is not necessary since only \mathbf{u}^b is used while discretizing (3.23).

This system of conservation laws contains $n+2$ equations on manifold $\Omega \subset \mathbb{R}^n$ and $n+3$ unknowns: $\rho^{(n)}$, $E^{(n)}$, $p^{(n)}$ and $m_i^{(n)}$ ($i = 1, 2, \dots, n$). Thus it needs to be closed with a constitutive relation. For this an equation of state is posed to relate pressure $p^{(n)}$ to the other variables. For this the ideal gas law is used:

$$p^{(n)} = (\gamma - 1) (E^{(n)} - E_{kin}^{(n)})$$

Where γ is the ratio of specific heats that is commonly encountered in compressible aerodynamics and $E_{kin}^{(n)}$ is the kinetic energy density, defined as:

$$E_{kin}^{(n)} = \frac{1}{2} \frac{\langle m^{(1,n)}, \star^\sharp m^{(1,n)} \rangle}{\star \rho^{(n)}} \quad (3.9)$$

Here \star^\sharp is the sharp Hodge \star mentioned above: It acts as the Hodge \star on the n -form leg of $m^{(1,n)}$ and as \sharp on its covector-valued leg; therefore it converts the covector-valued n -form $m^{(1,n)}$ to a vector-valued 0-form. The duality pairing in (3.9) is then the duality pairing between vector- and covector-valued differential forms given in definition 1.10.

3.3. Momentum- and kinetic energy-conserving methods

Accurately and correctly mimicking the (non)conservative behavior of (systems of) conservation laws is a central theme when constructing numerical methods. Numerical solutions are only approximations to the exact solutions of any problem; encoding correct conservation behavior into such an approximation is one way in which physically correct numerical solutions are sought. For a long time discretization methods have been derived that explicitly encode the conservation of primary variables. Examples of these numerical methods include finite volume- and finite element methods. These primary variables are the parameters for which the aforementioned conservation laws have been explicitly posed. For example, when applied to the (nonlinear) compressible Euler or Navier-Stokes equations the finite volume- and element methods simultaneously conserve the primary variables mass, linear momentum and either total or internal energy, depending on the model formulation that has been used for discretization.

While the conservation of these primary variables in any numerical solution is desirable and physically correct, the conservation of several secondary invariants of the Euler and Navier-Stokes models is not guaranteed by these methods. Within this thesis we focus on kinetic energy, one of the secondary invariants of the (inviscid) Euler equations. Kinetic energy conservation and minimizing numerical diffusion is not only important for the (long- and short-time) stability of a numerical simulation [39, 44] but also relevant for (among others) Direct Numerical Simulations (DNS) and for Large Eddy Simulations (LES), where subgrid models are used to model correct energy diffusion on scales that are not directly resolved by the simulation; numerical dissipation interferes with this, hence there is a need for low-dissipation methods.

Kinetic energy conservation in numerical methods for incompressible Euler flows can be achieved simultaneously with conservation of mass and momentum, by exploiting its divergence-free constraint. As shown by Arnold the incompressible Euler equations admit a Hamiltonian structure [4]; its solutions are precisely the shortest paths on the Lie group of divergence-free diffeomorphisms [3]. This structure has been used

throughout literature to obtain numerical solutions to the Euler equations that simultaneously conserve mass, momentum and kinetic energy. Examples of this include the work of: Evans and Hughes, who construct a mimetic isogeometric method for doing so [14]; Pavlov et al. and Gawlik et al., who approximate the Lie group of divergence-free diffeomorphisms with a finite Lie group subspace and apply variational analysis to this finite-sized group [17, 42]. In a somewhat different way, Palha & Gerritsma construct divergence-free solutions to the velocity-vorticity formulation of the Navier-Stokes equations: When viscosity is neglected this leads to conservation of mass, kinetic energy and vorticity [41].

In each of these methods the pointwise divergence-free nature of the numerical solutions leads to an equivalence between different forms of the advection operator in the incompressible Euler and Navier-Stokes models. In general, discretizing different forms of the nonlinear advection operator leads to formulation-dependent conservation behavior and accuracy. A handful of different (nonlinear) advection operator formulations are used throughout literature [12, 35]:

$$\begin{aligned}
&\text{Convective : } \mathbf{u} \cdot \nabla \mathbf{u} \\
&\text{Conservative : } \nabla \cdot (\mathbf{u} \otimes \mathbf{u}) = \mathbf{u} \cdot \nabla \mathbf{u} + (\nabla \cdot \mathbf{u}) \mathbf{u} \\
&\text{Skew-symmetric : } \mathbf{u} \cdot \nabla \mathbf{u} + \frac{1}{2} (\nabla \cdot \mathbf{u}) \mathbf{u}
\end{aligned} \tag{3.10}$$

In the (continuous) Euler and Navier-Stokes equations these all lead to the exact same solutions; switching between them is a matter of algebraic manipulation. It is only when discretizing or linearizing these operators that differences between them appear. Especially linearization destroys these equivalencies: Let $\tilde{\mathbf{u}}$ denote a linearized velocity vector field. Then the linearized advection operators become:

$$\begin{aligned}
&\text{Convective : } \tilde{\mathbf{u}} \cdot \nabla \mathbf{u} \\
&\text{Conservative : } \nabla \cdot (\tilde{\mathbf{u}} \otimes \mathbf{u}) \\
&\text{Skew-symmetric : } \tilde{\mathbf{u}} \cdot \nabla \mathbf{u} + \frac{1}{2} (\nabla \cdot \mathbf{u}) \tilde{\mathbf{u}}
\end{aligned} \tag{3.11}$$

These linearized operators behave differently than those in (3.10) unless $\tilde{\mathbf{u}} = \mathbf{u}$, which only occurs when the full nonlinear solution has been found up to machine precision. This poses a stringent requirement on computational cost, especially for large numerical systems. Whereas directly discretizing the linearized convective advection operator with a finite element method does not lead to conservation of momentum nor kinetic energy unless the full nonlinear solution is computed, using the linearized conservative form leads to a numerical scheme that does conserve momentum $(\rho) \mathbf{u}$ at all times. Similarly the linearized skew-symmetric form conserves kinetic energy $(\frac{1}{2} \rho) \mathbf{u}^2$.

For numerical solutions that are only weakly divergence-free the convective, conservative and skew-symmetric forms of the advection operator given in (3.10) are markedly different; only for strongly (i.e. pointwise) divergence-free solutions, with $\nabla \cdot \mathbf{u} = 0$ everywhere, the convective, conservative and skew-symmetric operators are identical. When this happens the conservation properties of both the conservative and skew-symmetric hold simultaneously; that is, both momentum and kinetic energy are conserved. Mass is conserved as well, since the divergence-free velocity field constraint is fulfilled.

This way of obtaining simultaneous conservation of momentum and kinetic energy does not generalize to compressible flows. Not only are the advection operators no longer posed purely in terms of \mathbf{u} (the momentum vector $\mathbf{m} = \rho \mathbf{u}$ appears, since the density field can no longer be moved outside all derivatives), but the velocity fields are no longer desired to be divergence-free as the continuity equation does not simplify to $\nabla \cdot \mathbf{u} = 0$. Hence the convective, conservative and skew-symmetric advection operators are markedly different and cannot be made equivalent anymore.

In the continuous Euler and Navier-Stokes models this is not an issue, since algebraic manipulation can be used to transition between advection operator forms. However as mentioned previously, after numerical discretization this is no longer possible and the conservation behavior of the discrete method depends on which advection form has been discretized. Often this means making a choice between conservation of momentum or kinetic energy. In the next section an approach is introduced that leads to discrete advection operators that

do simultaneously conserve mass, momentum and kinetic energy. This reflects part of the structure of advection: To merely move around mass, momentum and kinetic energy and not to produce or destroy any of it.

3.4. Roe variable decompositions

As already remarked by Roe in 1981 in his seminal paper [46], the variables and fluxes of the Euler equations in any number of dimensions can all be written as quadratic quantities. For example, following the convention of Rozema et al. [48] the following state vector can be defined for a two-dimensional Euler flow:

$$\mathbf{h} = \begin{bmatrix} \sqrt{\rho} \\ \sqrt{\rho}u \\ \sqrt{\rho}v \\ \sqrt{\rho e} \end{bmatrix} = \begin{bmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \end{bmatrix}$$

Where ρ is the fluid mass density, $\rho\mathbf{u} = [\rho u \quad \rho v]^T$ is the momentum vector and ρe is the internal energy density. The physical variables can be reconstructed as quadratic quantities:

$$\begin{aligned} \rho &= h_1^2 \\ \rho u &= h_1 h_2 \\ \rho v &= h_1 h_3 \\ E_{kin} &= \frac{1}{2} (h_2^2 + h_3^2) \\ e &= h_4^2 \\ E_{total} &= E_{kin} + e = h_4^2 + \frac{1}{2} (h_2^2 + h_3^2) \end{aligned}$$

Note specifically that the momentum components, kinetic- and internal energy are all quadratic variables. This fact can be exploited to yield advection operators that conserve both mass, momentum and kinetic-, internal- and total energy. To see this, let $\phi, \psi \in [h_1, h_2, h_3, h_4]$, each subject to an advection equation:

$$\begin{aligned} \frac{\partial \phi}{\partial t} + C(\mathbf{u})\phi &= 0 \\ \frac{\partial \psi}{\partial t} + C(\mathbf{u})\psi &= 0 \end{aligned}$$

With $C(\mathbf{u})$ an advection operator. Multiplying the advection equation for ϕ with ψ & vice versa and adding the two together then gives:

$$\underbrace{\psi \frac{\partial \phi}{\partial t} + \phi \frac{\partial \psi}{\partial t}}_{= \frac{\partial(\phi\psi)}{\partial t}} + \psi C(\mathbf{u})\phi + \phi C(\mathbf{u})\psi = 0 \quad (3.12)$$

$$\frac{\partial(\phi\psi)}{\partial t} + \psi C(\mathbf{u})\phi + \phi C(\mathbf{u})\psi = 0 \quad (3.13)$$

Suppose that $C(\mathbf{u})$ is anti-self-adjoint, such that:

$$\langle \phi, C(\mathbf{u})\psi \rangle = -\langle C(\mathbf{u})\phi, \psi \rangle$$

Then in (3.12) $\frac{\partial(\phi\psi)}{\partial t} = 0$, directly leading to conservation of the product $\phi\psi = h_i h_j$, with $h_i, h_j \in [h_1, h_2, h_3, h_4]$. Whereas this was derived in a continuous setting, discrete advection operators that retain this structure exist and can be constructed; for any matrix C it holds that:

$$\mathbf{v}^T C \mathbf{w} = -\mathbf{w}^T C \mathbf{v} \quad (3.14)$$

By noting that:

$$(\mathbf{v}^T C \mathbf{w})^T = \mathbf{w}^T C^T \mathbf{v}$$

It can be concluded that (3.14) holds for any vectors \mathbf{v} , \mathbf{w} if and only if $C^T = -C$. In other words, this holds if and only if matrix C is skew-symmetric. If discrete skew-symmetric advection operators are thus used, they can simultaneously conserve mass, momentum and kinetic energy. An additional condition for this to happen is that all conservation equations have to use the same matrix as advection operator.

This approach to discretizing conservative advection operators with skew-symmetric matrices was introduced by Rozema et al. in [47, 48] for finite volume methods. In this research it is extended to (isogeometric) mimetic finite element methods. Other uses throughout literature are scarce. While various of its elements appear in the approximate Roe solver [46], not much additional research seems to have gone into this modeling approach for advection in compressible fluid flows. Pettersson et al. in [43] use a similar decomposition for a stochastic finite element method in what they refer to as 'Roe variables'.

3.5. Obtaining a partial Roe variable formulation

In section 3.2 a non-variational formulation of the Euler equations was posed. A (partial) decomposition of this model can be derived in terms of Roe variables. Let Ω be a manifold in \mathbb{R}^n . For clarity the model given in (3.8) is repeated here:

$$\begin{aligned} \frac{\partial \rho^{(n)}}{\partial t} + \mathcal{L}_{\mathbf{u}} \rho^{(n)} &= 0 \\ \frac{\partial \langle m^{(1,n)}, \partial_i \rangle}{\partial t} + \mathcal{L}_{\mathbf{u}} \langle m^{(1,n)}, \partial_i \rangle + di_{\partial_i} p^{(n)} &= 0 \\ \frac{\partial E^{(n)}}{\partial t} + \mathcal{L}_{\mathbf{u}} E^{(n)} + \mathcal{L}_{\mathbf{u}} p^{(n)} &= 0 \end{aligned}$$

Starting with mass conservation we reformulate the density field as: $\rho^{(n)} = \sqrt{\rho^{(0)}} \wedge \sqrt{\rho^{(0)}} \wedge \sigma^{(n)} = \sqrt{\rho^{(0)}} \sqrt{\rho^{(0)}} \sigma^{(n)}$ with $\sigma^{(n)}$ the canonical volume form. Note that 0-forms are 'just functions': Due to their lack of association with any dual basis elements dx^i they can be moved around in wedge products at will. Applying this decomposition to the time derivative of the continuity equation gives:

$$\frac{\partial \rho^{(n)}}{\partial t} = \frac{\partial \sqrt{\rho^{(0)}} \sqrt{\rho^{(0)}} \sigma^{(n)}}{\partial t} = 2\sqrt{\rho^{(0)}} \frac{\partial \sqrt{\rho^{(0)}} \sigma^{(n)}}{\partial t}$$

Where $\sigma^{(n)}$ is explicitly taken to be time-independent, allowing for it to be moved in- and out of the time derivative as desired. Similarly, for the Lie derivative:

$$\begin{aligned} \mathcal{L}_{\mathbf{u}}(\rho^{(n)}) &= \mathcal{L}_{\mathbf{u}}(\sqrt{\rho^{(0)}}) \wedge \sqrt{\rho^{(0)}} \sigma^{(n)} + \sqrt{\rho^{(0)}} \wedge \mathcal{L}_{\mathbf{u}}(\sqrt{\rho^{(0)}} \sigma^{(n)}) \\ &= \sqrt{\rho^{(0)}} \left[\underbrace{\mathcal{L}_{\mathbf{u}}(\sqrt{\rho^{(0)}}) \sigma^{(n)} + \mathcal{L}_{\mathbf{u}}(\sqrt{\rho^{(0)}} \sigma^{(n)})}_{= \mathbb{L}_{\mathbf{u}}(\sqrt{\rho^{(0)}} \sigma^{(n)})} \right] \\ &= \sqrt{\rho^{(0)}} \mathbb{L}_{\mathbf{u}}(\sqrt{\rho^{(0)}} \sigma^{(n)}) \end{aligned}$$

Resulting in the following continuity equation for Roe variable $\sqrt{\rho^{(n)}}$:

$$\begin{aligned} 2\sqrt{\rho^{(0)}} \left[\frac{\partial \sqrt{\rho^{(0)}} \sigma^{(n)}}{\partial t} + \frac{1}{2} \mathbb{L}_{\mathbf{u}}(\sqrt{\rho^{(0)}} \sigma^{(n)}) \right] &= 0 \\ \frac{\partial \sqrt{\rho^{(0)}} \sigma^{(n)}}{\partial t} + \frac{1}{2} \mathbb{L}_{\mathbf{u}}(\sqrt{\rho^{(0)}} \sigma^{(n)}) &= 0 \end{aligned}$$

Which, by noting that $\sqrt{\rho^{(0)}} \sigma^{(n)} = \sqrt{\rho^{(n)}}$, is written as:

$$\frac{\partial \sqrt{\rho^{(n)}}}{\partial t} + \frac{1}{2} \mathbb{L}_{\mathbf{u}}(\sqrt{\rho^{(n)}}) = 0 \quad (3.15)$$

Here the Lie derivative is applied to 0-forms and volume forms. As was mentioned in example 5, when applied to a 0-form the Lie derivative results in the convective advection form, whereas applying it to a volume

form leads to the conservative formulation of the advection operator.

Turning to the momentum equation, the covector-valued momentum volume form $m^{(1,n)}$ is introduced:

$$m^{(1,n)} = \sum_i dx^i \otimes \underbrace{m_i^{(0)} \sigma^{(n)}}_{=m_i^{(n)}} = \sum_i dx^i \otimes m_i^{(n)}$$

For which the following Roe variable decomposition is used:

$$m^{(1,n)} = \sum_i dx^i \otimes m_i^{(n)} = \sum_i dx^i \otimes \sqrt{\rho}^{(0)} \phi_i^{(0)} \sigma^{(n)} = \sqrt{\rho}^{(0)} \sum_i dx^i \otimes \phi_i^{(0)} \sigma^{(n)} = \sqrt{\rho}^{(0)} \phi^{(1,n)} \quad (3.16)$$

Here $\phi_i^{(0)}$ is shorthand for $\sqrt{\rho} u_i$ in the Roe variable decomposition shown in section 3.4. The application of this decomposition to the momentum equation gives:

$$\begin{aligned} \frac{\partial \left(\sqrt{\rho}^{(0)} \phi_i^{(0)} \sigma^{(n)} \right)}{\partial t} &= \phi_i^{(0)} \sigma^{(n)} \frac{\partial \sqrt{\rho}^{(0)}}{\partial t} + \sqrt{\rho}^{(0)} \frac{\partial \left(\phi_i^{(0)} \sigma^{(n)} \right)}{\partial t} \\ \mathcal{L}_{\mathbf{u}} \left(\sqrt{\rho}^{(0)} \phi_i^{(0)} \sigma^{(n)} \right) &= \phi_i^{(0)} \sigma^{(n)} \mathcal{L}_{\mathbf{u}} \left(\sqrt{\rho}^{(0)} \right) + \sqrt{\rho}^{(0)} \mathcal{L}_{\mathbf{u}} \left(\phi_i^{(0)} \sigma^{(n)} \right) \end{aligned}$$

Which turn the momentum equation into:

$$\begin{aligned} \frac{\partial \langle m^{(1,n)}, \partial_i \rangle}{\partial t} + \mathcal{L}_{\mathbf{u}} \langle m^{(1,n)}, \partial_i \rangle + di_{\partial_i} p^{(n)} &= 0 \\ \underbrace{\phi_i^{(0)} \sigma^{(n)} \frac{\partial \sqrt{\rho}^{(0)}}{\partial t} + \sqrt{\rho}^{(0)} \frac{\partial \left(\phi_i^{(0)} \sigma^{(n)} \right)}{\partial t}}_{= \frac{\partial \langle m^{(1,n)}, \partial_i \rangle}{\partial t}} + \underbrace{\phi_i^{(0)} \sigma^{(n)} \mathcal{L}_{\mathbf{u}} \left(\sqrt{\rho}^{(0)} \right) + \sqrt{\rho}^{(0)} \mathcal{L}_{\mathbf{u}} \left(\phi_i^{(0)} \sigma^{(n)} \right)}_{= \mathcal{L}_{\mathbf{u}} \langle m^{(1,n)}, \partial_i \rangle} + di_{\partial_i} p^{(n)} &= 0 \end{aligned}$$

Now the continuity equation of (3.15) is inserted in place of the time derivative $\frac{\partial \sqrt{\rho}^{(0)}}{\partial t}$. In order to obtain the time derivative of $\sqrt{\rho}^{(0)}$ the Hodge \star operator is applied to the continuity equation:

$$\begin{aligned} \star \frac{\partial \sqrt{\rho}^{(n)}}{\partial t} + \frac{1}{2} \star \mathbb{L}_{\mathbf{u}} \left(\sqrt{\rho}^{(n)} \right) &= 0 \\ \frac{\partial \sqrt{\rho}^{(0)}}{\partial t} &= -\frac{1}{2} \star \mathbb{L}_{\mathbf{u}} \left(\sqrt{\rho}^{(n)} \right) \end{aligned}$$

Combining everything into the momentum equation then gives the following (intermediate) result:

$$\phi_i^{(0)} \sigma^{(n)} \left(-\frac{1}{2} \star \mathbb{L}_{\mathbf{u}} \left(\sqrt{\rho}^{(n)} \right) \right) + \sqrt{\rho}^{(0)} \frac{\partial \left(\phi_i^{(0)} \sigma^{(n)} \right)}{\partial t} + \underbrace{\phi_i^{(0)} \sigma^{(n)} \mathcal{L}_{\mathbf{u}} \left(\sqrt{\rho}^{(0)} \right) + \sqrt{\rho}^{(0)} \mathcal{L}_{\mathbf{u}} \left(\phi_i^{(0)} \sigma^{(n)} \right)}_I + di_{\partial_i} p^{(n)} = 0 \quad (3.17)$$

Where:

$$\begin{aligned} \phi_i^{(0)} \sigma^{(n)} \left(-\frac{1}{2} \star \mathbb{L}_{\mathbf{u}} \left(\sqrt{\rho}^{(0)} \sigma^{(n)} \right) \right) &= \phi_i^{(0)} \sigma^{(n)} \left(-\frac{1}{2} \star \left[\mathcal{L}_{\mathbf{u}} \left(\sqrt{\rho}^{(0)} \right) \sigma^{(n)} + \mathcal{L}_{\mathbf{u}} \left(\sqrt{\rho}^{(0)} \sigma^{(n)} \right) \right] \right) \\ &= -\frac{1}{2} \phi_i^{(0)} \sigma^{(n)} \left(\mathcal{L}_{\mathbf{u}} \left(\sqrt{\rho}^{(0)} \right) + \star \mathcal{L}_{\mathbf{u}} \left(\sqrt{\rho}^{(0)} \sigma^{(n)} \right) \right) \\ &= -\frac{1}{2} \phi_i^{(0)} \sigma^{(n)} \left(\mathcal{L}_{\mathbf{u}} \left(\sqrt{\rho}^{(0)} \right) + \star \mathcal{L}_{\mathbf{u}} \left(\sqrt{\rho}^{(0)} \right) \sigma^{(n)} + \star \sqrt{\rho}^{(0)} \mathcal{L}_{\mathbf{u}} \left(\sigma^{(n)} \right) \right) \\ &= -\frac{1}{2} \phi_i^{(0)} \sigma^{(n)} \left(\underbrace{2 \mathcal{L}_{\mathbf{u}} \left(\sqrt{\rho}^{(0)} \right)}_I + \star \sqrt{\rho}^{(0)} \mathcal{L}_{\mathbf{u}} \left(\sigma^{(n)} \right) \right) \\ &= -\phi_i^{(0)} \sigma^{(n)} \left(\underbrace{\mathcal{L}_{\mathbf{u}} \left(\sqrt{\rho}^{(0)} \right)}_I + \frac{1}{2} \star \sqrt{\rho}^{(0)} \mathcal{L}_{\mathbf{u}} \left(\sigma^{(n)} \right) \right) \end{aligned} \quad (3.18)$$

Note that it is explicitly used that $\star\sigma^{(n)} = 1$. The terms indicated with I in (3.17) and (3.18) cancel, which leaves the following:

$$\sqrt{\rho^{(0)}} \frac{\partial(\phi_i^{(0)}\sigma^{(n)})}{\partial t} + \underbrace{\sqrt{\rho^{(0)}} \mathcal{L}_{\mathbf{u}}(\phi_i^{(0)}\sigma^{(n)})}_{II} - \underbrace{\frac{1}{2}\phi_i^{(0)}\sigma^{(n)} \star \sqrt{\rho^{(0)}} \mathcal{L}_{\mathbf{u}}(\sigma^{(n)})}_{III} + di_{\partial_i} p^{(n)} = 0 \quad (3.19)$$

The terms indicated with II and III are those related to the advection operator. Focusing on the first of these terms it can be seen that:

$$\begin{aligned} \sqrt{\rho^{(0)}} \mathcal{L}_{\mathbf{u}}(\phi_i^{(0)}\sigma^{(n)}) &= \frac{1}{2}\sqrt{\rho^{(0)}} \left[\mathcal{L}_{\mathbf{u}}(\phi_i^{(0)}\sigma^{(n)}) + \mathcal{L}_{\mathbf{u}}(\phi_i^{(0)}\sigma^{(n)}) \right] \\ &= \frac{1}{2}\sqrt{\rho^{(0)}} \left[\mathcal{L}_{\mathbf{u}}(\phi_i^{(0)}\sigma^{(n)}) + (\sigma^{(n)} \mathcal{L}_{\mathbf{u}}(\phi_i^{(0)}) + \phi_i^{(0)} \mathcal{L}_{\mathbf{u}}(\sigma^{(n)})) \right] \end{aligned}$$

Whereas for the term marked with III we take a slightly longer detour, and start by remarking that:

$$\frac{1}{2}\phi_i^{(0)}\sigma^{(n)} \star \sqrt{\rho^{(0)}} \mathcal{L}_{\mathbf{u}}(\sigma^{(n)}) = \frac{1}{2}\phi_i^{(0)}\sqrt{\rho^{(0)}}\sigma^{(n)} \star \mathcal{L}_{\mathbf{u}}(\sigma^{(n)}) \quad (3.20)$$

Note that the canonical volume form is $\sigma^{(n)} = \sqrt{g} dx^1 \wedge dx^2 \wedge \dots \wedge dx^n$. From working out the Lie derivative it can be found that:

$$\begin{aligned} \sigma^{(n)} \star \mathcal{L}_{\mathbf{u}}(\sigma^{(n)}) &= \sigma^{(n)} \star di_{\mathbf{u}}(\sigma^{(n)}) \\ &= \sigma^{(n)} \star \left(\underbrace{\sum_i \frac{\partial\sqrt{g}u^i}{\partial x^i} dx^1 \wedge dx^2 \wedge \dots \wedge dx^n}_{=\mathcal{L}_{\mathbf{u}}(\sigma^{(n)})} \right) = \sigma^{(n)} \star \left(\sum_i \frac{\partial\sqrt{g}u^i}{\partial x^i} \underbrace{dx^1 \wedge dx^2 \wedge \dots \wedge dx^n}_{=\frac{\sigma^{(n)}}{\sqrt{g}}} \right) \\ &= \sigma^{(n)} \left(\frac{1}{\sqrt{g}} \sum_i \frac{\partial\sqrt{g}u^i}{\partial x^i} \right) \\ &= \frac{1}{\sqrt{g}} \sum_i \frac{\partial\sqrt{g}u^i}{\partial x^i} \sigma^{(n)} \\ &= \sum_i \frac{\partial\sqrt{g}u^i}{\partial x^i} dx^1 \wedge dx^2 \wedge \dots \wedge dx^n = \mathcal{L}_{\mathbf{u}}(\sigma^{(n)}) \end{aligned}$$

In short, $\sigma^{(n)} \star \mathcal{L}_{\mathbf{u}}(\sigma^{(n)}) = \mathcal{L}_{\mathbf{u}}(\sigma^{(n)})$ and thus it can be found from (3.20) that:

$$\frac{1}{2}\phi_i^{(0)}\sqrt{\rho^{(0)}}\sigma^{(n)} \star \mathcal{L}_{\mathbf{u}}(\sigma^{(n)}) = \frac{1}{2}\phi_i^{(0)}\sqrt{\rho^{(0)}}\mathcal{L}_{\mathbf{u}}(\sigma^{(n)})$$

At this point the modified versions of terms II and III are plugged back into (3.19):

$$\begin{aligned} \sqrt{\rho^{(0)}} \frac{\partial(\phi_i^{(0)}\sigma^{(n)})}{\partial t} + \underbrace{\frac{1}{2}\sqrt{\rho^{(0)}} \left[\mathcal{L}_{\mathbf{u}}(\phi_i^{(0)}\sigma^{(n)}) + (\sigma^{(n)} \mathcal{L}_{\mathbf{u}}(\phi_i^{(0)}) + \phi_i^{(0)} \mathcal{L}_{\mathbf{u}}(\sigma^{(n)})) \right]}_{II} \\ - \underbrace{\frac{1}{2}\phi_i^{(0)}\sqrt{\rho^{(0)}} \mathcal{L}_{\mathbf{u}}(\sigma^{(n)})}_{III} + di_{\partial_i} p^{(n)} = 0 \end{aligned}$$

The Lie derivatives of canonical volume form $\sigma^{(n)}$ are equal and of opposite sign, thereby canceling one another. This leads to the following formulation:

$$\sqrt{\rho^{(0)}} \frac{\partial(\phi_i^{(0)}\sigma^{(n)})}{\partial t} + \frac{1}{2}\sqrt{\rho^{(0)}} \underbrace{\left[\mathcal{L}_{\mathbf{u}}(\phi_i^{(0)}\sigma^{(n)}) + \mathcal{L}_{\mathbf{u}}(\phi_i^{(0)}\sigma^{(n)}) \right]}_{=\mathbb{L}_{\mathbf{u}}(\phi_i^{(0)}\sigma^{(n)})} + di_{\partial_i} p^{(n)} = 0$$

As was found for the continuity equation the advection operator is $\mathbb{L}_{\mathbf{u}}$, this time acting on $\phi_i^{(0)}\sigma^{(n)}$ instead of $\sqrt{\rho^{(0)}}\sigma^{(n)}$. The resulting Roe variable momentum conservation law is found by dividing with $\sqrt{\rho^{(0)}}$:

$$\frac{\partial(\phi_i^{(0)}\sigma^{(n)})}{\partial t} + \frac{1}{2}\mathbb{L}_{\mathbf{u}}(\phi_i^{(0)}\sigma^{(n)}) + \frac{1}{\sqrt{\rho^{(0)}}}di_{\partial_i}p^{(n)} = 0$$

Which by using covector-valued differential form $\phi^{(1,n)}$ defined in (3.16) becomes:

$$\frac{\partial\langle\partial_i, \phi^{(1,n)}\rangle}{\partial t} + \frac{1}{2}\mathbb{L}_{\mathbf{u}}(\langle\partial_i, \phi^{(1,n)}\rangle) + \frac{1}{\sqrt{\rho^{(0)}}}di_{\partial_i}p^{(n)} = 0$$

Note that dividing with $\sqrt{\rho^{(0)}}$ is not ill-defined, since $\sqrt{\rho^{(0)}}$ is the fluid mass density and is thus strictly positive as long as the assumption of a continuous medium is valid.

While the relationship between mass, momentum and kinetic energy (all three being quadratic functions of $\{\sqrt{\rho^{(n)}}, \phi_i^{(n)}\}$) leads to improvements in conservation behavior, no such advantage is obtained by formulating the (total) energy density $E^{(n)}$ as Roe variable. As such the original form of the energy equation is maintained. The (partial) Roe variable conservation laws are therefore:

$$\begin{aligned} \frac{\partial\sqrt{\rho^{(n)}}}{\partial t} + \frac{1}{2}\mathbb{L}_{\mathbf{u}}(\sqrt{\rho^{(n)}}) &= 0 \\ \frac{\partial\langle\partial_i, \phi^{(1,n)}\rangle}{\partial t} + \frac{1}{2}\mathbb{L}_{\mathbf{u}}(\langle\partial_i, \phi^{(1,n)}\rangle) + \frac{1}{\sqrt{\rho^{(0)}}}di_{\partial_i}p^{(n)} &= 0 \\ \frac{\partial E^{(n)}}{\partial t} + \mathcal{L}_{\mathbf{u}}E^{(n)} + \mathcal{L}_{\mathbf{u}}p^{(n)} &= 0 \end{aligned} \quad (3.21)$$

Where:

$$\mathbb{L}_{\mathbf{u}}(\sqrt{\rho^{(n)}}) = \mathcal{L}_{\mathbf{u}}(\sqrt{\rho^{(0)}})\sigma^{(n)} + \mathcal{L}_{\mathbf{u}}(\sqrt{\rho^{(0)}}\sigma^{(n)}) \quad (3.22)$$

And a similar relation holding for $\mathbb{L}_{\mathbf{u}}(\phi_i^{(n)})$. As can be seen some instances of $\sqrt{\rho^{(0)}}$ and $\phi_i^{(0)}$ still remain in (3.21) and (3.22). It can be noted that $\sqrt{\rho^{(0)}} = \star\sqrt{\rho^{(0)}}\sigma^{(n)}$ and $\phi_i^{(0)} = \star\phi_i^{(0)}\sigma^{(n)}$. Furthermore we note that:

$$\mathcal{L}_{\mathbf{u}}(\sqrt{\rho^{(0)}})\sigma^{(n)} = \star\mathcal{L}_{\mathbf{u}}(\star\sqrt{\rho^{(0)}}\sigma^{(n)})$$

And similar for $\mathcal{L}_{\mathbf{u}}(\phi_i^{(0)})\sigma^{(n)}$.

As remarked by Kreeft $\star\mathcal{L}_{\mathbf{u}}(\star\sqrt{\rho^{(0)}}\sigma^{(n)})$ is equal to the negative of the adjoint Lie derivative, since [31, p. 228]:

$$\star\mathcal{L}_{\mathbf{u}}(\star\sqrt{\rho^{(0)}}\sigma^{(n)}) = -\mathcal{L}_{\mathbf{u}}(\star\sqrt{\rho^{(0)}}\sigma^{(n)})$$

Noting that $\star\star\alpha^{(k)} = (-1)^{k(n-k)}\alpha^{(k)}$ for any k -form $\alpha^{(k)} \in \Lambda^{(k)}(\Omega)$ on n -dimensional manifold Ω it follows that:

$$\mathcal{L}_{\mathbf{u}}(\sqrt{\rho^{(0)}})\sigma^{(n)} = \star\mathcal{L}_{\mathbf{u}}(\star\sqrt{\rho^{(0)}}\sigma^{(n)}) = -\mathcal{L}_{\mathbf{u}}(\star\sqrt{\rho^{(0)}}\sigma^{(n)})$$

Where $\mathcal{L}_{\mathbf{u}}$ is the adjoint Lie derivative. Advection operator $\mathbb{L}_{\mathbf{u}}$ given in (3.22) can thus be stated as:

$$\mathbb{L}_{\mathbf{u}}(\sqrt{\rho^{(n)}}) = \mathcal{L}_{\mathbf{u}}(\sqrt{\rho^{(0)}})\sigma^{(n)} + \mathcal{L}_{\mathbf{u}}(\sqrt{\rho^{(0)}}\sigma^{(n)}) = \mathcal{L}_{\mathbf{u}}(\sqrt{\rho^{(0)}}\sigma^{(n)}) - \mathcal{L}_{\mathbf{u}}(\sqrt{\rho^{(0)}}\sigma^{(n)})$$

As such the combined advection operator $\mathbb{L}_{\mathbf{u}}$ is self-adjoint. Applying the aforementioned relations to (3.21) results in:

$$\begin{aligned} \frac{\partial\sqrt{\rho^{(n)}}}{\partial t} + \frac{1}{2}\star\mathcal{L}_{\mathbf{u}}(\star\sqrt{\rho^{(n)}}) + \frac{1}{2}\mathcal{L}_{\mathbf{u}}(\sqrt{\rho^{(n)}}) &= 0 \\ \frac{\partial\langle\partial_i, \phi^{(1,n)}\rangle}{\partial t} + \frac{1}{2}\star\mathcal{L}_{\mathbf{u}}(\star\langle\partial_i, \phi^{(1,n)}\rangle) + \frac{1}{2}\mathcal{L}_{\mathbf{u}}(\phi_i^{(n)}) + \frac{1}{\star\sqrt{\rho^{(0)}}}di_{\partial_i}p^{(n)} &= 0 \\ \frac{\partial E^{(n)}}{\partial t} + \mathcal{L}_{\mathbf{u}}E^{(n)} + \mathcal{L}_{\mathbf{u}}p^{(n)} &= 0 \end{aligned} \quad (3.23)$$

Advection operator \mathbb{L}_u thus satisfies the requirement that was set out in section 3.4: The Roe-decomposed continuity and momentum equations have the same anti-self-adjoint advection operator, making it possible to construct skew-symmetric discrete advection operators that simultaneously conserve mass, momentum and kinetic energy.

Velocity field \mathbf{u} is defined in a similar way as is the case in the non-Roe variable model given in (3.23):

$$\begin{aligned} \mathbf{u}^b &= \frac{\star\phi^{(1,n)}}{\star\sqrt{\rho}^{(n)}} = \frac{\sum_i dx^i \otimes \star(\phi_i dx^1 \wedge dx^2 \wedge \dots \wedge dx^n)}{\star(\sqrt{\rho} dx^1 \wedge dx^2 \wedge \dots \wedge dx^n)} \\ &= \frac{\sqrt{g} \sum_i dx^i \otimes \phi_i}{\sqrt{g} \sqrt{\rho}} \\ &= \sum_i dx^i \otimes \frac{\phi_i}{\sqrt{\rho}} \end{aligned}$$

Where again the metric contributions of the Hodge \star cancel.

The ideal gas law is again used as equation of state. Due to the Roe variables the definition of the kinetic energy is more straightforward, as it is now a quadratic quantity:

$$E_{kin}^{(n)} = \frac{1}{2} \langle m^{(1,n)}, \star^\# m^{(1,n)} \rangle \quad (3.24)$$

3.6. Aspects of discretization

Chapter 2 already introduced the mimetic isogeometric discretization methods and operators that will be used in this thesis. Having introduced most of the relevant theory and defined the continuous model with a Roe variable decomposition, several aspects of this discretization process are yet to be covered. While discretization of the Roe variable models is covered in chapters 7 (in one dimension) and 8 (in two dimensions), some of these aspects are also necessary for describing the discretization steps that have been used for the linear advection equation in chapter 5 and Burgers' equation in chapter 6; as such they are covered here. First is the approach for reconstructing physical variables from the Roe variables in section 3.6.1, followed by the time stepping methods that have been used in section 3.6.2. Lastly section 3.6.3 covers the linearization method that has been used to convert systems (3.8) and (3.23) into linear systems of equations.

3.6.1. Discrete reconstruction of physical variables

Isogeometric analysis employs B-splines to model solutions to physical continuum phenomena. For the model that is partially decomposed in Roe variables the numerical solution does not directly correspond to physical variables; these thus need to be reconstructed from the Roe variables. On manifold $\Omega \subset \mathbb{R}^2$ with cell complex Ω_h the partial Roe variable vector is $[\sqrt{\rho}^{(2)} \ \phi_1^{(2)} \ \phi_2^{(2)} \ E^{(2)}]$; thus the mass density, momentum density and kinetic energy density need to be reconstructed in some way. This can be done in an exact way with B-splines. Let (Ξ, Θ) be a tuple of knot vectors, each of length $m + p + 1$ and generating 2-form basis $\{\psi_i^{(2)}\}_{i=1}^{i=m+1}$, such that:

$$\begin{aligned} \sqrt{\rho}_h^{(2)} &= \sum_i \sqrt{\rho}_i \psi_i^{(2)} dx \wedge dy \\ \phi_{1,h}^{(2)} &= \sum_i \phi_i^1 \psi_i^{(2)} dx \wedge dy \\ \phi_{2,h}^{(2)} &= \sum_i \phi_i^2 \psi_i^{(2)} dx \wedge dy \end{aligned}$$

Note that it is required to expand all Roe variables in the same basis in order to preserve the simultaneous conservation of mass, momentum and kinetic energy. The discrete mass density field $\rho_h^{(2)}$ can be computed as follows:

$$\rho_h^{(2)} = \sqrt{\rho}_h^{(2)} \wedge \star \sqrt{\rho}_h^{(2)} = \sum_i \sum_j \sqrt{\rho}_i \sqrt{\rho}_j \psi_i^{(2)} \psi_j^{(2)} dx \wedge dy$$

Section 2.2.4 defines the algorithm put forth by Mørken in [36] that is used to exactly compute the product of any pair of B-splines as a higher-degree B-spline. For $\sqrt{\rho_h^{(2)}} \in \Lambda_{h,p}^{(2)}(\Omega_h)$ this method provides a map $\Lambda_{h,p}^{(2)}(\Omega_h) \times \Lambda_{h,p}^{(2)}(\Omega_h) \rightarrow \Lambda_{h,2p}^{(2)}(\Omega_h)$ for computing $\rho_h^{(2)} \in \Lambda_{h,2p}^{(2)}(\Omega_h)$. In the same way the 2-form legs of the covector-valued momentum 2-form $m^{(1,2)}$ can be calculated:

$$m_{1,h}^{(2)} = \sqrt{\rho_h^{(2)}} \wedge \star \phi_{1,h}^{(2)} = \sum_i \sum_j \sqrt{\rho_i} \phi_j^1 \psi_i^{(2)} \psi_j^{(2)} dx \wedge dy$$

$$m_{2,h}^{(2)} = \sqrt{\rho_h^{(2)}} \wedge \star \phi_{2,h}^{(2)} = \sum_i \sum_j \sqrt{\rho_i} \phi_j^1 \psi_i^{(2)} \psi_j^{(2)} dx \wedge dy$$

In this way the primary variables of the Euler equations can be reconstructed from the Roe variables that were defined in the previous section. As mentioned before, using this Roe variable decomposition leads to conservation of quadratic variables. With this way of exactly reconstructing the quadratic variables from the Roe variables the (quadratic) physical variables can be described as B-spline as well.

3.6.2. Time stepping

Spatial discretization presents only a portion of the discretization steps that have to be taken to arrive at a numerical method. Any advantageous properties of the spatial discretization need to be supported by appropriate time stepping methods, otherwise the conservation behavior found in the semi-discrete PDEs will not be present in the fully-discrete problem. Two different time stepping methods are used for the numerical methods derived in this research work. Before specifying these, their general class of time stepping methods is defined:

Definition 3.1. *Suppose we have a time-dependent variable $u(t)$ defined on domain $\Omega \times [0, T]$ with:*

$$\frac{\partial u}{\partial t} = F(t, u)$$

And let $t^{n+1}, t^n \in [0, T]$ (with $\Delta t = t^{n+1} - t^n$) be two time levels. Then an s -stage **Runge-Kutta method** used to find $u(t^{n+1}) = u^{n+1}$ from $u(t^n) = u^n$ is:

$$u^{n+1} = u^n + \Delta t \sum_{i=1}^s b_i k_i$$

Where s is the number of stages of the method, with:

$$k_i = F\left(t^n + c_i \Delta t, u^n + \Delta t \sum_{j=1}^s a_{ij} k_j\right)$$

The coefficients $b_i, c_i, a_{ij} \in \mathbb{R}$ fully specify the Runge-Kutta method used and are usually given in a **Butcher tableau**:

c_1	a_{11}	a_{12}	\dots	a_{1s}
c_2	a_{21}	a_{22}	\dots	a_{2s}
\vdots	\vdots	\vdots	\ddots	\vdots
c_s	a_{s1}	a_{s2}	\dots	a_{ss}
	b_1	b_2	\dots	b_s

Remark. *Runge-Kutta time stepping methods can be divided into implicit and explicit methods. For an explicit Runge-Kutta method with s stages the coefficients a_{ij} are equal to zero for $s \geq j \geq i$. As a consequence of this the j^{th} time stepping variable k_j can be computed from the variables k_i , with $0 \geq i < j$, in a fully decoupled way. Explicit Runge-Kutta methods thus allow one to generate the variables $k_1 \rightarrow k_2 \rightarrow \dots \rightarrow k_s$ successively. In contrast, implicit Runge-Kutta require one to solve a system of equations for the variables k_i . The higher the number of stages, the larger resulting linear system that needs to be solved for.*

The first of the methods used in this thesis is the implicit midpoint method:

Definition 3.2 (Sanderse [49]). *Suppose we have a time-dependent variable $u(t)$ defined on domain Ω with:*

$$\frac{\partial u}{\partial t} = F(u)$$

We define time levels t^{n+1} , t^n (with $\Delta t = t^{n+1} - t^n$) and apply a one-stage ($s = 1$) Runge-Kutta method with the following Butcher tableau:

$$\begin{array}{c|c} \frac{1}{2} & \frac{1}{2} \\ \hline & 1 \end{array}$$

*This results in the **implicit midpoint method**:*

$$u^{n+1} = u^n + \Delta t k_1 \quad (3.25)$$

Where:

$$k_1 = F\left(t^n + \frac{\Delta t}{2}, u^n + \frac{\Delta t}{2} k_1\right) \quad (3.26)$$

*Which presents a (non)linear system that is to be solved and depends on the form of F . The implicit midpoint method is the one-stage method of the broader class of **Gauss methods**. This group of methods is of order $\mathcal{O}(2s)$; the implicit midpoint method uses one stage and is thus second-order accurate.*

Remark. *The way the implicit midpoint method has been introduced here differs from the definitions often given in literature. To see that they are identical, reorder (3.25):*

$$k_1 = \frac{u^{n+1} - u^n}{\Delta t}$$

And apply this to the right-hand side of (3.26):

$$k_1 = F\left(t^n + \frac{\Delta t}{2}, u^n + \frac{\Delta t}{2} \frac{u^{n+1} - u^n}{\Delta t}\right) = F\left(\frac{1}{2}[t^{n+1} + t^n], \frac{1}{2}[u^{n+1} + u^n]\right)$$

The implicit midpoint method has been used in literature for structure-preserving discretizations of various flow types, including incompressible [30, 40, 41] and compressible [6] flows. As covered by Sanderse in [49] Gauss time stepping methods are energy-conserving, time-reversible and algebraically stable.

While carrying out numerical experiments with the implicit midpoint method applied to the two-dimensional Euler equations it was found that unphysical oscillations were propagating across the domain and polluting the numerical solutions. Hence it was decided to use a different time stepping approach, the Radau IIB method:

Definition 3.3 (Sanderse [49]). *Suppose that we again have a time-dependent variable $u(t)$, for which:*

$$\frac{\partial u}{\partial t} = F(u)$$

*The **Radau IIB method** is a two-stage implicit Runge-Kutta method with the following Butcher tableau:*

$$\begin{array}{c|cc} \frac{1}{3} & \frac{3}{8} & -\frac{1}{24} \\ 1 & \frac{7}{8} & \frac{1}{8} \\ \hline & \frac{3}{4} & \frac{1}{4} \end{array}$$

3.6.3. Linearization and nonlinear convergence

Depending on the formulation of the right-hand side function F introduced in the previous section, applying a time stepping method can result in a nonlinear scheme for u^{n+1} . This is the case in chapters 6, 7 and 8 of this thesis. Various approaches exist for linearizing a nonlinear model. In this work Picard linearization is used. Suppose that the following discrete system of equations has been obtained:

$$A(\mathbf{u})\mathbf{u} = \mathbf{f}$$

Where $A(\mathbf{u})$ is a matrix depending on solution vector \mathbf{u} and \mathbf{f} is the right-hand-side source vector. Picard linearization is an iterative method. Let k denote the index of the current iteration, such that \mathbf{u}_k is a known estimate for the nonlinear solution. Then Picard linearization works as follows:

$$A(\mathbf{u}_k)\mathbf{u}_{k+1} = \mathbf{f}$$

A new nonlinear solution estimate \mathbf{u}_{k+1} is thus obtained by applying current estimate \mathbf{u}_k to the system of equations where necessary. This results in a sequence of nonlinear solution estimates $\mathbf{u}_0 \rightarrow \mathbf{u}_1 \rightarrow \dots$. In this research the Picard iterations are considered to have converged when:

$$\|\mathbf{u}_{k+1} - \mathbf{u}_k\|_\infty < \epsilon$$

Where ϵ is the nonlinear convergence tolerance. A simulation (during any time step) is thus deemed to have converged when the maximum absolute difference between successive solution estimates $\mathbf{u}_k, \mathbf{u}_{k+1}$ is smaller than ϵ .

The Euler equations given in (3.8) and its partial Roe variable decomposition given in (3.23) both pose a coupled nonlinear system of equations. Since the nonlinearity for both models is contained in the velocity vector field \mathbf{u} , Picard linearization can be implemented in a straightforward way to yield a system of linear equations. Linearizing through \mathbf{u} removes the direct coupling between the continuity equation on one side and the momentum & energy equations on the other side. Letting k again denote the iteration index, the linearized regular Euler model becomes:

$$\begin{aligned} \frac{\partial \rho_{k+1}^{(n)}}{\partial t} + \mathcal{L}_{\mathbf{u}_k} \rho_{k+1}^{(n)} &= 0 \\ \frac{\partial \langle m_{k+1}^{(1,n)}, \partial_i \rangle}{\partial t} + \mathcal{L}_{\mathbf{u}_k} \langle m_{k+1}^{(1,n)}, \partial_i \rangle + (\gamma - 1) di \partial_i \left(E_{k+1}^{(n)} - \left(E_{kin}^{(n)} \right)_k \right) &= 0 \\ \frac{\partial E_{k+1}^{(n)}}{\partial t} + \gamma \mathcal{L}_{\mathbf{u}_k} E_{k+1}^{(n)} - (\gamma - 1) \mathcal{L}_{\mathbf{u}_k} \left(E_{kin}^{(n)} \right)_k &= 0 \end{aligned} \quad (3.27)$$

Where thus the only parameter linking the continuity equation to the momentum equations is the velocity vector field \mathbf{u}_h . For the Roe variable model the linearized system is:

$$\begin{aligned} \frac{\partial \sqrt{\rho}_{k+1}^{(n)}}{\partial t} + \frac{1}{2} \star \mathcal{L}_{\mathbf{u}_k} \left(\star \sqrt{\rho}_{k+1}^{(n)} \right) + \frac{1}{2} \mathcal{L}_{\mathbf{u}_k} \left(\sqrt{\rho}_{k+1}^{(n)} \right) &= 0 \\ \frac{\partial \langle \partial_i, \phi_{k+1}^{(1,n)} \rangle}{\partial t} + \frac{1}{2} \star \mathcal{L}_{\mathbf{u}_k} \left(\star \langle \partial_i, \phi_{k+1}^{(1,n)} \rangle \right) + \frac{1}{2} \mathcal{L}_{\mathbf{u}_k} \left(\langle \partial_i, \phi_{k+1}^{(1,n)} \rangle \right) + (\gamma - 1) \frac{1}{\star \sqrt{\rho}_{k+1}^{(0)}} di \partial_i \left(E_{k+1}^{(n)} - \left(E_{kin}^{(n)} \right)_k \right) &= 0 \\ \frac{\partial E_{k+1}^{(n)}}{\partial t} + \gamma \mathcal{L}_{\mathbf{u}_k} E_{k+1}^{(n)} - (\gamma - 1) \mathcal{L}_{\mathbf{u}_k} \left(E_{kin}^{(n)} \right)_k &= 0 \end{aligned} \quad (3.28)$$

Here the same is observed as for the regular model: The continuity and momentum equations become decoupled. Note that the last term in the momentum equation contains $1/\star \sqrt{\rho}_{k+1}^{(0)}$: This is required for momentum conservation. During any nonlinear iteration the continuity equation is thus to be solved for first, after which the updated solution $\sqrt{\rho}_{k+1}^{(0)}$ is used to construct the momentum equation.

Also note that in both models the kinetic energy $\left(E_{kin}^{(n)} \right)_k$ is resolved in an explicit way: This quantity is non-linear in itself, so resolving it implicitly in a fully linear way is not possible.

4

Reference models

Two external models have been used in the current research to benchmark the performance of the newly-proposed mimetic isogeometric finite element (FE) method. One of these is a package implementing the finite volume (FV) method called Clawpack [34] and the other is the nodal Discontinuous Galerkin (DG) method treated by Hesthaven and Warburton in [24]. One way to look at the relations between these three methods is to view the DG method as a hybrid between FV methods on the one hand and FE methods, under which the method derived in this research work falls, on the other hand. Figure 4.1 shows a Venn diagram that displays this relation.

