The Lagrangian Mimetic Spectral Element Method

Solving (non-)Linear Advection Problems with a Mimetic Method

Joey Dekker

The Lagrangian Mimetic Spectral Element Method

Solving (non-)Linear Advection Problems with a Mimetic Method

by

Joey Dekker

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Thursday December 22, 2022 at 2:00 PM.

Student number:	4538099	
Project duration:	June 16, 2021 – December 22, 2022	
Thesis committee:	Dr. M. I. Gerritsma,	TU Delft AE, supervisor
	Dr. B. Chen,	TU Delft AE
	Dr. I. Akkerman,	TU Delft 3ME
	Dr. M. Möller,	TU Delft EEMCS

Cover Image: Stock Photo from Jennylynn Fields/Shutterstock

Preface

In this report, I describe various aspects of my year-and-a-half-long journey through the world of mimetic methods. While numerical methods may be considered a branch of mathematics, one will find plenty of physics in this report as well. This unique opportunity of keeping physics in a numerical method is what drew me to this topic all this time ago and makes this topic interesting still.

My journey would not have been possible without the support of my family and friends, who have supported me by simply being there. I would also like to thank the members of my thesis committee for reading and assessing my work.

Last but not least, a special thank you goes out to Marc Gerritsma, without whom this project simply would not exist.

Joey Dekker
Delft, December 2022

Contents

List of Figures

List of Tables

List of Algorithms

List of Symbols and Abbreviations

Symbols

γ	Ratio of specific heats
λ	Lagrange multiplier
$\mathbb{E}$	Incidence matrix
$\mathbb{H}$	Hodge matrix
$\mathbb{M}$	Mass matrix
$\mathbb{R}^n$	n-dimensional Euclidean space
∂_t	Temporal derivative
Φ	Mapping
ϕ	Advected quantity
π	Momentum
ρ	Density (of mass or tracers)
$\star$	Hodge operator
τ	Reference temporal coordinate
Θ	Volume dilatation
ξ	Reference spatial coordinate
d	Exterior derivative
e	Edge polynomial
F	Deformation gradient
h	Lagrange polynomial
J	Jacobian matrix or determinant
K_s	Isentropic bulk modulus
K_t	Number of time steps
K_x	Number of spatial elements
L	Velocity gradient tensor
m	Mass
p	Pressure
p_t	Polynomial degree in time
p_x	Polynomial degree in space
S	Shock potential

t	Temporal coordinate
u	Velocity
V	Volume
X	Moving spatial coordinate/position
x	Spatial coordinate

Abbreviations

ALE	Arbitrary Lagrangian Eulerian
BC	Boundary Condition
CFD	Computational Fluid Dynamics
CFL	Courant–Friedrichs–Lewy
DEC	Discrete Exterior Calculus
FEEC	Finite Element Exterior Calculus
GLL	Gauss-Lobatto-Legendre
IC	Initial Condition
LMSEM	Lagrangian Mimetic Spectral Element Method
MFD	Mimetic Finite Difference
MSEM	Mimetic Spectral Element Method
ODE	Ordinary Differential Equation
PDE	Partial Differential Equation
VEM	Virtual Element Method

1

Introduction

The field of Computational Fluid Dynamics (CFD) has seen a tremendous amount of development over the past decades. As a result, CFD has been applied to many different problems, such as chaotic advection [1], the design of aeroplanes [2–4], and weather prediction [5], to name a few. However, despite the ubiquity of CFD, most of the fundamental algorithms and methods are (based on) those developed in the eighties and nineties [6, 7]. While this is not necessarily bad, it is well-known that such algorithms are far from perfect. Nevertheless, many current developments are aimed towards extensions in terms of complex geometries, parallel computing, and/or multi-physics flows [2, 8]. This causes two main problems. On the one hand, with the focus on complex problems, the fundamental algorithms receive less research attention, despite their shortcomings [9]. On the other hand, tackling more complex problems with a sub-optimal basis will be extra difficult due to the shortcomings of the underlying algorithms [9]. After all, adding new phenomena while the basis is not fully understood will lead to more unknowns and interactions.

One particular difficulty that affects nearly all CFD simulations is that of non-linear advection. Advection plays a fundamental role in fluid dynamics, and is involved in, if not responsible for, phenomena ranging from mixing to turbulence, shocks, and many more[1]. While these phenomena are what makes fluid dynamics both interesting and unique, their different characteristics and non-linearity also pose the greatest challenges. The consequence is that it is difficult to represent advection numerically and many attempts have thus failed.