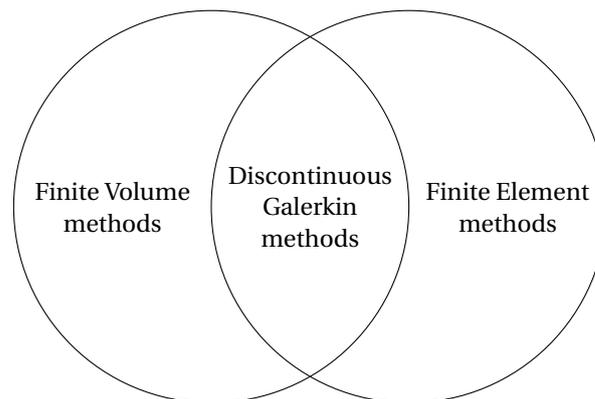


Figure 4.1: Venn diagram showing the relation between the various types of numerical discretization approaches covered in this research work; Discontinuous Galerkin methods combine aspects and ideas of FV- and FE methods

This chapter covers some of the basic theory of the aforementioned external methods in order to contrast their derivation and (dis)advantages for various types of physical models. To allow for a fair comparison with the method developed in this research work numerical aspects such as filtering, time stepping and boundary condition implementations are also covered. Section 4.1 will first cover the Clawpack FV package; part of this theory will come in useful in section 4.2, where the nodal DG method is covered. Following this section 4.3 briefly introduces the grid types that are used in the test cases on which both Clawpack and the DG method were used.

4.1. Clawpack

Clawpack¹ (short for Conservation Laws Package) [34] is an open-source package that implements a range of FV methods for linear and nonlinear hyperbolic systems of conservation laws. Section 4.1.1 will first cover the fundamental approach taken in finite volume discretization methods, after which several aspects of this approach are covered separately in order to provide a transparent overview of the algorithms from Clawpack

¹<http://www.clawpack.org/>

used as reference in this research. These aspects are time stepping, discrete flux functions & flux limiting and boundary condition implementation, covered in sections 4.1.2, 4.1.3 and 4.1.4 respectively. Whereas the FV discretization approach is presented here for one spatial dimension the same method can be used in any number of dimensions; wherever appropriate the extension from one to multiple spatial dimensions will be mentioned briefly. Publications discussing FV methods are numerous: Among others the books by R. Leveque [32, 33] provide a much more thorough derivation and theoretical background. Clawpack was used to create of the numerical examples and results given in the latter of these references.

4.1.1. Finite volume discretization

Consider the following equation on finite domain $\Omega \subset \mathbb{R}$:

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

Where u is an unknown variable and F is a (possibly nonlinear) flux. This equation is supplemented with a suitable initial condition and boundary values on $\partial\Omega$. We cover Ω with a finite set of non-overlapping cells $\{\Omega_k\}_{k=1}^{k=N}$ and integrate (4.1) over cell Ω_k :

$$\begin{aligned} \int_{\Omega_k} \frac{\partial u}{\partial t} d\Omega_k + \int_{\Omega_k} \frac{\partial F(u)}{\partial x} d\Omega_k &= 0 \\ \frac{\partial}{\partial t} \underbrace{\int_{\Omega_k} u d\Omega_k}_{=U_k} + F(u) \Big|_{\partial\Omega_k} &= 0 \\ \frac{\partial U_k}{\partial t} + F(u) \Big|_{\partial\Omega_k} &= 0 \end{aligned}$$

Where the divergence theorem was used to convert the integral of $\frac{\partial F}{\partial x}$ to a boundary evaluation. We define the degree of freedom associated with Ω_k to be the area-averaged integral of u over Ω_k , which we denote with U_k . The only things that are still needed to obtain a numerical method are a time stepping method and some way to compute $F(u) \Big|_{\partial\Omega_k}$, the flux operator on the two boundary sides of Ω_k . In multiple dimensions this boundary term is an integral over $\partial\Omega_k$, as per the divergence theorem. We denote the interface between cells Ω_k, Ω_{k+1} with $\partial\Omega_{k+1/2}$; note that the flux operator evaluated on this interface $F(U_{k+1/2})$ contributes positively to U_{k+1} and negatively to U_k or vice versa. We do not know the values of u on $\partial\Omega$. For cell Ω_k we thus need a way to compute the discrete flux functions $F(U_{k-1/2})$ and $F(U_{k+1/2})$ from the cell integrals $U_i, i \in [0, 1, \dots, N]$. This leads to the following semi-discrete formulation for FV methods:

$$\frac{\partial U_k}{\partial t} + \underbrace{F(U_{k+1/2})}_{=F_{k+1/2}} - \underbrace{F(U_{k-1/2})}_{=F_{k-1/2}} = 0$$

In other words: The time rate of change of U_k is equal to the difference between the flux entering and exiting Ω_k . Most finite volume methods follow the so-called REA approach to march a numerical solution from time level t^n to t^{n+1} : Reconstruct, Evolve, Average. This approach works as follows:

1. **Reconstruct:** Use the area-averaged integral degrees of freedom $U_k, k \in [0, 1, \dots, N]$ (which generate a piecewise-constant solution) to approximate or otherwise reconstruct the boundary fluxes $F_{k-1/2}, F_{k+1/2}$ for every cell k .
2. **Evolve:** Use the hyperbolic structure of advection to extrude the boundary fluxes $F_{k-1/2}(t), F_{k+1/2}(t)$ in and out of every cell boundary over the time step $t^n \rightarrow t^{n+1}$.
3. **Average:** Compute the area-averaged integral degrees of freedom U_k^{n+1} at time level $n+1$ from the evolved solution; the result is a new piecewise-constant solution with discontinuities on the cell boundaries. This solution can be used in the next reconstruction step.

This is the general approach with which FV methods resolve discretized conservation laws. Local conservation of the primary variables is achieved in each cell: U_k changes only because of the net flow across $\partial\Omega_k$. Similarly, U_{k+1} receives an inflow across $\partial\Omega_{k+1/2}$ that is exactly equal to the outflow of U_k across $\partial\Omega_{k+1/2}$,

dictated by the (discrete) flux function $F_{k+1/2}$.

The concepts covered here all generalize to two-, three- and higher-dimensional cases. Cell boundary ∂C_k will then consist of multiple patches, and flux functions are defined on each boundary patch separately. Leveque discusses these multidimensional extensions in chapter 18 of [32]. In case unknown u is a vector of variables instead of a scalar, U_k will also be a vector. The fluxes F of each variable will be different, though they all depend on the local velocity field for their extrusion.

4.1.2. Time stepping

Now that the spatial discretization approach has been introduced, we are left to discuss time discretization. With this the time steps are defined over which the boundary fluxes $F_{k-1/2}(t)$, $F_{k+1/2}(t)$ are extruded. Within this research Clawpack's implementation of the Lax-Wendroff method is used to generate the numerical solutions. This is a classic second-order method in space and time, and is discussed by Leveque in [33]. It is an explicit time stepping method; in other words, the solution at time level $n + 1$ is computed based solely on the solution and discrete fluxes computed at time level t^n . Since the Lax-Wendroff method is a higher-order method it displays dispersive behavior around discontinuities. To remedy this a flux limiter will be used to correct the discrete fluxes that result from the flux function formulation.

4.1.3. Flux functions and limiting

A great deal of research has gone into obtaining suitable flux functions and many have been proposed during the over sixty years of research into Finite Volume methods. Leveque gives an overview of many of the classical flux functions in [32, 33]; as mentioned previously, Clawpack was used to simulate many of the examples given in the latter reference. As it turns out the flux functions that produce monotone solutions in regions with high gradients and discontinuities cannot be more than first-order accurate [32, Thm. 15.6]. This first-order scaling behavior means that excessively fine meshes are required to resolve numerical simulations with a higher degree of accuracy, limiting the usefulness of these methods due to the required computational costs. The cause of this is the high amount of numerical dissipation produced by these monotone methods. Higher-order-accurate flux functions exist but display dispersive (oscillatory) behavior in high-gradient regions and around discontinuities. So-called 'high-resolution' schemes were created in an attempt to combine the advantages offered by higher-order schemes with the stability and monotonicity of monotone first-order flux functions. This section first covers the high-order flux functions that are used for the one-dimensional Euler equations, followed by some steps that result in a high-resolution scheme. Afterwards the schemes used for the two-dimensional Euler equations are briefly treated. The aim of this section is to provide a brief overview of the functions used from Clawpack to obtain the reference results that are given in chapters 7 and 8; a more elaborate discussion and derivation of these methods can be found in [32, 33].

In this research the high-order fluxes for the one-dimensional Euler equations are computed with the Harten-Lax-van Leer-Contact (HLLC) Riemann solver. This solver is an extension of the Harten-Lax-van Leer (HLL) method originally published in [22]. Riemann solvers are numerical methods used to solve the so-called Riemann problem, a (system of) conservation equation(s) with piecewise-constant initial conditions with a single discontinuity at $x = x_0$. This discontinuous initial condition results in a pattern of left- and right-running shock waves and discontinuities that each follow a characteristic direction of the underlying (system of) conservation law(s). The HLL method models this structure by only considering the fastest left- and right-running waves, resulting in three distinct regions around the discontinuity location x_0 .

If we let $x_0 = \partial\Omega_{k+1/2}$ and take U_k , U_{k+1} as initial conditions (recall that the degrees of freedom are cell-averaged and thus piecewise-constant) the discrete flux $F_{k+1/2}$ can be computed with the HLL Riemann solver. A weakness of the HLL method is that it does not take into account any waves between the fastest left- and right-running waves emanating from x_0 . This results in the absence of certain types of waves and the production of numerical diffusion, leading to certain waves experiencing unphysical damping. Various extensions have been proposed to remedy this, one of which is the Harten-Lax-van Leer-Contact (HLLC) Riemann solver that is used in this research work. This method was originally published by Toro et al. in [54] and supposes the existence of a contact wave between the fastest left- and right-running waves, all emanating from x_0 . Incidentally, as can be seen in chapter 7, this three-wave structure corresponds to the exact solution of the Euler equations for Sod's shock tube.

As mentioned this Riemann solver is used to compute the discrete fluxes $F_{k-1/2}, F_{k+1/2}$ at the cell boundaries $\partial\Omega_k$, which are then used in the Lax-Wendroff method to march the numerical solution between time levels $t^n \rightarrow t^{n+1}$. As the Lax-Wendroff method is second-order-accurate it is dispersive around discontinuities; to avoid this a high-resolution method is desired. These require a higher-order flux function $F_H(U)$ (such as the HLL or HLLC solvers mentioned above), a low-order flux function $F_L(U)$ (Clawpack uses the upwind scheme) and a flux limiter $\phi(U)$. The high-resolution flux function $F(U)$ at a cell boundary is then defined as:

$$F(U) = F_L(U) + \phi(U) (F_H(U) - F_L(U))$$

In other words: The high-resolution flux function F can be seen as the low-order flux function plus a higher-order correction. Flux limiter ϕ thus controls whether F_L or F_H dominates the high-resolution flux locally; the aim is to use F_H in smoother regions (because of its higher-order accuracy), and to switch to F_L in high-gradient regions and around discontinuities (because of its monotone behavior and stability). In this research Clawpack's implementation of the minmod limiter is used.

Definition 4.1 (Leveque [32, Eq. 16.53]). *Let r be a function that signals whether the local data is smooth. The **minmod limiter** ϕ_{mm} is defined as:*

$$\phi_{mm} = \max(0, \min(1, r))$$

Remark (Leveque [32, Eq. 16.15]). *To determine the high-resolution flux $F(U_{k+1/2}) = F_{k+1/2}$ at cell boundary $k + 1/2$ the signal function r_k is used; a possible formulation is based on the ratios of neighboring gradients:*

$$r_k = \frac{U_k - U_{k-1}}{U_{k+1} - U_k}$$

4.1.4. Boundary condition application

The zero-flux boundary conditions that are used in the one-dimensional test case (Sod's shock tube) can be applied to the FV method by simply setting the discrete flux functions on $\partial\Omega$ to zero.

Similarly, periodic boundary conditions can be imposed through the discrete flux functions as well. Suppose that a one-dimensional domain Ω is covered by cells Ω_k , $k = 1, 2, \dots, n$. The discrete flux functions that are used to compute the fluxes across the cell boundaries $\partial\Omega_{k+1/2}$ are also imposed on the cell boundaries that lie on $\partial\Omega$: $\partial\Omega_{n+1/2}$ on the right side of Ω and $\partial\Omega_{1/2}$ on the left side of the domain. Imposing that $\partial\Omega_{n+1/2} = \partial\Omega_{1/2}$ allows us to apply a discrete flux function on said cell boundaries that makes $\partial\Omega = \emptyset$.

The same can be done for a two-dimensional domain. Suppose that Ω is a square two-dimensional domain that is covered by cells $\Omega_{k,l}$, $k = 1, 2, \dots, n$, $l = 1, 2, \dots, m$. Then $\partial\Omega$ has boundaries on its north ($\partial\Omega_{k,m+1/2}$), east ($\partial\Omega_{n+1/2,l}$), south ($\partial\Omega_{k,1/2}$) and west ($\partial\Omega_{1/2,l}$) side. Letting $\partial\Omega_{1/2,l} = \partial\Omega_{n+1/2,l} \forall l \in 1, 2, \dots, m$ and $\partial\Omega_{k,1/2} = \partial\Omega_{k,m+1/2} \forall l \in 1, 2, \dots, n$ then results in a two-dimensional domain that's periodic in both directions.

4.2. Nodal Discontinuous Galerkin

The Discontinuous Galerkin method used as reference in this research is the nodal DG method given by Hesthaven and Warburton in [24]. Written below is a brief overview of some of the fundamentals of said DG method: Section 4.2.1 gives a general derivation of the DG approach, after which section 4.2.2 goes into some more detail about the flux functions that have been used in this research. The final three aspects covered here are the time stepping methods in section 4.2.3, filtering and flux limiting in section 4.2.4 and lastly the application of boundary conditions in section 4.2.5. A GitHub repository² contains an implementation of the nodal DG method as described by Hesthaven and Warburton in [24].

4.2.1. Discontinuous Galerkin discretization

Suppose that we want to simulate (4.1) again. We cover Ω with a set of non-overlapping cells Ω_k and suppose that the solution u on each cell Ω_k can be expanded in basis functions:

$$u_h^k(x, t) = \sum_{n=1}^{N_p} \hat{u}_n^k(t) \psi_n(x) \quad (4.2)$$

²<https://github.com/tcew/nodal-dg>

Where $\psi_n(x)$ is a local polynomial up to degree p ($N_p = p + 1$, $p = 0$ corresponds to a piecewise-constant solution). We multiply (4.1) with test function $\psi_n(x)$ and integrate over Ω_k to obtain a weak form:

$$\int_{\Omega_k} \frac{\partial u_h^k}{\partial t} \psi_n d\Omega_k + \int_{\Omega_k} \frac{\partial F(u_h^k)}{\partial x} \psi_n d\Omega_k = 0$$

Next we apply integration by parts to the second term on the left-hand side to arrive at:

$$\int_{\Omega_k} \frac{\partial u_h^k}{\partial t} \psi_n d\Omega_k - \int_{\Omega_k} \frac{\partial \psi_n}{\partial x} F(u_h^k) d\Omega_k = -F(u_h^k) \psi_n \Big|_{\partial\Omega_k} \quad (4.3)$$

Note that in contrast to FV methods we have a pointwise definition for u_h^k on Ω_k , given by basis function expansion (4.2). When gluing together the solutions defined on each cell we find that the solution on $\partial\Omega_k$ is multivalued: At the interface of cells Ω_k, Ω_{k+1} we have the solutions of u_h^k and u_h^{k+1} , both of which are equally valid. In order to decide on which solution is correct (or whether the correct value is a combination of both solutions) we introduce a flux function that depends on the local function values from u_h^k and u_h^{k+1} :

$$\int_{\Omega_k} \frac{\partial u_h^k}{\partial t} \psi_n d\Omega_k - \int_{\Omega_k} \frac{\partial \psi_n}{\partial x} F(u_h^k) d\Omega_k = -F^*(u_h^k) \psi_n \Big|_{\partial\Omega_k}$$

Where F^* is the imposed flux function. This defines a discrete weak form: The integrals are computed numerically with an appropriate quadrature rule and the coefficients \hat{u}_n^k are solved for.

What makes this DG method a nodal DG method is the choice of basis functions. In (4.2) the basis functions ψ_n are shown; while these are local polynomials, an equivalency exists with a nodal form:

$$u_h^k(x, t) = \sum_{n=1}^{N_p} \hat{u}_n^k(t) \psi_n(x) = \sum_{i=1}^{N_p} u_i^k(x_i^k, t) l_i^k(x) \quad (4.4)$$

Where l_i^k are Lagrange polynomials that have been defined through the N_p grid points x_i^k on every cell. The expansion coefficients \hat{u}_n^k and u_i^k can be converted into one another, as is explained in greater detail by Hesthaven & Warburton in [24]. An effect of this is that increasing the degree p of the basis function expansion is handled slightly differently than for many other finite element methods. In order to retain the equivalence between both forms shown in (4.4) degree increases are additive; the basis used for an expansion of degree p contains polynomials of degrees $0, 1, \dots, p$ and uses $N_p = p + 1$ degrees of freedom on each cell. Increasing the basis function degree thus also leads to an increase in the number of degrees of freedom, a process called k -refinement. This is unlike the other methods considered within this work; Clawpack can only do h -refinement (i.e. using smaller cells), and the numerical methods presented in this thesis can do only h - and p -refinement (i.e. substituting the degree of the basis functions used for a different degree). Whereas k -refinement is also possible with the newly-developed methods introduced in this work, this has not been implemented.

4.2.2. Flux functions

Discontinuous Galerkin methods are similar to FV methods when it comes to handling cell interfaces: Flux functions are defined in order to describe the influence that neighboring cells have on one another. In the same way as for FV methods there is some freedom in selecting a flux function formulation. In this research the Lax-Friedrichs flux function is used:

Definition 4.2 (Hesthaven & Warburton [24, p. 32]). *Let $F_{LF,k+1/2}^*(u_h^k, u_h^{k+1})$ denote the one-dimensional **Lax-Friedrichs flux** on cell boundary $\partial\Omega_{k+1/2}$, which depends on the local function values $u_h^k \Big|_{\partial\Omega_{k+1/2}}$ (on the left side of the cell boundary) and $u_h^{k+1} \Big|_{\partial\Omega_{k+1/2}}$ (on the right side of the cell boundary). Then:*

$$F_{LF,k+1/2}^*(u_h^k, u_h^{k+1}) = \frac{F(u_h^k) + F(u_h^{k+1})}{2} + \frac{C}{2} (u_h^k \Big|_{\partial\Omega_{k+1/2}} - u_h^{k+1} \Big|_{\partial\Omega_{k+1/2}})$$

Where F is the flux formulation of the original PDE (as given in (4.1)) and $C \in \mathbb{R}$ is a parameter that can be chosen appropriately. The two-dimensional Lax-Friedrichs flux formulation is given on page 208 of [24].

Remark. In this research C is taken as the maximum local characteristic velocity, which corresponds to the highest eigenvalue of the local Euler equations:

$$C = \sqrt{\frac{\gamma p}{\rho} + \left| \frac{m}{\rho} \right|}$$

Where γ is the ratio of specific heats, p is pressure, ρ fluid mass density and m momentum density.

4.2.3. Time stepping

Similar to the methods presented in this thesis the nodal DG method uses Runge-Kutta (RK) time stepping methods. More specifically, as is covered in section 5.7 of [24] Strong Stability-Preserving (SSP) explicit Runge-Kutta methods (SSP-RK) are used. For the one-dimensional Euler equations the following time stepping method is used:

Definition 4.3 (Gottlieb et al. [21, Prop. 4.1]). Suppose that we again have a time-dependent variable $u(t)$, for which:

$$\frac{\partial u}{\partial t} = F(u)$$

The **third-order Strong Stability-Preserving Runge-Kutta** (SSP-RK) method is a three-stage explicit Runge-Kutta method with the following Butcher tableau:

0	0	0	0
1	1	0	0
$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{4}$	0
	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{2}{3}$

As its name implies this time stepping method has third-order accuracy and is strong stability-preserving, meaning that it does not add any additional oscillations to an existing semi-discrete method, making it very suitable for cases with large gradients or discontinuous solutions.

Note that this method requires simultaneous storage of three solution arrays for every variable. In a similar way the following method is used for the two-dimensional Euler equations:

Definition 4.4 (Hesthaven and Warburton [24, Table 3.2]). Assume that we again have a time-dependent variable $u(t)$, for which:

$$\frac{\partial u}{\partial t} = F(u)$$

The **fourth-order Low-Storage Runge-Kutta** (LSRK) method is a five-stage explicit Runge-Kutta method with the following Butcher tableau:

c_1	a_1	0	0	0	0
c_2	0	a_2	0	0	0
c_3	0	0	a_3	0	0
c_4	0	0	0	a_4	0
c_5	0	0	0	0	a_5
	b_1	b_2	b_3	b_4	b_5

Where:

i	a_i	b_i	c_i
1	0	$\frac{1432997174477}{9575080441755}$	0
2	$-\frac{567301805773}{1357537059087}$	$\frac{5161836677717}{13612068292357}$	$\frac{1432997174477}{9575080441755}$
3	$-\frac{2404267990393}{2016746695238}$	$\frac{1720146321549}{2090206949498}$	$\frac{2526269341429}{6820363962896}$
4	$-\frac{3550918686646}{2091501179385}$	$\frac{3134564353537}{4481467310338}$	$\frac{2006345519317}{3224310063776}$
5	$-\frac{1275806237668}{842570457699}$	$\frac{2277821191437}{14882151754819}$	$\frac{2802321613138}{2924317926251}$

These coefficients are also given in table 3.2 of [24]. Again, as implied by its name this is a fourth-order explicit Runge-Kutta method. It is not strong stability-preserving, but this is not necessary for the isentropic vortex test case considered in this thesis for the two-dimensional Euler equations.

The Butcher tableau of this low-storage Runge-Kutta method shows that only one solution array has to be stored for every variable during any given time step. This is especially useful for higher-dimensional problems, where the amount of degrees of freedom can increase rapidly when meshes are refined.

4.2.4. Filtering and limiting

In order to increase the stability of the nodal DG method a slope limiter and a residual filtering approach have been implemented for the one- and two-dimensional Euler equations. This slope limiter is based on a minmod function. Note that this function is unrelated to the minmod limiter that was introduced for Clawpack in section 4.1.3.

Definition 4.5 (Hesthaven and Warburton [24, Eq. 5.24]). *Let $a_1, a_2, \dots, a_m \in \mathbb{R}$ be m input arguments and let s denote the average of their sign functions:*

$$s = \frac{1}{m} \sum_{i=1}^m \text{sign}(a_i)$$

Where the sign function is a function that return 1 or -1 for positive and negative arguments respectively. Let m denote the **minmod function**:

$$m(a_1, a_2, \dots, a_m) = \begin{cases} s \min_{1 \leq i \leq m} |a_i| & |s| = 1 \\ 0 & \text{otherwise} \end{cases}$$

Remark. Note that m thus returns a nonzero argument if and only if all input arguments a_1, a_2, \dots, a_m have the same sign; for any other case $|s| \neq 1$. When all input arguments have the same sign m will return the smallest input argument, with the same sign it had before. As remarked by Hesthaven and Warburton [24, p. 150] this function can be used as a slope limiter by inputting the linear slopes of the solution on three neighboring cells; if the slope signs of these three cells do not match m returns a zero, otherwise m returns the shallowest slope.

The slope limiter for the one-dimensional Euler equations is then constructed by following a specific order of steps [24, p. 152]:

1. Compute a linear approximation to the solution u_h^k on each cell; this approximation is defined to be equal to the cell average \bar{u}_h^k in the centroid of the cell (as to conserve u_h^k) and has slope $(u_h^k)_x$. This approximation results in cell boundary values v_l^k and v_r^k on the left- and right-side cell boundaries respectively.
2. If the approximated cell boundary values v_l^k and v_r^k match u_h^k on the left- and right-side cell boundaries no limiting needs to be done.
3. Otherwise use the previously-computed linear approximation to u_h^k and apply the minmod function to the cell slope $(u_h^k)_x$ by giving the cell averages of cell k and its neighbors as input: $m(\bar{u}_h^{k-1}, \bar{u}_h^k, \bar{u}_h^{k+1})$.

Since the cell averages are not affected by this slope limiting the slope-limited solution can be computed in each cell independently.

The case used in this research for the two-dimensional Euler equations does not contain any discontinuities. Nonetheless a simple filter is used in order that stabilizes the DG method especially when under-resolved solution features exist. For this a cut-off filter is used. Recall the expansion in term of local polynomial basis functions given in (4.2). Reducing the amplitude of the coefficients associated with the highest-degree basis functions on each cell by only 5% leads to a more stable method. Hesthaven and Warburton give a comparison between filtered and unfiltered results for the two-dimensional isentropic vortex in [24, Sec. 6.6.1].

4.2.5. Boundary condition application

Recall that in the DG formulation the variational problems on each cell are posed separately, after which the cell-specific problems are glued together with the numerical flux functions. As can be seen in (4.3) these fluxes are the only boundary terms present, and hence the zero-flux wall boundaries used in the one-dimensional Euler test case can be enforced by setting the numerical flux functions equal to zero on the domain boundaries.

The periodic boundary conditions that are used in the test case for the two-dimensional Euler equations are posed in an identical way to those used in Clawpack: The meshes used are constructed such that the eastern & western and the northern & southern boundaries have one-to-one correspondence of mesh nodes and edges. The usual numerical flux functions can then be posed across cell edge pairs on opposing boundary sides.

4.3. Grids and degrees of freedom

Clawpack and the DG method have different arrangements for their grids and the degrees of freedom that are defined on them. For Clawpack the meshes used in this research are straightforward: In one dimension a one-dimensional grid consisting of intervals of identical length is defined. This structure is extended to two dimensions in a tensor product way; as such the two-dimensional grid is uniform and Cartesian. Each cell in one and two dimensions has one degree of freedom associated to it for each scalar variable that is being solved for.

The discrete configuration of the DG method is a bit more complicated and depends on the maximum basis function degree p . Again in one dimension a covering of the computational domain by a set of identically-sized cells (intervals) is used. On each cell $p + 1$ degrees of freedom are defined; one degree of freedom is associated to each basis function degree $[0, 1, \dots, p]$. As covered in section 4.2.1 this is a defining feature of nodal DG methods, since it retains a one-to-one correspondence between the local polynomials and Lagrange polynomial bases on each cell.

In two dimensions triangular grids have been used for the DG method. These are shown in figure 4.2. Each grid can be obtained from a uniform Cartesian mesh by dividing each cell in two along one of its diagonals. The periodic boundary conditions can thus be applied in a straightforward way, since there is a direct one-to-one correspondence of mesh edges on the domain boundary. On each cell in this grid $\frac{1}{2}(p + 1)(p + 2)$ two-dimensional degrees of freedom have been defined.

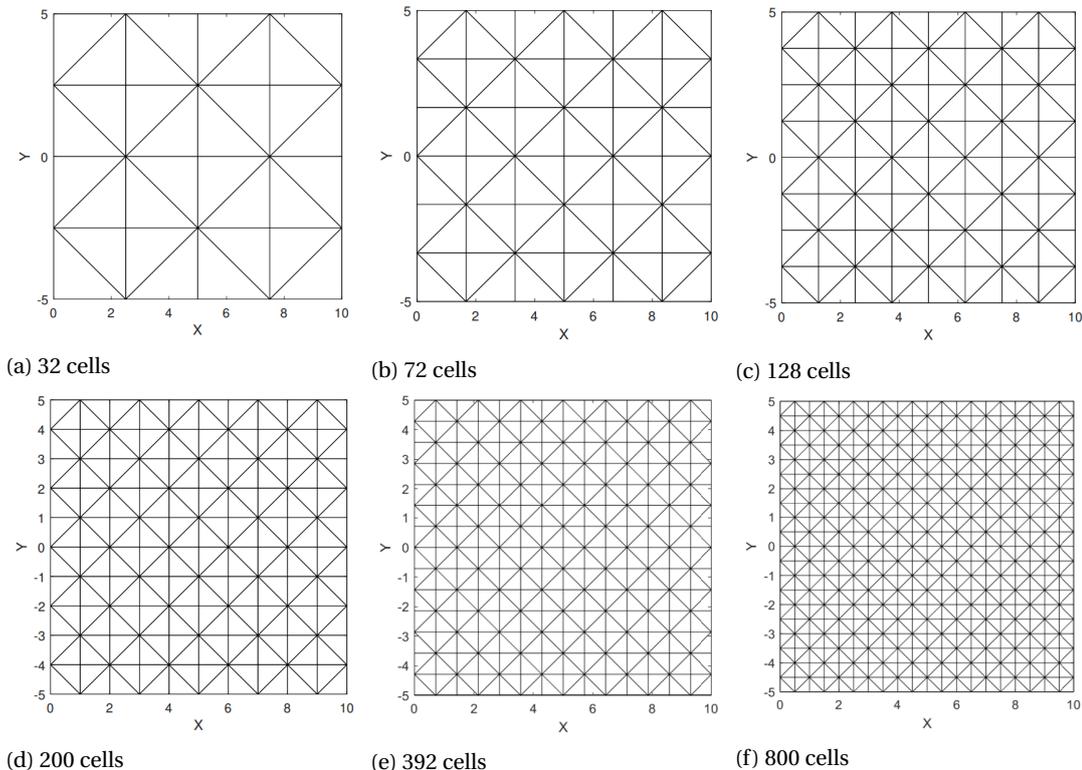


Figure 4.2: Various meshes used for the nodal Discontinuous Galerkin method to simulate the two-dimensional Euler equations in chapter 8

5

One-dimensional linear advection

A series of test cases have been selected in order to study the behavior of the proposed Euler model and discretization approach. Each test case described in this thesis increases the level of complexity with respect to previous tests. Relevant aspects of the proposed Euler model and its discretization are covered as they become relevant along the way. The first of these test cases is that of linear advection on periodic spatial domain $\Omega =]0, 1[$. In vector calculus notation the one-dimensional linear advection equation is written as:

$$\frac{\partial \alpha}{\partial t} + u \frac{\partial \alpha}{\partial x} = 0$$

This is a hyperbolic model problem that describes the physical (advective) transport of some quantity α due to an imposed velocity field u . Within this thesis the imposed advective velocity field will simply be a constant field of known value c . The exact solutions of the linear advection equation are well known. For a one-dimensional time-dependent solution $\alpha(x, t)$ and an imposed initial condition $\alpha(x, 0) = \alpha_0(x)$ on a periodic domain these are equal to:

$$\alpha(x, t) = \alpha(x - ct, 0) = \alpha_0(x - ct) \quad (5.1)$$

Section 5.1 will first define the two linear advection models that have been tested. After the definition of these models and the subsequent description of their discretization approach, two initial conditions are considered: The results obtained for the advection of a sine wave are covered in section 5.3. For this initial condition the error convergence and numerical conservation behavior are presented, establishing the accuracy and correctness of the implemented models.

5.1. Continuous models

Two formulations of the linear advection equation are considered on $\Omega =]0, 1[$:

$$\frac{\partial \alpha^{(1)}}{\partial t} + \mathcal{L}_{\mathbf{u}} \alpha^{(1)} = 0 \quad (5.2)$$

$$\frac{\partial \alpha^{(1)}}{\partial t} + \frac{1}{2} \mathcal{L}_{\mathbf{u}} \alpha^{(1)} + \frac{1}{2} \star \mathcal{L}_{\mathbf{u}} (\star \alpha^{(1)}) = 0 \quad (5.3)$$

Where $\alpha^{(1)} \in \Lambda^{(1)}(\Omega)$ is a one-form, \mathbf{u} is the advection velocity vector field, \mathcal{L} is the Lie derivative and \star is the Hodge \star operator. The difference between both models lies in their advection operators: As shown in example 5 the Lie derivative of a volume form (i.e. the advection operator in (5.2)) corresponds to the conservative formulation of the linear advection equation. On the other hand the advection operator in (5.3) is a combination of the Lie derivative and its adjoint. As will be shown in this chapter, this advection formulation results in a skew-symmetric discrete advection operator. For both models a weak form is obtained by taking the L^2 inner product (as defined in (1.19)) on Ω with one-form $v^{(1)} \in \Lambda^{(1)}(\Omega)$. This results in:

$$\left(v^{(1)}, \frac{\partial \alpha^{(1)}}{\partial t} \right)_{\Omega} + (v^{(1)}, \mathcal{L}_{\mathbf{u}} \alpha^{(1)})_{\Omega} = 0 \quad \forall v^{(1)} \in \Lambda^{(1)}(\Omega) \quad (5.4)$$

$$\left(v^{(1)}, \frac{\partial \alpha^{(1)}}{\partial t} \right)_{\Omega} + \frac{1}{2} (v^{(1)}, \mathcal{L}_{\mathbf{u}} \alpha^{(1)})_{\Omega} + \frac{1}{2} (v^{(1)}, \star \mathcal{L}_{\mathbf{u}} (\star \alpha^{(1)}))_{\Omega} = 0 \quad \forall v^{(1)} \in \Lambda^{(1)}(\Omega) \quad (5.5)$$

The presence of the Hodge \star operators in (5.5) implies undesirable metric dependence in the resulting discretization. To circumvent this the corresponding term will be manipulated before discretizing. First note that this term is equal to:

$$(v^{(1)}, \star \mathcal{L}_{\mathbf{u}}(\star \alpha^{(1)}))_{\Omega} = \int_{\Omega} v^{(1)} \wedge \star(\star \mathcal{L}_{\mathbf{u}}(\star \alpha^{(1)})) \quad (5.6)$$

From the properties of the Hodge \star operator it is known that $\star \star \beta^{(k)} = (-1)^{k(n-k)} \beta^{(k)}$ for all k -forms $\beta^{(k)} \in \Lambda^{(n)}(\Omega)$. In this case $k = n = 1$ since $\mathcal{L}_{\mathbf{u}} \alpha^{(1)} \in \Lambda^{(1)}(\Omega)$, and hence $k(n-k) = 0$. Therefore (5.6) becomes:

$$(v^{(1)}, \star \mathcal{L}_{\mathbf{u}}(\star \alpha^{(1)}))_{\Omega} = \int_{\Omega} v^{(1)} \wedge \star(\star \mathcal{L}_{\mathbf{u}}(\star \alpha^{(1)})) = \int_{\Omega} v^{(1)} \wedge \mathcal{L}_{\mathbf{u}}(\star \alpha^{(1)}) \quad (5.7)$$

The resulting term in (5.7) is not an L^2 -inner product; this is resolved by using the Lie derivative's Leibniz rule, with the intent of moving the Lie derivative from $\star \alpha^{(1)}$ to $v^{(1)}$. Doing this in (5.7) gives:

$$\int_{\Omega} v^{(1)} \wedge \mathcal{L}_{\mathbf{u}}(\star \alpha^{(1)}) = \int_{\Omega} \mathcal{L}_{\mathbf{u}}(v^{(1)} \wedge \star \alpha^{(1)}) - \int_{\Omega} \mathcal{L}_{\mathbf{u}}(v^{(1)}) \wedge \star \alpha^{(1)} \quad (5.8)$$

Recalling that the Lie derivative of any volume form is $\mathcal{L}_{\mathbf{u}} = di_{\mathbf{u}}$, Stokes' theorem can be used to turn the first term on the right side of (5.8) into a boundary integral. The result of the manipulations covered here is:

$$(v^{(1)}, \star \mathcal{L}_{\mathbf{u}}(\star \alpha^{(1)}))_{\Omega} = -(\mathcal{L}_{\mathbf{u}} v^{(1)}, \alpha^{(1)})_{\Omega} + \int_{\partial \Omega} i_{\mathbf{u}}(v^{(1)} \wedge \star \alpha^{(1)})$$

From which it follows that (5.3) can be rewritten as:

$$\left(v^{(1)}, \frac{\partial \alpha^{(1)}}{\partial t}\right)_{\Omega} + \frac{1}{2}(v^{(1)}, \mathcal{L}_{\mathbf{u}} \alpha^{(1)})_{\Omega} - \frac{1}{2}(\mathcal{L}_{\mathbf{u}} v^{(1)}, \alpha^{(1)})_{\Omega} + \frac{1}{2} \int_{\partial \Omega} i_{\mathbf{u}}(v^{(1)} \wedge \star \alpha^{(1)}) = 0 \quad \forall v^{(1)} \in \Lambda^{(1)}(\Omega)$$

Since $\Omega =]0, 1[$, $\partial \Omega = \emptyset$; as a result the boundary integral is exactly zero. The two models that are considered here are thus:

$$\left(v^{(1)}, \frac{\partial \alpha^{(1)}}{\partial t}\right)_{\Omega} + (v^{(1)}, \mathcal{L}_{\mathbf{u}} \alpha^{(1)})_{\Omega} = 0 \quad \forall v^{(1)} \in \Lambda^{(1)}(\Omega) \quad (5.9)$$

$$\left(v^{(1)}, \frac{\partial \alpha^{(1)}}{\partial t}\right)_{\Omega} + \frac{1}{2}(v^{(1)}, \mathcal{L}_{\mathbf{u}} \alpha^{(1)})_{\Omega} - \frac{1}{2}(\mathcal{L}_{\mathbf{u}} v^{(1)}, \alpha^{(1)})_{\Omega} = 0 \quad \forall v^{(1)} \in \Lambda^{(1)}(\Omega) \quad (5.10)$$

As will be shown in section 5.2 the discretization process for both models is identical. Nonetheless the different formulations used for the (continuous) advection operators lead to different discrete advection operators with different properties; a fact that will be exploited throughout this research work.

5.2. Discrete models

Knot vector Ξ is used to generate uniform periodic 1-form basis $\{\psi_i^{(1)}\}_{i=1}^{i=n}$ as given in section 2.2.6. This basis defines the space of discrete 1-forms $\Lambda_{h,p}^{(1)}(\Omega_h) \subset \Lambda^{(1)}(\Omega)$ of p^{th} degree. As a result the discrete 1-forms $\alpha_h^{(1)}$, $v_h^{(1)} \in \Lambda_{h,p}^{(1)}(\Omega_h)$ become:

$$\alpha_h^{(1)}(x, t) = \sum_{i=1}^n \alpha_i(t) \psi_i^{(1)}(x) = (\boldsymbol{\psi}^{(1)})^T \boldsymbol{\alpha}$$

$$v_h^{(1)}(x, t) = \sum_{i=1}^n v_i(t) \psi_i^{(1)}(x) = (\boldsymbol{\psi}^{(1)})^T \mathbf{v}$$

Picking $\Lambda_{h,p}^{(1)}(\Omega_h)$ also defines the space of discrete periodic 0-forms that conforms to the discrete De Rham complex. As is covered in chapter 2, this space is generated by the uniform periodic 0-form basis $\{\psi_i^{(0)}\}_{i=1}^{i=n}$. This basis spans $\Lambda_{h,p+1}^{(0)}(\Omega_h) \subset \Lambda^{(0)}(\Omega)$. With the discrete operators defined in section 2.3 (5.9) and (5.10) are reformulated as semi-discrete equations:

$$\mathbf{v}^T \mathbb{M}^{(1)} \frac{\partial \boldsymbol{\alpha}}{\partial t} + \mathbf{v}^T \mathbb{M}^{(1)} \mathbb{E}^{(1,0)} (\mathbb{M}^{(0)})^{-1} \mathbb{C}_{\mathbf{u}_h}^{(0,1)} \boldsymbol{\alpha} = 0 \quad \forall \mathbf{v} \in \mathbb{R}^n$$

$$\mathbf{v}^T \mathbb{M}^{(1)} \frac{\partial \boldsymbol{\alpha}}{\partial t} + \frac{1}{2} \mathbf{v}^T \mathbb{M}^{(1)} \mathbb{E}^{(1,0)} (\mathbb{M}^{(0)})^{-1} \mathbb{C}_{\mathbf{u}_h}^{(0,1)} \boldsymbol{\alpha} - \frac{1}{2} \mathbf{v}^T \left[\mathbb{M}^{(1)} \mathbb{E}^{(1,0)} (\mathbb{M}^{(0)})^{-1} \mathbb{C}_{\mathbf{u}_h}^{(0,1)} \right]^T \boldsymbol{\alpha} = 0 \quad \forall \mathbf{v} \in \mathbb{R}^n$$

Since these equations must hold for all $\mathbf{v} \in \mathbb{R}^n$, we set \mathbf{v} equal to each unit vector in \mathbb{R}^n ; doing this gives a set of n test functions that span $\Lambda_{h,p}^{(1)}(\Omega_h)$ and thereby results in an $n \times n$ linear system. Subsequently applying the implicit midpoint rule results in two fully-discrete numerical methods. The first of these corresponds to the conservative formulation of the linear advection, with the weak form given in (5.9):

$$\left[\mathbb{M}^{(1)} + \frac{\Delta t}{2} \mathbb{M}^{(1)} \mathbb{E}^{(1,0)} (\mathbb{M}^{(0)})^{-1} \mathbb{C}_{\mathbf{u}_h}^{(0,1)} \right] \boldsymbol{\alpha}^{k+1} = \left[\mathbb{M}^{(1)} - \frac{\Delta t}{2} \mathbb{M}^{(1)} \mathbb{E}^{(1,0)} (\mathbb{M}^{(0)})^{-1} \mathbb{C}_{\mathbf{u}_h}^{(0,1)} \right] \boldsymbol{\alpha}^k \quad (5.11)$$

This discrete model is thus referred to as the conservative method. The second numerical method corresponds to the linear advection formulation given in (5.3), with the weak form given in (5.10):

$$\left[\mathbb{M}^{(1)} + \frac{\Delta t}{4} \left(\left(\mathbb{M}^{(1)} \mathbb{E}^{(1,0)} (\mathbb{M}^{(0)})^{-1} \mathbb{C}_{\mathbf{u}_h}^{(0,1)} \right) - \left(\mathbb{M}^{(1)} \mathbb{E}^{(1,0)} (\mathbb{M}^{(0)})^{-1} \mathbb{C}_{\mathbf{u}_h}^{(0,1)} \right)^T \right) \right] \boldsymbol{\alpha}^{k+1} = \left[\mathbb{M}^{(1)} - \frac{\Delta t}{4} \left(\left(\mathbb{M}^{(1)} \mathbb{E}^{(1,0)} (\mathbb{M}^{(0)})^{-1} \mathbb{C}_{\mathbf{u}_h}^{(0,1)} \right) - \left(\mathbb{M}^{(1)} \mathbb{E}^{(1,0)} (\mathbb{M}^{(0)})^{-1} \mathbb{C}_{\mathbf{u}_h}^{(0,1)} \right)^T \right) \right] \boldsymbol{\alpha}^k \quad (5.12)$$

Here the superscripts k , $k+1$ indicate the time levels t^k , t^{k+1} and $\Delta t = t^{k+1} - t^k$. Recall that \mathbf{u}_h is given and independent of $\boldsymbol{\alpha}$; hence these numerical methods are linear and $\boldsymbol{\alpha}^{k+1}$ can be found by solving a linear system, provided that $\boldsymbol{\alpha}^k$ is known. Also note that the advection operators in (5.12) are skew-symmetric; hence this model is referred to as the skew-symmetric method.

5.3. Sine wave test case

As first test case the initial condition $\alpha^{(1)}(x, 0) = (1 + 0.25 \sin(2\pi x)) dx$ is imposed on $\Omega =]0, 1[$; the advection velocity field used is $\mathbf{u}_h = 1 \frac{\partial}{\partial x}$. The exact solution of this case is thus $\alpha(x, t) = 1 + 0.25 \sin(2\pi(x - t))$. The performance of the considered models is evaluated based on their relative L^2 errors and the accuracy of their respective conservation behavior. The L^2 errors are covered in section 5.3.1, after which section 5.3.2 briefly covers the conservation behavior and introduces the relevant conservation metrics.

As a typical example, running the numerical methods from $t = 0$ to $t = 10$ with $\Delta t = 0.01$ and $n = 25$ basis functions of degree $p = 2$ results in the solutions at $t = 10$ shown below in figure 5.1.

No visible differences exist between the solutions shown in figure 5.1. After ten advection periods both numerical solutions have a minor phase difference with respect to the exact solution.

5.3.1. Discretization error

Two sources of discretization error are present in these simulations, due to the decoupled space and time discretizations. The error induced by the implicit midpoint method scales with $\mathcal{O}(\Delta t^2)$, whereas the spatial discretization approach's error scales with $\mathcal{O}(\Delta x^{p+1})$ and thus depends on the degree of the basis functions.

Figure 5.2 shows the spatial L^2 error convergence of both numerical methods for various basis function degrees p . As can be seen both methods display optimal L^2 convergence rates up to $\mathcal{O}(10^{-8})$; at this point the error induced by the time step size $\Delta t = 10^{-4}$ starts to dominate the total L^2 error. The differences in L^2 error between both methods are only $\mathcal{O}(10^{-14})$; hence they are indistinguishable in figure 5.2.

In the same way the temporal L^2 error convergence obtained with different p is shown in figure 5.3 for $n = 50$. The relative L^2 error decreases with $\mathcal{O}(\Delta t^2)$ for all p , up to the point where the spatial discretization dominates the L^2 error.

5.3.2. Numerical conservation

As mentioned throughout this thesis, the numerical conservation behavior of various quantities is a major focus of the current research. More specifically, it is desirable that the discrete models mimic the conservation behavior of the continuous models. This is one of the cornerstones and aims of the modern development of

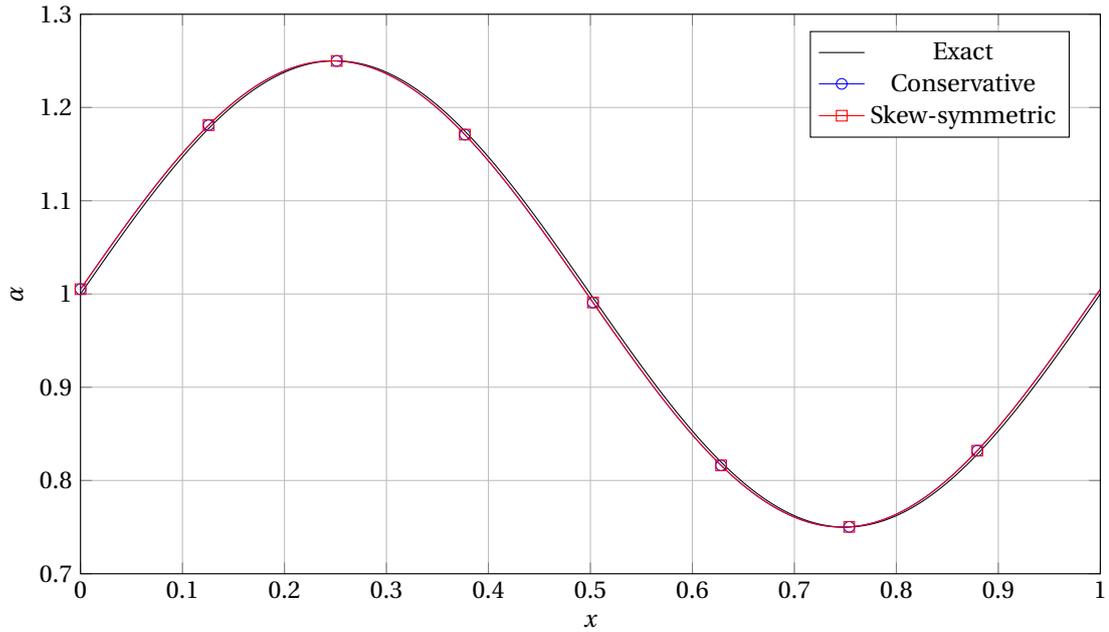


Figure 5.1: Numerical solutions at $t = 10$ (obtained with $\Delta t = 10^{-2}$, $n = 25$ basis functions of degree $p = 2$) for both methods discussed in this chapter

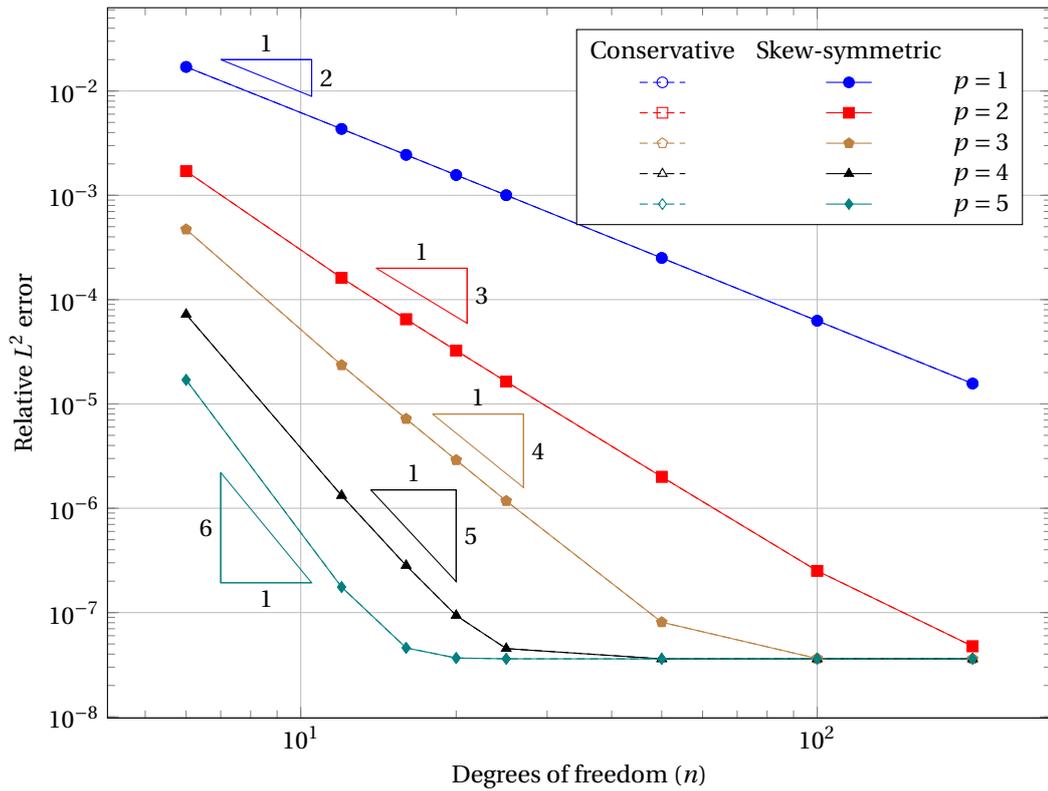


Figure 5.2: Convergence of the relative L^2 error of the conservative and skew-symmetric models at $t = 1$ for $\Delta t = 10^{-4}$, varying amounts of degrees of freedom n and basis function degrees p

numerical models. Various quantities will be considered, depending on structure of each continuous model. For the linear advection equation the conservation of the integrals of α and α^2 will be considered. In order to give a sense of scale and importance to the variations of these quantities, the relative changes of their integrals over time will be looked at. As such 'momentum' and 'kinetic energy' analogues are defined; while

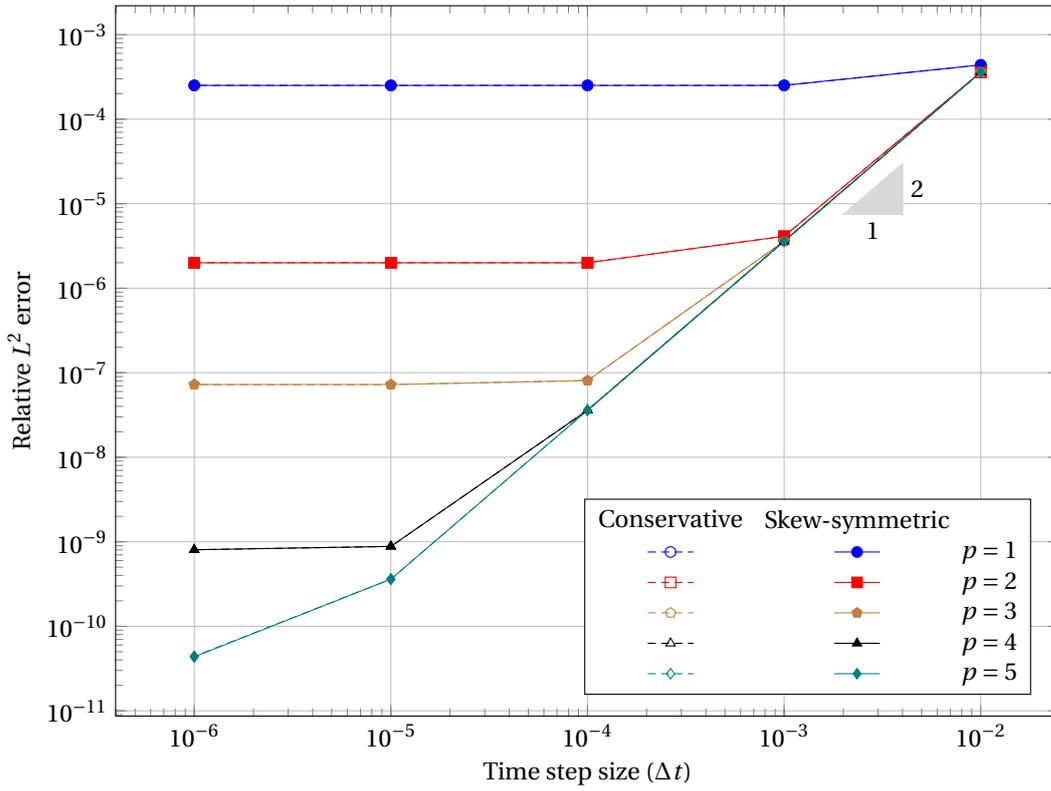


Figure 5.3: Convergence of the relative L^2 error of both considered numerical models at $t = 1$ for $n = 50$ and varying time step sizes Δt and basis function degrees p

these quantities don't have a physical interpretation for the linear advection equation within this research, they do for subsequent test cases. The following definitions are used in this chapter:

$$m(t) = \int_{\Omega} \alpha_h^{(1)}(x, t) \quad E(t) = \frac{1}{2} \int_{\Omega} \alpha_h^{(1)}(x, t) \wedge \star \alpha_h^{(1)}(x, t)$$

Where $m(t)$ is the momentum analogue and $E(t)$ is the kinetic energy analogue. Their changes over time are normalized with the momentum and kinetic energy analogues of the exact solution. The relative changes over time of $m(t)$ and $E(t)$ can thus be defined as:

$$\Delta m(t) = \frac{m(t) - m(0)}{m_{\text{exact}}(t)} \quad \Delta E(t) = \frac{E(t) - E(0)}{E_{\text{exact}}(t)}$$

Figure 5.4 shows the maximum absolute values of $\Delta m(t)$ and $\Delta E(t)$ for both models considered here. As can be seen both models conserve m and E up to $\mathcal{O}(10^{-13})$ with no visible trends as p or n are increased.

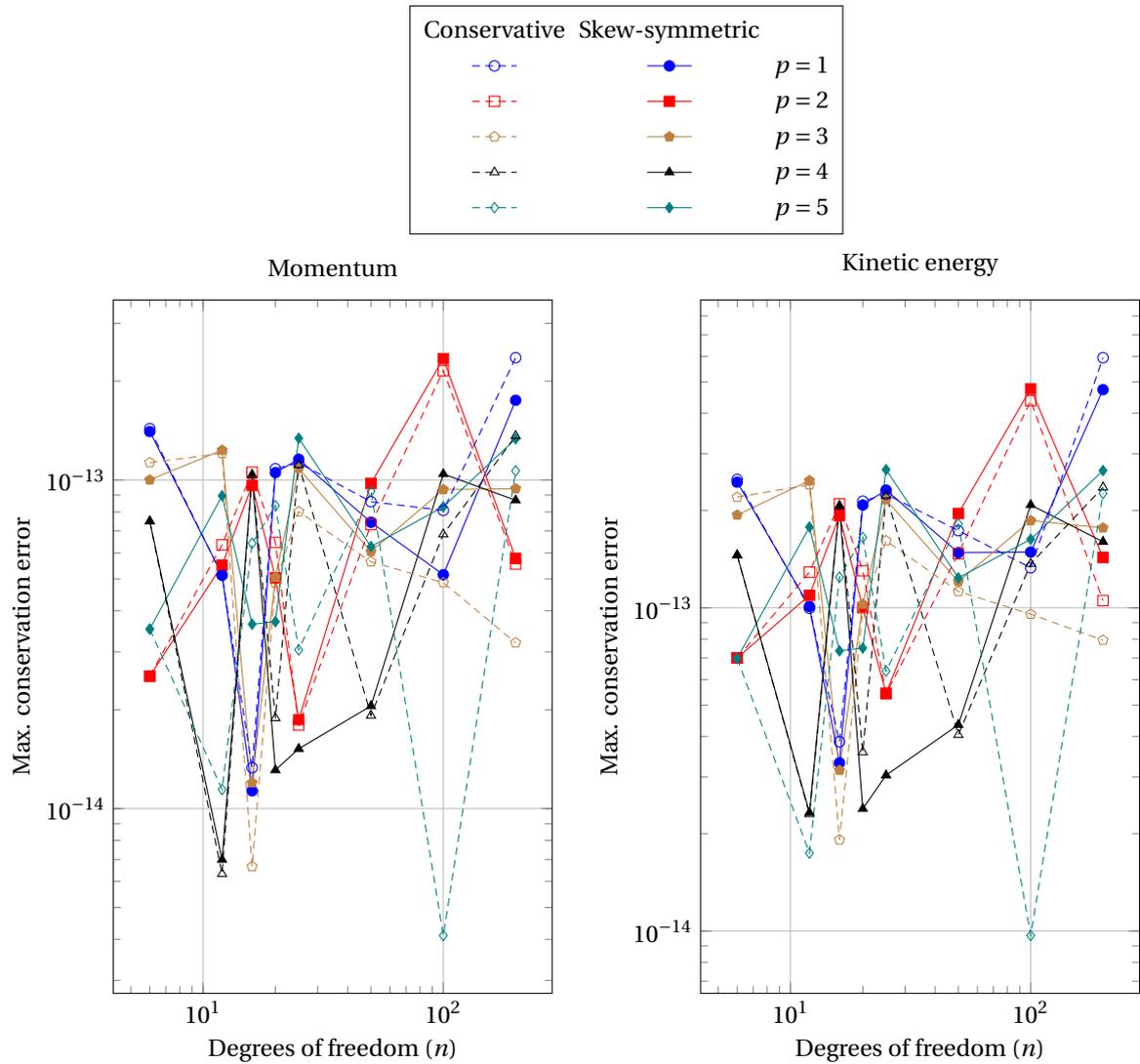


Figure 5.4: Maximum momentum and kinetic energy conservation errors of the conservative and skew-symmetric models for various basis function degrees p , $\Delta t = 10^{-4}$ was used

6

Burgers' equation

Burgers' equation presents a more complex test case. While the advective velocity field in the linear advection model is known and imposed upon the distribution of a variable, Burgers' equation is the simplest model where "self-advection" takes place. In other words, Burgers' equation describes the physical transport of a quantity that advects under its own influence. An example of this phenomenon in fluid mechanics is the advection term in the momentum equation of the incompressible Euler and Navier-Stokes models. Two formulations of (the inviscid form of) Burgers' equation are commonly given:

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} = 0 \quad (6.1)$$

$$\frac{\partial u}{\partial t} + \frac{1}{2} \frac{\partial u^2}{\partial x} = 0 \quad (6.2)$$

As was covered in section 3.3, (6.1) is commonly referred to as the advective or convective formulation, whereas (6.2) is the conservative formulation. One can easily check that both formulations are equivalent to one another by applying the product rule to the derivative $\frac{\partial u^2}{\partial x}$. One big difference between Burgers' equation and Euler & Navier-Stokes models is the relation between the various forms of the advection operator. In the momentum equations of the incompressible Euler and Navier-Stokes models the advective formulation $u \frac{\partial(\rho u)}{\partial x}$ is algebraically similar to the conservative form $\frac{\partial(\rho u u)}{\partial x}$: Switching between them can be done through substitution of the continuity equation into the momentum equations and appropriate algebraic manipulations. Since Burgers' equation consists of only a single equation, this algebraic similarity is absent.

Exact solutions to (the viscous and inviscid forms of) Burgers' equation exist and can be computed with the method of characteristics. A characteristic line emanating from initial condition $u_0(x)$ at point x_c is straight and has a slope of $u_0(x_c)$. Continuous initial conditions become discontinuous after some time if $\frac{\partial u_0}{\partial x} < 0$ at any point in the domain, due to the characteristic lines intersecting. Leveque [32, Chap. 3] discusses ways of obtaining exact solutions after such a discontinuity has formed.

6.1. Continuous models

Again $\Omega =]0, 1[$ is used as (periodic) domain. Let $\alpha^{(1)} \in \Lambda^{(1)}(\Omega)$ such that $\alpha^{(1)} = \alpha dx$ and recall that $\sharp : \Lambda^{(1)}(\Omega) \rightarrow T(\Omega)$ is the operator that transforms 1-forms into their dual vector fields, such that $\sharp \alpha^{(1)} = \mathbf{u} = \alpha \frac{\partial}{\partial x}$. Then:

$$\begin{aligned} \mathcal{L}_{\mathbf{u}} \alpha^{(1)} &= d i_{\mathbf{u}} (\alpha dx) = d(\alpha^2) = \frac{\partial \alpha^2}{\partial x} dx \\ \star \mathcal{L}_{\mathbf{u}} (\star \alpha^{(1)}) &= \star i_{\mathbf{u}} d(\star \alpha dx) = \star i_{\mathbf{u}} \left(\frac{\partial \alpha}{\partial x} dx \right) = \star \left(\alpha \frac{\partial \alpha}{\partial x} \right) = \alpha \frac{\partial \alpha}{\partial x} dx \end{aligned}$$

Hence the convective form corresponds to $\star \mathcal{L}_{\mathbf{u}} (\star \alpha^{(1)})$ and the conservative form corresponds to $\frac{1}{2} \mathcal{L}_{\mathbf{u}} \alpha^{(1)}$. This factor of $\frac{1}{2}$ affects the balance between the convective and conservative forms that is necessary for obtaining a skew-symmetric advection operator for Burgers' equation.

The two formulations of the Burgers' equation that are considered in this research are:

$$\frac{\partial \alpha^{(1)}}{\partial t} + \frac{1}{2} \mathcal{L}_u(\alpha^{(1)}) = 0 \quad (6.3)$$

$$\frac{\partial \alpha^{(1)}}{\partial t} + \frac{1}{2} \left((1-a) \mathcal{L}_u(\alpha^{(1)}) + 2a \star \mathcal{L}_u(\star \alpha^{(1)}) \right) = 0 \quad (6.4)$$

With $a = \frac{1}{3}$ (6.4) results in a skew-symmetric discretization, since in that case $1 - a = 2a$. To arrive at weak forms the same approach is taken as in section 5.1: The L^2 inner product is taken with 1-form $v^{(1)} \in \Lambda^{(1)}(\Omega)$ and the last term on the left side of (6.4) is rewritten with the help of (5.1). It follows that the weak forms of (6.3) and (6.4) are:

$$\left(v^{(1)}, \frac{\partial \alpha^{(1)}}{\partial t} \right)_{\Omega} + \frac{1}{2} \left(v^{(1)}, \mathcal{L}_u \alpha^{(1)} \right)_{\Omega} = 0 \quad \forall v^{(1)} \in \Lambda^{(1)}(\Omega) \quad (6.5)$$

$$\left(v^{(1)}, \frac{\partial \alpha^{(1)}}{\partial t} \right)_{\Omega} + \frac{1}{3} \left(v^{(1)}, \mathcal{L}_u \alpha^{(1)} \right)_{\Omega} + \frac{1}{3} \left(\mathcal{L}_u v^{(1)}, \alpha^{(1)} \right)_{\Omega} = 0 \quad \forall v^{(1)} \in \Lambda^{(1)}(\Omega) \quad (6.6)$$

6.2. Discrete models

Again knot vector Ξ is defined that generates a uniform periodic 1-form basis $\{\psi_i^{(1)}\}_{i=1}^{i=n}$ of degree p on Ω . $\alpha^{(1)}$ and $v^{(1)}$ are discretized with this basis:

$$\alpha_h^{(1)}(x, t) = \sum_{i=1}^n \alpha_i(t) \psi_i^{(1)}(x) = (\boldsymbol{\psi}^{(1)})^T \boldsymbol{\alpha}$$

$$v_h^{(1)}(x, t) = \sum_{i=1}^n v_i(t) \psi_i^{(1)}(x) = (\boldsymbol{\psi}^{(1)})^T \boldsymbol{v}$$

With $\alpha_h^{(1)}, v_h^{(1)} \in \Lambda_{h,p}^{(1)}(\Omega_h) \subset \Lambda^{(1)}(\Omega)$. The resulting semi-discrete models resemble those of the linear advection equation:

$$\mathbf{v}^T \mathbb{M}^{(1)} \frac{\partial \boldsymbol{\alpha}}{\partial t} + \frac{1}{2} \mathbf{v}^T \mathbb{M}^{(1)} \mathbb{E}^{(1,0)} (\mathbb{M}^{(0)})^{-1} \mathbb{C}_{\mathbf{u}_h}^{(0,1)} \boldsymbol{\alpha} = 0 \quad \forall \boldsymbol{v} \in \mathbb{R}^n \quad (6.7)$$

$$\mathbf{v}^T \mathbb{M}^{(1)} \frac{\partial \boldsymbol{\alpha}}{\partial t} + \frac{1}{3} \mathbf{v}^T \mathbb{M}^{(1)} \mathbb{E}^{(1,0)} (\mathbb{M}^{(0)})^{-1} \mathbb{C}_{\mathbf{u}_h}^{(0,1)} \boldsymbol{\alpha} - \frac{1}{3} \mathbf{v}^T \left[\mathbb{M}^{(1)} \mathbb{E}^{(1,0)} (\mathbb{M}^{(0)})^{-1} \mathbb{C}_{\mathbf{u}_h}^{(0,1)} \right]^T \boldsymbol{\alpha} = 0 \quad \forall \boldsymbol{v} \in \mathbb{R}^n \quad (6.8)$$

A big difference between these (semi)-discrete models and those of the advection equation covered in chapter 5 is that these models are no longer linear. Since $\mathbf{u}_h = (\alpha_h^{(1)})^{\sharp}$ is used as advecting velocity field the solution coefficient vector $\boldsymbol{\alpha}$ is both an input and an output of the discretized Burgers' equation. Before determining how to proceed with linearization, equations (6.7) and (6.8) are discretized in time with the implicit midpoint method. Factoring out the vector \boldsymbol{v}^T and applying this time stepping method results in the following discrete nonlinear equations:

$$\left[\mathbb{M}^{(1)} + \frac{\Delta t}{4} \mathbb{M}^{(1)} \mathbb{E}^{(1,0)} (\mathbb{M}^{(0)})^{-1} \mathbb{C}_{\mathbf{u}_h^{k+1/2}}^{(0,1)} \right] \boldsymbol{\alpha}^{k+1} = \left[\mathbb{M}^{(1)} - \frac{\Delta t}{4} \mathbb{M}^{(1)} \mathbb{E}^{(1,0)} (\mathbb{M}^{(0)})^{-1} \mathbb{C}_{\mathbf{u}_h^{k+1/2}}^{(0,1)} \right] \boldsymbol{\alpha}^k \quad (6.9)$$

$$\left[\mathbb{M}^{(1)} + \frac{\Delta t}{6} \left(\mathbb{M}^{(1)} \mathbb{E}^{(1,0)} (\mathbb{M}^{(0)})^{-1} \mathbb{C}_{\mathbf{u}_h^{k+1/2}}^{(0,1)} - \left[\mathbb{M}^{(1)} \mathbb{E}^{(1,0)} (\mathbb{M}^{(0)})^{-1} \mathbb{C}_{\mathbf{u}_h^{k+1/2}}^{(0,1)} \right]^T \right) \right] \boldsymbol{\alpha}^{k+1}$$

$$= \left[\mathbb{M}^{(1)} - \frac{\Delta t}{6} \left(\mathbb{M}^{(1)} \mathbb{E}^{(1,0)} (\mathbb{M}^{(0)})^{-1} \mathbb{C}_{\mathbf{u}_h^{k+1/2}}^{(0,1)} - \left[\mathbb{M}^{(1)} \mathbb{E}^{(1,0)} (\mathbb{M}^{(0)})^{-1} \mathbb{C}_{\mathbf{u}_h^{k+1/2}}^{(0,1)} \right]^T \right) \right] \boldsymbol{\alpha}^k \quad (6.10)$$

Where $\mathbf{u}_h^{k+1/2} = \frac{1}{2} (\mathbf{u}_h^k + \mathbf{u}_h^{k+1})$ is the source of nonlinearity, since $\mathbf{u}_h^{k+1} = (\alpha_h^{(1)})^{\sharp} = [(\boldsymbol{\psi}^{(1)})^T \boldsymbol{\alpha}^{k+1}]^{\sharp}$ contains the unknown left-hand side vector in systems (6.9) and (6.10). To be able to solve these systems a linearization approach is required. As mentioned in section 3.6.3 Picard linearization is used for this. The following estimate is used as initial guess $\boldsymbol{\alpha}_0^{k+1}$ for the first iteration at time step $k+1$:

$$\boldsymbol{\alpha}_0^{k+1} = \boldsymbol{\alpha}^k + (\boldsymbol{\alpha}^k - \boldsymbol{\alpha}^{k-1})$$

6.3. Sine wave test case

The same initial condition as was used for the linear advection equation in section 5.3 is imposed on $\Omega =]0, 1[$: $\alpha^{(1)}(x, 0) = (1 + \frac{1}{4} \sin(2\pi x)) dx$. This initial condition is advanced to $t = 1$ with the conservative and skew-symmetric discrete models given in (6.9) and (6.10) and compared to the exact solution. In this exact solution the sine wave initial condition develops a shock discontinuity starting at $t = \frac{2}{\pi} \approx 0.637$, owing to the gradient-steepening behavior of Burgers' equation; this is the initial time at which a pair of characteristics emanating from the initial condition intersect.

Figure 6.1 shows the exact solution and the numerical solutions obtained with both models presented in the previous section, at four different times. 25 basis functions of degree $p = 2$ were used, together with a time step size $\Delta t = 10^{-4}$ and nonlinear convergence tolerances $\epsilon = 10^{-6}$ and $\epsilon = 10^{-14}$. As can be seen the numerical solutions obtained with these inputs are indistinguishable from one another. The largest differences between the results of both models are $\mathcal{O}(10^{-3})$. This similarity is at least in part due to the values of ϵ used: Since the differences between successive nonlinear iterations decrease monotonically, ϵ acts as an estimate of an upper bound for the local error produced by using linearized solutions. As a result the pointwise difference between the results obtained with $\epsilon = 10^{-6}$ and $\epsilon = 10^{-14}$ is $\mathcal{O}(10^{-6})$.

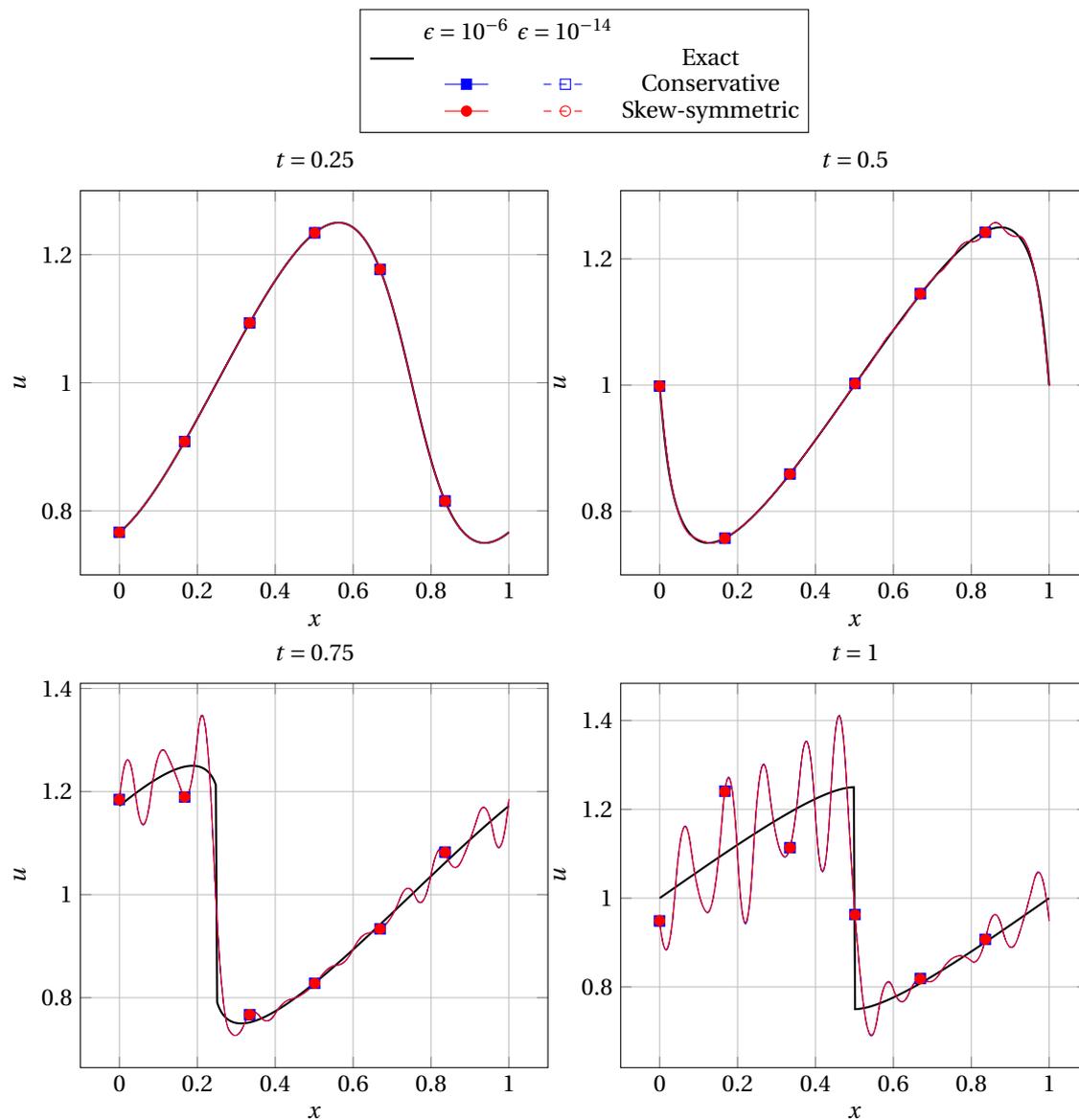


Figure 6.1: Solutions obtained with conservative and skew-symmetric models given in (6.9) and (6.10) respectively compared to the exact solution, at various times for $\Delta t = 10^{-4}$, $n = 25$ basis functions of degree $p = 2$ with nonlinear convergence tolerance $\epsilon = 10^{-14}$

6.3.1. Discretization error

Figure 6.2 shows the behavior of the L^2 errors for both models considered here at different time steps; $\epsilon = 10^{-6}$, $\epsilon = 10^{-14}$ and various basis function degrees p were used as the mesh is refined. Drastically different behavior is found at the different time steps shown. The results are covered in chronological order, differences between the two values for ϵ and the conservative and skew-symmetric models are discussed.

At $t = 0.25$ both the conservative and skew-symmetric models display similar L^2 error convergence to what was found in chapter 5 for linear advection, up to the point where time discretization errors start to dominate; with $\Delta t = 10^{-4}$ the L^2 error induced by the implicit midpoint time stepping method is $\mathcal{O}(10^{-8})$, the same as for the linear advection equation. The results of both models are identical for all combinations of n and p when $\epsilon = 10^{-14}$; with $\epsilon = 10^{-6}$ the relative L^2 errors of both methods start to drift apart as the L^2 errors reach $\mathcal{O}(10^{-7})$. In all points in this region the L^2 error of the skew-symmetric model is smaller than that of the conservative model.

At $t = 0.5$, the next time step shown, the relative L^2 errors have grown; from a factor of roughly four for both models with $p = 1$ to an increase with $p = 5$ between ten and a thousand times what it was at $t = 0.25$, depending on the mesh size. The error scales with n^2 (and thus $(\Delta x)^2$) for the coarsest meshes, independent of the basis function degree p . As the mesh is refined the convergence rates for the various p drift apart; the higher p , the higher the observed convergence rates. The differences between the L^2 errors observed with $\epsilon = 10^{-6}$ and $\epsilon = 10^{-14}$ are negligible for both models everywhere except for the region $n > 10^2$: Here the conservative model has a higher L^2 error than the skew-symmetric model when $\epsilon = 10^{-6}$ is used.

Time steps $t = 0.75$, $t = 1$ are both preceded by the time of initial wave break, $t_b \approx 0.637$. The exact solution has become discontinuous, and thus L^2 convergence is no longer guaranteed. As is also reflected in the representative solutions that are shown in figure 6.1, oscillations are created close to the moving discontinuity that are advected across the domain; these dominate the L^2 errors. The magnitudes of the L^2 errors have increased by factors between 10 and 100,000 compared to $t = 0.5$, depending on n and p . Refining the mesh or increasing the degree of the basis functions also no longer leads to a significant decrease in L^2 errors: The maximal error convergence slope is encountered at $t = 0.75$ for the coarsest meshes. This slope is only $\frac{1}{2}$ and nearly identical for all p .

Keeping the mesh size n constant ($n = 50$), changing the time step size and looking at the L^2 errors results in figure 6.3. At $t = 0.25$ the convergence behavior is identical to what was observed for linear advection: Initially convergence with $(\Delta t)^2$ is achieved, after which the spatial discretization error starts dominating the L^2 error behavior. At $t = 0.5$ the differences between degrees p have become significantly smaller, indicating that the spatial discretization dominates the total L^2 error even when large time steps ($\Delta t = 10^{-2}$) are used. This trend continues at $t = 0.75$ and $t = 1$, where the differences in L^2 error between different Δt or p are negligible compared to the total error magnitudes.

6.3.2. Numerical conservation

Momentum and kinetic energy analogues $m(t)$ and $E(t)$ are defined for Burgers' equation in the same way as was done for linear advection in section 5.3.2:

$$m(t) = \int_{\Omega} \alpha_h^{(1)}(x, t) \quad E(t) = \frac{1}{2} \int_{\Omega} \alpha_h^{(1)}(x, t) \wedge \star \alpha_h^{(1)}(x, t)$$

With their relative conservation errors:

$$\Delta m(t) = \frac{m(t) - m(0)}{m_{\text{exact}}(t)} \quad \Delta E(t) = \frac{E(t) - E(0)}{E_{\text{exact}}(t)}$$

Figure 6.4 shows the maximum absolute conservation errors of the same simulations with varying n that were discussed in the previous section. Due to the nonlinear nature of Burgers' equation (and the subsequent linearization thus necessary for numerical simulations) simultaneous conservation of m and E can only be obtained when the linearized solution solves the full nonlinear discrete system, i.e. when a nonlinear

convergence tolerance ϵ is used that is sufficiently close to machine precision. This is the only way in which equivalence between the convective and conservative advection operators can be achieved for this linearized equation.

First we focus on the skew-symmetric model: The behavior mentioned above can be observed in figure 6.4 by comparing the conservation results of the skew-symmetric model with $\epsilon = 10^{-6}$ and $\epsilon = 10^{-14}$; with $\epsilon = 10^{-14}$ both momentum and kinetic energy are conserved up to $\mathcal{O}(10^{-13})$, whereas using $\epsilon = 10^{-6}$ leads to momentum conservation errors up to $\mathcal{O}(10^{-6})$ while kinetic energy conservation is unchanged. The local growth in momentum conservation error as the mesh is refined is caused by the slowdown in convergence speed (and thus the nonlinear convergence error undershoot) on finer meshes. Because of this the actual nonlinear convergence with $\epsilon = 10^{-6}$ on the coarsest meshes is $\mathcal{O}(10^{-9})$ (due to the rapid convergence on coarse meshes), whereas on the fine meshes this is $\mathcal{O}(10^{-7})$. Skew-symmetric solutions on the coarse meshes are thus closer to the nonlinear solution, leading to better momentum conservation behavior.

Next we turn to the conservative method, which shows different behavior: Whereas it conserves momentum up to $\mathcal{O}(10^{-13})$ for both ϵ 's considered here, taking $\epsilon = 10^{-14}$ does not lead to similar levels of kinetic energy conservation error. In fact the differences in kinetic energy conservation error between $\epsilon = 10^{-6}$ and $\epsilon = 10^{-14}$ are only $\mathcal{O}(10^{-8})$, not even visible in figure 6.4. This is negligible compared to the actual magnitude of the kinetic energy conservation errors, which are $\mathcal{O}(10^{-4}) - \mathcal{O}(10^{-5})$. This is not only several orders of magnitude larger than the momentum conservation errors obtained with the skew-symmetric method for $\epsilon = 10^{-6}$, the conservation errors also do not decrease significantly as the nonlinear convergence tolerance is tightened.

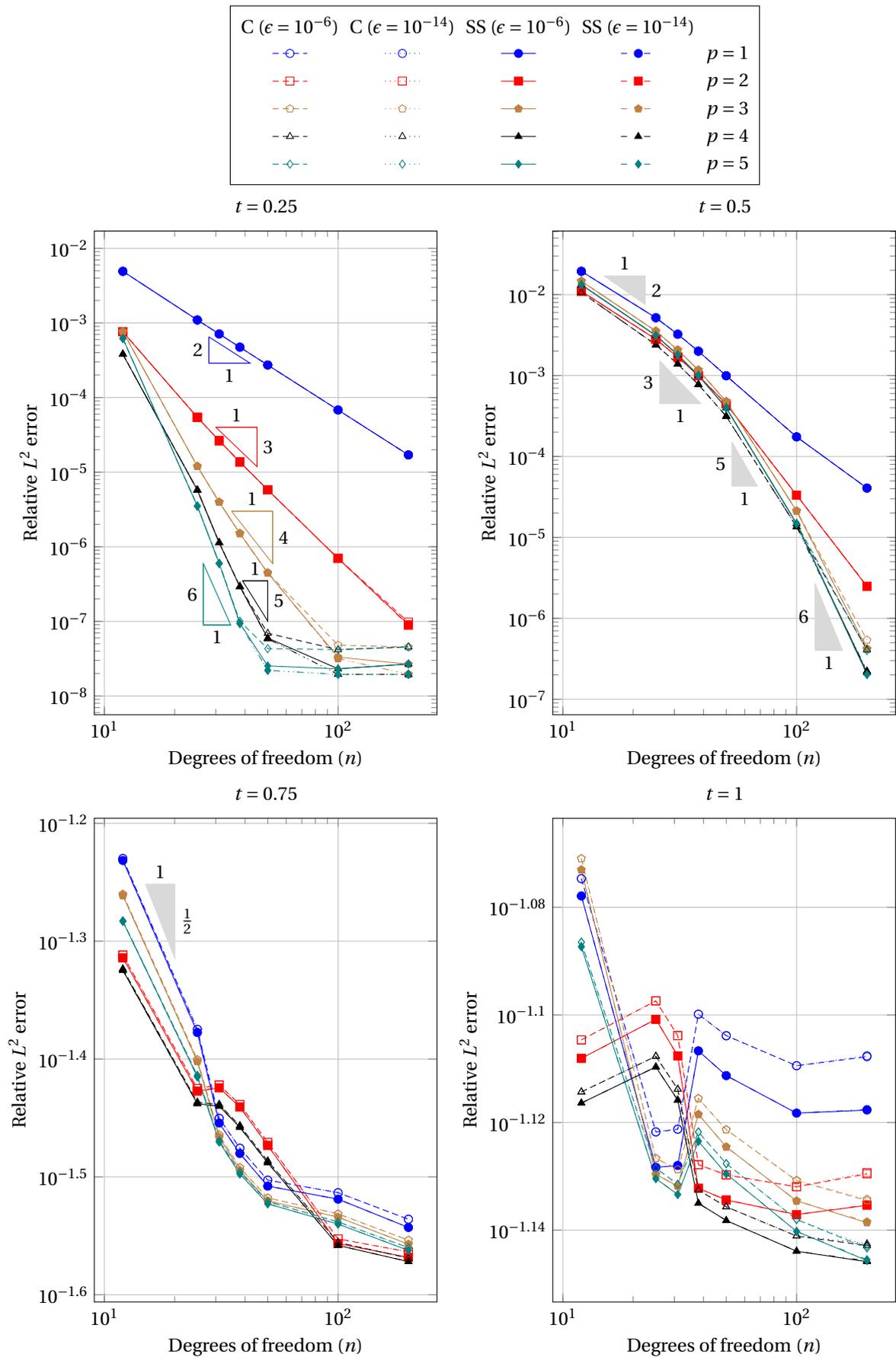


Figure 6.2: Convergence of the relative L^2 errors of the conservative (C) and skew-symmetric (SS) models at various times; $\Delta t = 10^{-4}$ and different basis function degrees p were used, together with $\epsilon = 10^{-6}$ and $\epsilon = 10^{-14}$

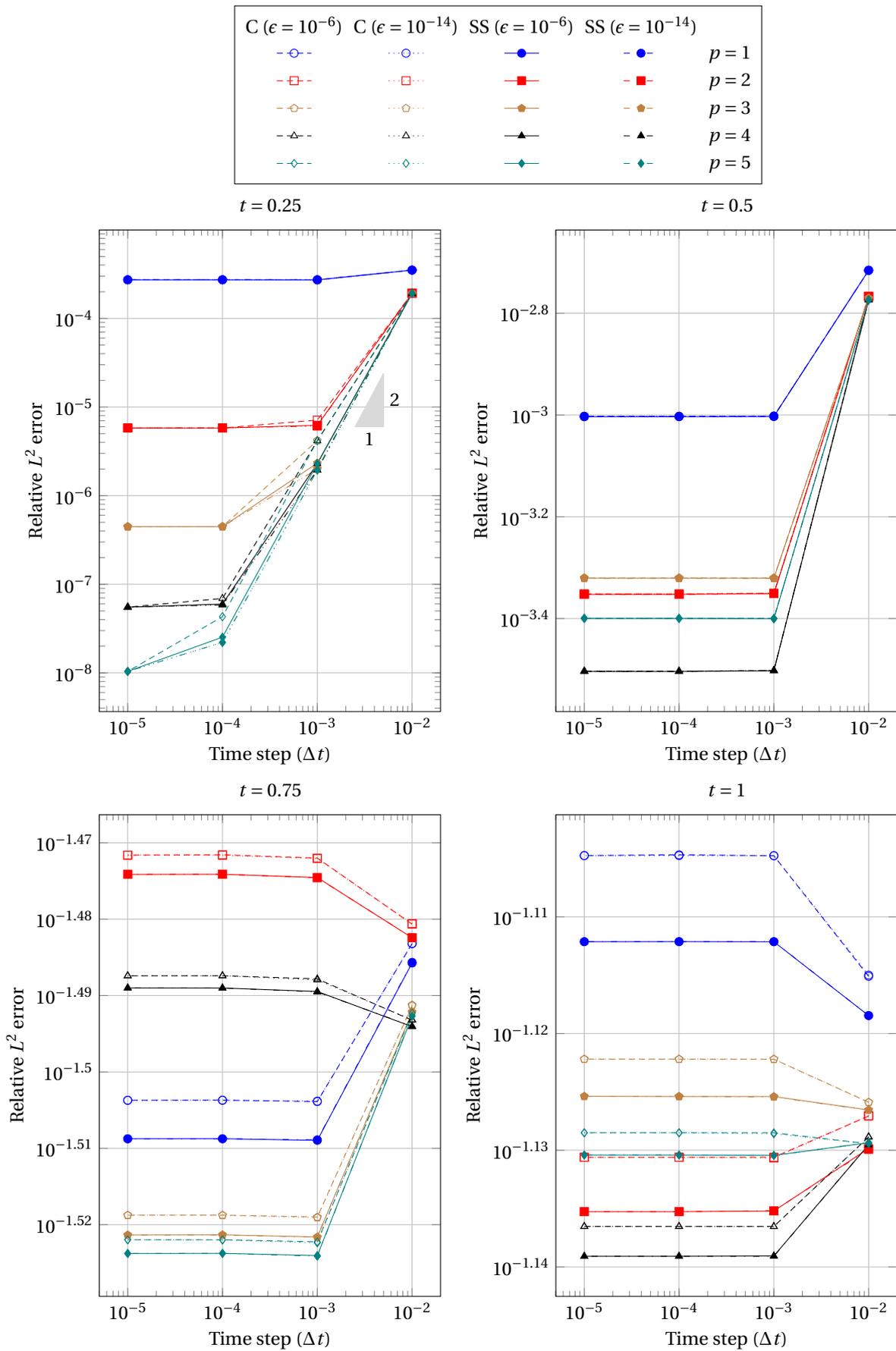


Figure 6.3: Convergence of the relative L^2 error of the conservative and skew-symmetric models at various times; $n = 50$ and different basis function degrees p were used, together with $\epsilon = 10^{-6}$ and $\epsilon = 10^{-14}$

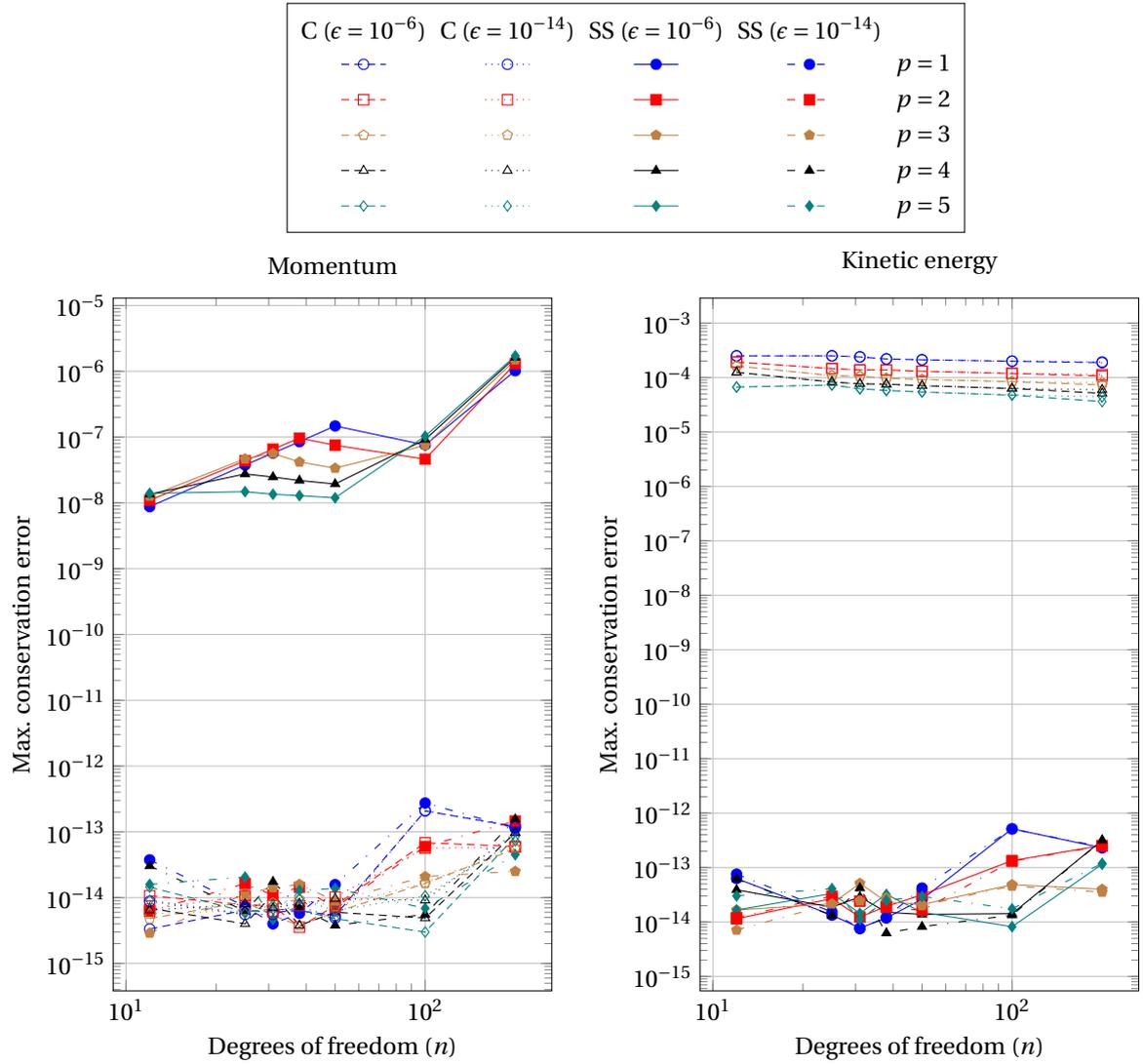


Figure 6.4: Maximum momentum and kinetic energy conservation errors of the conservative (C) and skew-symmetric (SS) models for various basis function degrees p ; $\Delta t = 10^{-4}$ was used, together with $\epsilon = 10^{-6}$ and $\epsilon = 10^{-14}$

7

One-dimensional Euler equations

Following the initial tests with linear advection and the inviscid (nonlinear) Burgers' equation, the Euler model derived in chapter 3 is put through its first test: Sod's shock tube. This one-dimensional test is a particular case of a wider family of Riemann problems and was first proposed by Sod in 1978 [50]. Physically speaking this case corresponds to a diaphragm inside of a tube, separating regions 1 and 5 of a gas at rest with different pressures and densities. At time $t = 0$ the diaphragm is removed, leading to a rush of gas from the high-pressure side (region 1) of the diaphragm to the low-pressure side (region 5). As a result a pattern forms consisting of a shock wave, contact discontinuity and expansion fan, all due to the discontinuous pressure and density profiles at the point where the diaphragm was originally located. This is a typical Riemann problem, for which exact solutions exist. Leveque covers the process of how to obtain its exact solutions in chapter 9 of [32].

Sod's shock tube is a popular (initial) test case for numerical methods for several reasons. First off, around the time it was initially proposed it was a challenging test case for (at the time) state-of-the-art numerical schemes for discontinuous solutions; the original paper in which Sod proposed it compared how accurately various well-known schemes solved it. Its emergence as a standard test case resulted in new numerical schemes and corrections for dealing with problems in gas dynamics. Depending on the initial conditions Sod's shock tube can simultaneously result in an expansion fan, contact discontinuity and shock wave next to one another. Secondly, exact solutions exist for Sod's shock tube and can be computed based on the initial pressure and density ratios at $t = 0$ at the diaphragm. This allows for a direct comparison of numerical methods with one another based on how well they resemble the exact solution and how well they satisfy its conservation behavior. Thirdly, it is only a one-dimensional case and therefore it is much quicker and simpler to implement numerical methods for solving it than for any two-dimensional case. It is often used as an initial test for new numerical methods precisely because of this.

The structure of this chapter is similar to that of the previous chapters: First the continuous mimetic isogeometric models and their weak forms will be given in section 7.1. As was done previously two forms of the Euler equations that were derived in chapter 3 are covered. These weak forms are then discretized in section 7.2. Following the construction of said discrete models tests with Sod's shock tube were carried out and are covered in section 7.3. Their results are compared to the exact solution of this shock tube problem and the numerical solutions obtained with Clawpack [34] and the nodal Discontinuous Galerkin method described by Hesthaven & Warburton in [24]. These methods were both covered in chapter 4.

7.1. Continuous models

The models derived in chapter 3 are considered here. First of these is the 'regular' formulation given in (3.8), repeated here for convenience:

$$\begin{aligned} \frac{\partial \rho^{(n)}}{\partial t} + \mathcal{L}_{\mathbf{u}} \rho^{(n)} &= 0 \\ \frac{\partial \langle m^{(1,n)}, \partial_i \rangle}{\partial t} + \mathcal{L}_{\mathbf{u}} \langle m^{(1,n)}, \partial_i \rangle + di_{\partial_i} p^{(n)} &= 0 \\ \frac{\partial E^{(n)}}{\partial t} + \mathcal{L}_{\mathbf{u}} E^{(n)} + \mathcal{L}_{\mathbf{u}} p^{(n)} &= 0 \end{aligned} \quad (7.1)$$

Here mass density $\rho^{(n)}$, energy density $E^{(n)}$, pressure $p^{(n)} \in \Lambda^{(n)}(\Omega)$ and momentum $m^{(1,n)} \in T^*(\Omega) \otimes \Lambda^{(n)}(\Omega)$. With only one principal direction (7.1) contains one momentum equation, resulting from duality pairing $\langle m^{(1,n)}, \partial_i \rangle = \langle \partial_x, dx \rangle \otimes \rho u_x dx = \rho u_x dx = m_x^{(1)}$. System (7.1) for one dimension is thus:

$$\begin{aligned} \frac{\partial \rho^{(1)}}{\partial t} + \mathcal{L}_{\mathbf{u}} \rho^{(1)} &= 0 \\ \frac{\partial m_x^{(1)}}{\partial t} + \mathcal{L}_{\mathbf{u}} m_x^{(1)} + di_{\partial_x} p^{(1)} &= 0 \\ \frac{\partial E^{(1)}}{\partial t} + \mathcal{L}_{\mathbf{u}} E^{(1)} + \mathcal{L}_{\mathbf{u}} p^{(1)} &= 0 \end{aligned} \quad (7.2)$$

This model will be used as benchmark for the main focus of the current research, the Roe variable formulation given in (3.23):

$$\begin{aligned} \frac{\partial \sqrt{\rho}^{(n)}}{\partial t} + \frac{1}{2} \star \mathcal{L}_{\mathbf{u}} (\star \sqrt{\rho}^{(n)}) + \frac{1}{2} \mathcal{L}_{\mathbf{u}} (\sqrt{\rho}^{(n)}) &= 0 \\ \frac{\partial (\phi_i^{(n)})}{\partial t} + \frac{1}{2} \star \mathcal{L}_{\mathbf{u}} (\star \phi_i^{(n)}) + \frac{1}{2} \mathcal{L}_{\mathbf{u}} (\phi_i^{(n)}) + \frac{1}{\star \sqrt{\rho}^{(0)}} di_{\partial_i} p^{(n)} &= 0 \\ \frac{\partial E^{(n)}}{\partial t} + \mathcal{L}_{\mathbf{u}} E^{(n)} + \mathcal{L}_{\mathbf{u}} p^{(n)} &= 0 \end{aligned}$$

Which in one dimension corresponds to:

$$\begin{aligned} \frac{\partial \sqrt{\rho}^{(1)}}{\partial t} + \frac{1}{2} \star \mathcal{L}_{\mathbf{u}} (\star \sqrt{\rho}^{(1)}) + \frac{1}{2} \mathcal{L}_{\mathbf{u}} (\sqrt{\rho}^{(1)}) &= 0 \\ \frac{\partial (\phi_x^{(1)})}{\partial t} + \frac{1}{2} \star \mathcal{L}_{\mathbf{u}} (\star \phi_x^{(1)}) + \frac{1}{2} \mathcal{L}_{\mathbf{u}} (\phi_x^{(1)}) + \frac{1}{\star \sqrt{\rho}^{(0)}} di_{\partial_x} p^{(1)} &= 0 \\ \frac{\partial E^{(1)}}{\partial t} + \mathcal{L}_{\mathbf{u}} E^{(1)} + \mathcal{L}_{\mathbf{u}} p^{(1)} &= 0 \end{aligned}$$

These models are supplemented with the equation of state for $p^{(1)}$ that was mentioned in chapter 7:

$$p^{(1)} = (\gamma - 1) (E^{(1)} - E_{kin}^{(1)})$$

To obtain the weak forms corresponding to (7.2) and (7.1), define $\Omega \subset \mathbb{R}$ and let test function $v^{(1)} \in \Lambda^{(1)}(\Omega)$. Taking the L^2 inner product of (7.2) with $v^{(1)}$ results in the following weak form:

$$\begin{aligned} \left(v^{(1)}, \frac{\partial \rho^{(1)}}{\partial t} \right) + (v^{(1)}, \mathcal{L}_{\mathbf{u}} \rho^{(1)}) &= 0 \\ \left(v^{(1)}, \frac{\partial m_x^{(1)}}{\partial t} \right) + (v^{(1)}, \mathcal{L}_{\mathbf{u}} m_x^{(1)}) + (v^{(1)}, di_{\partial_x} p^{(1)}) &= 0 \\ \left(v^{(1)}, \frac{\partial E^{(1)}}{\partial t} \right) + (v^{(1)}, \mathcal{L}_{\mathbf{u}} E^{(1)}) + (v^{(1)}, \mathcal{L}_{\mathbf{u}} p^{(1)}) &= 0 \end{aligned} \quad (7.3)$$

Similarly, for the Roe variable model of (7.1):

$$\begin{aligned} & \left(v^{(1)}, \frac{\partial \sqrt{\rho}^{(1)}}{\partial t} \right)_{\Omega} + \frac{1}{2} \left(v^{(1)}, \star \mathcal{L}_{\mathbf{u}} \left(\star \sqrt{\rho}^{(1)} \right) \right)_{\Omega} + \frac{1}{2} \left(v^{(1)}, \mathcal{L}_{\mathbf{u}} \left(\sqrt{\rho}^{(1)} \right) \right)_{\Omega} = 0 \\ & \left(v^{(1)}, \frac{\partial \phi_x^{(1)}}{\partial t} \right)_{\Omega} + \frac{1}{2} \left(v^{(1)}, \star \mathcal{L}_{\mathbf{u}} \left(\star \phi_x^{(1)} \right) \right)_{\Omega} + \frac{1}{2} \left(v^{(1)}, \mathcal{L}_{\mathbf{u}} \left(\phi_x^{(1)} \right) \right)_{\Omega} + \left(v^{(1)}, \frac{1}{\star \sqrt{\rho}^{(1)}} di_{\partial_i} p^{(1)} \right)_{\Omega} = 0 \quad (7.4) \\ & \left(v^{(1)}, \frac{\partial E^{(1)}}{\partial t} \right)_{\Omega} + \left(v^{(1)}, \mathcal{L}_{\mathbf{u}} E^{(1)} \right)_{\Omega} + \left(v^{(1)}, \mathcal{L}_{\mathbf{u}} p^{(1)} \right)_{\Omega} = 0 \end{aligned}$$

As was done in chapters 5 and 6 the advection terms containing the Hodge \star operator in (7.4) are rewritten. The relation shown in (5.1) is used to obtain the following weak form for the Roe variable model:

$$\begin{aligned} & \left(v^{(1)}, \frac{\partial \sqrt{\rho}^{(1)}}{\partial t} \right)_{\Omega} + \frac{1}{2} \left(v^{(1)}, \mathcal{L}_{\mathbf{u}} \left(\sqrt{\rho}^{(1)} \right) \right)_{\Omega} - \frac{1}{2} \left(\mathcal{L}_{\mathbf{u}} \left(v^{(1)} \right), \sqrt{\rho}^{(1)} \right)_{\Omega} + \frac{1}{2} \int_{\partial \Omega} i_{\mathbf{u}} \left(v^{(1)} \wedge \star \sqrt{\rho}^{(1)} \right) = 0 \\ & \left(v^{(1)}, \frac{\partial \phi_x^{(1)}}{\partial t} \right)_{\Omega} + \frac{1}{2} \left(v^{(1)}, \mathcal{L}_{\mathbf{u}} \left(\phi_x^{(1)} \right) \right)_{\Omega} - \frac{1}{2} \left(\mathcal{L}_{\mathbf{u}} \left(v^{(1)} \right), \phi_x^{(1)} \right)_{\Omega} + \frac{1}{2} \int_{\partial \Omega} i_{\mathbf{u}} \left(v^{(1)} \wedge \star \phi_x^{(1)} \right) + \left(v^{(1)}, \frac{1}{\star \sqrt{\rho}^{(1)}} di_{\partial_i} p^{(1)} \right)_{\Omega} = 0 \\ & \left(v^{(1)}, \frac{\partial E^{(1)}}{\partial t} \right)_{\Omega} + \left(v^{(1)}, \mathcal{L}_{\mathbf{u}} E^{(1)} \right)_{\Omega} + \left(v^{(1)}, \mathcal{L}_{\mathbf{u}} p^{(1)} \right)_{\Omega} = 0 \quad (7.5) \end{aligned}$$

7.2. Discrete models

With the weak forms in place the next step is to discretize them. First the weak form of the regular model as given in (7.3) is discretized, followed by the model posed in terms of Roe variables given in (7.5). For both models closed knot vectors are used to generate the finite-dimensional bases for $\Lambda_{h,p+1}^{(0)}(\Omega_h) \subset \Lambda^{(0)}(\Omega)$, $\Lambda_{h,p}^{(1)}(\Omega_h) \subset \Lambda^{(1)}(\Omega)$. Once the discrete models have been defined we will take a look at two specific aspects that come up when implementing these discretizations: Boundary condition application, which is covered in section 7.2.3, and kinetic energy modeling, covered in section 7.2.4.

7.2.1. Regular model

We define:

$$\begin{aligned} v_h^{(1)} &= \sum_{i=1}^N v_i \psi_i^{(1)} = \mathbf{v}^T \boldsymbol{\psi}^{(1)} & \rho_h^{(1)} &= \sum_{i=1}^N \rho_i \psi_i^{(1)} = \boldsymbol{\rho}^T \boldsymbol{\psi}^{(1)} \\ (m_x^{(1)})_h &= \sum_{i=1}^N (m_x)_i \psi_i^{(1)} = \mathbf{m}_x^T \boldsymbol{\psi}^{(1)} & E_h^{(1)} &= \sum_{i=1}^N E_i \psi_i^{(1)} = \mathbf{E}^T \boldsymbol{\psi}^{(1)} \end{aligned}$$

Where $v_h^{(1)}, \rho_h^{(1)}, (m_x^{(1)})_h, E_h^{(1)} \in \Lambda_{h,p}^{(1)}(\Omega_h)$. The discrete kinetic energy is computed according to (3.9):

$$\left(E_{kin}^{(1)} \right)_h = \frac{1}{2} \frac{\left\langle (m_x^{(1)})_h, \star (m_x^{(1)})_h \right\rangle}{\star \rho_h^{(1)}} \quad (7.6)$$

In section 2.2.4 a method was introduced that allows one to construct B-splines that are exact products of lower-degree B-splines. While that method can be applied here to resolve the product $\langle m_h^{(1,n)}, \star^{\sharp} m_h^{(1,n)} \rangle$ this leaves us with $1/(\star \rho_h^{(1)})$. This implies that $\left(E_{kin}^{(1)} \right)_h$ does not fall within any B-spline-discretized subspace of $\Lambda^{(1)}(\Omega)$ but is in fact a Non-Uniform Rational B-Spline (NURBS). The question of how to discretize the operators acting on $\left(E_{kin}^{(1)} \right)_h$ is covered in section 7.2.4, for now the composition of these operators will be denoted with \mathbb{K} . For velocity field \mathbf{u} a similar thing holds, since we know that:

$$\mathbf{u} = \left(\frac{m_x^{(1)}}{\star \rho^{(1)}} \right)^{\sharp} \approx \mathbf{u}_h = \left(\frac{(m_x^{(1)})_h}{\star \rho_h^{(1)}} \right)^{\sharp}$$

As such \mathbf{u}_h is a NURBS as well and \mathbf{u}_h^b (the 1-form that is used to construct contraction matrices $\mathbb{C}_{\mathbf{u}}^{(k+1,k)}$) falls outside of the 1-form space of the discrete (B-spline) De Rham complex. We can use this NURBS directly in the discretized weak form since \mathbf{u}_h is only used in the nonlinear component of the contraction matrix of the discrete interior product; it being a NURBS does not affect discrete interior product map $\Lambda_{h,p}^{(1)}(\Omega_h) \rightarrow \Lambda_{h,p+1}^{(0)}(\Omega_h)$. Discretizing the exterior derivatives and interior products in the usual way then gives:

$$\begin{aligned} & \mathbf{v}^T \mathbb{M}^{(1)} \frac{\partial \boldsymbol{\rho}}{\partial t} + \mathbf{v}^T \mathbb{M}^{(1)} \mathbb{E}^{(1,0)} (\mathbb{M}^{(0)})^{-1} \mathbb{C}_{\mathbf{u}_h}^{(0,1)} \boldsymbol{\rho} = 0 \\ \mathbf{v}^T \mathbb{M}^{(1)} \frac{\partial \mathbf{m}_x}{\partial t} + \mathbf{v}^T \mathbb{M}^{(1)} \mathbb{E}^{(1,0)} (\mathbb{M}^{(0)})^{-1} \mathbb{C}_{\mathbf{u}_h}^{(0,1)} \mathbf{m}_x + (\gamma - 1) \mathbf{v}^T \mathbb{M}^{(1)} \mathbb{E}^{(1,0)} (\mathbb{M}^{(0)})^{-1} \mathbb{C}_{\partial_x}^{(0,1)} \mathbf{E} \\ & \quad - (\gamma - 1) \mathbf{v}^T \mathbb{K}_{\partial_x} \mathbf{E}_{kin} = 0 \\ \mathbf{v}^T \mathbb{M}^{(1)} \frac{\partial \mathbf{E}}{\partial t} + \gamma \mathbf{v}^T \mathbb{M}^{(1)} \mathbb{E}^{(1,0)} (\mathbb{M}^{(0)})^{-1} \mathbb{C}_{\mathbf{u}_h}^{(0,1)} \mathbf{E} - (\gamma - 1) \mathbf{v}^T \mathbb{K}_{\mathbf{u}_h} \mathbf{E}_{kin} = 0 \end{aligned} \quad (7.7)$$

Which should hold for all $\mathbf{v} \in \mathbb{R}^N$ and defines the semi-discrete problem. To reiterate, the formulation and structure of the matrices denoted with \mathbb{K} will be covered in section 7.2.4. Note that the momentum and energy equations are coupled through linear and nonlinear terms, whereas the coupling between the continuity and momentum equations is exclusively nonlinear through \mathbf{u}_h . As is covered in section 3.6.3 the linearization step will decouple the continuity and momentum equations during each nonlinear iteration, allowing us to solve both equations separately from one another before updating velocity field \mathbf{u}_h . Applying the implicit midpoint method to (7.7) gives the following nonlinear system to be solved:

$$\begin{aligned} & \begin{bmatrix} \mathbb{M}^{(1)} + \frac{\Delta t}{2} \mathbb{M}^{(1)} \mathbb{L}_{\mathbf{u}_h^{k+1/2}} & 0 & 0 \\ 0 & \mathbb{M}^{(1)} + \frac{\Delta t}{2} \mathbb{M}^{(1)} \mathbb{L}_{\mathbf{u}_h^{k+1/2}} & \frac{(\gamma-1)\Delta t}{2} \mathbb{M}^{(1)} \mathbb{L}_{\partial_x} \\ 0 & 0 & \mathbb{M}^{(1)} + \frac{\gamma\Delta t}{2} \mathbb{M}^{(1)} \mathbb{L}_{\mathbf{u}_h^{k+1/2}} \end{bmatrix} \begin{bmatrix} \boldsymbol{\rho}^{k+1} \\ \mathbf{m}_x^{k+1} \\ \mathbf{E}^{k+1} \end{bmatrix} \\ = & \begin{bmatrix} \mathbb{M}^{(1)} - \frac{\Delta t}{2} \mathbb{M}^{(1)} \mathbb{L}_{\mathbf{u}_h^{k+1/2}} & 0 & 0 \\ 0 & \mathbb{M}^{(1)} - \frac{\Delta t}{2} \mathbb{M}^{(1)} \mathbb{L}_{\mathbf{u}_h^{k+1/2}} & -\frac{(\gamma-1)\Delta t}{2} \mathbb{M}^{(1)} \mathbb{L}_{\partial_x} \\ 0 & 0 & \mathbb{M}^{(1)} - \frac{\gamma\Delta t}{2} \mathbb{M}^{(1)} \mathbb{L}_{\mathbf{u}_h^{k+1/2}} \end{bmatrix} \begin{bmatrix} \boldsymbol{\rho}^k \\ \mathbf{m}_x^k \\ \mathbf{E}^k \end{bmatrix} \\ & + (\gamma - 1) \Delta t \begin{bmatrix} 0 \\ \mathbb{K}_{\partial_x} \\ \mathbb{K}_{\mathbf{u}_h^{k+1/2}} \end{bmatrix} \mathbf{E}_{kin}^{k+1/2} \end{aligned} \quad (7.8)$$

Where the following notation has been introduced in order to improve the legibility of (7.8):

$$\begin{aligned} \mathbb{L}_{\mathbf{u}_h^{k+1/2}} &= \mathbb{E}^{(1,0)} (\mathbb{M}^{(0)})^{-1} \mathbb{C}_{\mathbf{u}_h^{k+1/2}}^{(0,1)} \\ \mathbb{L}_{\partial_x} &= \mathbb{E}^{(1,0)} (\mathbb{M}^{(0)})^{-1} \mathbb{C}_{\partial_x}^{(0,1)} \end{aligned}$$

The half-time step velocity field $\mathbf{u}_h^{k+1/2}$ is computed as:

$$\mathbf{u}_h^{k+1/2} = \left(\frac{\left(m_x^{(1)} \right)_h^{k+1/2}}{\star \left(\rho^{(1)} \right)_h^{k+1/2}} \right)^\# = \left(\frac{\left(m_x^{(1)} \right)_h^k + \left(m_x^{(1)} \right)_h^{k+1}}{\star \left(\left(\rho^{(1)} \right)_h^k + \left(\rho^{(1)} \right)_h^{k+1} \right)} \right)^\#$$

As can be seen in (7.8) the continuity equation is only coupled to the momentum and energy equations through the nonlinear velocity contribution contained in discrete advection operator $\mathbb{L}_{\mathbf{u}_h^{k+1/2}}$. Linearizing (7.8) through the velocity field $\mathbf{u}_h^{k+1/2}$ allows us to perform nonlinear Picard iterations by first estimating $\left(m_x^{(1)} \right)_h^{k+1}$ and $\left(\rho^{(1)} \right)_h^{k+1}$ such that $\mathbf{u}_h^{k+1/2}$ can be approximated; this velocity estimate is then used in solving the continuity equation:

$$\left(\mathbb{M}^{(1)} + \frac{\Delta t}{2} \mathbb{M}^{(1)} \mathbb{L}_{\mathbf{u}_h^{k+1/2}} \right) \boldsymbol{\rho}^{k+1} = \left(\mathbb{M}^{(1)} - \frac{\Delta t}{2} \mathbb{M}^{(1)} \mathbb{L}_{\mathbf{u}_h^{k+1/2}} \right) \boldsymbol{\rho}^k$$

After which the same velocity field estimate is used to solve the coupled momentum-energy system:

$$\begin{aligned} & \begin{bmatrix} \mathbb{M}^{(1)} + \frac{\Delta t}{2} \mathbb{M}^{(1)} \mathbb{L}_{\mathbf{u}_h^{k+1/2}} & \frac{(\gamma-1)\Delta t}{2} \mathbb{M}^{(1)} \mathbb{L}_{\partial_x} \\ 0 & \mathbb{M}^{(1)} + \frac{\gamma\Delta t}{2} \mathbb{M}^{(1)} \mathbb{L}_{\mathbf{u}_h^{k+1/2}} \end{bmatrix} \begin{bmatrix} \mathbf{m}_x^{k+1} \\ \mathbf{E}^{k+1} \end{bmatrix} \\ = & \begin{bmatrix} \mathbb{M}^{(1)} - \frac{\Delta t}{2} \mathbb{M}^{(1)} \mathbb{L}_{\mathbf{u}_h^{k+1/2}} & -\frac{(\gamma-1)\Delta t}{2} \mathbb{M}^{(1)} \mathbb{L}_{\partial_x} \\ 0 & \mathbb{M}^{(1)} - \frac{\gamma\Delta t}{2} \mathbb{M}^{(1)} \mathbb{L}_{\mathbf{u}_h^{k+1/2}} \end{bmatrix} \begin{bmatrix} \mathbf{m}_x^k \\ \mathbf{E}^k \end{bmatrix} + (\gamma-1)\Delta t \begin{bmatrix} \mathbb{K}_{\partial_x} \\ \mathbb{K}_{\mathbf{u}_h^{k+1/2}} \end{bmatrix} \mathbf{E}_{kin}^{k+1/2} \end{aligned}$$

This gives new solution estimates $\boldsymbol{\rho}^{k+1}$ and \mathbf{m}_x^{k+1} that can be used to update the velocity field estimate $\mathbf{u}_h^{k+1/2}$. When the nonlinear convergence tolerance outlined in section 3.6.3 has been met the current time step is finished and the solution is advanced to the next time step.

7.2.2. Skew-symmetric Roe variable model

Similar to what was done for the regular model we define the following discretized quantities for the variables given in (7.5):

$$\begin{aligned} v_h^{(1)} &= \sum_{i=1}^N v_i \psi_i^{(1)} = \mathbf{v}^T \boldsymbol{\psi}^{(1)} & \sqrt{\rho}_h^{(1)} &= \sum_{i=1}^N \sqrt{\rho}_i \psi_i^{(1)} = \sqrt{\boldsymbol{\rho}}^T \boldsymbol{\psi}^{(1)} \\ (\phi_x^{(1)})_h &= \sum_{i=1}^N (\phi_x)_i \psi_i^{(1)} = \boldsymbol{\phi}_x^T \boldsymbol{\psi}^{(1)} & E_h^{(1)} &= \sum_{i=1}^N E_i \psi_i^{(1)} = \mathbf{E}^T \boldsymbol{\psi}^{(1)} \end{aligned}$$

With $v_h^{(1)}, \sqrt{\rho}_h^{(1)}, (\phi_x^{(1)})_h, E_h^{(1)} \in \Lambda_{h,p}^{(1)}(\Omega_h)$. As was defined in (3.24), kinetic energy $(E_{kin}^{(1)})_h$ is a squared quantity in this model:

$$(E_{kin}^{(1)})_h = \frac{1}{2} \langle (\phi_x^{(1)})_h, \star (\phi_x^{(1)})_h \rangle \quad (7.9)$$

It can thus be computed exactly from $(\phi_x^{(1)})_h$ with the B-spline product operation outlined in section 2.2.4 but still falls outside of the function spaces of the discrete De Rham complex. As was done for the regular model the matrices \mathbb{K} will be used to indicate compositions of operators acting on the discrete kinetic energy; in section 7.2.4 these matrices are defined in more detail. Velocity field \mathbf{u}_h is computed in similar fashion to the regular model:

$$\mathbf{u} = \left(\frac{\phi_x^{(1)}}{\star \sqrt{\rho}^{(1)}} \right)^\# \approx \mathbf{u}_h = \left(\frac{(\phi_x^{(1)})_h}{\star \sqrt{\rho}_h^{(1)}} \right)^\#$$

And as such \mathbf{u}_h is again a NURBS. The aforementioned basis function expansions are applied to (7.5), resulting in the following semi-discrete system:

$$\begin{aligned} \mathbf{v}^T \mathbb{M}^{(1)} \frac{\partial \sqrt{\boldsymbol{\rho}}}{\partial t} + \frac{1}{2} \mathbf{v}^T \mathbb{M}^{(1)} \mathbb{E}^{(1,0)} (\mathbb{M}^{(0)})^{-1} \mathbb{C}_{\mathbf{u}_h}^{(0,1)} \sqrt{\boldsymbol{\rho}} - \frac{1}{2} \mathbf{v}^T \left[\mathbb{M}^{(1)} \mathbb{E}^{(1,0)} (\mathbb{M}^{(0)})^{-1} \mathbb{C}_{\mathbf{u}_h}^{(0,1)} \right]^T \sqrt{\boldsymbol{\rho}} + \frac{1}{2} \int_{\partial\Omega} i_{\mathbf{u}_h} \left(v_h^{(1)} \wedge \star \sqrt{\rho}_h^{(1)} \right) &= 0 \\ \mathbf{v}^T \mathbb{M}^{(1)} \frac{\partial \boldsymbol{\phi}_x}{\partial t} + \frac{1}{2} \mathbf{v}^T \mathbb{M}^{(1)} \mathbb{E}^{(1,0)} (\mathbb{M}^{(0)})^{-1} \mathbb{C}_{\mathbf{u}_h}^{(0,1)} \boldsymbol{\phi}_x - \frac{1}{2} \mathbf{v}^T \left[\mathbb{M}^{(1)} \mathbb{E}^{(1,0)} (\mathbb{M}^{(0)})^{-1} \mathbb{C}_{\mathbf{u}_h}^{(0,1)} \right]^T \boldsymbol{\phi}_x + \frac{1}{2} \int_{\partial\Omega} i_{\mathbf{u}_h} \left(v_h^{(1)} \wedge \star (\phi_x^{(1)})_h \right) \\ + (\gamma-1) \mathbf{v}^T \mathbb{D}_{\sqrt{\boldsymbol{\rho}}} \mathbb{E}^{(1,0)} (\mathbb{M}^{(0)})^{-1} \mathbb{C}_{\partial_x}^{(0,1)} \mathbf{E} - (\gamma-1) \mathbf{v}^T \mathbb{K}_{\partial_x} \mathbf{E}_{kin} &= 0 \\ \mathbf{v}^T \mathbb{M}^{(1)} \frac{\partial \mathbf{E}}{\partial t} + \gamma \mathbf{v}^T \mathbb{M}^{(1)} \mathbb{E}^{(1,0)} (\mathbb{M}^{(0)})^{-1} \mathbb{C}_{\mathbf{u}_h}^{(0,1)} \mathbf{E} - (\gamma-1) \mathbf{v}^T \mathbb{K}_{\mathbf{u}_h} \mathbf{E}_{kin} &= 0 \end{aligned} \quad (7.10)$$

Where $\mathbb{D}_{\sqrt{\boldsymbol{\rho}}}$ is a matrix with entries $\int_{\Omega} \frac{\psi_i^{(1)} \psi_j^{(1)}}{\star \sqrt{\rho}_h^{(1)}}$. In other words, its entries are similar to those of mass matrix $\mathbb{M}^{(1)}$ apart from the mass density term in the denominator. The boundary integral terms in (7.10) are left in their current formulations for reasons that are covered in section 7.2.3. Applying the implicit midpoint rule

to (7.10) leads to:

$$\begin{aligned}
& \begin{bmatrix} \mathbb{M}^{(1)} + \frac{\Delta t}{4} \tilde{\mathbb{L}}_{\mathbf{u}_h^{k+1/2}} & 0 & 0 \\ 0 & \mathbb{M}^{(1)} + \frac{\Delta t}{4} \tilde{\mathbb{L}}_{\mathbf{u}_h^{k+1/2}} & (\gamma-1)\mathbb{D}^{(1)}_{\sqrt{\rho}^{k+1/2}} \mathbb{L}_{\partial_x} \\ 0 & 0 & \mathbb{M}^{(1)} + \gamma \mathbb{M}^{(1)} \mathbb{L}_{\mathbf{u}_h^{k+1/2}} \end{bmatrix} \begin{bmatrix} \sqrt{\rho}^{k+1} \\ \phi_x^{k+1} \\ \mathbf{E}^{k+1} \end{bmatrix} \\
& = \begin{bmatrix} \mathbb{M}^{(1)} - \frac{\Delta t}{4} \tilde{\mathbb{L}}_{\mathbf{u}_h^{k+1/2}} & 0 & 0 \\ 0 & \mathbb{M}^{(1)} - \frac{\Delta t}{4} \tilde{\mathbb{L}}_{\mathbf{u}_h^{k+1/2}} & -(\gamma-1)\mathbb{D}^{(1)}_{\sqrt{\rho}^{k+1/2}} \mathbb{L}_{\partial_x} \\ 0 & 0 & \mathbb{M}^{(1)} - \gamma \mathbb{M}^{(1)} \mathbb{L}_{\mathbf{u}_h^{k+1/2}} \end{bmatrix} \begin{bmatrix} \sqrt{\rho}^k \\ \phi_x^k \\ \mathbf{E}^k \end{bmatrix} \\
& \quad + (\gamma-1)\Delta t \begin{bmatrix} 0 \\ \mathbb{K}_{\partial_x} \\ \mathbb{K}_{\mathbf{u}_h^{k+1/2}} \end{bmatrix} \mathbf{E}_{kin}^{k+1/2}
\end{aligned} \tag{7.11}$$

Where again additional notation was used to simplify (7.11):

$$\begin{aligned}
\mathbb{L}_{\mathbf{u}_h^{k+1/2}} &= \mathbb{E}^{(1,0)} (\mathbb{M}^{(0)})^{-1} \mathbb{C}_{\mathbf{u}_h^{k+1/2}}^{(0,1)} \\
\mathbb{L}_{\partial_x} &= \mathbb{E}^{(1,0)} (\mathbb{M}^{(0)})^{-1} \mathbb{C}_{\partial_x}^{(0,1)} \\
\tilde{\mathbb{L}}_{\mathbf{u}_h^{k+1/2}} &= \mathbb{M}^{(1)} \mathbb{L}_{\mathbf{u}_h^{k+1/2}} - \left[\mathbb{M}^{(1)} \mathbb{L}_{\mathbf{u}_h^{k+1/2}} \right]^T + \int_{\partial\Omega} i_{\mathbf{u}_h^{k+1/2}} \left(\boldsymbol{\psi}^{(1)} \wedge \star (\boldsymbol{\psi}^{(1)})^T \right)
\end{aligned}$$

Similar to the regular model the half-time step velocity $\mathbf{u}_h^{k+1/2}$ is computed with:

$$\mathbf{u}_h^{k+1/2} = \left(\frac{(\phi_x^{(1)})_h^{k+1/2}}{\star (\sqrt{\rho}^{(1)})_h^{k+1/2}} \right)^\# = \left(\frac{(\phi_x^{(1)})_h^k + (\phi_x^{(1)})_h^{k+1}}{\star \left((\sqrt{\rho}^{(1)})_h^k + (\sqrt{\rho}^{(1)})_h^{k+1} \right)} \right)^\#$$

Similarly entry (i, j) of matrix $\mathbb{D}^{(1)}_{\sqrt{\rho}^{k+1/2}}$ is computed with the half-time step mass density:

$$\left(\mathbb{D}^{(1)}_{\sqrt{\rho}^{k+1/2}} \right)_{ij} = \int_{\Omega} \frac{\psi_i^{(1)} \psi_j^{(1)}}{\star \frac{1}{2} \left[\left(\sqrt{\rho}_h^{(1)} \right)^k + \left(\sqrt{\rho}_h^{(1)} \right)^{k+1} \right]}$$

The continuity equation becomes decoupled from the momentum and energy equations just as was the case for the regular model. Using the same approach (Picard iterations) to linearize the system through the velocity field $\mathbf{u}_h^{k+1/2}$ then allows one to iteratively solve the continuity equation and the momentum-energy system separately. The linearized continuity equation is:

$$\left(\mathbb{M}^{(1)} + \frac{\Delta t}{4} \tilde{\mathbb{L}}_{\mathbf{u}_h^{k+1/2}} \right) \sqrt{\rho}^{k+1} = \left(\mathbb{M}^{(1)} - \frac{\Delta t}{4} \tilde{\mathbb{L}}_{\mathbf{u}_h^{k+1/2}} \right) \sqrt{\rho}^k$$

Whereas the combined momentum-energy system is:

$$\begin{aligned}
& \begin{bmatrix} \mathbb{M}^{(1)} + \frac{\Delta t}{4} \tilde{\mathbb{L}}_{\mathbf{u}_h^{k+1/2}} & (\gamma-1)\mathbb{D}^{(1)}_{\sqrt{\rho}^{k+1/2}} \mathbb{L}_{\partial_x} \\ 0 & \mathbb{M}^{(1)} + \gamma \mathbb{M}^{(1)} \mathbb{L}_{\mathbf{u}_h^{k+1/2}} \end{bmatrix} \begin{bmatrix} \phi_x^{k+1} \\ \mathbf{E}^{k+1} \end{bmatrix} \\
& = \begin{bmatrix} \mathbb{M}^{(1)} - \frac{\Delta t}{4} \tilde{\mathbb{L}}_{\mathbf{u}_h^{k+1/2}} & -(\gamma-1)\mathbb{D}^{(1)}_{\sqrt{\rho}^{k+1/2}} \mathbb{L}_{\partial_x} \\ 0 & \mathbb{M}^{(1)} - \gamma \mathbb{M}^{(1)} \mathbb{L}_{\mathbf{u}_h^{k+1/2}} \end{bmatrix} \begin{bmatrix} \phi_x^k \\ \mathbf{E}^k \end{bmatrix} + (\gamma-1)\Delta t \begin{bmatrix} \mathbb{K}_{\partial_x} \\ \mathbb{K}_{\mathbf{u}_h^{k+1/2}} \end{bmatrix} \mathbf{E}_{kin}^{k+1/2}
\end{aligned}$$

A key difference between the skew-symmetric method and the regular model that was covered in the previous section is the matrix $\mathbb{D}^{(1)}_{\sqrt{\rho}^{k+1/2}}$. After solving the continuity equation the newly-found solution for $\sqrt{\rho}^{k+1}$ is to be used in constructing $\mathbb{D}^{(1)}_{\sqrt{\rho}^{k+1/2}}$ for the momentum-energy system. This is required in order to maintain discrete momentum conservation. After having solved the momentum-energy system it is to be checked whether the nonlinear convergence criteria have been fulfilled; if this is the case the current time step terminates and the next time step is initialized.

7.2.3. Application of boundary conditions

Whereas the test cases of chapters 5 and 6 were periodic and used explicitly periodic spline bases, the shock tube case considered in this chapter is modeled with wall boundaries on either side of $\Omega = [0, 1]$. These wall boundaries correspond to zero-flux conditions that are imposed on the regular- and skew-symmetric (Roe variable) models given in (7.8) and (7.11) respectively. This will be done in various ways.

First off, the boundary integrals present in Roe variable system of equations (7.11) physically correspond to the fluxes of $\sqrt{\rho}_h^{(1)}$ and $(\phi_x^{(1)})_h$ over $\partial\Omega$, induced by velocity field \mathbf{u}_h . Since zero-flux conditions are imposed the boundary integrals are equal to zero and thus removed from (7.11).

Furthermore we remark that wall boundaries not only imply zero-flux boundary conditions (i.e. prohibit transport across said boundary), but also imply zero wall-normal velocity. Naturally these two conditions are related to one another, but imply different things and are applied differently to the numerical system. To ensure that $\mathbf{u}_h|_{\partial\Omega} = 0$ we impose homogeneous Dirichlet conditions on $(m_x^{(1)})_h$ and $(\phi_x^{(1)})_h$. With the open knot vector basis defined in section 2.2.5 this implies that the expansion coefficients of the first and last basis function are zero.

Lastly, zero-flux boundary conditions are applied to the discrete interior products. Recall that the discrete interior product is resolved with a weak form: To compute the discrete $k-1$ -form corresponding to a k -form (and a vector field) the basis of the space of discrete $k-1$ -forms is used as test functions. In this one-dimensional context:

$$\mathbb{M}^{(0)} \boldsymbol{\beta} = \mathbb{C}_{\mathbf{u}}^{(0,1)} \boldsymbol{\alpha}$$

Where $\boldsymbol{\beta}$, $\boldsymbol{\alpha}$ are the coefficient vectors of $\beta_h^{(0)}$, $\alpha_h^{(1)}$ respectively:

$$\beta_h^{(0)} = \sum_{i=1}^N \beta_i \psi_i^{(0)} = \boldsymbol{\beta}^T \boldsymbol{\psi}^{(0)} \quad \alpha_h^{(1)} = \sum_{i=2}^N \alpha_i \psi_i^{(1)} = \boldsymbol{\alpha}^T \boldsymbol{\psi}^{(1)}$$

Such that $\beta_h^{(0)} = i_{\mathbf{u}_h} \alpha_h^{(1)}$. Zero-flux boundary conditions imply that $\beta_h^{(0)}|_{\partial\Omega} = 0$ and thus correspond to Dirichlet boundary conditions on $\beta_h^{(0)}$. With closed knot vectors only $\psi_1^{(0)}$ and $\psi_N^{(0)}$ are nonzero on $\partial\Omega$, and thus 0-form coefficients β_1 and β_N are directly defined to be zero by the zero-flux conditions. This allows us to slightly reduce the size of the matrices that encode the discrete interior product. Whereas normally $\mathbb{M}^{(0)} \in \mathbb{R}^{(N) \times (N)}$ and $\mathbb{C}_{\mathbf{u}}^{(0,1)} \in \mathbb{R}^{(N) \times (N-1)}$ elimination of the boundary conditions results in $\mathbb{M}^{(0)} \in \mathbb{R}^{(N-2) \times (N-2)}$ and $\mathbb{C}_{\mathbf{u}}^{(0,1)} \in \mathbb{R}^{(N-2) \times N}$, with 0-form coefficient vector $\boldsymbol{\beta}$ as follows:

$$\boldsymbol{\beta} = \begin{bmatrix} 0 \\ (\mathbb{M}^{(0)})^{-1} \mathbb{C}_{\mathbf{u}}^{(0,1)} \boldsymbol{\alpha} \\ 0 \end{bmatrix}$$

7.2.4. Kinetic energy advection modeling

As alluded to in sections 7.2.1 and 7.2.2 discrete kinetic energy $(E_{kin}^{(1)})_h \notin \Lambda_{h,p}^{(1)}(\Omega_h)$. Thus a different way is needed of applying the Lie derivative (consisting of the interior product and exterior derivative) to $(E_{kin}^{(1)})_h$. In both models covered here the kinetic energy term contains the square of each momentum component $(m_i^{(1)})_h$ and $(\phi_i^{(1)})_h$. Recall from section 2.2.4 that there exists an exact method for computing the product of two B-splines as another, higher-degree B-spline. Denote for now this operation with χ .

For the regular model the kinetic energy is then computed by dividing this product spline with the mass density $\star \rho_h^{(1)}$, thereby making discrete kinetic energy a NURBS. While the discrete B-spline spaces $\Lambda_{h,p+1}^{(0)}(\Omega_h)$ and $\Lambda_{h,p}^{(1)}(\Omega_h)$ were defined such that they form a discrete De Rham complex, currently no such construction is known for the more general NURBS functions. Let $\Gamma_{h,p,q}^{(1)}(\Omega_h) \subset \Lambda^{(1)}(\Omega)$ denote for now the space of NURBS-discrete 1-forms of degree p in the numerator and degree q in its denominator. Resolving the kinetic energy $(E_{kin}^{(1)})_h$ and its Lie derivative in the regular model is carried out in the way shown in the following

commutative diagram, which relates the various function spaces used:

$$\begin{array}{ccc}
 \Lambda_{h,p+1}^{(0)}(\Omega_h) & \xrightarrow{d} & \Lambda_{h,p}^{(1)}(\Omega_h) \\
 & \searrow^{i_{\mathbf{u}_h}} & \downarrow \chi \\
 & & \Lambda_{h,2p}^{(1)}(\Omega_h) \\
 & & \downarrow \star \rho_h^{(1)} \\
 & & \Gamma_{h,2p,p}^{(1)}(\Omega_h)
 \end{array}$$

Starting with $(m_x^{(1)})_h \in \Lambda_p^{(1)}(\Omega_h)$ the square $\langle (m_x^{(1)})_h, \star(m_x^{(1)})_h \rangle \in \Lambda_{h,2p}^{(1)}(\Omega_h)$ can be computed exactly with χ , followed by division with $\star \rho_h^{(1)}$ (and application of appropriate scalar factors) to compute $(E_{kin}^{(1)})_h \in \Gamma_{h,2p,p}^{(1)}(\Omega_h)$. To retain the discrete De Rham complex the discrete interior product $i_{\mathbf{u}_h}$ acting on $(E_{kin}^{(1)})_h$ is made to map onto $\Lambda_{h,p+1}^{(0)}(\Omega_h)$. Recall the approach taken in section 2.3.4 for the discrete interior product that maps k -forms to $k-1$ -forms: A weak form is constructed with test functions $\psi_i^{(k-1)}$. In the current context the discrete kinetic energy can thus be mapped to $\Lambda_{h,p+1}^{(0)}(\Omega_h)$ by defining a weak form for $i_{\mathbf{u}_h}(E_{kin}^{(1)})_h$ with the 0-forms basis functions. The subsequent application of the exterior derivative d is then still exact, and the resulting discrete Lie derivative maps $\Gamma_{h,2p,p}^{(1)}(\Omega_h) \rightarrow \Lambda_p^{(1)}(\Omega_h)$.

The discrete operators \mathbb{K}_{∂_x} and $\mathbb{K}_{\mathbf{u}_h^{k+1/2}}$ given in (7.8) are then defined as:

$$\begin{aligned}
 \mathbb{K}_{\partial_x} &= \mathbb{M}^{(1)} \mathbb{E}^{(1,0)} (\mathbb{M}^{(0)})^{-1} \mathbb{C}_{\partial_x}^{(0,\bar{1})} \\
 \mathbb{K}_{\mathbf{u}_h^{k+1/2}} &= \mathbb{M}^{(1)} \mathbb{E}^{(1,0)} (\mathbb{M}^{(0)})^{-1} \mathbb{C}_{\mathbf{u}_h^{k+1/2}}^{(0,\bar{1})}
 \end{aligned}$$

Where the superscript $\bar{1}$ is used to denote the space of NURBS 1-forms in contraction matrices \mathbb{C} . These operators are thus a composition of the discrete Lie derivative and mass matrix $\mathbb{M}^{(1)}$ due to the weak form that is used to discretize the model.

In the skew-symmetric (Roe variable) model the discrete kinetic energy is a product spline $(E_{kin}^{(1)})_h \in \Lambda_{h,2p}^{(1)}(\Omega_h)$. While it can be used directly it is nonetheless still not contained in any of the spline spaces that are part of the discrete De Rham complex. This leaves two options: The first option would be to use the discrete interior product to project $(E_{kin}^{(1)})_h \in \Lambda_{h,2p}^{(1)}(\Omega_h)$ to $\Lambda_{h,p+1}^{(0)}(\Omega_h)$, similar to the construction used in the regular model that was covered previously. In the second option the discrete interior product maps $(E_{kin}^{(1)})_h$ to higher-degree 0-form spline space $\Lambda_{h,2(p+1)}^{(0)}(\Omega_h)$, which does form a discrete De Rham complex with $\Lambda_{h,2p}^{(1)}(\Omega_h)$. This does not lead to a commutative relation between the product 0- and 1-form spline spaces, since the exterior derivative does not map $\Lambda_{2(p+1)}^{(0)}(\Omega_h) \rightarrow \Lambda_{2p}^{(1)}(\Omega_h)$ (as $2(p+1) \neq 2p+1$). These two constructions can be summarized in commutative diagrams; the left-hand diagram below shows the structure of the first option, which has been dubbed the 'approximate approach', whereas the right-hand diagram shows the second option, that will be referred to as the 'exact approach':

$$\begin{array}{ccc}
 \Lambda_{h,p+1}^{(0)}(\Omega_h) & \xrightarrow{d} & \Lambda_{h,p}^{(1)}(\Omega_h) \\
 \downarrow \chi & \dashrightarrow^{i_{\mathbf{u}_h}} & \downarrow \chi \\
 \Lambda_{h,2(p+1)}^{(0)}(\Omega_h) & \xrightarrow{i_{\mathbf{u}_h}} & \Lambda_{h,2p}^{(1)}(\Omega_h)
 \end{array}
 \qquad
 \begin{array}{ccc}
 \Lambda_{h,p+1}^{(0)}(\Omega_h) & \dashrightarrow^{d} & \Lambda_{h,p}^{(1)}(\Omega_h) \\
 \downarrow \chi & \dashrightarrow^{i_{\mathbf{u}_h}} & \downarrow \chi \\
 \Lambda_{h,2(p+1)}^{(0)}(\Omega_h) & & \Lambda_{h,2p}^{(1)}(\Omega_h) \\
 & \nearrow^{d} & \nwarrow^{i_{\mathbf{u}_h}} \\
 & \Lambda_{h,2p+1}^{(0)}(\Omega_h) &
 \end{array}$$

In both commutative diagrams the spline product operators, interior products and exterior derivatives used to resolve the kinetic energy are denoted with solid arrows. The dashed arrows indicate other operators that are shown for illustrative purposes but are not used within the current context. One can see that in both diagrams the discrete kinetic energy is modeled in $\Lambda_{h,2p}^{(1)}(\Omega_h)$, where it is found through the aforementioned B-spline product map χ that was defined in section 2.2.4. The resulting discrete Lie derivative of the first option mentioned above (indicated in the left diagram) maps $\Lambda_{h,2p}^{(1)}(\Omega_h) \rightarrow \Lambda_{h,p}^{(1)}(\Omega_h)$, whereas in the second option a Lie derivative map $\Lambda_{h,2p}^{(1)}(\Omega_h) \rightarrow \Lambda_{h,2p}^{(1)}(\Omega_h)$ is used. For the approximate kinetic energy advection modeling approach the following matrices have been indicated with \mathbb{K} in (7.11):

$$\begin{aligned}\mathbb{K}_{\partial_x} &= \mathbb{D}_{\sqrt{\rho}^{k+1/2}}^{(1)} \mathbb{E}^{(1,0)} (\mathbb{M}^{(0)})^{-1} \mathbb{C}_{\partial_x}^{(0,\tilde{1})} \\ \mathbb{K}_{\mathbf{u}_h^{k+1/2}} &= \mathbb{M}^{(1)} \mathbb{E}^{(1,0)} (\mathbb{M}^{(0)})^{-1} \mathbb{C}_{\mathbf{u}_h^{k+1/2}}^{(0,\tilde{1})}\end{aligned}$$

Whereas in the exact kinetic energy advection modeling approach:

$$\begin{aligned}\mathbb{K}_{\partial_x} &= \mathbb{D}_{\sqrt{\rho}^{k+1/2}}^{(1,\tilde{1})} \mathbb{E}^{(\tilde{1},\tilde{0})} (\mathbb{M}^{(\tilde{0})})^{-1} \mathbb{C}_{\partial_x}^{(\tilde{0},\tilde{1})} \\ \mathbb{K}_{\mathbf{u}_h^{k+1/2}} &= \mathbb{M}^{(1,\tilde{1})} \mathbb{E}^{(\tilde{1},\tilde{0})} (\mathbb{M}^{(\tilde{0})})^{-1} \mathbb{C}_{\mathbf{u}_h^{k+1/2}}^{(\tilde{0},\tilde{1})}\end{aligned}$$

Tildes indicate the spline product 0- and 1-form spaces $\Lambda_{h,2p+1}^{(0)}(\Omega_h)$, $\Lambda_{h,2p}^{(1)}(\Omega_h)$ and the subscript $(1, \tilde{1})$ indicates a (non-square) mass matrix with integrals of the products $\psi_i^{(1)} \psi_j^{(\tilde{1})}$. The results produced with the two discretization approaches covered here for kinetic energy advection in the skew-symmetric model will be compared in the next section.

7.3. Application to Sod's shock tube

The models described above are used to solve Sod's shock tube problem as described in [50]. For this problem the domain is $\Omega = [0, 1]$. Sod's original formulation uses an initial condition presented in terms of pressure p , mass density ρ and velocity u ; the conversion to primary variables is given by the physical definition $m_x = \rho u$ and the equation of state, which defines $E = \frac{p}{\gamma-1} + \frac{1}{2} \rho u^2$. Figure 7.1 shows the initial condition used to test the numerical models. As mentioned in the introduction of this chapter, the setup of Sod's shock tube physically represents what happens when a diaphragm at $x_0 = 0.5$ that separates two regions with vastly different pressures and mass densities breaks. A right-running shock wave is formed, together with a right-running contact discontinuity and a left-running expansion fan, all originating from the diaphragm location x_0 . The exact solution to this shock tube problem at any time $t > 0$ can be computed with one-dimensional gas dynamics theory; Leveque goes into some detail about this in chapter 9 of [32].

The two external numerical methods that have been used to simulate this shock tube problem (Clawpack and a nodal DG method) have been described in chapter 4. Their results are compared to the simulation results obtained with the regular and skew-symmetric models for various mesh sizes and basis function degrees with respect to the following aspects:

- Solutions and predicted locations of flow features (shock wave, contact discontinuity, expansion fan) at $t = 0.21$
- Maximum (absolute) conservation errors w.r.t. exact solution

The simulation end time $t = 0.21$ is used for several of these comparisons since it is subject to all time steps that come before it; hence any drift and growth in errors over the time span of each simulation is accumulated and included in the results presented here. Flow feature locations are relevant performance indicators, since they indicate (in)accurate behavior of the numerical methods when it comes to predicting the travel speeds of various types of discontinuities in supersonic flows. The maximum conservation errors show upper bounds of how accurate each method follows the evolution of physical quantities of interest over the whole time interval $[0, 0.21]$. Suppose r is a physical quantity of interest, we define the conservation error as:

$$\Delta r(t) = (r_{num}(t) - r_{num}(0)) - (r_{exact}(t) - r_{exact}(0))$$

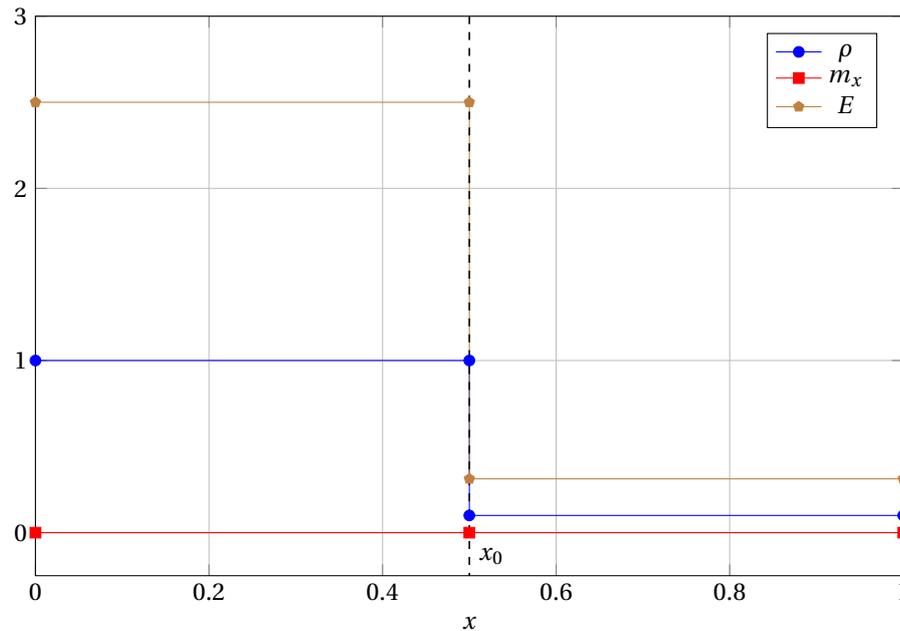


Figure 7.1: Initial condition of Sod's shock tube problem as given in [50], converted to primary variables

For mass and total energy the exact amounts don't change over time and thus the difference ($r_{exact}(t) - r_{exact}(0)$) is zero and drops out. For kinetic energy and momentum the exact and numerical amounts at $t = 0$ are equal to zero, causing $r_{num}(0)$ and $r_{exact}(0)$ to drop out. The only quantity for which all of these variables are nonzero is the internal energy E_{int} . The maximum conservation error is then the maximum of $|\Delta r(t^n)|$ over all time levels $t^n \in [0, 0.21]$. Due to the discontinuities in the exact solution no L^2 error convergence is expected to take place; hence no quantitative comparisons are made with respect to this accuracy criterion.

Three sets of comparisons are made between the different methods, in the following order:

1. Both kinetic energy advection modeling approaches for the skew-symmetric method that were covered in section 7.2.4, for $\epsilon = 10^{-6}$ and $\epsilon = 10^{-12}$
2. The regular and skew-symmetric methods, for $\epsilon = 10^{-6}$ and $\epsilon = 10^{-12}$
3. Clawpack, the nodal DG method and the regular and skew-symmetric methods, the latter two again for $\epsilon = 10^{-6}$

The first of these is covered in section 7.3.1, followed by the comparison in section 7.3.2 between both numerical methods derived in section 7.2. Finally in section 7.3.3 the results of both these methods with $\epsilon = 10^{-6}$ are compared to the results obtained with Clawpack and the nodal DG method that were described in chapter 4. For all results shown a time step size of $\Delta t = 10^{-3}$ was used.

7.3.1. Kinetic energy modeling comparison

As mentioned above the first comparison covered for the one-dimensional Euler equations is made between the two kinetic energy advection modeling approaches for the skew-symmetric method that were discussed in section 7.2.4. We start by looking at the solutions at $t = 0.21$ obtained with both approaches; $n = 100$ basis functions and nonlinear tolerances $\epsilon = 10^{-6}$ and $\epsilon = 10^{-12}$ were used. Said mass density, momentum and total energy density solutions are shown in figure 7.3 together with the exact solution in black.

Oscillatory behavior

It is immediately obvious that both approaches result in oscillatory solutions throughout the entire domain. No matter what basis function order p is used the magnitude of the oscillations remains constant. Two types of oscillations are produced during the simulations. To help illustrate this figure 7.2 shows the mass density solutions of the skew-symmetric method with exact kinetic energy advection modeling and $\epsilon = 10^{-12}$ at time

instants $t = 0.025$, $t = 0.05$ and $t = 0.1$. These three solution instants display the types of oscillations that are being produced. Around $x = 0.6 - 0.7$ the solution at $t = 0.025$ contains several right-moving oscillations that have originated from the initial discontinuity at $x = 0.5$; these are the same oscillations that are shown around $x = 0.8 - 0.9$ at $t = 0.05$. A similar group of left-moving oscillations can be seen around $x = 0.35 - 0.45$ at $t = 0.025$ and around $x = 0.15 - 0.25$ at $t = 0.05$. Both these groups reflect off of the domain walls and travel back towards the center of the domain. At $t = 0.1$ the initially-right-moving oscillations can already be seen to interact with the solution around the two right-moving discontinuities.

The second type of oscillations are those that travel with the local shock speed; these can be seen at $x = 0.55 - 0.6$ at $t = 0.05$ and $x = 0.6 - 0.7$ at $t = 0.1$. They do not cross the discontinuities on either side of them and have a maximum amplitude that does not increase significantly over time, although some interactions occur with the first group of (traveling) oscillations that makes the amplitudes time-varying. Note that the oscillations shown in figures 7.2 and 7.3 are bounded by the discontinuous step sizes adjacent to them on either side; the only place where this does not hold is in the direct vicinity of the rightmost discontinuity. A brief example of how all these oscillations can be (partially) dealt with is given in section 7.3.3, where a comparison is made between the results obtained with the numerical methods derived in this chapter and external methods.

Flow feature locations

Whereas the locations of the expansion wave and the left discontinuity are resolved correctly with both methods, the travel speed of the right discontinuity is being over-predicted by both methods for all orders p . Moreover, the local solutions do not display any kind of monotonic behavior: Going right-to-left across the rightmost discontinuity we first see a local increase in mass density, momentum and total energy density (similar to what is visible in the exact solution) followed by an identical decrease for $p = 2$, $p = 3$. Only after this local minimum do the solutions start oscillating around a mean value that's (approximately) equal to the exact local value. Note that increasing the order of the basis functions does not reduce the oscillatory behavior, nor does it improve the travel speed prediction of the right discontinuity. Lastly we note that there are no visible differences between the results obtained with $\epsilon = 10^{-6}$ and $\epsilon = 10^{-12}$.

Accuracy of conservation behavior

Next we look at the conservation behavior obtained with both kinetic energy advection modeling approaches, shown in figure 7.4. Both mass (in the top left corner) and total energy (at the bottom) are conserved by each method up to $\mathcal{O}(10^{-15})$ for all p and n , with no significant differences between both methods. For the other three physical parameters (momentum, kinetic energy and internal energy) identical trends can be seen: The exact kinetic energy modeling approach leads to slightly lower conservation errors. Whereas for momentum the differences between both methods are constant over the whole range of mesh sizes and a monotonic (and constant) decrease in the conservation error can be seen, for both kinetic and internal energy refining the mesh leads to seemingly asymptotic convergence to non-zero conservation errors with each method. Recall that the total energy is conserved up to $\mathcal{O}(10^{-15})$; since

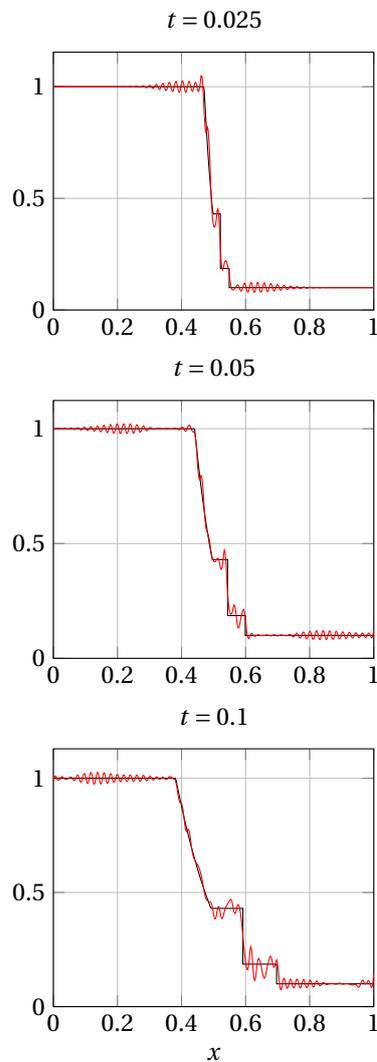


Figure 7.2: Mass density solutions at $t = 0.025$, $t = 0.05$ and $t = 0.1$ of the skew-symmetric model with exact kinetic energy modeling for $\Delta t = 10^{-3}$, $p = 2$ and $\epsilon = 10^{-12}$ compared to the exact solution

total energy is the sum of kinetic and internal energy the graphs of these two energy components are identical.

While the exact kinetic energy advection approach has a small advantage over the approximate approach on the coarser half of the mesh size range for finer meshes this advantage disappears. It can be seen that using $p = 1$ with the exact advection approach gives the best results: For momentum, kinetic energy and internal energy this gives the smallest conservation errors, though the differences with higher-order solutions are small. Apart from this one outlier ($p = 1$ with the exact approach) using higher-order basis functions does not improve conservation behavior. Lastly it is noted that the differences between the results with $\epsilon = 10^{-6}$ and $\epsilon = 10^{-12}$ are negligible compared to the other variations: The only parameters where any difference is visible are mass and total energy, where round-off errors dominate the conservation error.

Conclusions

While small differences exist between the results that were obtained with the approximate and exact kinetic energy advection approaches, these do not justify the differences in computational cost between them. The exact approach requires the construction of significantly larger contraction matrices than the approximate approach. As the degree p is increased the difference in computational cost between the two approaches also increases. The existence of only small differences between the results of both models does not warrant this increase in computational cost, and hence the approximate kinetic energy advection approach was used in lieu of the exact approach.

7.3.2. Regular & skew-symmetric model comparison

Next we compare the performance of the regular and skew-symmetric models with $\epsilon = 10^{-6}$ and $\epsilon = 10^{-12}$. For the latter the approximate kinetic energy advection model was used.

Flow feature locations

We start by comparing their solutions at $t = 0.21$, which are shown in figure 7.5. As can be seen both methods again result in oscillatory solutions, though the skew-symmetric model has notably more oscillations; this is especially noticeable for $x < 0.7$ (on the left side of the left discontinuity) and to the right of the right discontinuity. Between the two discontinuities the amplitude of the oscillations is similar for both models. Both the regular and skew-symmetric models sharply capture the discontinuous gradients at either ends of the expansion wave, and have high gradients at or near both discontinuities. Moreover, the regular model's prediction of the location of the right discontinuity is significantly more accurate than that of the skew-symmetric model and unlike for the skew-symmetric model the right discontinuity is captured monotonically (except for Gibbs' phenomenon). No differences are visible for either method between the results obtained with $\epsilon = 10^{-6}$ and $\epsilon = 10^{-12}$.

Accuracy of conservation behavior

Following this we compare the conservation errors of both methods. As can be seen in figure 7.6 both mass and total energy are again conserved up to $\mathcal{O}(10^{-15})$ for both the regular and skew-symmetric methods: Hence the kinetic and internal energy graphs are identical for both. The mass and total energy graphs are also the places where any differences between $\epsilon = 10^{-6}$ and $\epsilon = 10^{-12}$ are visible, due to differences in round-off errors.

Looking at the momentum graph in the top right corner of figure 7.6, the regular method has momentum conservation errors that are nearly an order of magnitude smaller than those of the skew-symmetric method along the entire mesh size range. A similar trend can be seen when comparing the kinetic and internal energy conservation errors: Although the difference between both methods is smaller than it is for momentum the regular method consistently has lower conservation errors. With $p = 1$ the regular method seems to perform best, whereas there are no noticeable differences between its various higher-order solutions.

Conclusions

In all aspects considered here the regular model outperforms the skew-symmetric model. Firstly it does not have the same erroneous behavior in resolving the travel speed of the rightmost discontinuity. Secondly it is less oscillatory than the skew-symmetric model in most of the domain, although their behavior is seemingly identical between the two discontinuities that are present. Thirdly the conservation errors of the regular model are smaller than those of the skew-symmetric method. This is especially noticeable for momentum but also present for kinetic energy and internal energy; similar performance is seen for mass and total energy.

7.3.3. Comparison with established methods

The last comparison to be made for the one-dimensional Euler equations is between the regular and skew-symmetric methods with $\epsilon = 10^{-6}$ which were compared in section 7.3.2, Clawpack and the nodal DG method of Hesthaven and Warburton; the latter two have been covered briefly in chapter 4.

Solution quality & flow feature locations

Figure 7.7 shows the solutions at $t = 0.21$ for Clawpack and the DG method in the left column and the regular and skew-symmetric models in the right column. We focus on the left column after noting that the results of the regular and skew-symmetric models were already covered in the previous comparisons. It can be seen that the results of Clawpack and the DG method are dissipative instead of dispersive like the regular and skew-symmetric models. Very few oscillations are present in the numerical solutions, but this comes at a cost: The solutions at the discontinuous gradients at either end of the expansion wave and both right-moving discontinuities are smeared out over a small area, resulting in a loss of accuracy. While this effect is most noticeable for Clawpack the DG method is also affected by it and exceeds the dissipative behavior displayed by Clawpack here and there. Increasing the order p used by the DG method also increases the area over which discontinuities (and solutions around discontinuous gradients) are smeared out instead of improving the DG method's resolution. These solutions are also not completely free of unphysical oscillations, as can be seen around $x \approx 0.45$; here the mass density and momentum results of the DG method contain some oscillations whereas the exact solution is monotonic. However, the size and effect of these oscillations pale in comparison to those seen in the results of the regular and skew-symmetric methods.

Despite the oscillations present in the (unfiltered, non-limited) regular and skew-symmetric methods the results reproduce the features present in the exact solution. This raises the possibility of removing (part of) the oscillations through post-processing steps after a simulation has completed. A simple example of such an approach consists of applying a moving average-filter; averaging the numerical solution over a small region reduces the sharpness with which solution features are resolved while removing a (portion of) the oscillations. Figure 7.8 shows how the results of the regular and skew-symmetric methods change when a moving average-filter with a width of $\Delta x = 0.0375$ is applied to their solutions at $t = 0.21$ with $p = 2$. This moving average-filter works by simply taking 400 equally-spaced samples on Ω and averaging in each point all sample values within a distance of $\Delta x/2 = 0.0375/2$.

For comparison the solutions of Clawpack and the DG method (with $p = 2$) are also shown, together with the exact solution in black. While small oscillations can still be seen at $x < 0.25$ and to the right of the rightmost discontinuity, their amplitudes have decreased significantly. The same can be seen throughout the expansion wave, at the cost of discontinuous gradients at either ends of said expansion wave being resolved less sharply. Some oscillations are still present in the skew-symmetric solution between $x = 0.45 - 0.65$, although these are also much smaller in amplitude than those of the non-averaged solution. Both discontinuities have been smeared over a small area as well, in a similar way to the results of Clawpack and the DG method. Between these discontinuities both methods still show oscillatory behavior, but as in the rest of the domain the oscillation amplitudes that remain are significantly smaller than those of the non-averaged solutions shown in figure 7.7. Lastly we note that it is now more clearly visible that the skew-symmetric method overpredicts the travel speed of the rightmost discontinuity, whereas the regular method matches the exact solution, Clawpack and the DG method much closer.

Accuracy of conservation behavior

Turning to the conservation errors shown in figure 7.9 large differences between the various models can be seen for some of the physical variables. Nearly identical behavior is shown for mass and total energy: Whereas the regular & skew-symmetric models and Clawpack conserve both up to $\mathcal{O}(10^{-15})$, the DG method has conservation errors up to $\mathcal{O}(10^{-2})$ on the coarsest meshes for all basis function degrees shown. Although these conservation errors decrease to $\mathcal{O}(10^{-14})$ for $p = 1$ the higher-order solutions seem to converge asymptotically to $\mathcal{O}(10^{-10})$, even when up to four times as many degrees of freedom are used as in the finest simulations of Clawpack and the regular & skew-symmetric models.

For momentum the DG method shows the same behavior: Conservation errors up to $\mathcal{O}(10^{-4})$ are produced on the coarsest grids and its higher-order solutions converge to conservation errors of $\mathcal{O}(10^{-9})$ on the finest grids, again despite using up to nearly four times as many degrees of freedom as Clawpack and the regular &

skew-symmetric methods. With $p = 1$ the DG method shows a decrease of its momentum conservation error from $\mathcal{O}(10^{-4})$ on the coarsest mesh to $\mathcal{O}(10^{-16})$ on the finest meshes. Clawpack shows the same trend for smaller amounts of degrees of freedom. Both methods clearly outperform the regular and skew-symmetric models over the entire mesh size range when it comes to the accuracy of their momentum conservation; only on the coarsest DG meshes with $p = 2$, $p = 3$ do the regular & skew-symmetric methods come close. Whereas the differences between them are relatively small for the coarsest meshes, the low convergence rate of the regular and skew-symmetric models result in differences of at least four orders of magnitude on the finest meshes.

The last physical variables to be compared are kinetic and internal energy. Due to the significant total energy conservation errors of the DG method these graphs are not identical to one another, unlike in the previous comparisons. Looking at kinetic energy the best-performing method on the coarse half of the mesh size range is the regular method, followed by the skew-symmetric method and Clawpack and the DG method following at some distance for $p = 1$. Whereas the regular and skew-symmetric models converge asymptotically to kinetic energy conservation errors of $\mathcal{O}(10^{-3})$ for all p as the mesh is refined, Clawpack and the DG method both display monotonic behavior with a constant order of convergence. Clawpack's conservation error is smaller than that of the DG method even with $p = 1$; as p is increased in the DG method its kinetic energy conservation error grows. Despite this the slopes of the conservation error graphs are the same for $p = 1, 2, 3$. For $p = 1$ the DG method this results in the DG method having a smaller kinetic energy conservation error than the skew-symmetric method on its coarsest meshed, while for $p > 1$ its errors are still larger. Note however that the DG method's results still display a constant monotonic decrease, whereas the regular & skew-symmetric methods have seemingly bottomed out.

The overall picture is similar for the internal energy conservation errors, except for the results of the DG method. As mentioned before this method does not always conserve total energy up to machine precision and this is reflected in the internal energy conservation errors: On the coarsest mesh the DG method's conservation errors are an order of magnitude larger than the next-highest errors. As the mesh is refined and the total energy errors decrease rapidly the internal energy errors approach the conservation errors of the kinetic energy: Since the DG method's total energy conservation error is smaller than $\mathcal{O}(10^{-6})$ for $n > 10^2$ it is negligible with respect to the kinetic energy and internal energy errors, which are both $\mathcal{O}(10^{-3})$.

Conclusions

Of the four methods compared here Clawpack seems to perform best for this case. While it is the most diffusive method around discontinuities it is effective with the smallest amount of degrees of freedom. It unconditionally conserves mass and total energy up to machine precision on all grids and displays monotonic convergence in its momentum, kinetic energy and internal energy conservation errors. While the DG method generally also shows diffusive (and not dispersive) behavior it has some unphysical behavior around $x = 0.5$ and has large conservation errors on its coarsest grids for especially mass and total energy. While these errors decrease in size as its mesh is refined, they are still significantly larger than those found with the other methods. The DG method also has the highest kinetic and internal energy conservation errors for most of its results, especially with $p > 1$. Only when it comes to momentum conservation does it consistently outperform the regular and skew-symmetric methods.

Lastly it can be concluded that the regular and skew-symmetric methods in their current implementations are outperformed by Clawpack and the DG method in most aspects for this case. Their unfiltered solutions are extremely oscillatory; using a simple moving average-filter damps out the worst of it, but in the region between the two discontinuities of the exact solution this is not sufficient. Additionally the skew-symmetric model wrongly predicts the location of the rightmost discontinuity. At this point it should be pointed out that both Clawpack and the DG method use some form of filtering or flux limiting as stabilization; without this additional stabilization none of their simulations ran successfully to $t = 0.21$. The fact that both the regular and skew-symmetric methods successfully converged despite the extreme oscillations present in their solutions and without any form of stabilization is testament to their stability.

In terms of conservation behavior both models performs similarly to Clawpack for mass and total energy on all meshes, as well as for kinetic and internal energy on the coarsest half of the meshes considered here. It is postulated here that the lack of convergence in terms of kinetic and internal energy conservation errors is

being driven by the formation and transport of oscillations on the finest meshes. However, the biggest flaw in terms of conservation behavior of these methods is found for momentum; these conservation errors are significantly larger than those of Clawpack and the DG method.

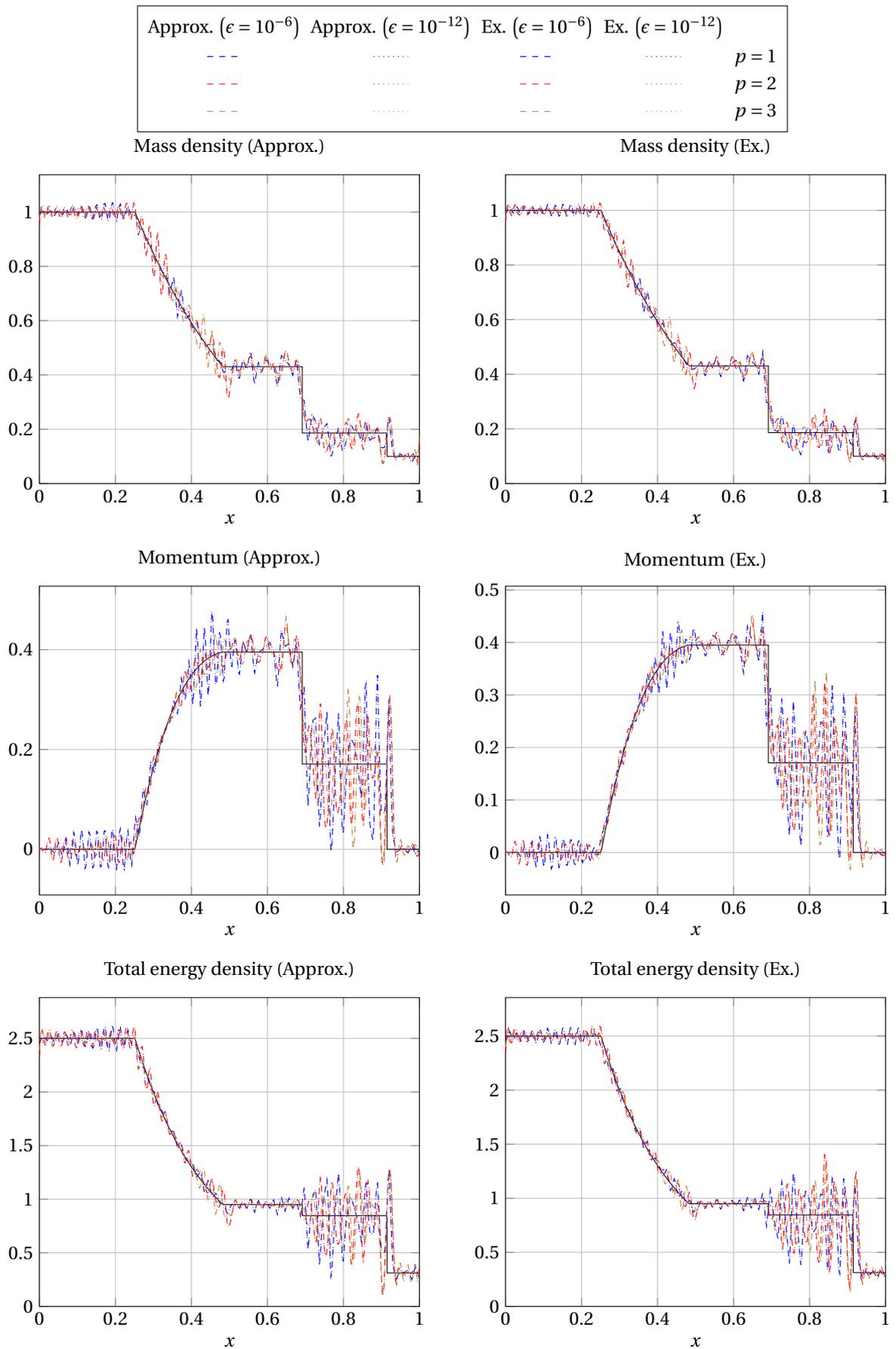


Figure 7.3: Solutions at $t = 0.21$ of primary physical variables of the skew-symmetric model with exact (Ex.) and approximate (Approx.) kinetic energy modeling for $\Delta t = 10^{-3}$, different solution orders p , $\epsilon = 10^{-6}$ and $\epsilon = 10^{-12}$

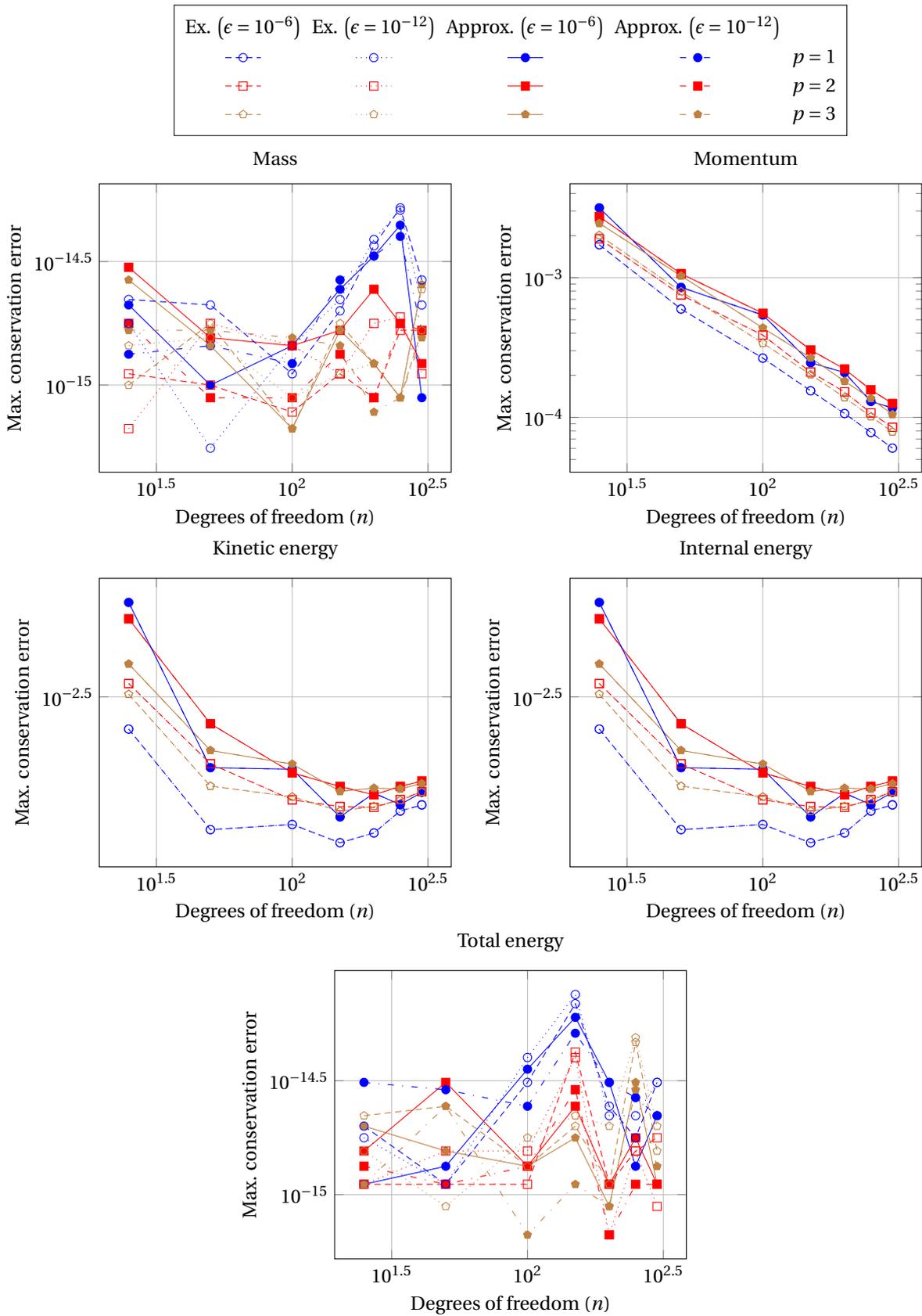


Figure 7.4: Maximum conservation errors for several physical variables of the skew-symmetric model with exact (Ex.) and approximate (Approx.) kinetic energy modeling for $\Delta t = 10^{-3}$, different solution orders p , $\epsilon = 10^{-6}$ and $\epsilon = 10^{-12}$

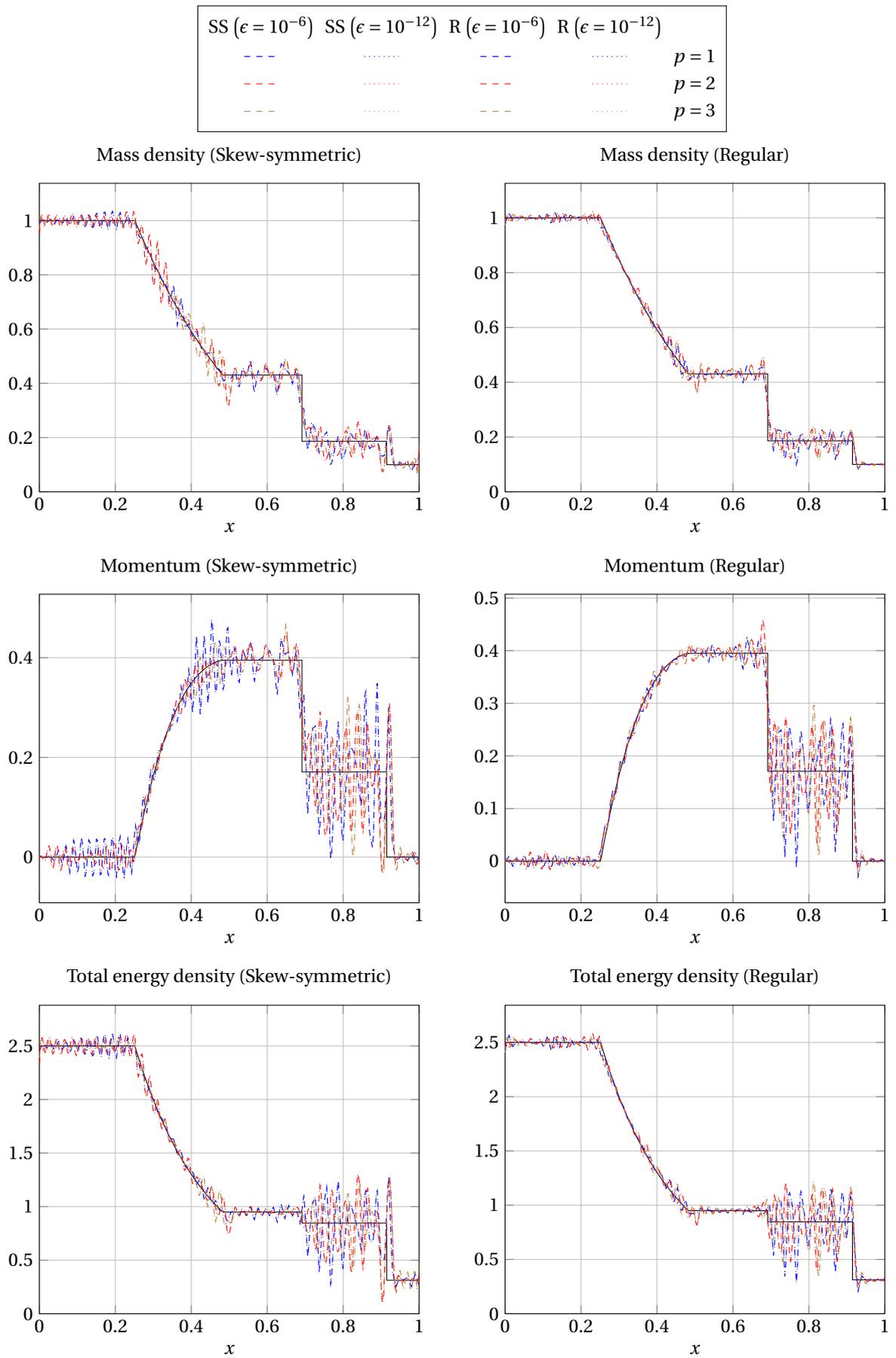


Figure 7.5: Solutions at $t = 0.21$ of primary physical variables of the regular (R) and skew-symmetric (SS) (with approximate kinetic energy advection) models for $\Delta t = 10^{-3}$, different solution orders p , $\epsilon = 10^{-6}$ and $\epsilon = 10^{-12}$

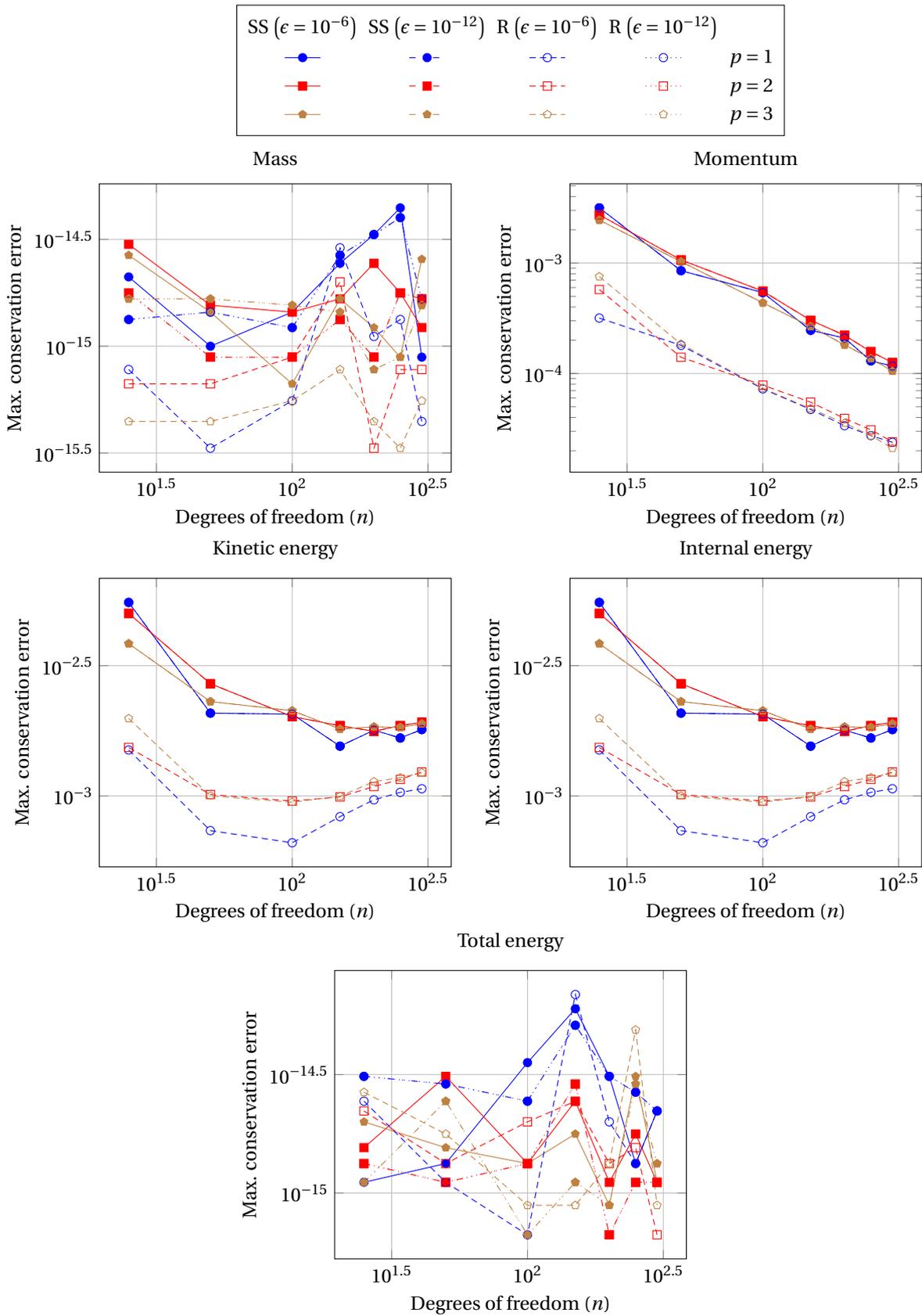


Figure 7.6: Maximum conservation errors for several physical variables of the regular (R) and skew-symmetric (SS) (with approximate kinetic energy advection) models for $\Delta t = 10^{-3}$, different solution orders p , $\epsilon = 10^{-6}$ and $\epsilon = 10^{-12}$

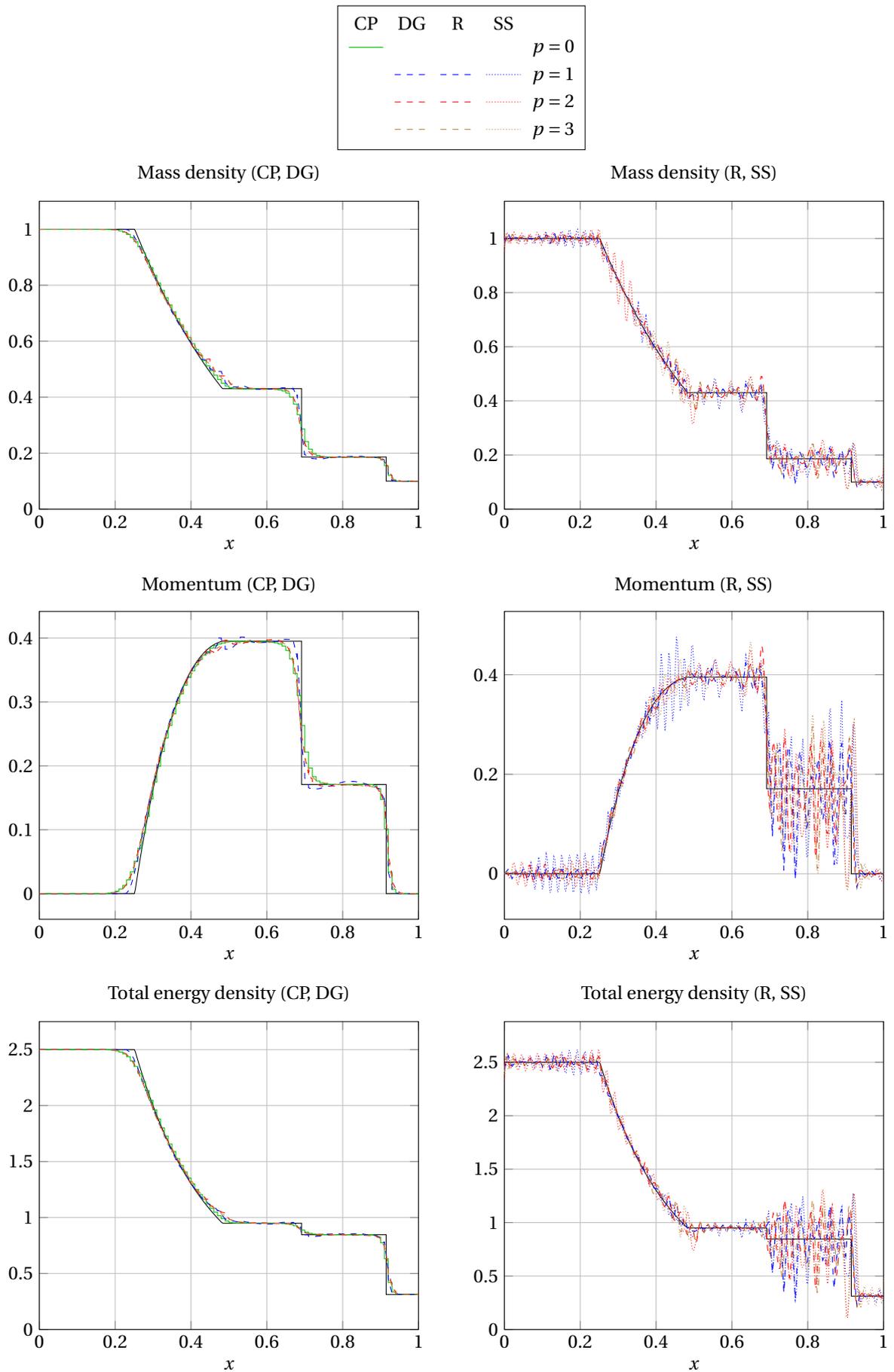


Figure 7.7: Solutions at $t = 0.21$ of primary physical variables of the regular (R) and skew-symmetric (SS) models with $\epsilon = 10^{-6}$, Clawpack and the DG method, all with $\Delta t = 10^{-3}$ and different solution orders p where possible

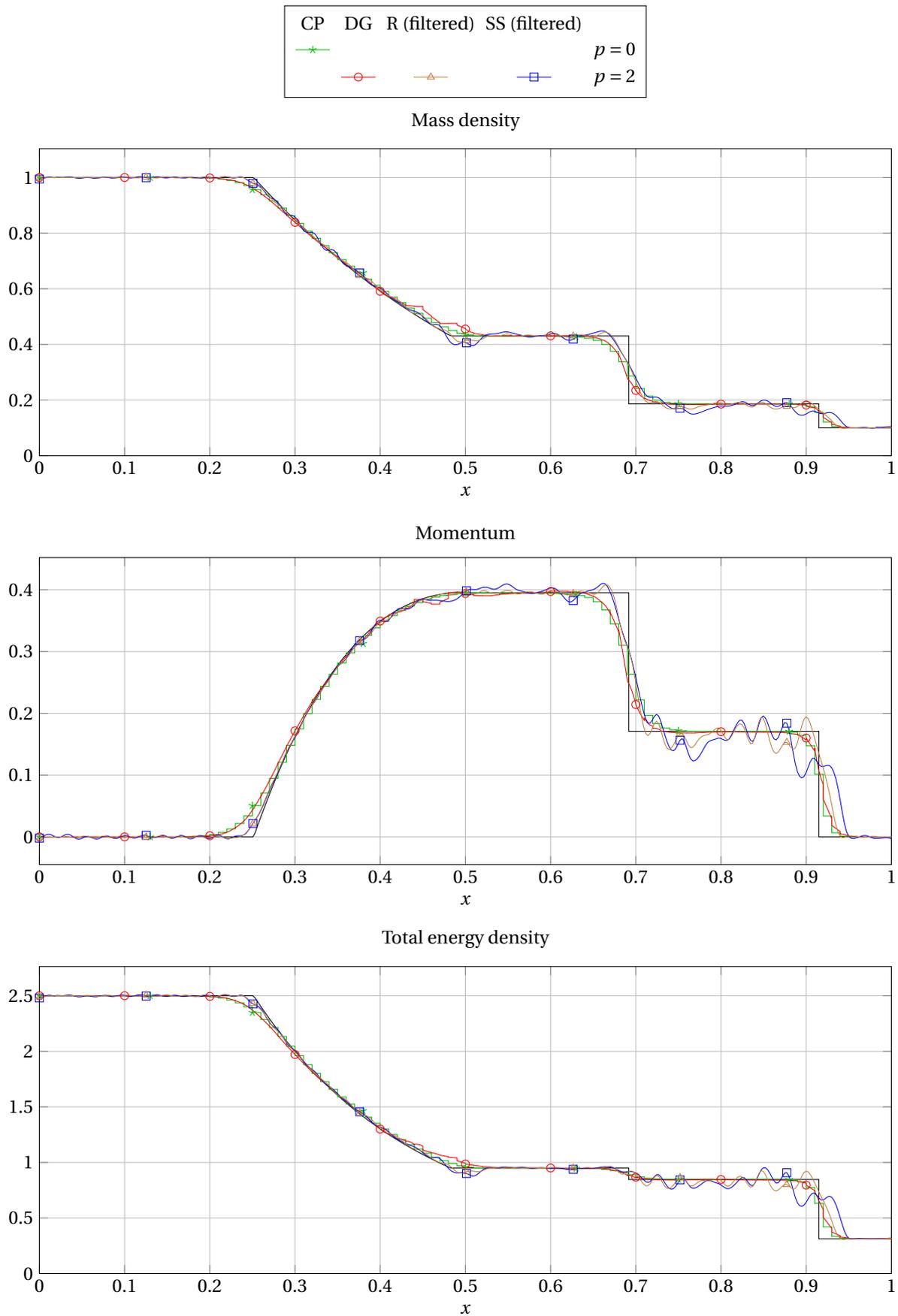


Figure 7.8: Solutions at $t = 0.21$ of primary physical variables of the regular (R) and skew-symmetric (SS) models with $\epsilon = 10^{-6}$, Clawpack and the DG method, all with $\Delta t = 10^{-3}$ and different solution orders p where possible

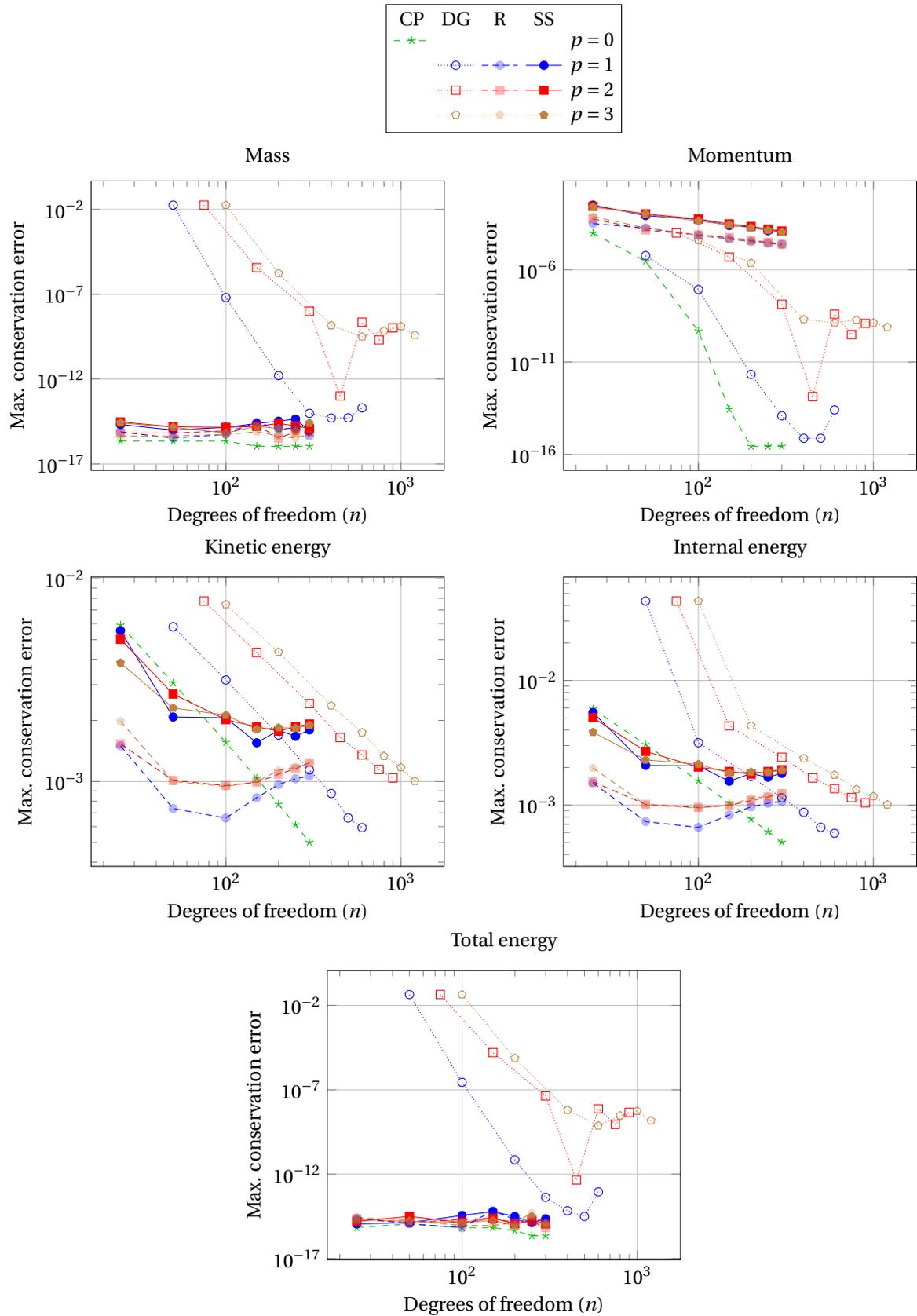
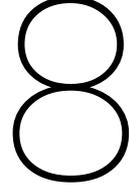


Figure 7.9: Maximum conservation errors for several physical variables of the regular (R) and skew-symmetric (SS) models with $\epsilon = 10^{-6}$, the DG method and Clawpack (CP), all with $\Delta t = 10^{-3}$ and different solution orders p where possible



Two-dimensional Euler equations

After the initial one-dimensional Euler model test with Sod's shock tube a two-dimensional test is to be carried out. Two variations of the same isentropic vortex test case are carried out on a doubly-periodic domain: First a static vortex is simulated, followed by a moving vortex. The structure of this chapter is largely similar to those of the previous test cases. First the continuous models for the regular and skew-symmetric (Roe variable) formulations are defined in section 8.1. These are then discretized in section 8.2. Following this section 8.3 introduces the two-dimensional isentropic vortex test case, after which sections 8.4 and 8.5 cover the results that were obtained for a the aforementioned static and moving vortices respectively.

8.1. Continuous models

The models introduced in chapter 7 are extended to two spatial dimensions. As is done throughout this thesis the regular model is covered first, followed by the skew-symmetric Roe variable model. Since their original definitions were given for an arbitrary number of dimensions this extension is straightforward; the main differences are the types of differential forms that are used (the volume-forms are now 2-forms, for example) and the number of momentum equations that have to be solved (two instead of one). The regular model in two spatial dimensions is:

$$\begin{aligned}
 \frac{\partial \rho^{(2)}}{\partial t} + \mathcal{L}_{\mathbf{u}} \rho^{(2)} &= 0 \\
 \frac{\partial \langle m^{(1,2)}, \partial_x \rangle}{\partial t} + \mathcal{L}_{\mathbf{u}} \langle m^{(1,2)}, \partial_x \rangle + di_{\partial_x} p^{(2)} &= 0 \\
 \frac{\partial \langle m^{(1,2)}, \partial_y \rangle}{\partial t} + \mathcal{L}_{\mathbf{u}} \langle m^{(1,2)}, \partial_y \rangle + di_{\partial_y} p^{(2)} &= 0 \\
 \frac{\partial E^{(2)}}{\partial t} + \mathcal{L}_{\mathbf{u}} E^{(2)} + \mathcal{L}_{\mathbf{u}} p^{(2)} &= 0
 \end{aligned} \tag{8.1}$$

Where x, y are spatial coordinates on $\Omega \subset \mathbb{R}^2$ and the notation of the other physical variables is consistent with previous chapters. Now momentum $m^{(1,2)} \in T^*(\Omega) \otimes \Lambda^{(2)}(\Omega)$, with:

$$\begin{aligned}
 \langle m^{(1,2)}, \partial_x \rangle &= \langle \partial_x, dx \rangle \otimes m_x dx \wedge dy + \langle \partial_x, dy \rangle \otimes m_y dx \wedge dy = m_x dx \wedge dy = m_x^{(2)} \\
 \langle m^{(1,2)}, \partial_y \rangle &= \langle \partial_y, dx \rangle \otimes m_x dx \wedge dy + \langle \partial_y, dy \rangle \otimes m_y dx \wedge dy = m_y dx \wedge dy = m_y^{(2)}
 \end{aligned}$$

Although in this case the principal directions of the momentum equations align exactly with the coordinate grid, any pointwise linearly independent pair of directions could have been defined on Ω to model momentum conservation; each momentum equation would then have consisted of (possibly spatially-varying) linear combinations of $m_x^{(2)}$ and $m_y^{(2)}$. Note that we now have to distinguish between two kinds of 1-forms, one associated with unit covector dx and one with dy .

Similarly the Roe variable model in two spatial dimensions is:

$$\begin{aligned}
& \frac{\partial \sqrt{\rho}^{(2)}}{\partial t} + \frac{1}{2} \mathcal{L}_{\mathbf{u}}(\sqrt{\rho}^{(2)}) + \frac{1}{2} \star \mathcal{L}_{\mathbf{u}}(\star \sqrt{\rho}^{(2)}) = 0 \\
& \frac{\partial \langle \phi^{(1,2)}, \partial_x \rangle}{\partial t} + \frac{1}{2} \mathcal{L}_{\mathbf{u}}(\langle \phi^{(1,2)}, \partial_x \rangle) + \frac{1}{2} \star \mathcal{L}_{\mathbf{u}}(\star \langle \phi^{(1,2)}, \partial_x \rangle) + \frac{1}{\star \sqrt{\rho}^{(2)}} di_{\partial_x} p^{(2)} = 0 \\
& \frac{\partial \langle \phi^{(1,2)}, \partial_y \rangle}{\partial t} + \frac{1}{2} \mathcal{L}_{\mathbf{u}}(\langle \phi^{(1,2)}, \partial_y \rangle) + \frac{1}{2} \star \mathcal{L}_{\mathbf{u}}(\star \langle \phi^{(1,2)}, \partial_y \rangle) + \frac{1}{\star \sqrt{\rho}^{(2)}} di_{\partial_y} p^{(2)} = 0 \\
& \frac{\partial E^{(2)}}{\partial t} + \mathcal{L}_{\mathbf{u}} E^{(2)} + \mathcal{L}_{\mathbf{u}} p^{(2)} = 0
\end{aligned} \tag{8.2}$$

Where:

$$\begin{aligned}
\langle \phi^{(1,2)}, \partial_x \rangle &= \langle \partial_x, dx \rangle \otimes \phi_x dx \wedge dy + \langle \partial_x, dy \rangle \otimes \phi_y dx \wedge dy = \phi_x dx \wedge dy = m_x^{(2)} \\
\langle \phi^{(1,2)}, \partial_y \rangle &= \langle \partial_y, dx \rangle \otimes \phi_x dx \wedge dy + \langle \partial_y, dy \rangle \otimes \phi_y dx \wedge dy = \phi_y dx \wedge dy = \phi_y^{(2)}
\end{aligned}$$

Like before both models are extended with the ideal gas law, in order to express the pressure $p^{(2)}$ in terms of the other variables:

$$p^{(2)} = (\gamma - 1) (E^{(2)} - E_{kin}^{(2)})$$

Once again we take the L^2 inner products of the regular and Roe variable models with test function $v^{(2)} \in \Lambda^{(2)}(\Omega)$. For the regular model that was defined in (8.1) this leads to the following weak form:

$$\begin{aligned}
& \left(v^{(2)}, \frac{\partial \rho^{(2)}}{\partial t} \right)_{\Omega} + (v^{(2)}, \mathcal{L}_{\mathbf{u}} \rho^{(2)})_{\Omega} = 0 \\
& \left(v^{(2)}, \frac{\partial m_x^{(2)}}{\partial t} \right)_{\Omega} + (v^{(2)}, \mathcal{L}_{\mathbf{u}} m_x^{(2)})_{\Omega} + (v^{(2)}, di_{\partial_x} p^{(2)})_{\Omega} = 0 \\
& \left(v^{(2)}, \frac{\partial m_y^{(2)}}{\partial t} \right)_{\Omega} + (v^{(2)}, \mathcal{L}_{\mathbf{u}} m_y^{(2)})_{\Omega} + (v^{(2)}, di_{\partial_y} p^{(2)})_{\Omega} = 0 \\
& \left(v^{(2)}, \frac{\partial E^{(2)}}{\partial t} \right)_{\Omega} + (v^{(2)}, \mathcal{L}_{\mathbf{u}} E^{(2)})_{\Omega} + (v^{(2)}, \mathcal{L}_{\mathbf{u}} p^{(2)})_{\Omega} = 0
\end{aligned}$$

The final step in obtaining the weak form that will be discretized is to apply the equation of state:

$$\begin{aligned}
& \left(v^{(2)}, \frac{\partial \rho^{(2)}}{\partial t} \right)_{\Omega} + (v^{(2)}, \mathcal{L}_{\mathbf{u}} \rho^{(2)})_{\Omega} = 0 \\
& \left(v^{(2)}, \frac{\partial m_x^{(2)}}{\partial t} \right)_{\Omega} + (v^{(2)}, \mathcal{L}_{\mathbf{u}} m_x^{(2)})_{\Omega} + (\gamma - 1) (v^{(2)}, di_{\partial_x} E^{(2)})_{\Omega} = (\gamma - 1) (v^{(2)}, di_{\partial_x} E_{kin}^{(2)})_{\Omega} \\
& \left(v^{(2)}, \frac{\partial m_y^{(2)}}{\partial t} \right)_{\Omega} + (v^{(2)}, \mathcal{L}_{\mathbf{u}} m_y^{(2)})_{\Omega} + (\gamma - 1) (v^{(2)}, di_{\partial_y} E^{(2)})_{\Omega} = (\gamma - 1) (v^{(2)}, di_{\partial_y} E_{kin}^{(2)})_{\Omega} \\
& \left(v^{(2)}, \frac{\partial E^{(2)}}{\partial t} \right)_{\Omega} + \gamma (v^{(2)}, \mathcal{L}_{\mathbf{u}} E^{(2)})_{\Omega} = (\gamma - 1) (v^{(2)}, \mathcal{L}_{\mathbf{u}} E_{kin}^{(2)})_{\Omega}
\end{aligned} \tag{8.3}$$

In the same way the weak form of the Roe variable model defined in (8.2) is:

$$\begin{aligned}
& \left(v^{(2)}, \frac{\partial \sqrt{\rho}^{(2)}}{\partial t} \right)_{\Omega} + \frac{1}{2} (v^{(2)}, \mathcal{L}_{\mathbf{u}}(\sqrt{\rho}^{(2)}))_{\Omega} + \frac{1}{2} (v^{(2)}, \star \mathcal{L}_{\mathbf{u}}(\star \sqrt{\rho}^{(2)}))_{\Omega} = 0 \\
& \left(v^{(2)}, \frac{\partial \phi_x^{(2)}}{\partial t} \right)_{\Omega} + \frac{1}{2} (v^{(2)}, \mathcal{L}_{\mathbf{u}}(\phi_x^{(2)}))_{\Omega} + \frac{1}{2} (v^{(2)}, \star \mathcal{L}_{\mathbf{u}}(\star \phi_x^{(2)}))_{\Omega} + \left(v^{(2)}, \frac{1}{\star \sqrt{\rho}^{(2)}} di_{\partial_x} p^{(2)} \right)_{\Omega} = 0 \\
& \left(v^{(2)}, \frac{\partial \phi_y^{(2)}}{\partial t} \right)_{\Omega} + \frac{1}{2} (v^{(2)}, \mathcal{L}_{\mathbf{u}}(\phi_y^{(2)}))_{\Omega} + \frac{1}{2} (v^{(2)}, \star \mathcal{L}_{\mathbf{u}}(\star \phi_y^{(2)}))_{\Omega} + \left(v^{(2)}, \frac{1}{\star \sqrt{\rho}^{(2)}} di_{\partial_y} p^{(2)} \right)_{\Omega} = 0 \\
& \left(v^{(2)}, \frac{\partial E^{(2)}}{\partial t} \right)_{\Omega} + (v^{(2)}, \mathcal{L}_{\mathbf{u}} E^{(2)})_{\Omega} + (v^{(2)}, \mathcal{L}_{\mathbf{u}} p^{(2)})_{\Omega} = 0
\end{aligned} \tag{8.4}$$

Since $\sqrt{\rho}^{(2)}$, $\phi_x^{(2)}$ and $\phi_y^{(2)}$ are all volume-forms the advection terms containing the Hodge-star operator (for example: $(v^{(2)}, \star \mathcal{L}_{\mathbf{u}}(\star \sqrt{\rho}^{(2)}))_{\Omega}$) are rewritten in the exact same way as was shown in section 5.1 for a one-dimensional model; recall that the same approach was used in all one-dimensional tests treated so far, including the Euler model covered in chapter 7. Applying this to (8.4) leads to the following model:

$$\begin{aligned}
& \left(v^{(2)}, \frac{\partial \sqrt{\rho}^{(2)}}{\partial t} \right)_{\Omega} + \frac{1}{2} \left(v^{(2)}, \mathcal{L}_{\mathbf{u}}(\sqrt{\rho}^{(2)}) \right)_{\Omega} - \frac{1}{2} \left(\mathcal{L}_{\mathbf{u}}(v^{(2)}), \sqrt{\rho}^{(2)} \right)_{\Omega} + \frac{1}{2} \int_{\partial \Omega} i_{\mathbf{u}}(v^{(2)} \wedge \star \sqrt{\rho}^{(2)}) = 0 \\
& \left(v^{(2)}, \frac{\partial \phi_x^{(2)}}{\partial t} \right)_{\Omega} + \frac{1}{2} \left(v^{(2)}, \mathcal{L}_{\mathbf{u}}(\phi_x^{(2)}) \right)_{\Omega} - \frac{1}{2} \left(\mathcal{L}_{\mathbf{u}}(v^{(2)}), \phi_x^{(2)} \right)_{\Omega} + \frac{1}{2} \int_{\partial \Omega} i_{\mathbf{u}}(v^{(2)} \wedge \star \phi_x^{(2)}) \\
& \qquad \qquad \qquad + \left(v^{(2)}, \frac{1}{\star \sqrt{\rho}^{(2)}} di_{\partial_x} p^{(2)} \right)_{\Omega} = 0 \\
& \left(v^{(2)}, \frac{\partial \phi_y^{(2)}}{\partial t} \right)_{\Omega} + \frac{1}{2} \left(v^{(2)}, \mathcal{L}_{\mathbf{u}}(\phi_y^{(2)}) \right)_{\Omega} - \frac{1}{2} \left(\mathcal{L}_{\mathbf{u}}(v^{(2)}), \phi_y^{(2)} \right)_{\Omega} + \frac{1}{2} \int_{\partial \Omega} i_{\mathbf{u}}(v^{(2)} \wedge \star \phi_y^{(2)}) \\
& \qquad \qquad \qquad + \left(v^{(2)}, \frac{1}{\star \sqrt{\rho}^{(2)}} di_{\partial_y} p^{(2)} \right)_{\Omega} = 0 \\
& \left(v^{(2)}, \frac{\partial E^{(2)}}{\partial t} \right)_{\Omega} + (v^{(2)}, \mathcal{L}_{\mathbf{u}} E^{(2)})_{\Omega} + (v^{(2)}, \mathcal{L}_{\mathbf{u}} p^{(2)})_{\Omega} = 0
\end{aligned} \tag{8.5}$$

Since domain Ω is periodic, $\partial \Omega = \emptyset$; thus the boundary terms vanish. Applying the ideal gas law to (8.5) results in the weak form of the skew-symmetric (Roe variable) model that is to be discretized:

$$\begin{aligned}
& \left(v^{(2)}, \frac{\partial \sqrt{\rho}^{(2)}}{\partial t} \right)_{\Omega} + \frac{1}{2} \left(v^{(2)}, \mathcal{L}_{\mathbf{u}}(\sqrt{\rho}^{(2)}) \right)_{\Omega} - \frac{1}{2} \left(\mathcal{L}_{\mathbf{u}}(v^{(2)}), \sqrt{\rho}^{(2)} \right)_{\Omega} = 0 \\
& \left(v^{(2)}, \frac{\partial \phi_x^{(2)}}{\partial t} \right)_{\Omega} + \frac{1}{2} \left(v^{(2)}, \mathcal{L}_{\mathbf{u}}(\phi_x^{(2)}) \right)_{\Omega} - \frac{1}{2} \left(\mathcal{L}_{\mathbf{u}}(v^{(2)}), \phi_x^{(2)} \right)_{\Omega} \\
& \qquad \qquad \qquad + (\gamma - 1) \left(v^{(2)}, \frac{1}{\star \sqrt{\rho}^{(2)}} di_{\partial_x} E^{(2)} \right)_{\Omega} = (\gamma - 1) \left(v^{(2)}, \frac{1}{\star \sqrt{\rho}^{(2)}} di_{\partial_x} E_{kin}^{(2)} \right)_{\Omega} \\
& \left(v^{(2)}, \frac{\partial \phi_y^{(2)}}{\partial t} \right)_{\Omega} + \frac{1}{2} \left(v^{(2)}, \mathcal{L}_{\mathbf{u}}(\phi_y^{(2)}) \right)_{\Omega} - \frac{1}{2} \left(\mathcal{L}_{\mathbf{u}}(v^{(2)}), \phi_y^{(2)} \right)_{\Omega} \\
& \qquad \qquad \qquad + (\gamma - 1) \left(v^{(2)}, \frac{1}{\star \sqrt{\rho}^{(2)}} di_{\partial_y} E^{(2)} \right)_{\Omega} = (\gamma - 1) \left(v^{(2)}, \frac{1}{\star \sqrt{\rho}^{(2)}} di_{\partial_y} E_{kin}^{(2)} \right)_{\Omega} \\
& \left(v^{(2)}, \frac{\partial E^{(2)}}{\partial t} \right)_{\Omega} + \gamma (v^{(2)}, \mathcal{L}_{\mathbf{u}} E^{(2)})_{\Omega} = (\gamma - 1) \left(v^{(2)}, \mathcal{L}_{\mathbf{u}} E_{kin}^{(2)} \right)_{\Omega}
\end{aligned} \tag{8.6}$$

8.2. Discrete models

We convert the weak forms defined in section 8.1 into linear systems by expanding all quantities in (8.3) and (8.6) in two-dimensional basis functions. As mentioned in chapter 2 the higher-dimensional discrete bases are formed by taking Cartesian products of the discrete one-dimensional 0- and 1-form basis functions. In the rest of this section we use lexicographic ordering for the basis functions and their associated expansion coefficients; for two-dimensional basis functions this ordering maps (x, y) -index $(i_x, i_y) \rightarrow i = i_x + (i_y - 1)N_x$, with N_x the number of basis functions in x -direction. While it is possible to have unequal numbers of basis functions in the x - and y -directions ($N_x \neq N_y$) this research will feature only meshes for which $N_x = N_y$.

8.2.1. Regular model

The quantities given in (8.3) are expanded in the 2-form basis $\Lambda_p^{(2)}(\Omega_h)$ as follows:

$$\begin{aligned} v_h^{(2)} &= \sum_{i=1}^{N^2} v_i \psi_i^{(2)} = \mathbf{v}^T \boldsymbol{\psi}^{(2)} & \rho_h^{(2)} &= \sum_{i=1}^{N^2} \rho_i \psi_i^{(2)} = \boldsymbol{\rho}^T \boldsymbol{\psi}^{(2)} \\ (m_x^{(2)})_h &= \sum_{i=1}^{N^2} (m_x)_i \psi_i^{(2)} = \mathbf{m}_x^T \boldsymbol{\psi}^{(2)} & (m_y^{(2)})_h &= \sum_{i=1}^{N^2} (m_y)_i \psi_i^{(2)} = \mathbf{m}_y^T \boldsymbol{\psi}^{(2)} \\ E_h^{(2)} &= \sum_{i=1}^{N^2} E_i \psi_i^{(2)} = \mathbf{E}^T \boldsymbol{\psi}^{(2)} \end{aligned}$$

Note that N^2 two-dimensional basis functions are used, together with the aforementioned lexicographic index ordering. Kinetic energy $(E_{kin}^{(2)})_h$ is modeled as:

$$(E_{kin}^{(2)})_h = \frac{1}{2} \frac{\langle (m_x^{(2)})_h, \star (m_x^{(2)})_h \rangle + \langle (m_y^{(2)})_h, \star (m_y^{(2)})_h \rangle}{\star \rho_h^{(2)}}$$

As was done in section 7.2.4, $(E_{kin}^{(2)})_h \in \Gamma_{h,2p,p}^{(2)}(\Omega_h) \subset \Lambda^{(2)}(\Omega)$ denotes the space of 2-form NURBS on Ω_h . The squares of $\phi_x^{(2)}$ and $\phi_y^{(2)}$ are again resolved with the exact B-spline multiplication algorithm that was covered in section 2.2.4. Applying the basis function expansions to (8.3) then gives:

$$\begin{aligned} \mathbf{v}^T \mathbb{M}^{(2)} \frac{\partial \boldsymbol{\rho}}{\partial t} + \mathbf{v}^T \mathbb{M}^{(2)} \begin{bmatrix} \mathbb{E}^{(2,1x)} & \mathbb{E}^{(2,1y)} \end{bmatrix} \begin{bmatrix} (\mathbb{M}^{(1x)})^{-1} \mathbb{C}_{\mathbf{u}_h}^{(1x,2)} \\ -(\mathbb{M}^{(1y)})^{-1} \mathbb{C}_{\mathbf{u}_h}^{(1y,2)} \end{bmatrix} \boldsymbol{\rho} &= 0 \\ \mathbf{v}^T \mathbb{M}^{(2)} \frac{\partial \mathbf{m}_x}{\partial t} + \mathbf{v}^T \mathbb{M}^{(2)} \begin{bmatrix} \mathbb{E}^{(2,1x)} & \mathbb{E}^{(2,1y)} \end{bmatrix} \begin{bmatrix} (\mathbb{M}^{(1x)})^{-1} \mathbb{C}_{\mathbf{u}_h}^{(1x,2)} \\ -(\mathbb{M}^{(1y)})^{-1} \mathbb{C}_{\mathbf{u}_h}^{(1y,2)} \end{bmatrix} \mathbf{m}_x \\ -(\gamma - 1) \mathbf{v}^T \mathbb{M}^{(2)} \mathbb{E}^{(2,1y)} (\mathbb{M}^{(1y)})^{-1} \mathbb{C}_{\partial_x}^{(1y,2)} \mathbf{E} &= -(\gamma - 1) \mathbf{v}^T \mathbb{M}^{(2)} \mathbb{E}^{(2,1y)} (\mathbb{M}^{(1y)})^{-1} \mathbb{C}_{\partial_x}^{(1y,\bar{2})} \mathbf{E}_{kin} \\ \mathbf{v}^T \mathbb{M}^{(2)} \frac{\partial \mathbf{m}_y}{\partial t} + \mathbf{v}^T \mathbb{M}^{(2)} \begin{bmatrix} \mathbb{E}^{(2,1x)} & \mathbb{E}^{(2,1y)} \end{bmatrix} \begin{bmatrix} (\mathbb{M}^{(1x)})^{-1} \mathbb{C}_{\mathbf{u}_h}^{(1x,2)} \\ -(\mathbb{M}^{(1y)})^{-1} \mathbb{C}_{\mathbf{u}_h}^{(1y,2)} \end{bmatrix} \mathbf{m}_y \\ +(\gamma - 1) \mathbf{v}^T \mathbb{M}^{(2)} \mathbb{E}^{(2,1x)} (\mathbb{M}^{(1x)})^{-1} \mathbb{C}_{\partial_y}^{(1x,2)} \mathbf{E} &= (\gamma - 1) \mathbf{v}^T \mathbb{M}^{(2)} \mathbb{E}^{(2,1x)} (\mathbb{M}^{(1x)})^{-1} \mathbb{C}_{\partial_y}^{(1x,\bar{2})} \mathbf{E}_{kin} \\ \mathbf{v}^T \mathbb{M}^{(2)} \frac{\partial \mathbf{E}}{\partial t} + \gamma \mathbf{v}^T \mathbb{M}^{(2)} \begin{bmatrix} \mathbb{E}^{(2,1x)} & \mathbb{E}^{(2,1y)} \end{bmatrix} \begin{bmatrix} (\mathbb{M}^{(1x)})^{-1} \mathbb{C}_{\mathbf{u}_h}^{(1x,2)} \\ -(\mathbb{M}^{(1y)})^{-1} \mathbb{C}_{\mathbf{u}_h}^{(1y,2)} \end{bmatrix} \mathbf{E} \\ &= (\gamma - 1) \mathbf{v}^T \mathbb{M}^{(2)} \begin{bmatrix} \mathbb{E}^{(2,1x)} & \mathbb{E}^{(2,1y)} \end{bmatrix} \begin{bmatrix} (\mathbb{M}^{(1x)})^{-1} \mathbb{C}_{\mathbf{u}_h}^{(1x,\bar{2})} \\ -(\mathbb{M}^{(1y)})^{-1} \mathbb{C}_{\mathbf{u}_h}^{(1y,\bar{2})} \end{bmatrix} \mathbf{E}_{kin} \end{aligned} \quad (8.7)$$

Note that superscript $\bar{2}$ denotes the discrete 2-form NURBS space $\Gamma_{h,2p,p}^{(2)}(\Omega_h)$. Test vector \mathbf{v}^T can be factored out without loss of generality. Before applying the time stepping method we introduce some \mathbf{v} notation to make the resulting linear systems more compact:

$$\begin{aligned} \mathbb{L}_{\mathbf{u}_h} &= \begin{bmatrix} \mathbb{E}^{(2,1x)} & \mathbb{E}^{(2,1y)} \end{bmatrix} \begin{bmatrix} (\mathbb{M}^{(1x)})^{-1} \mathbb{C}_{\mathbf{u}_h}^{(1x,2)} \\ -(\mathbb{M}^{(1y)})^{-1} \mathbb{C}_{\mathbf{u}_h}^{(1y,2)} \end{bmatrix} \\ \mathbb{L}_{\partial_x} &= \mathbb{E}^{(2,1y)} (\mathbb{M}^{(1y)})^{-1} \mathbb{C}_{\partial_x}^{(1y,2)} \\ \mathbb{L}_{\partial_y} &= \mathbb{E}^{(2,1x)} (\mathbb{M}^{(1x)})^{-1} \mathbb{C}_{\partial_y}^{(1x,2)} \end{aligned}$$

3.6.3 Picard linearization is used in this research, with $\mathbf{u}_h^{k_1}$ and $\mathbf{u}_h^{k_2}$ as the linearized terms. The variables at time levels k_1 and k_2 can be used to compute the solution at the end of the current time step, leading to estimates for the solution vectors in each Picard iteration.

8.2.2. Skew-symmetric Roe variable model

The same basis function expansions are used for the skew-symmetric model as for the regular model:

$$\begin{aligned} \mathbf{v}_h^{(2)} &= \sum_{i=1}^{N^2} v_i \psi_i^{(2)} = \mathbf{v}^T \boldsymbol{\psi}^{(2)} & \sqrt{\rho}_h^{(2)} &= \sum_{i=1}^{N^2} \sqrt{\rho}_i \psi_i^{(2)} = \sqrt{\boldsymbol{\rho}}^T \boldsymbol{\psi}^{(2)} \\ (\phi_x^{(2)})_h &= \sum_{i=1}^{N^2} (\phi_x)_i \psi_i^{(2)} = \boldsymbol{\phi}_x^T \boldsymbol{\psi}^{(2)} & (\phi_y^{(2)})_h &= \sum_{i=1}^{N^2} (\phi_y)_i \psi_i^{(2)} = \boldsymbol{\phi}_y^T \boldsymbol{\psi}^{(2)} \\ E_h^{(2)} &= \sum_{i=1}^{N^2} E_i \psi_i^{(2)} = \mathbf{E}^T \boldsymbol{\psi}^{(2)} \end{aligned}$$

Note that in this case the Roe variables have been expanded in the same basis as the physical variables in the regular model. Kinetic energy $(E_{kin}^{(2)})_h$ is now modeled as:

$$(E_{kin}^{(2)})_h = \frac{1}{2} \left(\langle (\phi_x^{(2)})_h, \star (\phi_x^{(2)})_h \rangle + \langle (\phi_y^{(2)})_h, \star (\phi_y^{(2)})_h \rangle \right)$$

In this case $(E_{kin}^{(2)})_h \in \Lambda_{h,2p}^{(2)}(\Omega_h) \subset \Lambda^{(2)}(\Omega)$. These basis function expansions then lead to the following semi-discrete weak form with skew-symmetric advection operators:

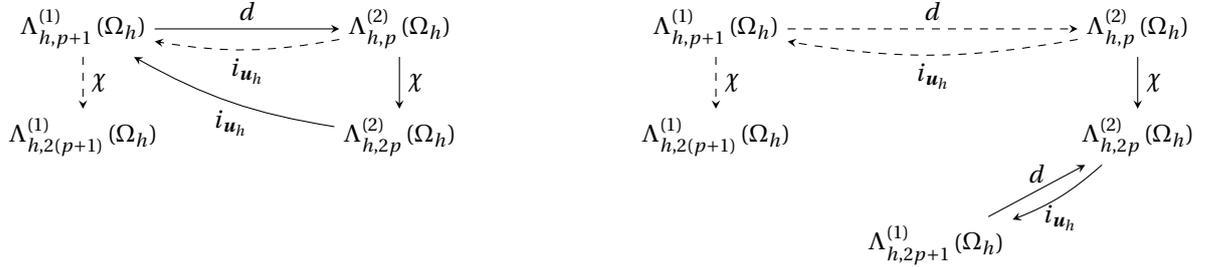
$$\begin{aligned} & \mathbf{v}^T \mathbb{M}^{(2)} \frac{\partial \sqrt{\boldsymbol{\rho}}}{\partial t} + \frac{1}{2} \mathbf{v}^T \mathbb{M}^{(2)} \begin{bmatrix} \mathbb{E}^{2,1x} & \mathbb{E}^{2,1y} \end{bmatrix} \begin{bmatrix} (\mathbb{M}^{(1,x)})^{-1} \mathbb{C}_{\mathbf{u}_h}^{(1,x,2)} \\ -(\mathbb{M}^{(1,y)})^{-1} \mathbb{C}_{\mathbf{u}_h}^{(1,y,2)} \end{bmatrix} \sqrt{\boldsymbol{\rho}} \\ & \quad - \frac{1}{2} \mathbf{v}^T \left(\mathbb{M}^{(2)} \begin{bmatrix} \mathbb{E}^{2,1x} & \mathbb{E}^{2,1y} \end{bmatrix} \begin{bmatrix} (\mathbb{M}^{(1,x)})^{-1} \mathbb{C}_{\mathbf{u}_h}^{(1,x,2)} \\ -(\mathbb{M}^{(1,y)})^{-1} \mathbb{C}_{\mathbf{u}_h}^{(1,y,2)} \end{bmatrix} \right)^T \sqrt{\boldsymbol{\rho}} = 0 \\ & \mathbf{v}^T \mathbb{M}^{(2)} \frac{\partial \boldsymbol{\phi}_x}{\partial t} + \frac{1}{2} \mathbf{v}^T \mathbb{M}^{(2)} \begin{bmatrix} \mathbb{E}^{2,1x} & \mathbb{E}^{2,1y} \end{bmatrix} \begin{bmatrix} (\mathbb{M}^{(1,x)})^{-1} \mathbb{C}_{\mathbf{u}_h}^{(1,x,2)} \\ -(\mathbb{M}^{(1,y)})^{-1} \mathbb{C}_{\mathbf{u}_h}^{(1,y,2)} \end{bmatrix} \boldsymbol{\phi}_x \\ & \quad - \frac{1}{2} \mathbf{v}^T \left(\mathbb{M}^{(2)} \begin{bmatrix} \mathbb{E}^{2,1x} & \mathbb{E}^{2,1y} \end{bmatrix} \begin{bmatrix} (\mathbb{M}^{(1,x)})^{-1} \mathbb{C}_{\mathbf{u}_h}^{(1,x,2)} \\ -(\mathbb{M}^{(1,y)})^{-1} \mathbb{C}_{\mathbf{u}_h}^{(1,y,2)} \end{bmatrix} \right)^T \boldsymbol{\phi}_x - (\gamma - 1) \mathbf{v}^T \mathbb{D}_{\sqrt{\boldsymbol{\rho}}}^{(2)} \mathbb{E}^{(2,1y)} (\mathbb{M}^{(1,y)})^{-1} \mathbb{C}_{\boldsymbol{\phi}_x}^{(1,y,2)} \mathbf{E} \\ & \quad = -(\gamma - 1) \mathbf{v}^T \mathbb{K}_{\partial_x, \sqrt{\boldsymbol{\rho}}} \mathbf{E}_{kin} \\ & \mathbf{v}^T \mathbb{M}^{(2)} \frac{\partial \boldsymbol{\phi}_y}{\partial t} + \frac{1}{2} \mathbf{v}^T \mathbb{M}^{(2)} \begin{bmatrix} \mathbb{E}^{2,1x} & \mathbb{E}^{2,1y} \end{bmatrix} \begin{bmatrix} (\mathbb{M}^{(1,x)})^{-1} \mathbb{C}_{\mathbf{u}_h}^{(1,x,2)} \\ -(\mathbb{M}^{(1,y)})^{-1} \mathbb{C}_{\mathbf{u}_h}^{(1,y,2)} \end{bmatrix} \boldsymbol{\phi}_y \\ & \quad - \frac{1}{2} \mathbf{v}^T \left(\mathbb{M}^{(2)} \begin{bmatrix} \mathbb{E}^{2,1x} & \mathbb{E}^{2,1y} \end{bmatrix} \begin{bmatrix} (\mathbb{M}^{(1,x)})^{-1} \mathbb{C}_{\mathbf{u}_h}^{(1,x,2)} \\ -(\mathbb{M}^{(1,y)})^{-1} \mathbb{C}_{\mathbf{u}_h}^{(1,y,2)} \end{bmatrix} \right)^T \boldsymbol{\phi}_y + (\gamma - 1) \mathbf{v}^T \mathbb{D}_{\sqrt{\boldsymbol{\rho}}}^{(2)} \mathbb{E}^{(2,1x)} (\mathbb{M}^{(1,x)})^{-1} \mathbb{C}_{\boldsymbol{\phi}_y}^{(1,x,2)} \mathbf{E} \\ & \quad = (\gamma - 1) \mathbf{v}^T \mathbb{K}_{\partial_y, \sqrt{\boldsymbol{\rho}}} \mathbf{E}_{kin} \\ & \mathbf{v}^T \mathbb{M}^{(2)} \frac{\partial \mathbf{E}}{\partial t} + \gamma \mathbf{v}^T \mathbb{M}^{(2)} \begin{bmatrix} \mathbb{E}^{2,1x} & \mathbb{E}^{2,1y} \end{bmatrix} \begin{bmatrix} (\mathbb{M}^{(1,x)})^{-1} \mathbb{C}_{\mathbf{u}_h}^{(1,x,2)} \\ -(\mathbb{M}^{(1,y)})^{-1} \mathbb{C}_{\mathbf{u}_h}^{(1,y,2)} \end{bmatrix} \mathbf{E} = (\gamma - 1) \mathbf{v}^T \mathbb{K}_{\mathbf{u}_h} \mathbf{E}_{kin} \end{aligned} \tag{8.11}$$

Where the superscript $\tilde{2}$ denotes the space of product 2-forms $\Lambda_{h,2p}^{(2)}(\Omega_h)$ that contains $(E_{kin}^{(2)})_h$. Like before we factor out \mathbf{v}^T without loss of generality. The operators acting on the kinetic energy terms are denoted with $\mathbb{K}_{\partial_x, \sqrt{\boldsymbol{\rho}}}$, $\mathbb{K}_{\partial_y, \sqrt{\boldsymbol{\rho}}}$ and $\mathbb{K}_{\mathbf{u}_h}$; their exact formulations are covered in the next section. Matrix $\mathbb{D}_{\sqrt{\boldsymbol{\rho}}}^{(2)}$ corresponds to the 2-form mass matrix as follows:

$$\left(\mathbb{D}_{\sqrt{\boldsymbol{\rho}}}^{(2)} \right)_{ij} = \int_{\Omega} \frac{\psi_i^{(2)} \psi_j^{(2)}}{\star \sqrt{\boldsymbol{\rho}}_h^{(2)}}$$

8.2.3. Kinetic energy modeling

As was covered in section 7.2.4 for the one-dimensional Euler equations, the skew-symmetric model gives room to define the discretized advection of kinetic energy in two different ways. Depending on the choice of advection operators the Lie derivative acting on the kinetic energy term maps between different discrete spline spaces. Let χ again denote the spline product algorithm covered in section 2.2.4, then:



Here the left diagram shows the 'approximate' kinetic energy advection approach, while the right diagram shows the 'exact' approach. These are wholly identical to those of the one-dimensional Euler equations, apart from the types of k -forms that are involved. Note again that the solid lines denote operators that are used in the kinetic energy advection approaches; the dashed lines are only present to provide context and to denote other existing operators between these spaces.

The various matrices that were denoted with \mathbb{K} in the previous section then become, for the approximate kinetic energy advection approach:

$$\begin{aligned}\mathbb{K}_{\partial_x, \sqrt{\rho}} &= \mathbb{D}_{\sqrt{\rho}}^{(2)} \mathbb{E}^{(2,1y)} (\mathbb{M}^{(1y)})^{-1} \mathbb{C}_{\partial_x}^{(1y, \tilde{2})} \\ \mathbb{K}_{\partial_y, \sqrt{\rho}} &= \mathbb{D}_{\sqrt{\rho}}^{(2)} \mathbb{E}^{(2,1x)} (\mathbb{M}^{(1x)})^{-1} \mathbb{C}_{\partial_y}^{(1x, \tilde{2})} \\ \mathbb{K}_{u_h} &= \mathbb{M}^{(2)} \begin{bmatrix} \mathbb{E}^{2,1x} & \mathbb{E}^{2,1y} \end{bmatrix} \begin{bmatrix} (\mathbb{M}^{(1x)})^{-1} \mathbb{C}_{u_h}^{(1x, \tilde{2})} \\ -(\mathbb{M}^{(1y)})^{-1} \mathbb{C}_{u_h}^{(1y, \tilde{2})} \end{bmatrix}\end{aligned}$$

And for the exact kinetic energy advection approach:

$$\begin{aligned}\mathbb{K}_{\partial_x, \sqrt{\rho}} &= \mathbb{D}_{\sqrt{\rho}}^{(2, \tilde{2})} \mathbb{E}^{(2, \tilde{1}y)} (\mathbb{M}^{(\tilde{1}y)})^{-1} \mathbb{C}_{\partial_x}^{(\tilde{1}y, \tilde{2})} \\ \mathbb{K}_{\partial_y, \sqrt{\rho}} &= \mathbb{D}_{\sqrt{\rho}}^{(2, \tilde{2})} \mathbb{E}^{(2, \tilde{1}x)} (\mathbb{M}^{(\tilde{1}x)})^{-1} \mathbb{C}_{\partial_y}^{(\tilde{1}x, \tilde{2})} \\ \mathbb{K}_{u_h} &= \mathbb{M}^{(2, \tilde{2})} \begin{bmatrix} \mathbb{E}^{\tilde{2}, \tilde{1}x} & \mathbb{E}^{\tilde{2}, \tilde{1}y} \end{bmatrix} \begin{bmatrix} (\mathbb{M}^{(\tilde{1}x)})^{-1} \mathbb{C}_{u_h}^{(\tilde{1}x, \tilde{2})} \\ -(\mathbb{M}^{(\tilde{1}y)})^{-1} \mathbb{C}_{u_h}^{(\tilde{1}y, \tilde{2})} \end{bmatrix}\end{aligned}$$

Recall that the tilde superscripts $\tilde{2}$, $\tilde{1}_x$, $\tilde{1}_y$ are used to denote the spaces corresponding to the product 1- and 2-forms. Like in the previous chapter, double superscripts such as those used for mass matrix $\mathbb{M}^{(2, \tilde{2})}$ contain integrals of the products $\psi_i^{(2)} \psi_j^{(\tilde{2})}$.

8.3. Two-dimensional isentropic vortex

Within the current research work the two-dimensional test considered is that of the so-called isentropic vortex. This case will be considered on $\Omega =]0, 10[^2$, although other domain sizes and non-periodic domains are also used throughout literature. A freestream velocity field (u_∞, v_∞) is defined on Ω together with temperature T_∞ . Perturbations are imposed linearly on the freestream conditions. As given by Yee et al. in [59] these perturbations can be defined as:

$$\begin{aligned}\delta u &= -\frac{\beta}{2\pi} e^{\frac{1}{2}(1-r^2)} (y - y_0) & \delta v &= \frac{\beta}{2\pi} e^{\frac{1}{2}(1-r^2)} (x - x_0) \\ \delta T &= -\frac{(\gamma - 1)\beta^2}{8\gamma\pi^2} e^{1-r^2}\end{aligned}$$

Where $\beta \in \mathbb{R}$ is the vortex strength, (x_0, y_0) are the initial coordinates of the center of the vortex, $r^2 = (x - x_0)^2 + (y - y_0)^2$ is the distance to the center of said vortex and γ is the ratio of specific heats. In the current research (x_0, y_0) will always be taken at the center of the domain. Knowing that $p = \rho^\gamma$ and $T = p/\rho$ allows us to apply the perturbation δT to ρ , by noting that thus $\rho = T^{1/(\gamma-1)}$. These perturbations are imposed onto the freestream conditions, from which the initial conditions follow:

$$u = u_\infty + \delta u = u_\infty - \frac{\beta}{2\pi} e^{\frac{1}{2}(1-r^2)} (y - y_0) \quad v = v_\infty + \delta v = v_\infty + \frac{\beta}{2\pi} e^{\frac{1}{2}(1-r^2)} (x - x_0) \quad (8.15)$$

$$\rho = [T_\infty + \delta T]^{\frac{1}{\gamma-1}} = \left[T_\infty - \frac{(\gamma-1)\beta^2}{8\gamma\pi^2} e^{1-r^2} \right]^{\frac{1}{\gamma-1}} \quad E = \frac{p}{\gamma-1} + \frac{1}{2}\rho(u^2 + v^2) \quad (8.16)$$

Spiegel et al. remark in [51] that this formulation of the vortex perturbation indeed leads to an initial condition without entropy perturbation; since $S = p/\rho^\gamma$ and $p = \rho^\gamma$ the perturbations do not affect the entropy distribution: $\delta S = 0$. By setting (u_∞, v_∞) , T_∞ , (x_0, y_0) , γ and vortex strength β we can thus fully define an isentropic vortex test. Within this research work (x_0, y_0) will always be the domain's center: $(x_0, y_0) = (5, 5)$. Furthermore $\gamma = 1.4$ is used, together with $T_\infty = 1$. Lastly vortex strength $\beta = 5$ will be used for all tests in this chapter. As remarked by Spiegel et al. in section IV.D of [51] the vortex definition in (8.15) is not periodic. Imposing periodic boundary conditions onto this non-periodic initial condition will induce some error if the velocity perturbations at the boundaries are not zero. Within this research the aforementioned parameters result in velocity perturbation components of $\mathcal{O}((e^{-12})) = \mathcal{O}((10^{-6}))$ on $\partial\Omega$.

Two vortex tests will be covered in this thesis: In section 8.4 the results obtained for a static vortex (zero freestream velocity) are discussed, followed by a moving vortex with $(u_\infty, v_\infty) = (1, 1)$ in section 8.5. For both test cases three sets of comparisons are made, in the following order:

1. The two kinetic energy advection modeling approaches of the skew-symmetric model with $\epsilon = 10^{-6}$, $\epsilon = 10^{-12}$;
2. The regular and skew-symmetric models with $\epsilon = 10^{-6}$, $\epsilon = 10^{-12}$;
3. Clawpack, the nodal DG method of Hesthaven & Warburton and the regular and skew-symmetric models with $\epsilon = 10^{-6}$.

The aim of this three-staged result comparison approach is to establish: The (lack of) differences between the kinetic energy modeling approaches of the skew-symmetric model; consistency of the skew-symmetric model and the regular model, to show that the decomposition into Roe variables does not negatively affect the accuracy of the model; the performance of the skew-symmetric model relative to the regular model, Clawpack and the aforementioned nodal DG model.

Said comparisons will be made in terms of the relative L^2 errors of the primary variables (mass density, x -momentum, y -momentum, total energy density) and the conservation errors of each method. The L^2 errors are normalized by using the L^2 norms of the exact variables (mass density, momentum components and energy density). Since the exact isentropic vortex solutions conserve mass, momentum in both directions, kinetic energy, internal energy and total energy it is straightforward to determine how well each numerical method approximates this conservation behavior: For a physical quantity of interest r the relative non-conservation error Δr is defined in the same way as was done in previous chapters:

$$\Delta r(t) = \frac{(r_{num}(t) - r_{num}(0)) - (r_{exact}(t) - r_{exact}(0))}{r_{exact}(t)} = \frac{r_{num}(t) - r_{num}(0)}{r_{exact}(0)}$$

Here the second equality follows because (as mentioned above) the exact solutions conserve all physical quantities of interest considered in this research. The equation above defines the relative kinetic energy conservation errors that are used to compare the amounts of numerical dissipation present in each method. Lastly the absolute conservation errors of the primary variables are compared; these are not normalized, since the exact solution of the static vortex test that is covered in section 8.4 has x - and y -momenta that are exactly zero. For the sake of consistency the primary variables are thus compared in terms of their absolute conservation errors $r_{num}(t) - r_{num}(0)$.

8.4. Static isentropic vortex test

The first of the two-dimensional test cases considered in this research work is the isentropic vortex without background velocity: $(u_\infty, v_\infty) = (0, 0)$. This corresponds to an exact solution that is entirely static and time-independent. Section 8.4.1 covers the first comparison, that of the two kinetic energy modeling approaches of the skew-symmetric model which were both covered in section 8.2.3. Following this initial comparison the skew-symmetric model's results are compared to the regular model that is described in section 8.2 as well, for different nonlinear convergence tolerances ϵ . Lastly the results of both of these models (with $\epsilon = 10^{-6}$) are compared to Clawpack and the nodal DG method of Hesthaven & Warburton; these reference models have been described in chapter 4.

8.4.1. Kinetic energy modeling comparison

As mentioned, we start the static vortex numerical experiments by comparing the results obtained with the two kinetic energy modeling approaches possible with the skew-symmetric model.

Relative L^2 errors

Figure 8.2 shows the L^2 errors of the exact and approximate kinetic energy advection modeling approaches for $\epsilon = 10^{-6}$ and $\epsilon = 10^{-12}$, obtained with various basis function degrees p . As can be seen the differences between both approaches are indiscernible for most combinations of mesh sizes and degrees, indicating that the kinetic energy advection modeling approaches show similar scaling behavior. Whereas all L^2 errors of the simulations with odd-degree basis functions decrease monotonically, the even-degree simulations are more fickle and do not show monotonic L^2 convergence for the mass density field. This non-monotonic convergence appears during the simulation itself, as is evidenced by figure 8.1. In this figure the relative L^2 errors of the mass density field are shown. Monotonic convergence is shown for all degrees p is shown over most of the mesh size range.

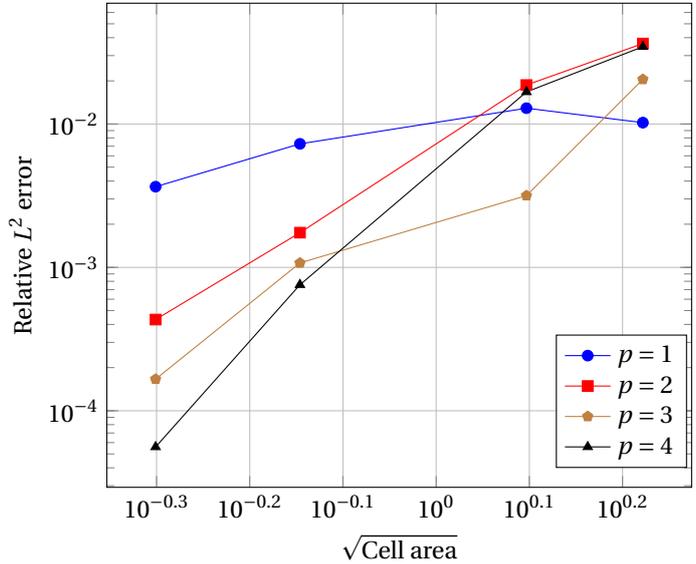


Figure 8.1: Convergence of the relative mass density L^2 errors of the initial condition for different degrees p

Numerical diffusion

The same trend is shown in figure 8.3, where the maximum deviation of the total kinetic energy on Ω with respect to the initial kinetic energy is shown. Note that the deviations shown have been normalized with the exact total kinetic energy, which is $\mathcal{O}(1)$ for this static vortex. These deviations are a measure of the amount of numerical dissipation induced by the numerical methods. Both modeling approaches again show nearly-identical behavior, which indicates that using higher-degree spaces to resolve the advection of kinetic energy does not provide any gain. Again only the simulations with odd-degree basis functions seem to converge monotonically.

Primary non-conservation errors

In fact, there is an apparent downside to using these higher-degree spaces for modeling kinetic energy advection: Round-off errors become more pronounced, leading to degrading conservation behavior. This can be seen in figure 8.4, which shows the maximum mass, momentum and (total) energy deviations relative to their respective initial values. Whereas mass is conserved up to $\mathcal{O}(10^{-13})$ for each model and all considered p and ϵ , this is not the case for the momentum components and the total energy. Both kinetic energy models suffer from worse conservation performance as the basis function degree p is increased, but this effect is more pronounced with the exact kinetic energy advection model. Momentum and total energy conservation up to $\mathcal{O}(10^{-13})$ is attained with $p=1$ for both models, with nearly identical outcomes. While their results are still largely the same for $p=2$ a further degree increase to $p=3$ and $p=4$ comes with significant differences

between both models: The approximate model has a maximum non-conservation of $\mathcal{O}(10^{-12})$ for $p = 3$ and $\mathcal{O}(10^{-11})$ for $p = 4$, whereas the exact kinetic energy advection model shows conservation errors of $\mathcal{O}(10^{-11})$ for $p = 3$ and $\mathcal{O}(10^{-10})$ for $p = 4$. These conservation errors appear to be independent of mesh size, indicating that they originate from round-off errors that appear when constructing the discrete operators used in each simulation.

Conclusions

Two preliminary conclusions can be drawn based on these results. The first conclusion is that the approximate kinetic energy advection modeling approach is preferable over the exact approach; whereas the differences in L^2 error (accuracy) and kinetic energy non-conservation (numerical diffusion) are nearly identical for both approaches, the approximate approach has significantly smaller conservation errors for when higher-degree basis functions are used. Furthermore the approximate approach leads to a numerical scheme that has a lower computational cost, since smaller matrices are used for some of the discrete kinetic energy advection operators.

The second conclusion that follows from these results is that the effect of ϵ within the range $[10^{-12}, 10^{-6}]$ is seemingly negligible. As far as any noticeable differences exist between the results obtained with $\epsilon = 10^{-6}$ and $\epsilon = 10^{-12}$, these are dwarfed by the differences between each basis function degree p . Hence the extra computational cost induced by running each time step until the difference between successive iterations is smaller than $\epsilon = 10^{-12}$ is not offset by an increase in accuracy or decrease in numerical diffusion.

8.4.2. Regular & skew-symmetric model comparison

Now that the efficacy of the two kinetic energy advection modeling approaches of section 8.2.3 has been compared and the choice has been made for the approximate approach, we turn to comparing the regular and skew-symmetric models that were derived in section 8.2.

Relative L^2 errors

The L^2 errors of the regular and skew-symmetric models are shown in figure 8.6 for $\epsilon = 10^{-6}$ and $\epsilon = 10^{-12}$. Note that the skew-symmetric results were also used in the previous section. Taking these skew-symmetric results as a baseline, we can see that the results of the regular model are similar for $p = 1, p = 3$. For $p = 2$ and $p = 4$ however, the regular model suffers from non-monotonic convergence for all variables. While this is also the case for the mass density convergence of the skew-symmetric model, the L^2 errors of the regular model are significantly higher for even-valued p , indicating that it is less stable than the skew-symmetric model. This is further supported by noting that the regular model diverged with $p = 4$ on the finest mesh, hence the missing data point in figure 8.6. These L^2 errors are produced during the simulation; they are not present in the discretized initial conditions, as is evidenced by figure 8.5. In this figure the relative L^2 errors of the discrete mass density initial conditions are shown for the regular and skew-symmetric methods. It is clear that the behavior of both methods is similar; the differences between them thus form during the simulation itself. It is also worth pointing out in figure 8.6 that the regular model, like the skew-symmetric model, is insensitive to whether $\epsilon = 10^{-6}$ or $\epsilon = 10^{-12}$ is used: The results with both ϵ 's are indistinguishable from one another.

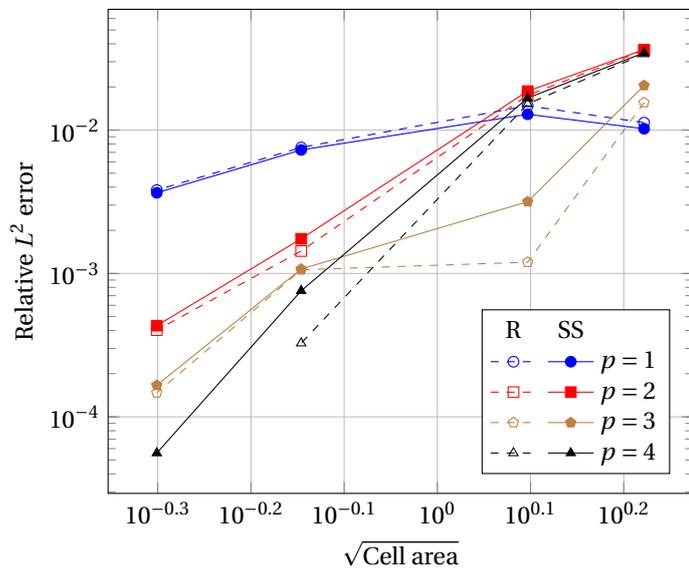


Figure 8.5: Convergence of relative mass density L^2 errors of initial conditions of the regular (R) and skew-symmetric (SS) models for different degrees p

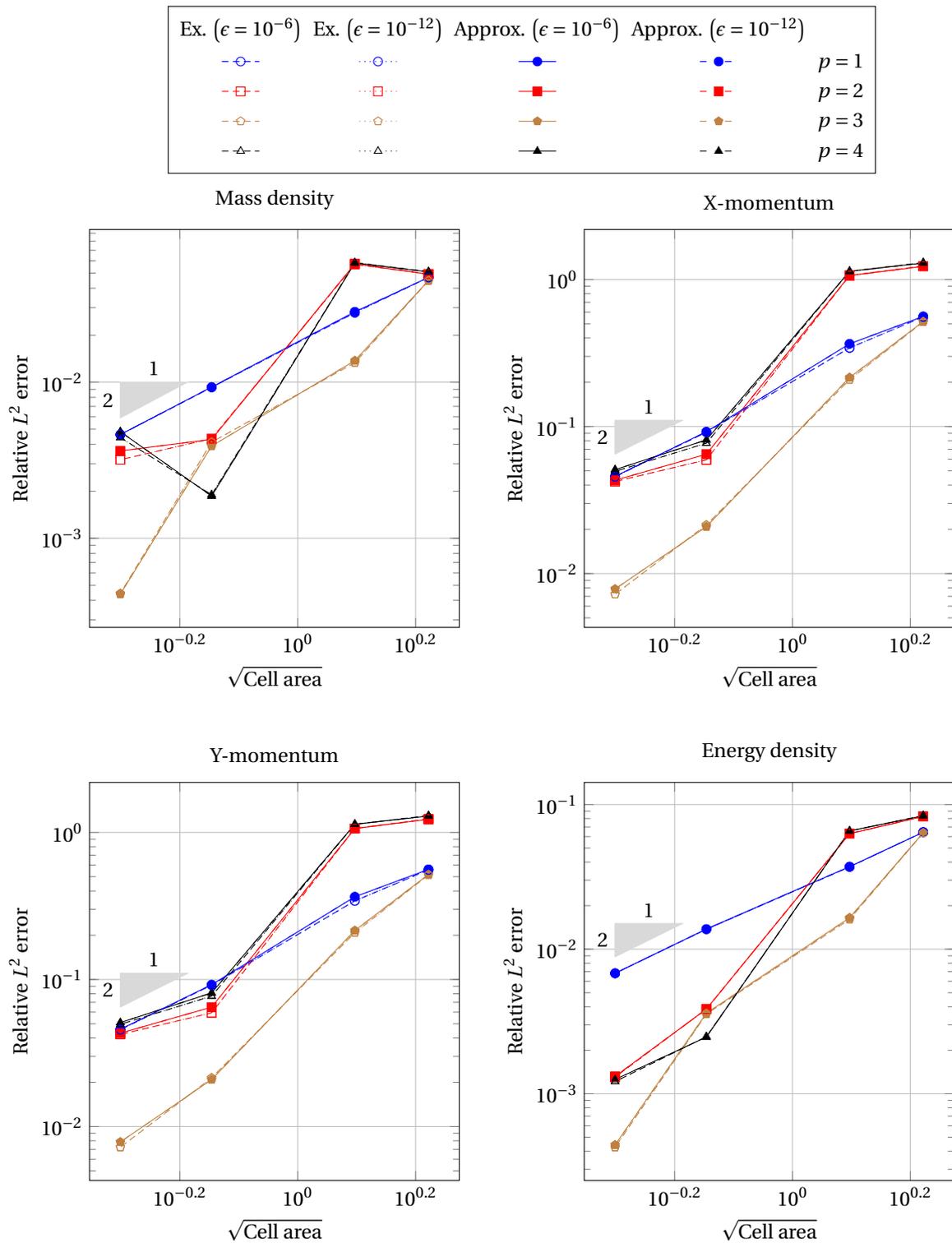


Figure 8.2: Convergence of the relative L^2 errors at $t = 10$ of the skew-symmetric model with exact (Ex.) and approximate (Approx.) kinetic energy modeling for $\Delta t = 5 \cdot 10^{-2}$, $(u_\infty, v_\infty) = (0, 0)$, different solution degrees p , $\epsilon = 10^{-6}$ and $\epsilon = 10^{-12}$

Numerical diffusion

Looking at the numerical diffusion (kinetic energy non-conservation) a similar picture is painted: As figure 8.7 shows, both the regular and skew-symmetric model have (nearly) identical amounts of numerical diffusion for $p = 1$ and very similar amounts for $p = 3$. For $p = 2$, $p = 4$ the same pattern can be seen that was noted for

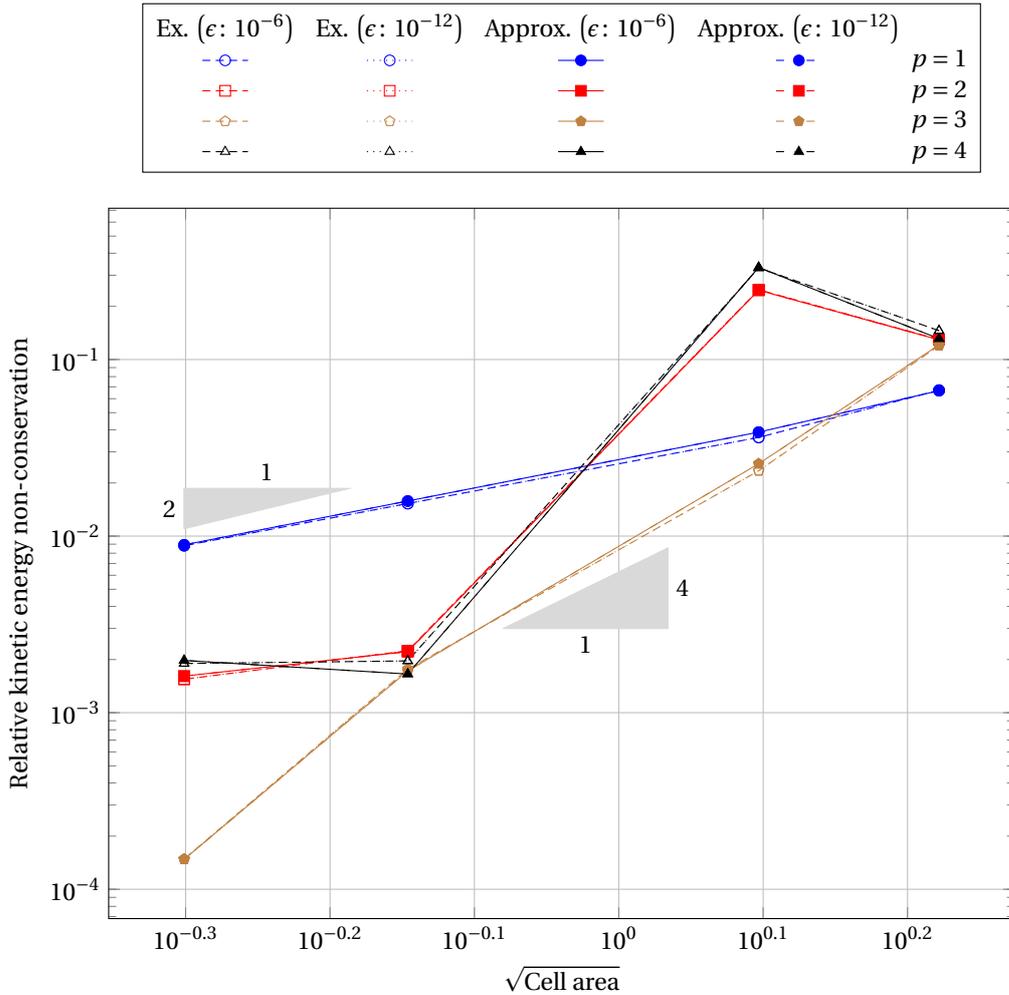


Figure 8.3: Maximum kinetic energy non-conservation errors of the skew-symmetric model with exact (Ex.) and approximate (Approx.) kinetic energy modeling for $\Delta t = 5 \cdot 10^{-2}$, $(u_\infty, v_\infty) = (0, 0)$, different solution degrees p , $\epsilon = 10^{-6}$ and $\epsilon = 10^{-12}$

the L^2 errors, with the regular model having kinetic energy conservation errors that are at least a full order of magnitude higher than that of the skew-symmetric model. This trend is also independent of ϵ .

Primary non-conservation errors

Lastly, we compare the conservation errors of the primary variables of both models. These are shown in figure 8.8. As can be noted when looking at the mass (in the top left corner) and total energy (in the bottom right corner) conservation errors, the regular model is more sensitive to the round-off errors that appear in the continuity and energy equations when p is increased. Especially for $p = 3$ ($\mathcal{O}(10^{-12})$) and $p = 4$ ($\mathcal{O}(10^{-11})$) the mass and energy conservation errors are at least an order of magnitude larger than those of the skew-symmetric model ($\mathcal{O}(10^{-13})$ and $\mathcal{O}(10^{-12})$ respectively). When looking at X-momentum (in the top right corner) and Y-momentum (in the bottom left corner) it is apparent that both methods have similar performance: The conservation error is raised by similar amounts as p is increased. One can furthermore observe that the value of ϵ once again has only a small effect on the results; only for $p = 1$, $p = 2$ there are noticeable difference, though these are only small and are negligible compared to the variations that are encountered when refining the grid or increasing the basis function degree p .

Conclusions

Based on these observations it can be concluded that the skew-symmetric model slightly outperforms the regular model. Both models are insensitive to the values of ϵ used, and as long as basis functions of odd degrees are used their performance is largely similar; the only things distinguishing the skew-symmetric model is its smaller sensitivity to round-off errors. For even-degree solutions a different picture can be seen: The

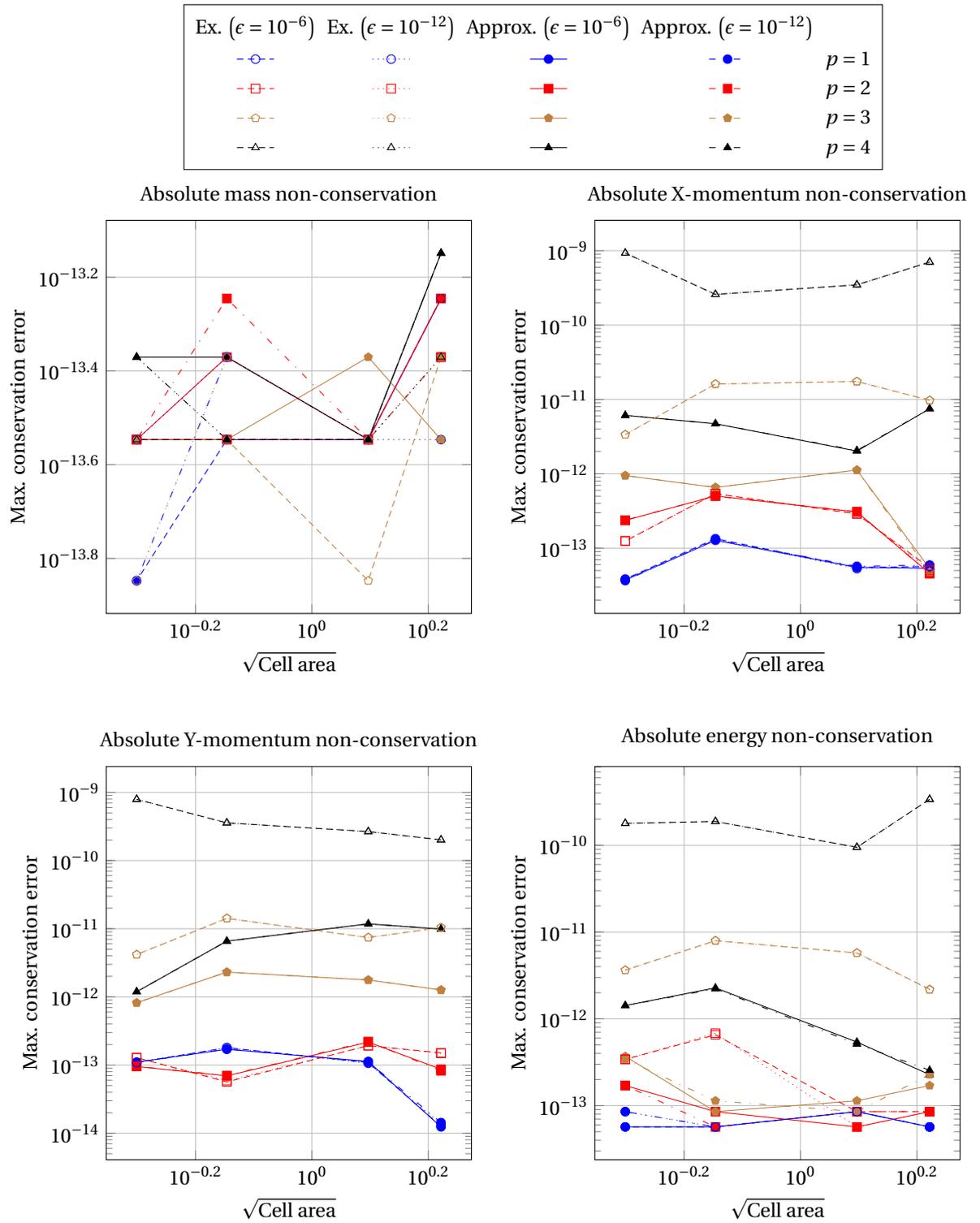


Figure 8.4: Maximum non-conservation of primary variables of the skew-symmetric model with exact (Ex.) and approximate (Approx.) kinetic energy modeling for $\Delta t = 5 \cdot 10^{-2}$, $(u_\infty, v_\infty) = (0, 0)$, different solution degrees p , $\epsilon = 10^{-6}$ and $\epsilon = 10^{-12}$

regular method has significant issues (no convergence on the finest mesh for $p = 4$) and does not have monotonic L^2 convergence for any of the variables shown. Furthermore its L^2 errors are larger and it has a full order of magnitude more numerical diffusion than the skew-symmetric method for this static vortex case.

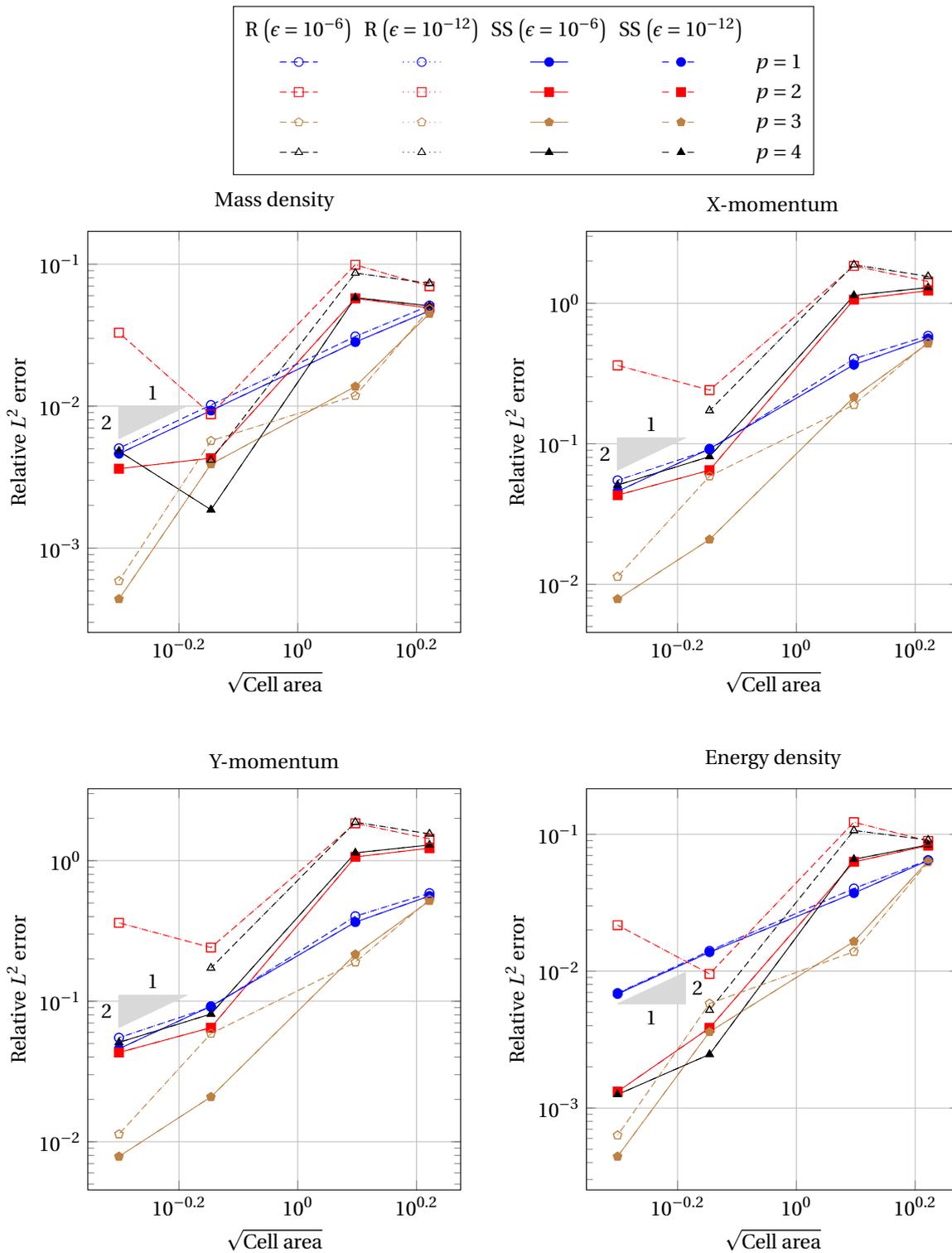


Figure 8.6: Convergence of the relative L^2 errors at $t = 10$ of the regular (R) and skew-symmetric (SS) models for $\Delta t = 5 \cdot 10^{-2}$, $(u_\infty, v_\infty) = (0, 0)$, different solution degrees p , $\epsilon = 10^{-6}$ and $\epsilon = 10^{-12}$

8.4.3. Comparison with established methods

The last comparisons made for the static vortex test case are those between the regular and skew-symmetric numerical models on the one hand, and Clawpack and the nodal DG method of Hesthaven & Warburton on the other hand. As mentioned before, these external methods have been covered in chapter 4. The aim of this

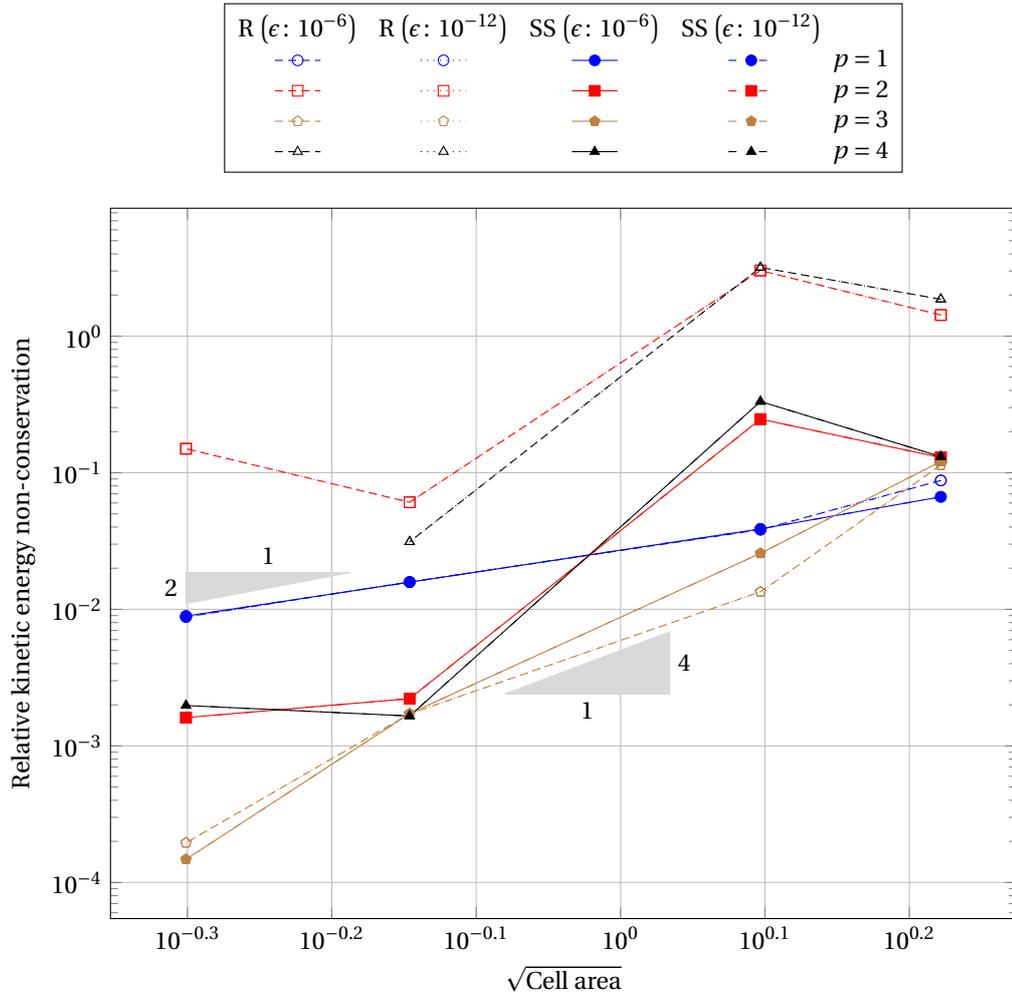


Figure 8.7: Maximum kinetic energy non-conservation errors of the regular (R) and skew-symmetric (SS) models for $\Delta t = 5 \cdot 10^{-2}$, $(u_\infty, v_\infty) = (0, 0)$, different solution degrees p , $\epsilon = 10^{-6}$ and $\epsilon = 10^{-12}$

comparison is to establish the efficacy of the methods presented in this research, relative to existing methods. It is worth noting that both Clawpack and the nodal DG method require filtering or flux limiting in order to converge to useful results: It was found that without the added stabilization this filtering/flux limiting provides, the DG method would not produce any results. However, even with this stabilization enabled the DG method failed to converge on the coarsest meshes for $p = 3$. Clawpack's low-order methods result in solutions that have been dissipated greatly: The original vortex at the center of the domain is no longer recognizable. Hence Clawpack's high-resolution methods had to be used in order to obtain useful results, which entails the use of a flux limiter.

Relative L^2 errors

Figure 8.9 shows the relative L^2 errors at $t = 10$ of the regular and skew-symmetric models with nonlinear tolerance $\epsilon = 10^{-6}$ and of Clawpack and the aforementioned DG method, both for different mesh sizes and solution degrees p . Since Clawpack uses piecewise-constant function approximations it is interpreted to be of degree $p = 0$. With $p = 1$ and $p = 3$ the models presented in this research outperform the DG method on the entire mesh size range considered here; moreover, for $p = 3$ the DG model diverged on the coarsest meshes. As mentioned in previous sections as well, the even-powered degrees $p = 2$ and $p = 4$ are more problematic for the regular and skew-symmetric models; the mass density L^2 errors of both even-powered degrees do not converge monotonically and for $p = 4$ the L^2 errors of the other primary variables are also considerably larger on the finest meshes than their counterparts in the DG model. For the regular model this trend is larger than for the skew-symmetric model. Whereas Clawpack has monotonic convergence with a seemingly increasing order of accuracy as the mesh is refined, it also has the shallowest slope and highest L^2 errors of all methods

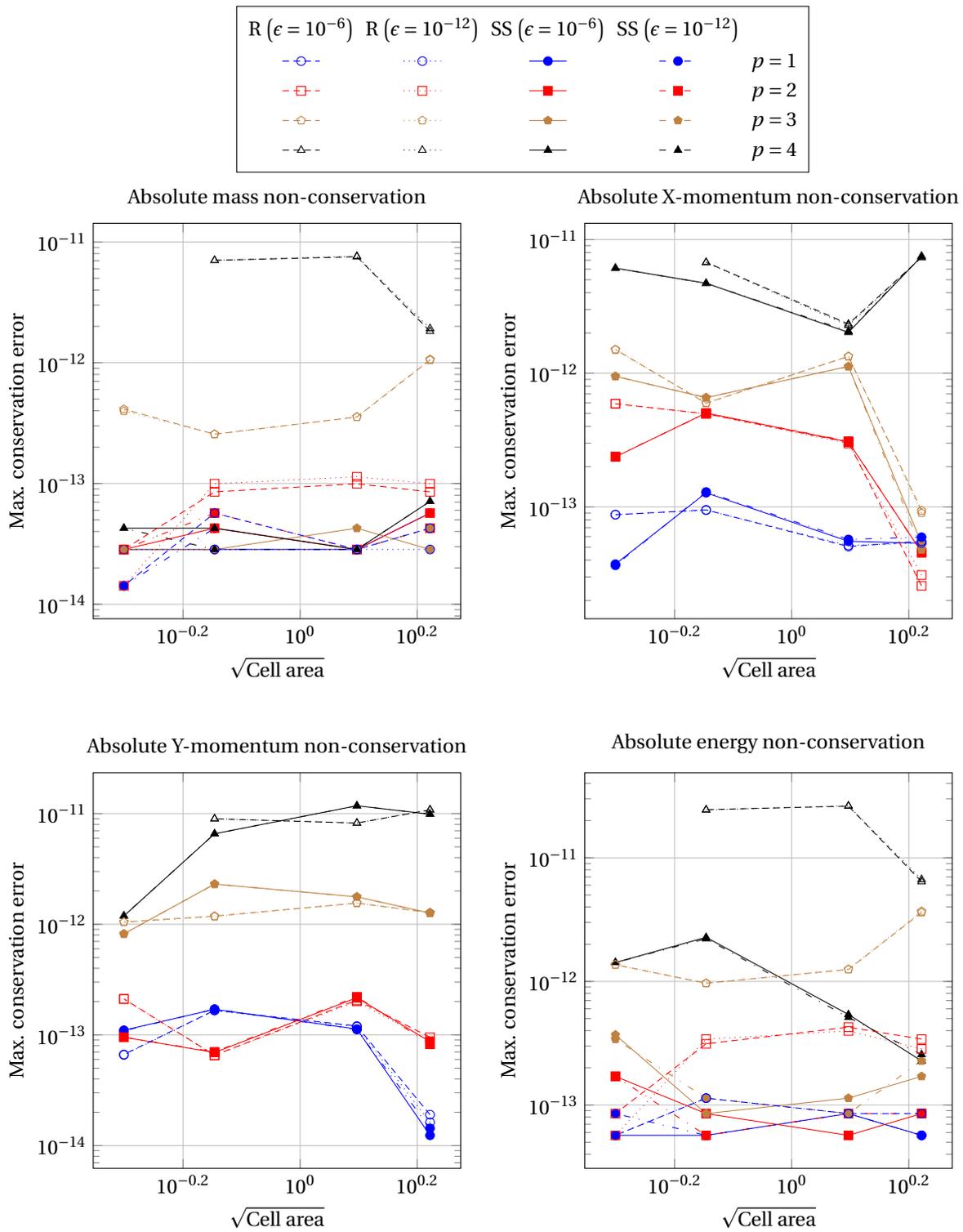


Figure 8.8: Conservation errors of primary variables of the regular (R) and skew-symmetric (SS) models for $\Delta t = 5 \cdot 10^{-2}$, $(u_\infty, v_\infty) = (0, 0)$, different solution degrees p , $\epsilon = 10^{-6}$ and $\epsilon = 10^{-12}$

shown here.

Numerical diffusion

Following the L^2 error comparison, we look at the numerical diffusion of these methods by comparing their maximum relative kinetic energy conservation errors. Figure 8.10 contains these conservation errors, graphed

against the square root of the cell area. One can see that Clawpack (again) is the method that performs poorest; over the entire range of cell sizes considered it has the highest kinetic energy conservation error of all methods considered in this work, except for the regular method with $p = 2$, $p = 4$ on the coarsest meshes. The regular and skew-symmetric models outperform the DG method for $p = 1$, $p = 3$ along most of the mesh sizes shown, though the DG method catches up on the finest mesh with $p = 3$. For $p = 2$ and $p = 4$ the regular and skew-symmetric models show non-monotonic convergence of the kinetic energy conservation error. Whereas the performance of the skew-symmetric model is at times similar, better or worse than the DG method when looking at these even-valued p , the regular model is worse on most of the cell size range considered. Recall that the regular model also did not converge for $p = 4$ on one of the meshes, hence one of its data points is missing; this is an additional sign of its problematic stability with even-degree cases for this static vortex.

A different picture emerges when comparing the numerical diffusion of the various methods versus the number of degrees of freedom used, rather than versus a cell area metric. This is shown in figure 8.11. As mentioned in section 4.2 the nodal DG method uses more degrees of freedom per cell when the degree p is increased; a process called k -refinement. Hence the same mesh with $p = 1$ and $p > 1$ results in two different amounts of degrees of freedom that are used to resolve the DG method's numerical system. When judging the performance of the methods in this way it is apparent that the numerical models presented in this thesis are more efficient (in terms of numerical diffusion) than Clawpack and the DG method. Clawpack is still largely outperformed by the three other methods over the mesh size range considered and is unlikely to catch up to the other methods on finer meshes, while the DG method needs $\mathcal{O}(10^3)$ number of degrees of freedom to reduce its numerical diffusion to amounts lower than $\mathcal{O}(10^{-2})$. Compared to at most $4 \cdot 10^2$ degrees of freedom that the regular and skew-symmetric models need for even the lowest-degree solutions to attain this the mesh size of the DG method presents a significant computational cost, as every degree of freedom constitutes an additional equation that needs to be handled.

Primary non-conservation errors

As with the other comparisons, the final parameters that are considered here are the conservation errors of the primary variables. These are shown in figure 8.12. We start by looking at the mass and energy conservation errors, shown in the top left and bottom right corners respectively. Clawpack's performance for these parameters is similar to the lower-degree simulations performed with the regular and skew-symmetric methods. The DG method shows the same pattern as the regular and skew-symmetric methods: Increasing conservation errors as the degree p is increased, although the magnitude of this increase in mass and energy conservation errors is limited and does not exceed $\mathcal{O}(10^{-13})$ for mass and $\mathcal{O}(10^{-12})$ for energy. It can also be noted that the conservation errors of the DG method increase as the mesh size is increased. Lastly, we note that the regular model with $p = 4$ has significantly higher conservation errors than the other methods shown.

When looking at the X- and Y-momentum conservation errors (in the top right and bottom left corners of figure 8.12 respectively) it is immediately apparent that Clawpack outperforms the other methods by at least an order of magnitude over the range of mesh sizes considered here. For X-momentum the DG method is on par with the regular and skew-symmetric models for $p = 1$, but for $p > 1$ there are noticeable differences between them as the mesh is refined; this is due to the apparent near-monotonic growth of momentum conservation errors of the DG method. With the finest mesh and $p = 4$ the DG method reaches an X-momentum conservation error of $\mathcal{O}(10^{-10})$, which is (nearly) an order of magnitude larger than that of the skew-symmetric model. Turning to Y-momentum, it can be seen that even with lower-degree basis functions the DG method has conservation errors that are (nearly) an order of magnitude larger than those of the regular and skew-symmetric models - save for one point with $p = 2$, where similar performance of the three models is observed. For the higher-degree basis functions the conservation errors of these three methods are similar on the coarsest meshes, while the conservation error growth with mesh refinement found in the DG method leads to large differences on finer meshes.

Conclusions

To recap this comparison between Clawpack, the nodal DG method of Hesthaven & Warburton and the regular and skew-symmetric models presented in section 8.2, the following can be noted:

- Clawpack suffers the most from numerical diffusion for this case, as is evident by its large L^2 errors

and kinetic energy conservation errors compared to the other methods. Of the four methods being compared Clawpack has the poorest performance on the metrics considered here, with the exception of the conservation errors of the primary variables.

- The performance of the DG method and the regular & skew-symmetric models for this case is comparable when comparing them against the square-root of their cell sizes. When comparing their results in terms of the number of degrees of freedom the DG method performs markedly worse than the regular & skew-symmetric models; it uses significantly more degrees of freedom to attain similar L^2 errors and kinetic energy conservation tolerances. This effect is most apparent with $p = 3$, $p = 4$, since these basis function degrees result in the highest number of degrees of freedom for the DG method.
- Of these four methods the skew-symmetric model seems to be the most useful and robust; the other three methods all require some sort of stabilization and/or diverged during one or more of the simulations. The regular method diverged with $p = 4$ on the finest mesh considered (20×20), while the DG method did not converge on the coarsest meshes with $p = 3$ and requires filtering as well as flux limiting in order to converge. Lastly, Clawpack's high-resolution schemes require flux limiters to stabilize the solution in regions with high gradients.

8.5. Moving isentropic vortex test

Following the static vortex test a freestream velocity of $(u_\infty, v_\infty) = (1, 1)$ is used. This makes the vortex traverse Ω once every 10 seconds and raises the total kinetic energy on Ω to $\mathcal{O}(10^2)$. Note that the perturbation parameters are unchanged, as they are simply superimposed on the freestream conditions. The same comparisons are made for this case as were made for the static vortex: Section 8.5.1 first covers the results obtained with the skew-symmetric model's two kinetic energy modeling approaches that were covered in section 8.2.3. Then section 8.5.2 compares the skew-symmetric model with the regular model, after which a comparison of these models with the nodal DG method of Hesthaven & Warburton and Clawpack is made in section 8.5.3. All these comparisons are made in terms of L^2 error (as measure of accuracy), relative kinetic energy non-conservation (as measure of numerical diffusion) and non-conservation of the primary variables (as measure of physical correctness), identical to what was done for the static vortex test case.

8.5.1. Kinetic energy modeling comparison

Like before the first comparison made for the moving vortex case is between the two kinetic energy modeling approaches for the skew-symmetric model that were covered in section 8.2.3.

Relative L^2 errors

Figure 8.16 shows the L^2 errors obtained for these approaches, with $\epsilon 10^{-6}$ and $\epsilon 10^{-12}$. As one can see the differences between the two approaches and the ϵ 's used is negligible compared to the differences of the various basis function degrees p . Moving from $p = 1$ to $p = 2$ gives the largest decreases in the L^2 errors, including an apparent higher order of accuracy; further increases in p seemingly do not increase the order of accuracy but do result in a somewhat lower L^2 error, albeit this decrease is smaller than that found by increasing $p = 1$ to $p = 2$. Hence L^2 errors are not a reason to prefer one kinetic energy modeling approach over another. Unlike the results of the static vortex, the numerical methods show monotonic convergence for all p ; only for the coarsest meshes using $p = 1$ does not result in monotonic convergence.

Numerical diffusion

These same trends are also visible when looking at the amount of kinetic energy non-conservation, shown in figure 8.14: The amount of kinetic energy non-conservation (and thus the numerical diffusion) decreases monotonically for all p , with the results of $p = 1$ having a markedly different order of convergence than those for $p > 1$. Note that again that the differences between the higher-degree basis functions are small. Furthermore they do not seem to obtain orders of accuracy higher than 2, except for the finest meshes considered here. The differences between the kinetic energy modeling approaches and ϵ 's used are negligible compared to the basis function degrees p and mesh refinements. Again, based on these results neither of the two kinetic energy modeling approaches has an edge over the other.

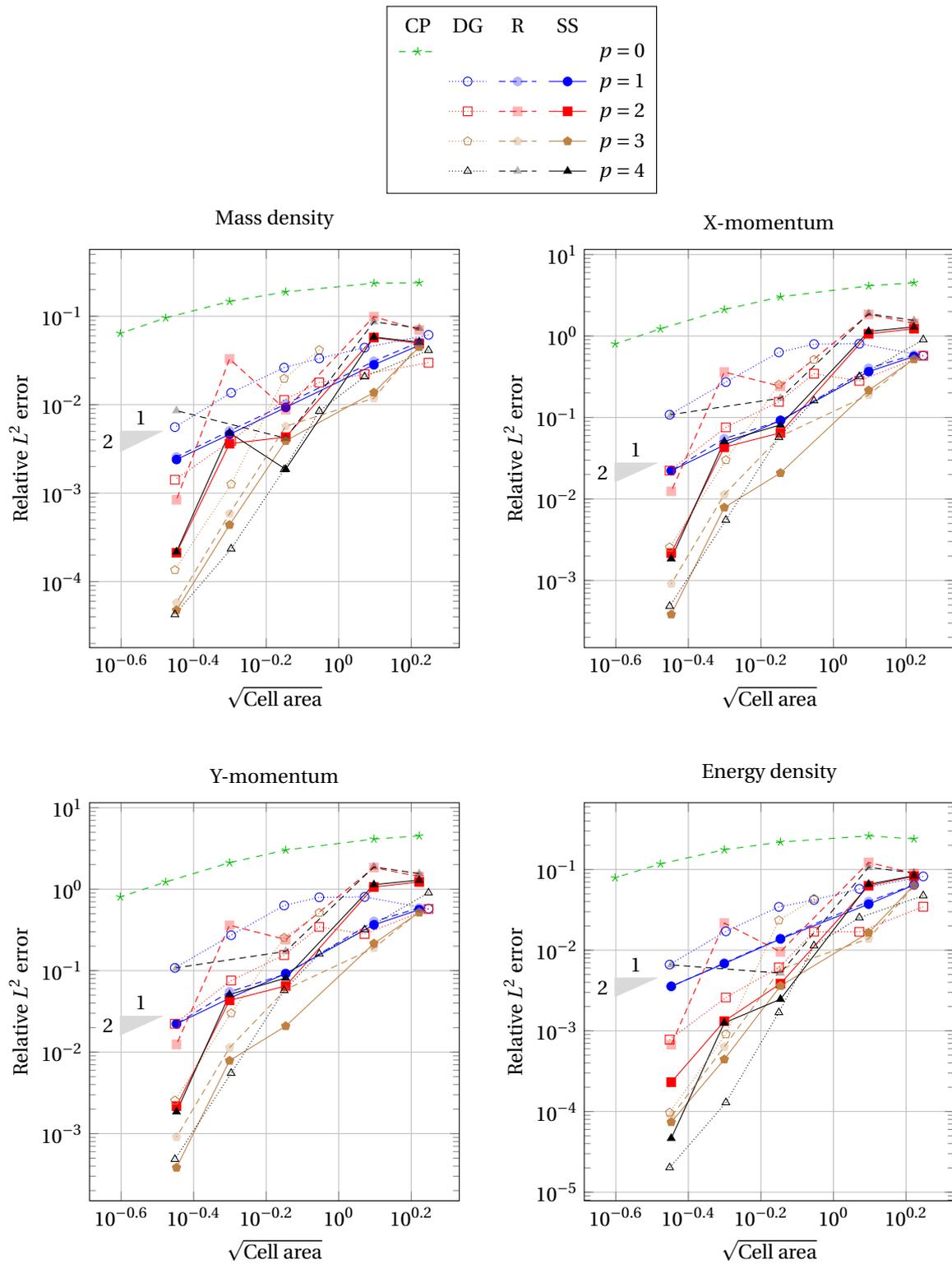


Figure 8.9: Convergence of the relative L^2 errors at $t = 10$ of the regular (R) and skew-symmetric (SS) models for $\Delta t = 5 \cdot 10^{-2}$ and $\epsilon = 10^{-6}$, the DG model (DG) for $\Delta t = 1 \cdot 10^{-2}$ and Clawpack (CP) for $\Delta t = 5 \cdot 10^{-3}$; $(u_\infty, v_\infty) = (0, 0)$ and different solution degrees p were used

Primary non-conservation errors

Lastly the primary variables' non-conservation of both modeling approaches is compared; this is shown in figure 8.15. What is immediately noticeable is the degradation in conservation performance as the degrees p of both methods are increased. For mass conservation the magnitude of this effect is negligible, but for X- &

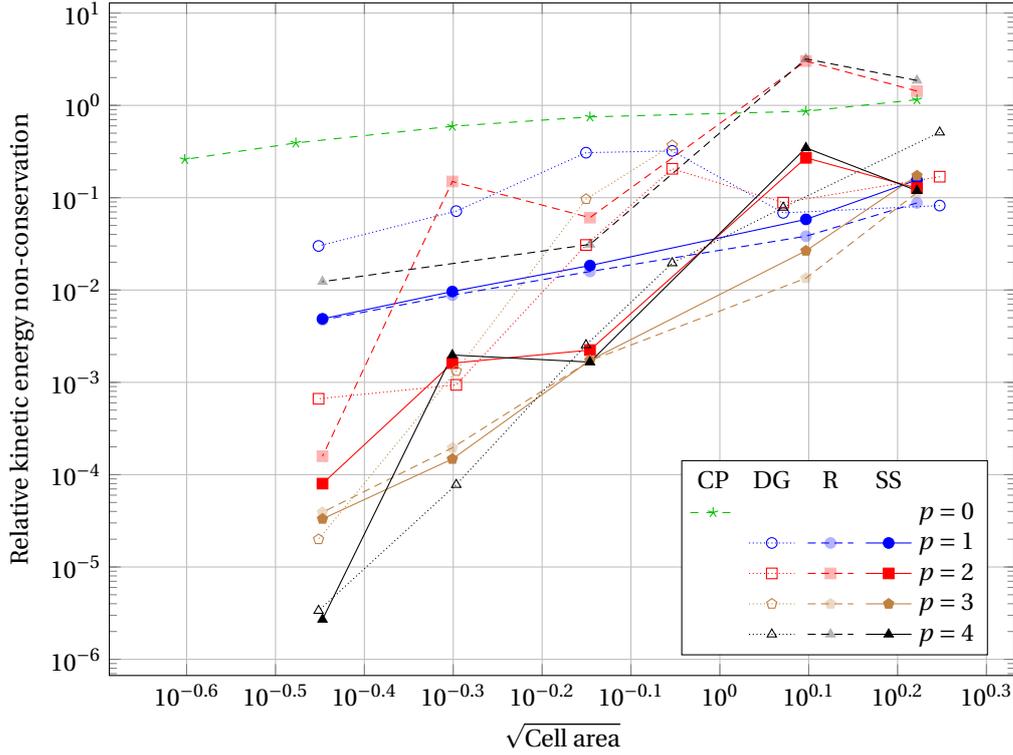


Figure 8.10: Maximum relative kinetic energy non-conservation versus the square root of the cell area for the regular (R) and skew-symmetric (SS) models with $\Delta t = 5 \cdot 10^{-2}$ and $\epsilon = 10^{-6}$, the DG model (DG) with $\Delta t = 1 \cdot 10^{-2}$ and Clawpack (CP) with $\Delta t = 5 \cdot 10^{-3}$; $(u_\infty, v_\infty) = (0, 0)$ and different solution degrees p were used

Y-momentum and energy conservation a (near-)monotonic increase in non-conservation can be seen. This increase is significantly larger for the exact kinetic energy advection modeling approach, leading to conservation errors of $\mathcal{O}(10^{-8})$ with $p = 4$ for X- & Y-momentum and (total) energy, compared to only $\mathcal{O}(10^{-11})$ for the approximate kinetic energy advection modeling approach. The (near-)monotonic increase of the conservation error with p combined with its independence from cell size indicates that the primary drivers behind these conservation errors are the round-off errors encountered when constructing the various discrete operators. Also note that the absolute (not relative) non-conservation of the primary variables is plotted here; the X- & Y-momentum and energy conservation errors of the moving vortex are $\mathcal{O}(10)$ times as large as those found for the static vortex.

Conclusions

As with the static vortex, the main conclusion that can be drawn based on these results is that the approximate kinetic energy advection modeling approach is preferable. There is no significant difference between the exact and approximate approaches when it comes to L^2 errors or numerical diffusion; only the conservation error at higher degrees is different between both methods, with the approximate approach having significantly smaller conservation errors, as mentioned above. Furthermore the approximate approach has a lower computational cost, since some of the discrete operators that have to be constructed once for every nonlinear iteration are smaller.

A secondary conclusion is that the value of the nonlinear convergence tolerance ϵ does not have a noticeable impact on L^2 errors, numerical diffusion and conservation of the primary variables for the range tested here ($\epsilon = 10^{-6}$, $\epsilon = 10^{-12}$). For both kinetic energy modeling approaches figures 8.16, 8.14 and 8.15 show that the extra computational cost used to drive the nonlinear convergence of each time step to $\epsilon = 10^{-12}$ does not come with a significant increase in any of the performance measures considered here.

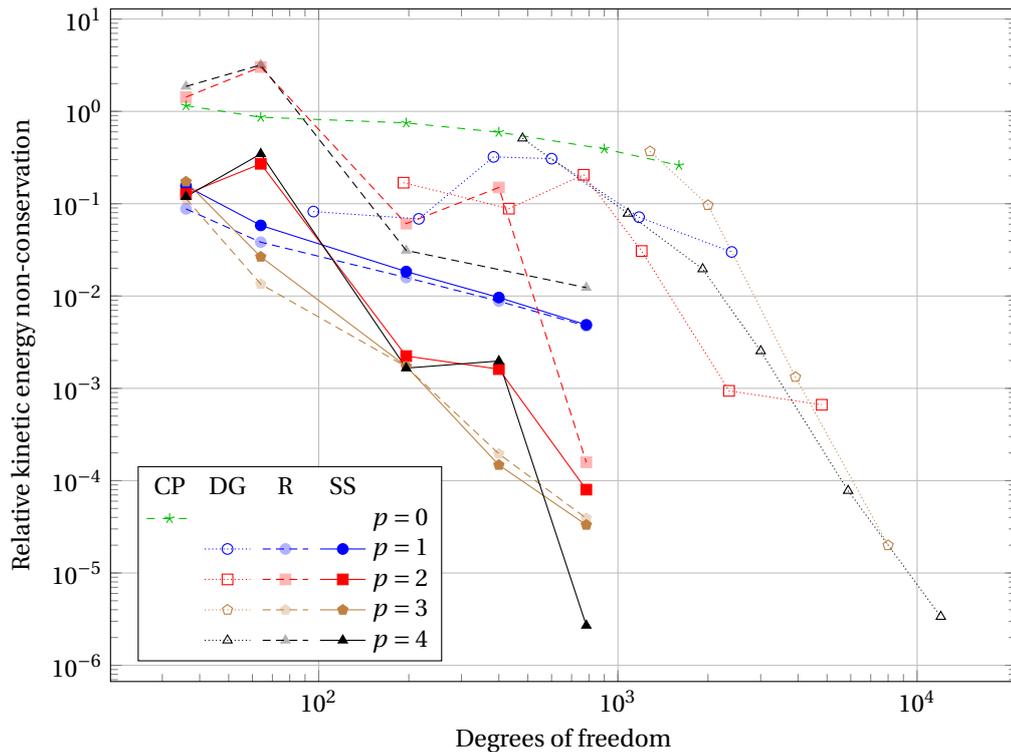


Figure 8.11: Maximum relative kinetic energy non-conservation versus the amount of degrees of freedom for the regular (R) and skew-symmetric (SS) models with $\Delta t = 5 \cdot 10^{-2}$ and $\epsilon = 10^{-6}$, the DG model (DG) with $\Delta t = 1 \cdot 10^{-2}$ and Clawpack (CP) with $\Delta t = 5 \cdot 10^{-3}$; $(u_\infty, v_\infty) = (0, 0)$ and different solution degrees p were used

8.5.2. Regular & skew-symmetric model comparison

Following the comparison of the two kinetic energy advection modeling approaches we turn to a comparison between the regular and skew-symmetric model (with the approximate kinetic energy approach).

Relative L^2 errors

Figure 8.16 shows the relative L^2 errors of several physical parameters for both models. Unlike the results of the static vortex case, with a moving vortex the regular and skew-symmetric models perform similarly for all mesh sizes and degrees p . In this case the L^2 convergence of both methods is monotonic for all p . For all but the coarsest mesh $p = 1$ leads to the highest errors for every parameter, while for the skew-symmetric method $p = 4$ consistently has the lowest error on all meshes except the coarsest. For the regular method the L^2 errors with $p = 4$ show some oscillations, causing them to be larger than those found with $p = 2$, $p = 3$ on some of the intermediate meshes. Both methods have similar behavior when it comes to $p = 2$, $p = 3$: The order with the lowest L^2 error changes per mesh. Overall both methods behave very similarly as far as the L^2 errors for this moving vortex are concerned, and (as was also the case with the static vortex) whether $\epsilon = 10^{-6}$ or $\epsilon = 10^{-12}$ has been used does not seem to noticeably affect the results.

Numerical diffusion

Following the L^2 errors, we look at the kinetic energy conservation errors; these are displayed in figure 8.17. Some differences exist between the results obtained with both methods. On the coarse end of the mesh size spectrum considered here the skew-symmetric model outperforms the regular method when higher-degree ($p = 3$, $p = 4$) basis functions are used, whereas the differences between both methods decrease as the mesh is refined. A similar trend is seen for $p = 1$ and $p = 2$, but for $p = 2$ the regular method displays some oscillatory behavior, with a seemingly varying order of convergence. Lastly we note that again, the two values used for ϵ result in indistinguishable results.

Primary non-conservation error

The final aspects of the comparison between the regular and skew-symmetric models are the conservation errors of the primary physical variables, shown in figure 8.18. As is shown in the top left corner of this figure,

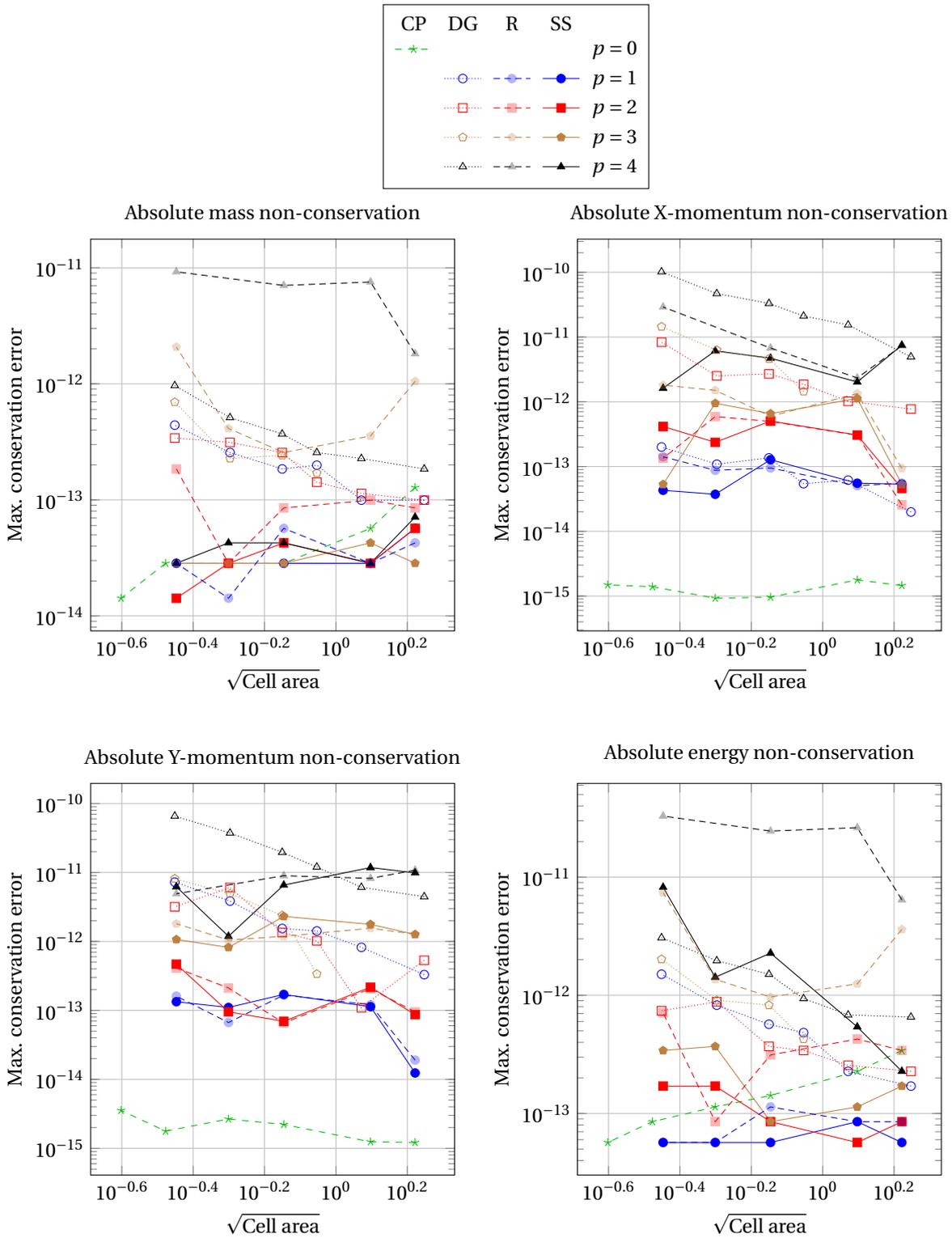


Figure 8.12: Conservation errors of primary variables for the regular (R) and skew-symmetric (SS) models with $\Delta t = 5 \cdot 10^{-2}$ and $\epsilon = 10^{-6}$, the DG model (DG) with $\Delta t = 1 \cdot 10^{-2}$ and Clawpack (CP) with $\Delta t = 5 \cdot 10^{-3}$; $(u_\infty, v_\infty) = (0, 0)$ and different solution degrees p were used

the regular model has significantly higher conservation errors than the skew-symmetric model, with the differences between both models rising as p is increased; the most extreme comparison is $\mathcal{O}(10^{-11})$ with $p = 4$ for the regular model and $\mathcal{O}(10^{-14})$ for the skew-symmetric model. Different values of ϵ do not seem to affect the conservation errors of the regular model whereas for the skew-symmetric model some differences

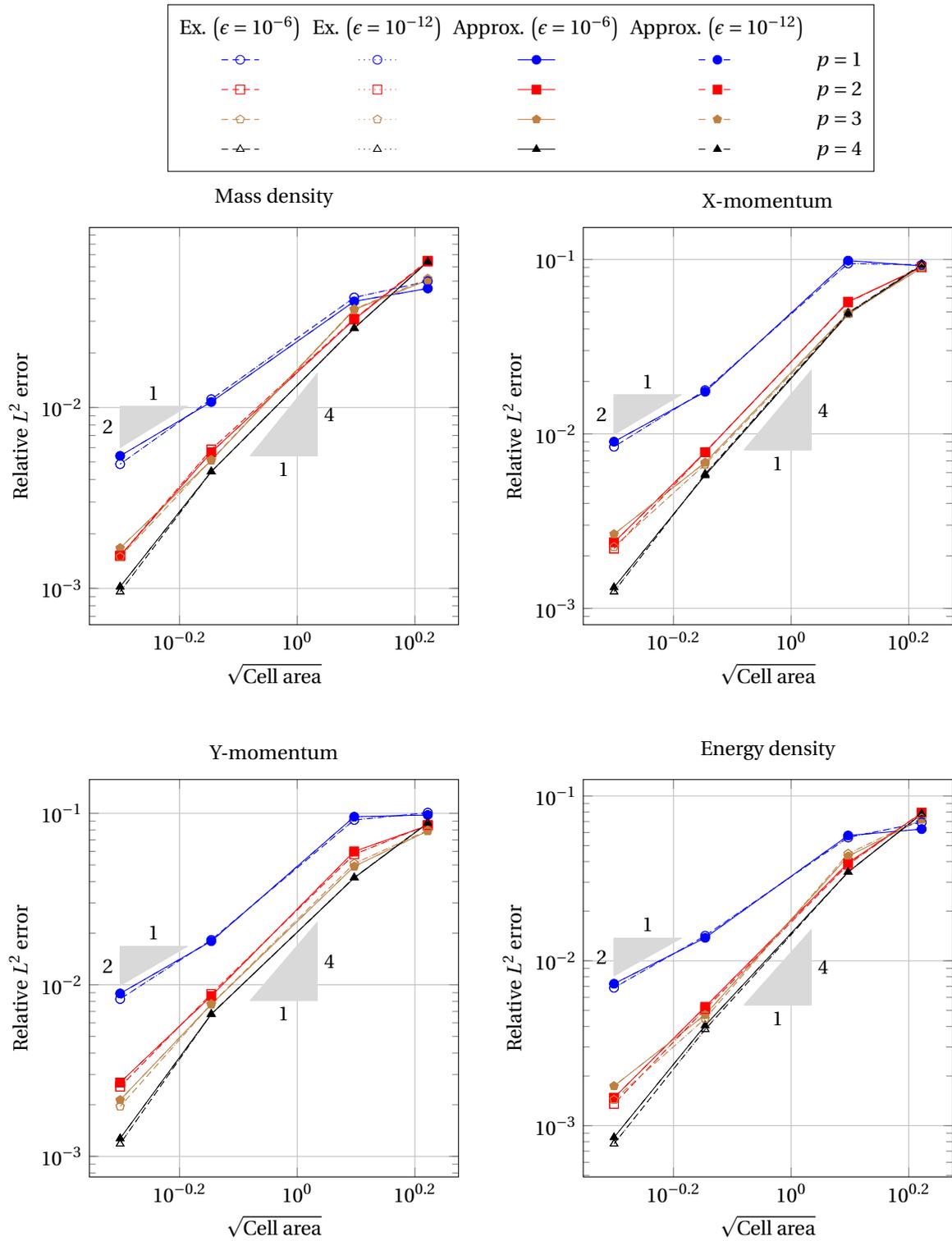


Figure 8.13: Convergence of the relative L^2 errors at $t = 10$ of the skew-symmetric model with exact (Ex.) and approximate (Approx.) kinetic energy modeling for $\Delta t = 2.5 \cdot 10^{-2}$, $(u_\infty, v_\infty) = (1, 1)$, different solution degrees p , $\epsilon = 10^{-6}$ and $\epsilon = 10^{-12}$

exist, likely due to round-off errors while iterating during a time step. For the other variables a different trend can be seen, although the overall conclusion is similar. Both models show increases in conservation errors as p is increased. Barring some individual points the regular model has higher conservation errors for all p considered.

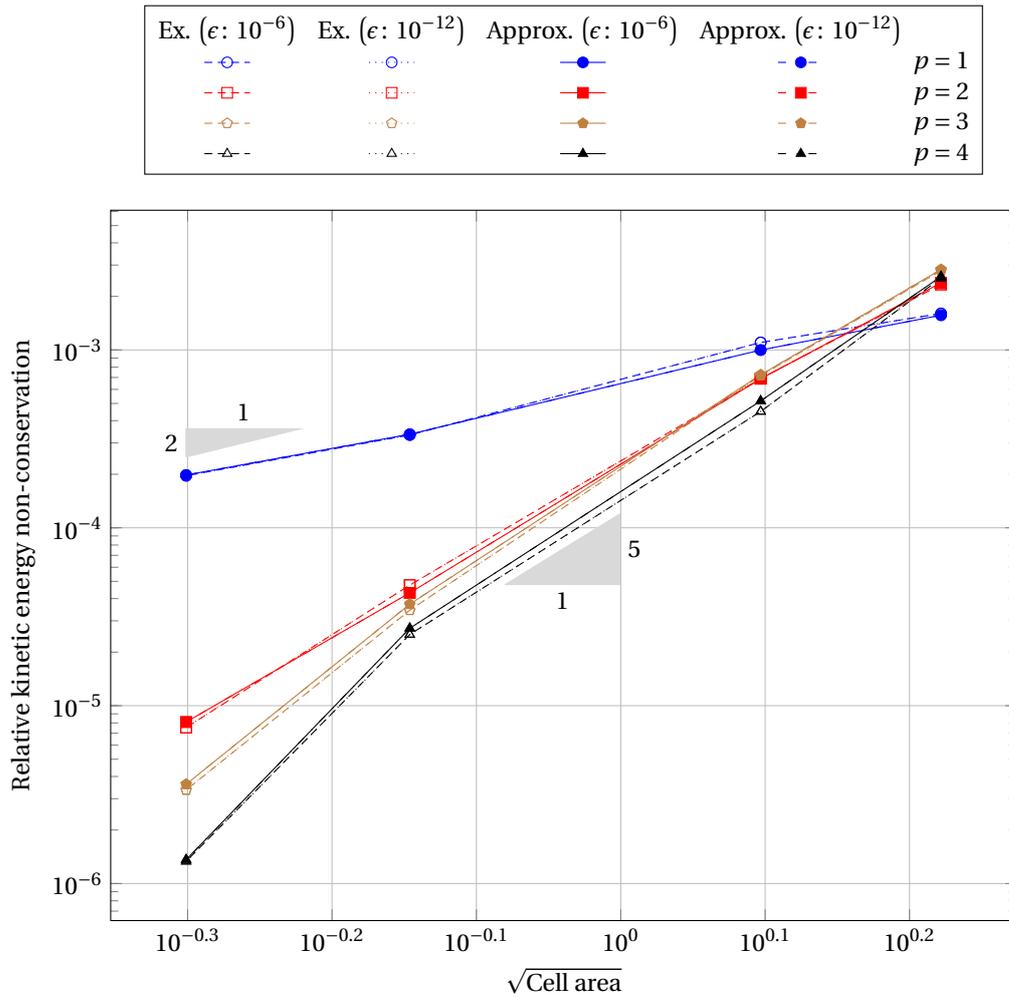


Figure 8.14: Maximum kinetic energy non-conservation errors of the skew-symmetric model with exact (Ex.) and approximate (Approx.) kinetic energy modeling for $\Delta t = 2.5 \cdot 10^{-2}$, $(u_\infty, v_\infty) = (1, 1)$, different solution degrees p , $\epsilon = 10^{-6}$ and $\epsilon = 10^{-12}$

Conclusions

Unlike what was found for the static vortex, the differences between the results of both methods are generally small. With this moving vortex monotonic L^2 convergence is seen for all parameters, while the L^2 errors of both methods are similar. The same can be said about the numerical diffusion, to a certain extent: On the coarser meshes shown the skew-symmetric method has a slight advantage over the regular method, but these differences become smaller as the mesh becomes finer. As was found with the static vortex the regular method is seemingly more sensitive to round-off errors than the skew-symmetric method.

8.5.3. Comparison with established methods

As with the static vortex the last comparison that will be made for the moving vortex is one between the methods presented in this thesis and the two external methods: Clawpack and the nodal DG method of Hesthaven & Warburton, both of which are covered in chapter 4. Recall that both external methods use some form of extra stabilization; this is also covered in the aforementioned chapter. The regular and skew-symmetric methods do not use any form of filtering, flux-limiting or additional stabilization.

Relative L^2 errors

First up is a comparison of the relative accuracy of all methods: Figure 8.19 shows the relative L^2 errors of the mass density, momentum components and total energy density. It is immediately apparent that Clawpack's results have much higher L^2 errors for all parameters shown over the entire range of cell sizes considered here. One can conclude that Clawpack needs a significantly finer mesh in order to resolve the moving vortex to the same levels of accuracy as the other methods. For $p = 1$ and $p = 2$ the regular and skew-symmetric

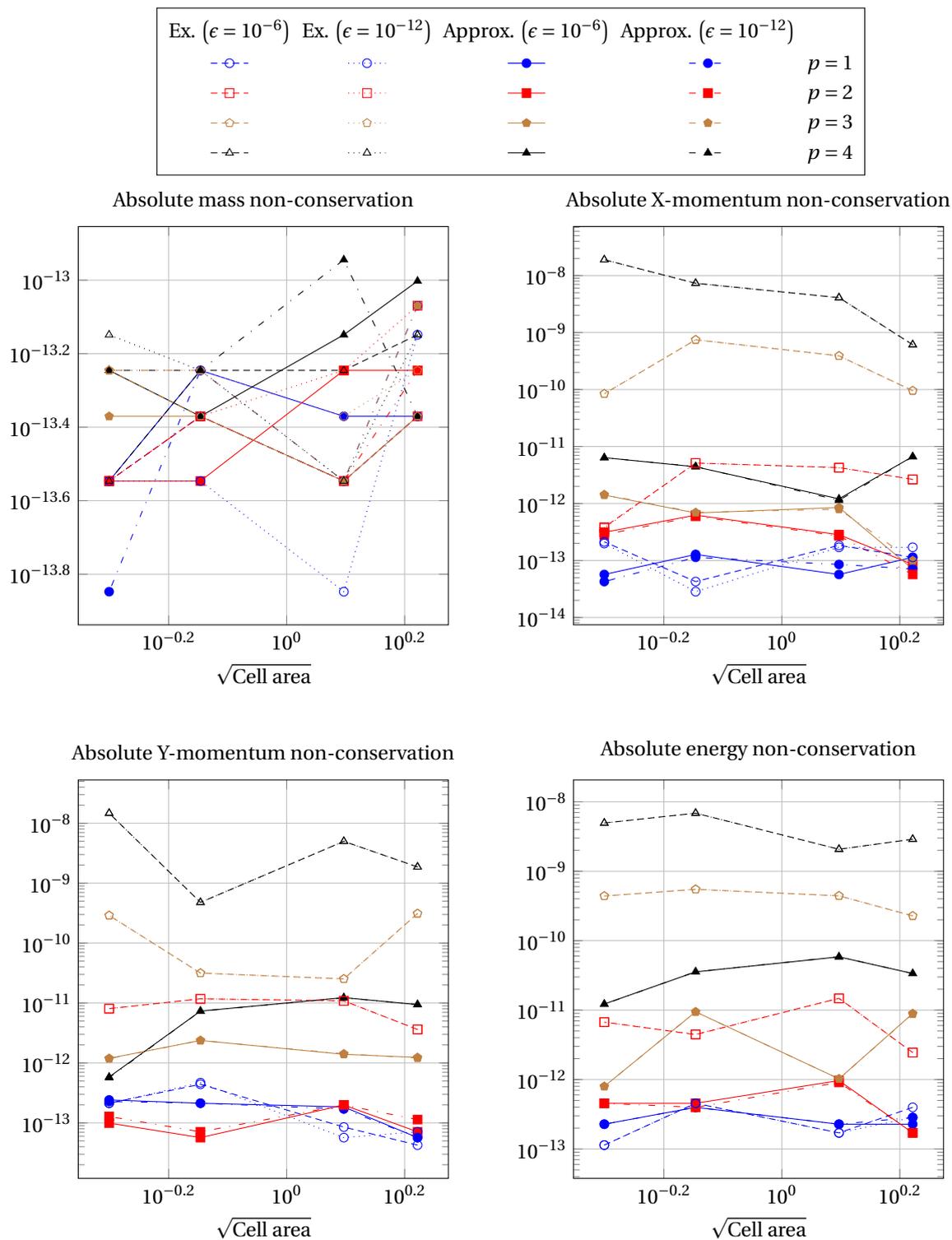


Figure 8.15: Maximum non-conservation of primary variables of the skew-symmetric model with exact (Ex.) and approximate (Approx.) kinetic energy modeling for $\Delta t = 2.5 \cdot 10^{-2}$, $(u_\infty, v_\infty) = (1, 1)$, different solution degrees p , $\epsilon = 10^{-6}$ and $\epsilon = 10^{-12}$

methods outperform the DG method on all but the coarsest meshes, whereas for $p = 3$ and $p = 4$ the DG method has slightly (for $p = 3$) or significantly (for $p = 4$) lower L^2 errors on most of the mesh size range shown in figure 8.19; only for the finest mesh shown the regular and skew-symmetric models have similar or better performance than the DG method, depending on the physical variable. With $p = 3$ energy density is the only parameter for which the DG method and the regular and skew-symmetric models have nearly-identical

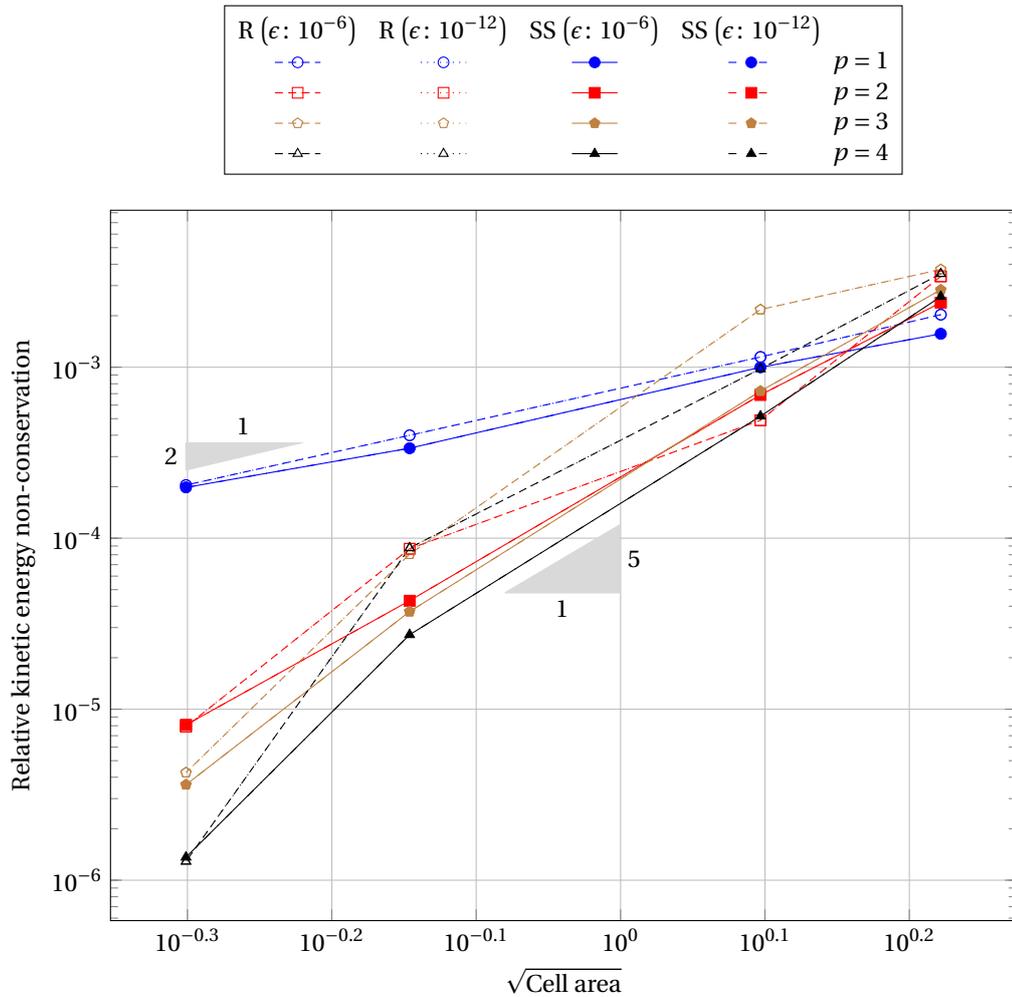


Figure 8.17: Maximum kinetic energy non-conservation errors of the regular (R) and skew-symmetric (SS) models for $\Delta t = 2.5 \cdot 10^{-2}$, $(u_\infty, v_\infty) = (1, 1)$, different solution degrees p , $\epsilon = 10^{-6}$ and $\epsilon = 10^{-12}$

other methods; not only does it have the highest kinetic energy non-conservation of all methods shown, its non-conservation error on the finest mesh is $\mathcal{O}(10^{-2})$. This is nearly an order of magnitude larger than the non-conservation errors of the other methods on their coarsest meshes. Whereas the numerical diffusion of the DG method with $p = 1$ initially increases as the mesh is refined it plateaus and converges again, reaching similar non-conservation errors on the finest mesh as were found on its coarsest mesh. In contrast, the regular and skew-symmetric methods both show (near)-constant second-order convergence for $p = 1$ with non-conservation errors that are at least an order of magnitude lower than those of the DG method. A similar comparison can be made for $p = 2$: In this case the DG method shows monotonic convergence, but its numerical diffusion is an order of magnitude larger than those of the regular and skew-symmetric methods while all methods have similar orders of convergence. Note that the results of the regular method with $p = 2$ show variations in the order of convergence.

Looking at $p = 3$ for these three methods it can be noted that they have similar performance levels throughout the mesh size range shown; which method comes out on top changes depending on the data point. For $p = 4$ it can be noted that the skew-symmetric method has a slight advantage over the DG method and the regular model for coarser meshes, whereas the DG method improves relative to the other two models as the mesh is refined. For the finest meshes the skew-symmetric model shows superlinear convergence and again (slightly) outperforms the other two methods, but the differences between all three methods at this point are small. Note that the skew-symmetric is the only method that displays somewhat-constant orders of convergence, with a clear difference between the left half (superlinear convergence) and right half (fifth-order convergence) of the mesh size range.

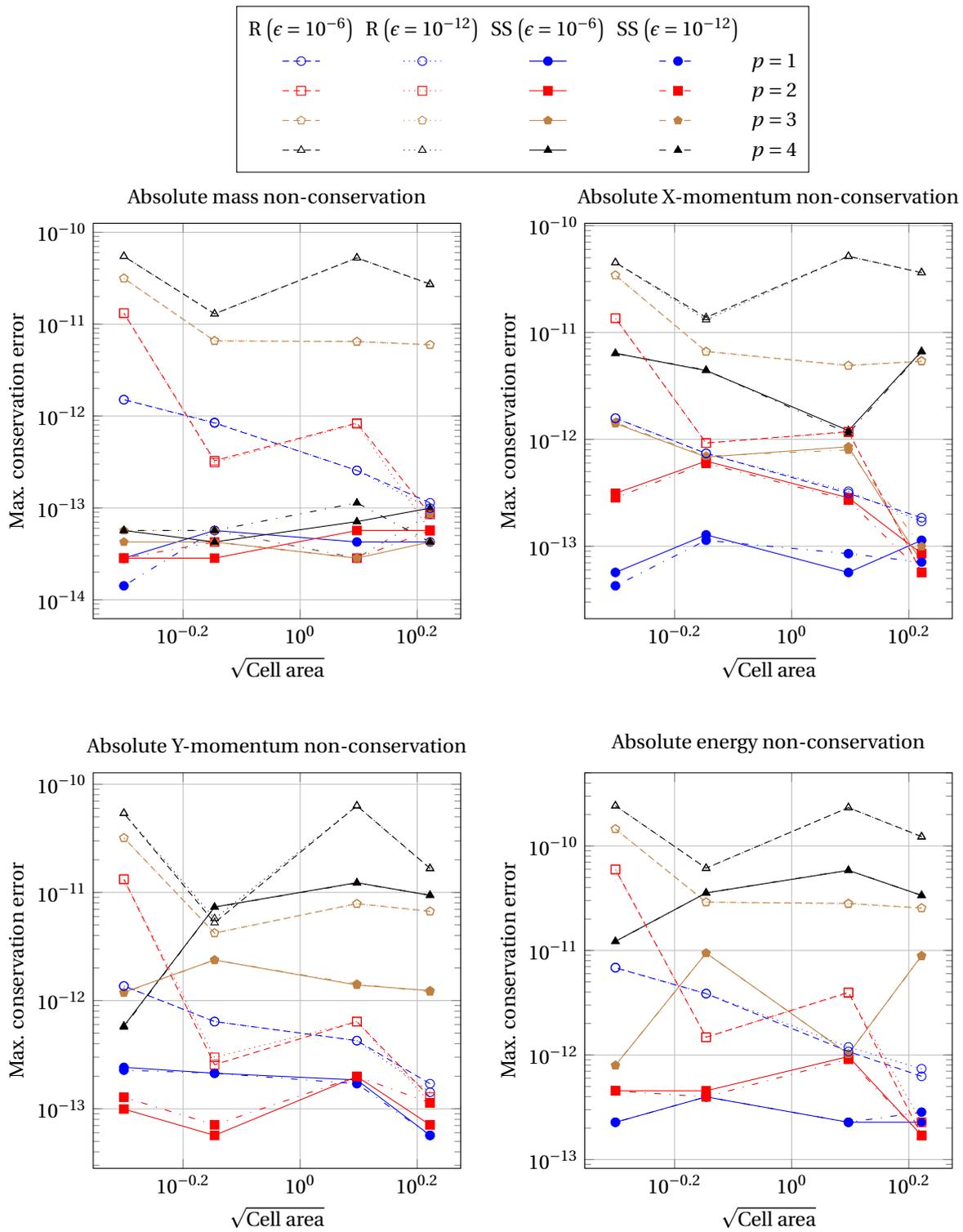


Figure 8.18: Conservation errors of primary variables of the regular (R) and skew-symmetric (SS) models for $\Delta t = 2.5 \cdot 10^{-2}$, $(u_\infty, v_\infty) = (1, 1)$, different solution degrees p , $\epsilon = 10^{-6}$ and $\epsilon = 10^{-12}$

These same relative kinetic energy conservation errors can be graphed against the number of degrees of freedom used, as is shown in figure 8.21. This places the results of figure 8.20 in a different light. Clawpack still performs the poorest out of all four methods. Whereas the relative performance of the regular and skew-symmetric methods have not changed (since they use the same meshes) it is apparent that the DG method

needs significantly more degrees of freedom in order to reduce the kinetic energy conservation errors to similar levels as those of the regular and skew-symmetric models. The differences between the DG model on the one hand and the regular and skew-symmetric models on the other hand grow as p is increased, due to the k -refinement happening in the DG model.

Primary non-conservation errors

Lastly the conservation errors of the primary physical parameters are compared: These are the same variables for which the L^2 errors were compared before. Figure 8.22 shows these conservation errors. We start by looking at mass non-conservation in the top left corner of figure 8.22. There is a clear divide between the methods considered in this work: On the one hand Clawpack and the skew-symmetric model have conservation errors of $\mathcal{O}(10^{-14})$ for all p (in the case of the skew-symmetric model), whereas the regular model and the DG method show higher conservation errors as p is increased. For the X- and Y-momenta the differences between the methods are smaller; although Clawpack still generally has the lowest conservation errors its performance is on par with the low-degree results of the skew-symmetric model. This model now also shows an increase in conservation errors when p is increased (in contrast to the mass conservation results), but the magnitudes of these conservation errors are generally smaller than those of the regular model and the DG method. Which of these two models performs best for any given degree varies per data point. Finally, looking at the energy conservation errors it can be seen that again Clawpack performs best, while for $p = 1$ and $p = 2$ the skew-symmetric model outperforms the regular model and the DG method. For $p = 3$ and $p = 4$ the regular model is outperformed by the DG method and the skew-symmetric model, which both seem to perform similarly.

Conclusions

Summarizing the conclusions drawn in this section, the following can be noted:

- Clawpack suffers from excessive dissipation, which affects its accuracy (high L^2 errors compared to the other methods) and its numerical dissipation (high kinetic energy non-conservation). The rate at which this is reduced as the mesh is refined is very low compared to the other methods as well. The only metric in which Clawpack outperforms the other methods is the magnitude of the conservation errors of primary variables, but since these are $\mathcal{O}(10^{-10})$ at most for the highest-degree solutions this is only a small advantage.
- Performance differences are found when comparing the DG method and the regular & skew-symmetric models against the square-root cell area. For $p = 1$ and $p = 2$ the regular and skew-symmetric models have somewhat lower L^2 errors while also having numerical dissipation that is approximately an order of magnitude smaller than that of the DG method. The performance differences with $p = 3$ and $p = 4$ are smaller, with all three models performing similarly. A different picture emerges when looking at their performances relative to the amount of degrees of freedom used: The DG method is outperformed by the regular and skew-symmetric models by nearly two orders of magnitude (for $p = 1$) to approximately four orders of magnitude (with $p = 4$) when it comes to numerical diffusion. For the highest-degree solutions considered here the conservation errors of the primary variables (as shown in figure 8.22) of the three methods are of similar levels; these are also the ones that are highest in magnitude ($\mathcal{O}(10^{-10})$). For $p = 1$ and $p = 2$ the skew-symmetric model seems to outperform the regular model and the DG method, but the magnitudes of the conservation errors in these case are smaller than the maximal values encountered (with $p = 4$).
- Of the four methods considered here the regular and skew-symmetric models appear to perform best; this is especially apparent when comparing the results of each method with respect to the number of degrees of freedom used. These are also the models that seem to be the most robust, as they do not use any filtering or stabilization steps. Whereas Clawpack does not strictly require any flux limiting if a first-order scheme is used, the results found when using this scheme were significantly worse, due to additional numerical dissipation. Whereas this generally stabilizes numerical methods it also results in a degradation of accuracy.

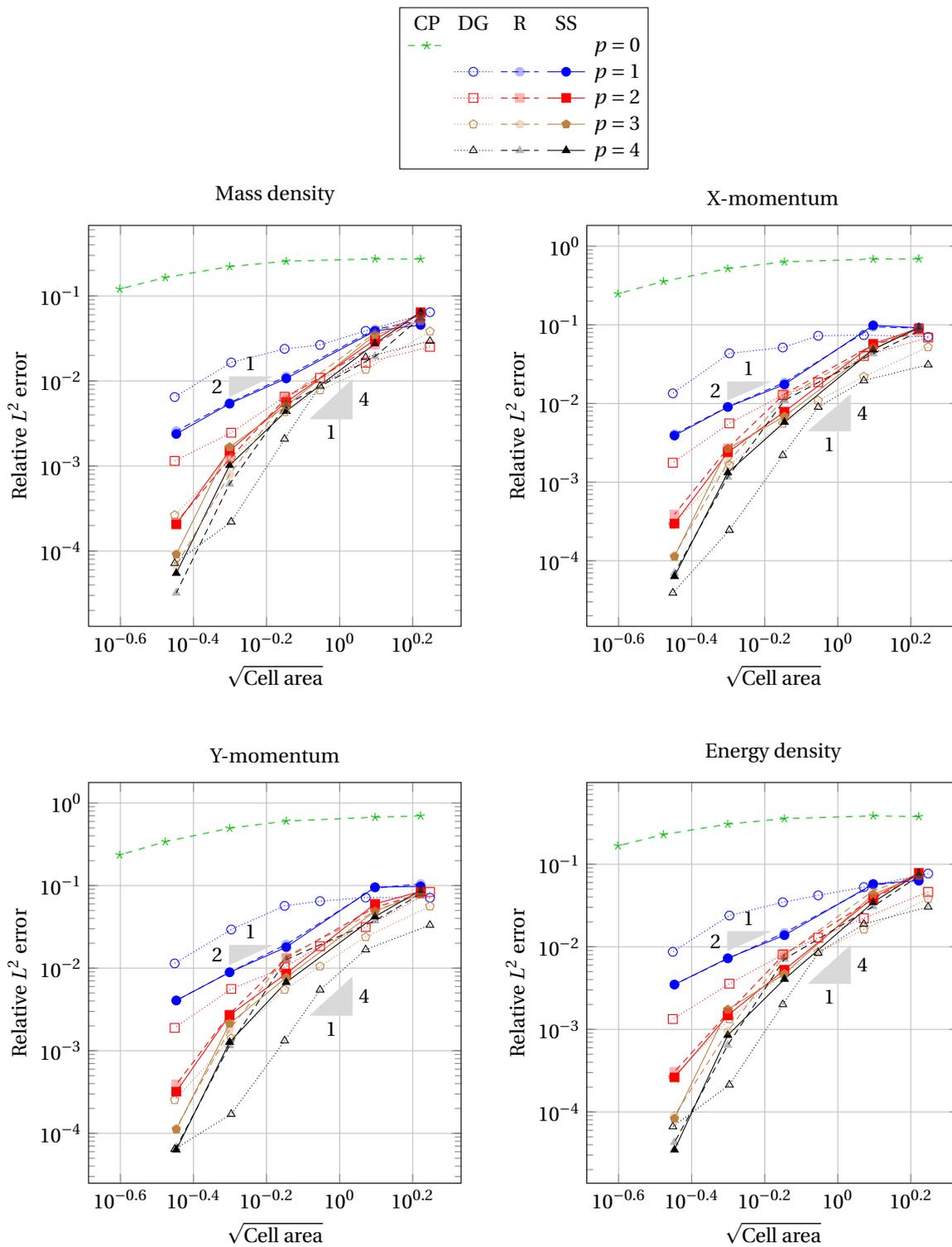


Figure 8.19: Convergence of the relative L^2 errors at $t = 10$ of the Roe variable model with $\epsilon = 10^{-6}$, $\Delta t = 2.5 \cdot 10^{-2}$ and the DG model with $\Delta t = 10^{-2}$, for $(u_\infty, v_\infty) = (1, 1)$ and different solution degrees p

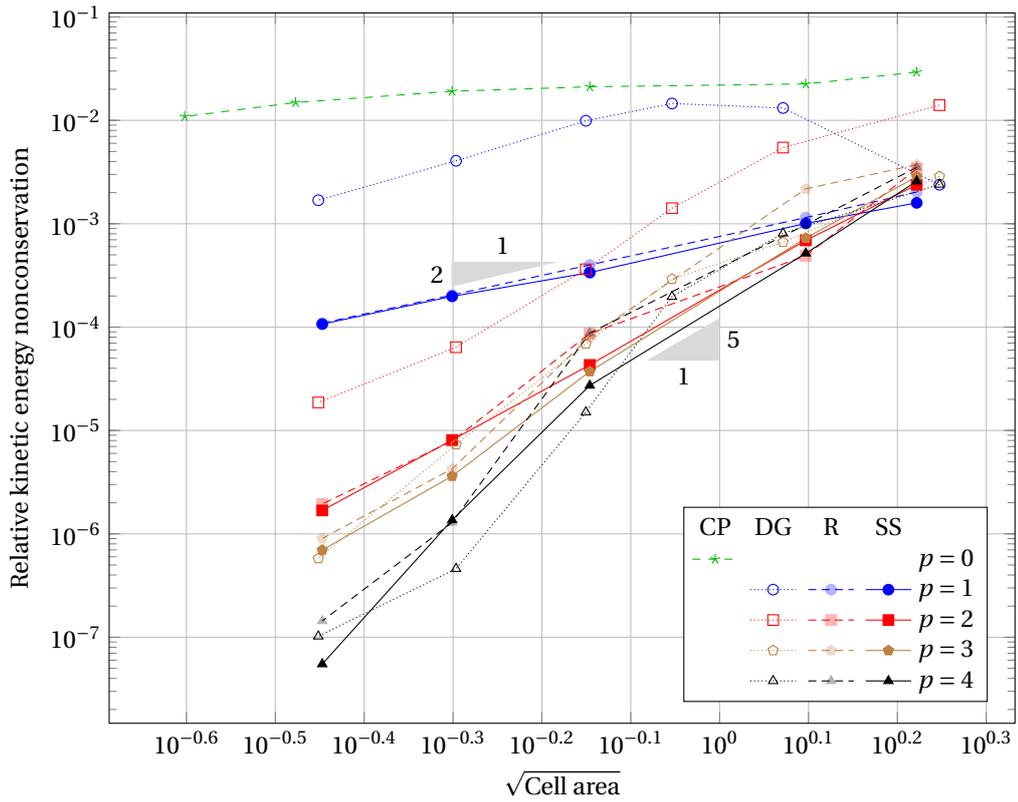


Figure 8.20: Kinetic energy non-conservation errors of different models for $(u_\infty, v_\infty) = (1, 1)$ and different solution degrees p , graphed against the square-root of the cell area; $\Delta t = 2.5 \cdot 10^{-2}$ for the skew-symmetric model, $\Delta t = 1 \cdot 10^{-2}$ for the DG model

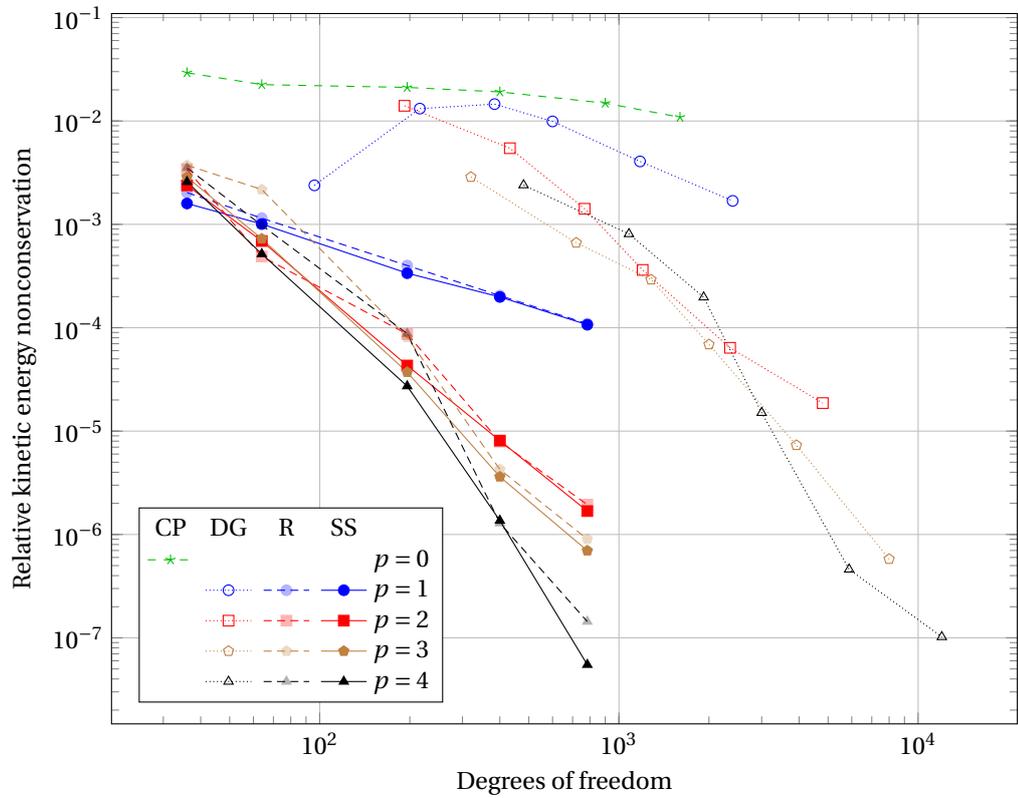


Figure 8.21: Kinetic energy non-conservation errors of different models for $(u_\infty, v_\infty) = (1, 1)$ and different solution degrees p , graphed against the number of degrees of freedom used; $\Delta t = 2.5 \cdot 10^{-2}$ for the skew-symmetric model, $\Delta t = 1 \cdot 10^{-2}$ for the DG model

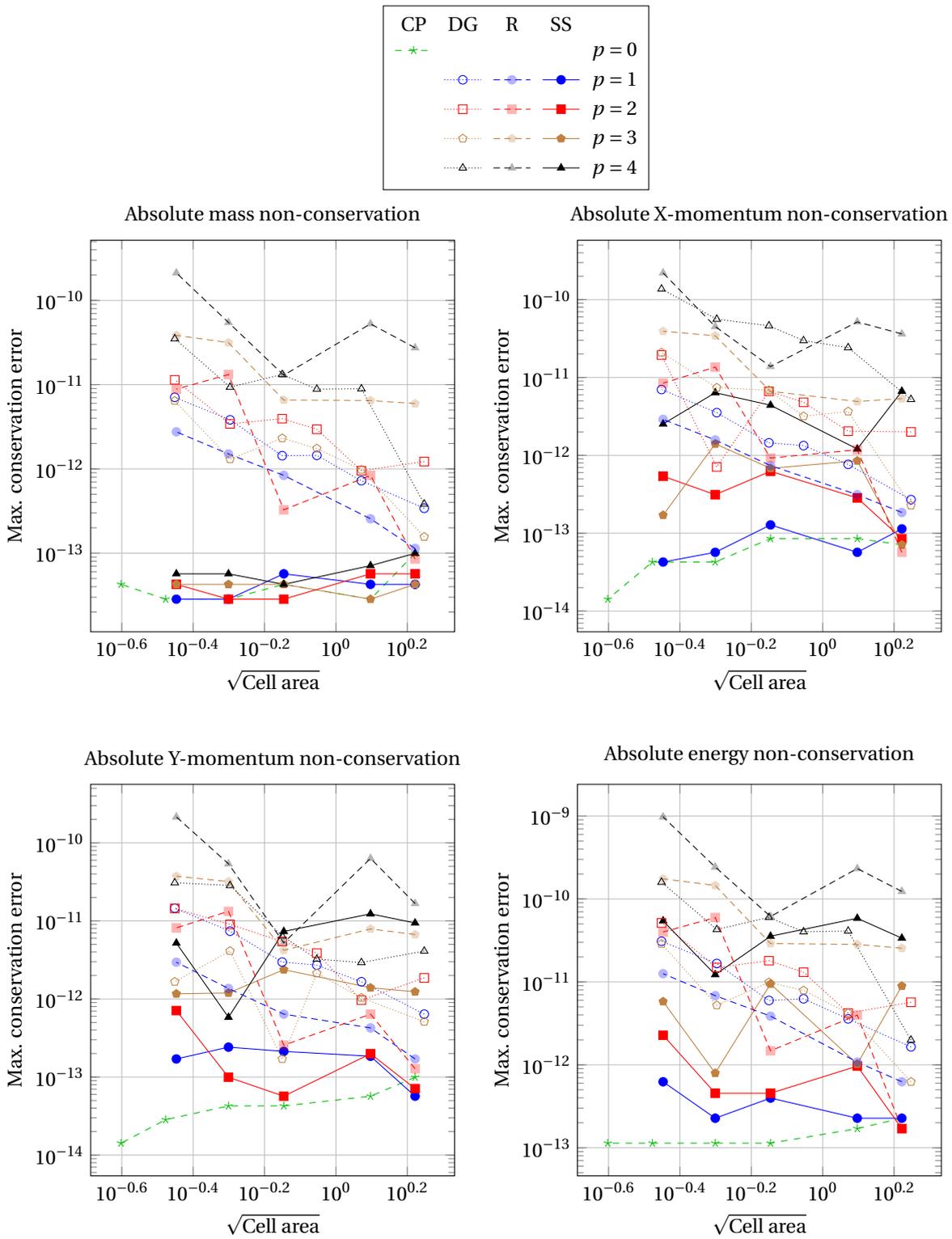


Figure 8.22: Conservation errors of primary variables for the regular (R) and skew-symmetric (SS) models with $\Delta t = 2.5 \cdot 10^{-2}$ and $\epsilon = 10^{-6}$, the DG model (DG) with $\Delta t = 1 \cdot 10^{-2}$ and Clawpack (CP) with $\Delta t = 5 \cdot 10^{-3}$; $(u_\infty, v_\infty) = (1, 1)$ and different solution degrees p were used

Conclusions

This thesis focuses on mimetic isogeometric methods for compressible Euler flows. Its aims were threefold:

- A new geometric model for sub- and supersonic compressible Euler flows has been posed.
- A partial decomposition into Roe variables was applied to this model.
- This geometric model and the model resulting from the partial Roe decomposition have been discretized with mimetic isogeometric methods, subjected to a range of test cases and compared to external methods.

The first chapters of this thesis introduce the relevant theory of mimetic isogeometric discretization methods. Isogeometric spline spaces can be chosen for differential forms such that the discrete exterior derivative is resolved exactly on any grid, independent of its shape or deformation. While the interior product was discretized in a metric-dependent way - through a weak form - the De Rham complex of the continuous differential form spaces is retained on the discrete level.

Chapter 3 covered the difficulties behind simultaneous conservation of momentum and kinetic energy in simulations of (in)compressible Euler flows and covered some existing approaches for this. Whereas suitable ways of achieving this exist for incompressible flow models, they all exploit the divergence-free nature of admissible solutions. No methods currently exist that simultaneously conserve mass, momentum and kinetic energy in compressible Euler flows; hence an attempt is made in this research to make a step towards this. A geometric model for compressible Euler flows has been posed, based on observations made in literature. This model resolves the energy density, pressure and mass density as volume forms while momentum is modeled as a covector-valued volume form; it is referred to as the 'regular' model.

Advection is a phenomenon that by its nature does not produce any mass, momentum or energy; it merely redistributes field densities. In an attempt to retain this structure in discrete settings a partial Roe variable decomposition was applied to the posed geometric Euler model. This approach has seen (limited) use in literature for finite volume methods, but had not been applied in any way to mimetic finite element methods yet. The idea behind applying this decomposition is to construct discrete advection operators that simultaneously conserve mass, momentum, kinetic energy, internal energy and total energy for compressible Euler flows in any number of dimensions. Said decomposition leads to advection operators that are self-adjoint when used in a weak form. Whereas this construction leads to advection operators that contain the Hodge \star in various ways, algebraic manipulation of said weak forms results in a (self-adjoint) combination of L^2 inner products. Discretizing the weak forms of these self-adjoint advection operators results in skew-symmetric matrices, which conserve squared variables exactly in discrete settings. The power of the partial Roe decomposition is then that mass, momentum and kinetic energy are all products of Roe variables. As long as the continuity and momentum equations of the Euler model are discretized with the same advection operators, these will then conserve mass, momentum and kinetic energy simultaneously. This requires mass density and momentum to be expanded in the same bases, and a Bubnov-Galerkin finite element approach to be used.

As was found during initial numerical tests, formulating advection operators in this self-adjoint/skew-symmetric way does not affect the numerical results in a negative way: During initial tests with one-dimensional linear advection and Burgers' equation on periodic domains the solutions and L^2 errors of both the normal and skew-symmetric advection were largely similar. Only in chapter 7 was it found that the discrete (self-adjoint/skew-symmetric) Roe variable model gave wrong shock speed predictions in Sod's shock tube; this is likely due to an interaction effect in the Roe decomposed Euler model, since the shock speeds were resolved correctly and in line with exact solutions when testing with Burgers' equation in chapter 6. This is further reinforced by the results of the discretized version of the regular model: It correctly resolved the locations of all shocks. The solutions of the regular and decomposed models for this one-dimensional Euler test were

extremely oscillatory. Despite this no stability issues were encountered and no damping or filtering of the numerical solution was necessary. Applying a simple moving average-filter as post-processing step removed most of the unphysical oscillations. Some oscillations remained in the region between both discontinuities in the exact solution. This suggests that a more sophisticated post-processing method could be used to satisfactorily improve solution quality.

Compared to the finite volume package Clawpack and a nodal Discontinuous Galerkin (DG) method the regular and decomposed models have decent performance on coarse meshes in terms of numerical conservation errors relative to the exact solution. Especially the DG method suffers from large mass and energy conservation errors, whereas Clawpack and the two methods developed in this research do not. Furthermore, Clawpack and the DG method both require flux limiting in order to achieve sufficient stability: Without flux limiters no solutions were obtained for Sod's shock tube for either method, suggesting that the regular and decomposed models have superior stability properties. While no hard data has been gathered on this, it is suggested that the lack of monotonically-decreasing kinetic energy and internal energy conservation error on finer meshes is due to the existence of the aforementioned oscillations, which increase in frequency as the mesh is refined.

The last test cases covered in this report are two-dimensional isentropic vortices on a twice-periodic domain. Both the regular geometric Euler model and its Roe variable decomposition generally outperform the reference methods for both stationary and moving vortices. Only for even-degree ($p = 2$, $p = 4$) solutions with the stationary vortex case do the models presented in this work have some issues with non-monotonic convergence - the regular model is affected by this more than the decomposed model. Clawpack displays overly diffusive behavior, resulting in large L^2 errors and high amounts of numerical diffusion. While the L^2 errors of the DG method are comparable to those of the two models developed in this work for all basis function orders considered, it requires significantly more degrees of freedom to attain this. Since the amount of degrees of freedom used by the DG method scales with the maximum degree of its basis functions this effect is especially pronounced for higher-degree solutions: To obtain similar levels of numerical diffusion the DG method needs up to ten times as many degrees of freedom as the methods presented in the current research.

All in all, both the regular Euler model and its Roe variable decomposition perform well versus Clawpack and the nodal DG method used as reference methods. While oscillatory solutions are obtained for Sod's shock tube it is to be remarked that the regular Euler model and its Roe variable decomposition are the only models that do not use any form of damping or flux limiting. They thus seemingly have superior stability properties over Clawpack and the DG method. For the two-dimensional isentropic vortex the methods developed in this work both outperform the reference methods, especially when comparing them with respect to their numbers of degrees of freedom. The performance of the regular Euler model and the Roe variable decomposition are similar in most aspects. Standout differences between them are the inaccurate prediction of shock advection speed by the decomposed model (while the regular model's prediction was accurate) and the increased sensitivity to stability issues and round-off error of the regular model.

Recommendations for further research

Over the course of this research certain choices had to be made and various avenues have been left unexplored due to time constraints. Several suggestions can be made for possible improvements or alternative approaches to improve upon the work done in this thesis:

- In the current formulation of the Roe decomposed model all linear terms have been resolved implicitly. It could be beneficial if instead the pressure terms of the momentum and energy equations would be linearized in an explicit way. This turns the Picard-linearized model into a system of coupled advection equations with forcing terms on the right-hand side. The advection terms have already been made to conserve mass, momentum, kinetic energy, internal energy and total energy; perhaps these right-hand-side forcing terms can be constructed in such a way that the total system conserves these quantities as well.
- Alternatively a time-staggered dual system of equations can be set up to march the current linear system in time. By staggering the two systems of equations with respect to one another the half-time step solutions can be used to march the solution of each system in an alternating way. Palha & Gerritsma [41] have already applied this to a mimetic discretization of incompressible flow.
- Both the regular and Roe decomposed models displayed excessive dispersive behavior for Sod's shock tube. This can be dealt with in two ways: Either a damping term needs to be added to the model (this is not desirable since it introduces an unphysical influence) or a post-processing step can be added to improve solution quality without affecting the algorithms themselves. For the former approach the recent work on the Variational Multiscale method and Discontinuity Capturing by Ten Eikelder et al. is suggested [52]; this provides a way of damping oscillations near discontinuities that is grounded in approximation theory. For the latter the work of Gelb & Tadmor [18, 19] or similar approaches are suggested.
- A recent approach by Jain et al. [29] is to discretize the Hodge \star operator as the (inverses of) mass matrices $\mathbb{M}^{(k)}$. Doing this results in increased sparsity of the linear systems and gives a way of explicitly defining the dual De Rham complex; this structure and the possibilities it gives could be exploited for discretizing certain operators (such as the interior product) in a metric-free way.
- Some works suggest using the covariant derivative to describe advection instead of the Lie derivative. These include Gilbert & Vanneste [20], Kreeft [31] and Boyland [9]. By using linear mappings to orthogonal spaces the covariant derivative could be pulled back such that the Christoffel symbols become trivial, foregoing the need to calculate them. This would circumvent one of the big challenges to discretizing the covariant derivative.

Bibliography

- [1] A. Back and E. Sonnendrücker. Spline discrete differential forms. application to maxwell's equations. Technical report, IRMA, CNRS and Université de Strasbourg, 2011.
- [2] A. Abbà and L. Bonaventura. A mimetic finite difference discretization for the incompressible Navier–Stokes equations. *International Journal for Numerical Methods in Fluids*, 56(8):1101–1106, March 2008.
- [3] V. Arnold. Sur la géométrie différentielle des groupes de Lie de dimension infinie et ses applications à l'hydrodynamique des fluides parfaits. *Annales de l'institut Fourier*, 16(1):319–361, 1966.
- [4] V.I. Arnold and B.A. Khesin. *Topological Methods in Hydrodynamics*. Number 125 in Applied Mathematical Sciences. Springer-Verlag New York, Inc, 1998. ISBN 0-387-94947-X.
- [5] A. Back and E. Sonnendrücker. Finite Element Hodge for spline discrete differential forms. Application to the Vlasov–Poisson system. *Applied Numerical Mathematics*, 79:124–136, May 2014. ISSN 01689274.
- [6] T.M. Bendall, T.H. Gibson, J. Shipton, C.J. Cotter, and B. Shipway. A compatible finite-element discretisation for the moist compressible Euler equations. *Quarterly Journal of the Royal Meteorological Society*, pages 1–19, July 2020.
- [7] P.B. Bochev and J.M. Hyman. Principles of mimetic discretizations of differential operators. In D.N. Arnold, P.B. Bochev, R.B. Lehoucq, R.A. Nicolaides, and M. Shashkov, editors, *Compatible Spatial Discretizations*, volume 142 of *IMA Volumes in Mathematics and its Applications*. Springer Science+Business Media, LLC, 2006.
- [8] D. Boffi, F. Brezzi, and M. Fortin. *Mixed Finite Element Methods and Applications*, volume 44 of *Springer Series in Computational Mathematics*. Springer-Verlag Berlin Heidelberg, 2013.
- [9] P. Boyland. Fluid mechanics and mathematical structures. In R. L. Ricca, editor, *An Introduction to the Geometry and Topology of Fluid Flows*, pages 105–134. Springer Netherlands, Dordrecht, 2001. ISBN 978-94-010-0446-6. doi: 10.1007/978-94-010-0446-6_6. URL https://doi.org/10.1007/978-94-010-0446-6_6.
- [10] A. Buffa and G. Sangalli, editors. *IsoGeometric Analysis: A New Paradigm in the Numerical Approximation of PDEs*, volume 2161 of *Lecture Notes in Mathematics*. Springer International Publishing, 2016. ISBN 978-3-319-42308-1 978-3-319-42309-8.
- [11] A. Buffa, G. Sangalli, and R. Vázquez. Isogeometric methods for computational electromagnetics: B-spline and T-spline discretizations. *Journal of Computational Physics*, 257:1291–1320, January 2014.
- [12] S. Charnyi, T. Heister, M. A. Olshanskii, and L. G. Rebholz. On conservation laws of Navier–Stokes Galerkin discretizations. *Journal of Computational Physics*, 337:289–308, May 2017.
- [13] J.A. Cottrell, T.J.R. Hughes, and Y. Bazilevs. *Isogeometric Analysis: Toward Integration of CAD and FEA*. John Wiley & Sons, Ltd, September 2009. ISBN 978-0-470-74873-2.
- [14] J.A. Evans and T.J.R. Hughes. Isogeometric Compatible Discretizations for Viscous Incompressible Flow. In A. Buffa and G. Sangalli, editors, *IsoGeometric Analysis: A New Paradigm in the Numerical Approximation of PDEs*, volume 2161 of *Lecture Notes in Mathematics*, pages 155–193. Springer International Publishing, Pavia, Italy, 2016. ISBN 978-3-319-42308-1 978-3-319-42309-8.
- [15] T. Frankel. *The Geometry of Physics: An Introduction*. Cambridge University Press, second edition, 2004.
- [16] E.S. Gawlik and F. Gay-Balmaz. A Variational Finite Element Discretization of Compressible Flow. *Foundations of Computational Mathematics*, 21:961–1001, 2021.

- [17] E.S. Gawlik, P. Mullen, D. Pavlov, J.E. Marsden, and M. Desbrun. Geometric, variational discretization of continuum theories. *Physica D: Nonlinear Phenomena*, 240(21):1724–1760, October 2011. ISSN 01672789. doi: 10.1016/j.physd.2011.07.011.
- [18] A. Gelb and E. Tadmor. Detection of Edges in Spectral Data. *Applied and Computational Harmonic Analysis*, 7(1):101–135, 1999.
- [19] A. Gelb and E. Tadmor. Enhanced spectral viscosity approximations for conservation laws. *Applied Numerical Mathematics*, 33(1-4):3–21, May 2000.
- [20] A. D. Gilbert and J. Vanneste. Geometric generalised Lagrangian-mean theories. *Journal of Fluid Mechanics*, 839:95–134, March 2018. doi: 10.1017/jfm.2017.913.
- [21] S. Gottlieb, C.W. Shu, and E. Tadmor. Strong Stability-Preserving High-Order Time Discretization Methods. *SIAM Review*, 43(1):89–112, January 2001.
- [22] A.M. Harten, P.D. Lax, and B. van Leer. On Upstream Differencing and Godunov-Type Schemes for Hyperbolic Conservation Laws. *SIAM Review*, 25(1):35–61, January 1983.
- [23] Hatcher, A. *Algebraic Topology*. Cambridge University Press, 2002.
- [24] J. S. Hesthaven and T. Warburton. *Nodal Discontinuous Galerkin Methods - Algorithms, Analysis, and Applications*. Number 54 in Texts in Applied Mathematics. Springer Science+Business Media, LLC, first edition, 2008. ISBN 978-0-387-72067-8.
- [25] R.R. Hiemstra, R.H.M. Huijsmans, and M.I. Gerritsma. High order gradient, curl and divergence conforming spaces, with an application to NURBS-based IsoGeometric Analysis. *arXiv e-prints*, Sep 2012.
- [26] A. N. Hirani. *Discrete Exterior Calculus*. Doctoral, California Institute of Technology, Pasadena, California, May 2003.
- [27] T.J.R. Hughes, J.A. Cottrell, and Y. Bazilevs. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Computer Methods in Applied Mechanics and Engineering*, 194(39-41):4135–4195, October 2005. ISSN 00457825.
- [28] J.M. Hyman and J.C. Scovel. Deriving Mimetic Difference Approximations To Differential Operators Using Algebraic Topology. Technical report, Los Alamos National Laboratory, Los Alamos, 1988.
- [29] V. Jain, Y. Zhang, A. Palha, and M. Gerritsma. Construction and application of algebraic dual polynomial representations for finite element methods on quadrilateral and hexahedral meshes. *Computers & Mathematics with Applications*, 95:101–142, August 2021.
- [30] S. Janssen. Structure-Preserving Isogeometric Analysis for Computational Fluid Dynamics. Master’s thesis, Delft University of Technology, Delft, November 2016.
- [31] J. J. Kreeft. *Mimetic Spectral Element Method: A discretization of Geometry and Physics*. PhD thesis, Delft University of Technology, October 2013.
- [32] R. J. LeVeque. *Numerical methods for conservation laws*. Lectures in mathematics ETH Zürich. Birkhäuser Verlag, 2nd ed edition, 1992. ISBN 978-3-7643-2723-1 978-0-8176-2723-2.
- [33] R.J. Leveque. *Finite-volume methods for hyperbolic problems*. Cambridge University Press, first edition, 2002.
- [34] K. T. Mandli, A. J. Ahmadi, M. Berger, D. Calhoun, D. L. George, Y. Hadjimichael, D. I. Ketcheson, G. I. Lemoine, and R. J. LeVeque. Clawpack: building an open source ecosystem for solving hyperbolic pdes. *PeerJ Computer Science*, 2:e68, 2016. doi: 10.7717/peerj-cs.68.
- [35] Y. Morinishi. Skew-symmetric form of convective terms and fully conservative finite difference schemes for variable density low-Mach number flows. *Journal of Computational Physics*, page 25, 2010.
- [36] K. Mørken. Some identities for products and degree raising of splines. *Constructive Approximation*, 7(1):195–208, December 1991. ISSN 0176-4276, 1432-0940. doi: 10.1007/BF01888153.

- [37] M. Nakahara. *Geometry, Topology and Physics*. Graduate Student Series in Physics. Institute of Physics Publishing Ltd., second edition, 2003.
- [38] C. Nash and S. Sen. *Topology and Geometry for Physicists*. Dover Books on Mathematics. Dover Publications, Inc., dover edition, 2011.
- [39] A. Natale and C.J. Cotter. Scale-selective dissipation in energy-conserving finite element schemes for two-dimensional turbulence. *Quarterly Journal of the Royal Meteorological Society*, 143(705):1734–1745, April 2017.
- [40] G.T. Oud, D.R. van der Heul, C. Vuik, and R.A.W.M. Henkes. A fully conservative mimetic discretization of the Navier–Stokes equations in cylindrical coordinates with associated singularity treatment. *Journal of Computational Physics*, 325:314–337, November 2016.
- [41] A. Palha and M. Gerritsma. A mass, energy, enstrophy and vorticity conserving (MEEVC) mimetic spectral element discretization for the 2D incompressible Navier-Stokes equations. *Journal of Computational Physics*, 328:200–220, January 2017.
- [42] D. Pavlov, P. Mullen, Y. Tong, E. Kanso, J.E. Marsden, and M. Desbrun. Structure-preserving discretization of incompressible fluids. *Physica D: Nonlinear Phenomena*, 240(6):443–458, March 2011.
- [43] P. Pettersson, G. Iaccarino, and J. Nordström. A stochastic Galerkin method for the Euler equations with Roe variable transformation. *Journal of Computational Physics*, 257:481–500, January 2014.
- [44] N.A. Phillips. An Example of Non-Linear Computational Instability. *The Atmosphere and the Sea in Motion*, pages 501–504, 1959.
- [45] L. Piegl and W. Tiller. *The NURBS Book*. Monographs in Visual Communication. Springer-Verlag Berlin Heidelberg, second edition, 1997.
- [46] P.L. Roe. Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes. *Journal of Computational Physics*, 43:357–372, 1981.
- [47] W. Rozema. *Low-dissipation methods and models for the simulation of turbulent subsonic flow: theory and applications*. PhD thesis, University of Groningen, 2015. ISBN: 9789036781886 OCLC: 922569558.
- [48] W. Rozema, R. W. C. P. Verstappen, A. E. P. Veldman, and J. C. Kok. Low-Dissipation Simulation Methods and Models for Turbulent Subsonic Flow. *Archives of Computational Methods in Engineering*, 27(1): 299–330, January 2020. ISSN 1134-3060, 1886-1784. doi: 10.1007/s11831-018-09307-7.
- [49] B. Sanderse. Energy-conserving Runge–Kutta methods for the incompressible Navier–Stokes equations. *Journal of Computational Physics*, 233:100–131, January 2013.
- [50] G. A. Sod. A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws. *Journal of Computational Physics*, 27(1):1–31, April 1978.
- [51] S. C. Spiegel, H. T. Huynh, and J. R. DeBonis. A Survey of the Isentropic Euler Vortex Problem using High-Order Methods. In *22nd AIAA Computational Fluid Dynamics Conference*, Dallas, TX, June 2015. American Institute of Aeronautics and Astronautics. ISBN 978-1-62410-366-7.
- [52] M.F.P. ten Eikelder, Y. Bazilevs, and I. Akkerman. A theoretical framework for discontinuity capturing: Joining variational multiscale analysis and variation entropy theory. *Computer Methods in Applied Mechanics and Engineering*, 359:1–29, February 2020. ISSN 00457825. doi: 10.1016/j.cma.2019.112664.
- [53] E. Tonti. *The Mathematical Structure of Classical and Relativistic Physics: A General Classification Diagram*. Modeling and Simulation in Science, Engineering and Technology. Birkhäuser, 2013. ISBN 978-1-4614-7421-0.
- [54] E. F. Toro, M. Spruce, and W. Speares. Restoration of the contact surface in the HLL-Riemann solver. *Shock Waves*, 4(1):25–34, July 1994. ISSN 0938-1287, 1432-2153.
- [55] D. Toshniwal. A Geometric Approach Towards Momentum Conservation. master thesis, Delft University of Technology, 2012.

- [56] D. Toshniwal, R. H. M. Huijsmans, and M. I. Gerritsma. A Geometric Approach Towards Momentum Conservation. In Mejdî Azaïez, Henda El Fekih, and Jan S. Hesthaven, editors, *Spectral and High Order Methods for Partial Differential Equations - ICOSAHOM 2012*, Lecture Notes in Computational Science and Engineering, pages 393–402. Springer International Publishing, 2014. ISBN 978-3-319-01601-6. doi: 10.1007/978-3-319-01601-6_32.
- [57] J. Van Kan, A. Segal, and F. Vermolen. *Numerical Methods in Scientific Computing*. Delft Academic Press, Delft, second edition, 2014. ISBN 978-90-6562-363-8.
- [58] A. Yavari. On geometric discretization of elasticity. *Journal of Mathematical Physics*, 49(2):022901, February 2008.
- [59] H.C Yee, N.D Sandham, and M.J Djomehri. Low-Dissipative High-Order Shock-Capturing Methods Using Characteristic-Based Filters. *Journal of Computational Physics*, 150(1):199–238, March 1999.