This difficulty is related to two conflicting types of error; dispersion and dissipation errors. Dispersion errors often lead to instabilities and blow-ups, while dissipation errors are more stable but remove information from the solution. In most CFD simulations, numerical methods with additional dissipation (artificial viscosity) are used, due to their stable and robust nature. The consequence of this, however, is that these simulations are only useful for short-duration simulations and sensitive problems. Flows involving flow instabilities, especially over longer time periods, for example, cannot be reliably tackled due to such artificial stabilisation terms. Reducing the strength of the stabilisation term may improve the result by reducing the negative effect of diffusion, but can lead to dispersion errors.

The aim of this project is, therefore, to create a numerical advection scheme that can accurately and reliably handle advection, without generating the common dispersion or dissipation errors. This is a formidable task in itself, so only one-dimensional (but unsteady) problems will be considered. Additionally, in order to best represent the physical process of advection, a so-called mimetic method will be used (such methods are briefly explained in Section 1.1 and in detail in Chapter 2). In particular, this will be the Mimetic Spectral Element Method (MSEM) [10–14]. On top of that, this project will also use a Lagrangian formulation of advection (and fluid dynamics) and will create a space-time method. The reason for choosing these specific aspects will be explained in Chapter 3. The result is the Lagrangian Mimetic Spectral Element Method (LMSEM), hence the title of this work.

---

[1] Even the cover image on this report could not be made without non-linear advection. Using so-called 'fluid art' (see `https://mymodernmet.com/acrylic-pouring-fluid-art-basics/`), pigments are mixed and poured over a surface, creating the most interesting patterns. While the use of pouring and tilting, combined with hard-to-characterise, viscous fluids makes such a phenomenon virtually impossible to predict, it can lead to mesmerising images and artwork.

1.1. Mimetic Methods and the Role of Geometry

Mimetic methods ”aim to preserve key mathematical and physical properties of continuum physics models in their finite-dimensional algebraic representations.” [15, p.1039]. Simply put, the mathematical and physical properties of the problem must also be present when discretised. This is of course the goal of any numerical method, though doing this consistently and reliably requires a strong basis. It turns out that geometry can provide such as basis. This geometrical basis is an important aspect of the success of mimetic methods and plays a key role in their conservation and stability properties.

Geometrical and topological aspects have been important in the study of fluid dynamics since the early days [16], and have played fundamental roles in the discovery of new relations, invariants, and patterns [16, 17]. One can even find elements of geometry and (geometric) similarity in Leonardo da Vinci’s work on fluid mechanics [18]. Nevertheless, the importance and role of geometry have varied considerably over time. Nowadays, geometry plays only a minor role, if any, in most treatments of fluid dynamics, though Chapter 2 will show that by only using a few basic geometric elements one can build very powerful numerical methods. While the content of Chapter 2 is sufficient for this work, further geometric relations may be found in the literature, see for example [19].

Besides the general effectiveness of mimetic methods, one important aspect to consider when using a Lagrangian formulation is the movement of the computational mesh. There are various ways of dealing with this aspect, though many conventional numerical methods lose their accuracy when the mesh deforms. However, mimetic methods have been shown to work well on highly deformed (even self-overlapping) meshes [12, 20, 21], non-smooth grids [22, 23], and keep their properties on moving meshes [24]. It is, therefore, expected that the moving mesh aspect will not lead to severe problems, though the implementation is new and requires development.

1.2. Research Question

The aim of this project was already mentioned. Nevertheless, a research question was also formulated at the start of this project and is as follows.

> Can the Lagrangian view of advection be used to create a feasible, space-time numerical method for advection in the mimetic spectral element framework?

In support of the main question, the following sub-questions were also formulated.

- Does the method produce the correct result? That is, is the solution sufficiently close to an exact or reference solution (in a suitable norm)?
- Does the method converge when increasing the number of mesh elements or the polynomial order? If so, what is the order of the method?
- Can the method do the above for problems with shock formation as well?

The main question is clear in itself, and the sub-questions aim to make the main question somewhat more tangible. The sub-questions are particularly relevant as an explanation of the feasibility term in the main question. The suitability of the research question was established in a separate literature study, though the main elements will be reiterated in Chapter 3.

From the way in which the questions are posed, one can deduce that this work is intended as a proof of concept. As such, the most important aspects of this work are showing that the method produces the right result and trying to explain why this is and how the method works. A clear understanding of the underlying ideas is thus considered more important than running many example problems. This is particularly true when developing a mimetic method, where the underlying physical phenomena play an important role. Additionally, while new methods are often compared to existing methods, the focus of this work is developing the new method, so comparisons to other methods are left for future work.

Taking an advance on some of the terminology introduced in Chapter 2, this project aims to create a topological advection scheme. This means that advection can be handled without approximation and is completely independent of basis functions or metric elements. This is an important question, as the available mimetic schemes in the literature were not successful in this regard. This will also be detailed in Chapter 3.

1.3. Computational Notes

The programs created in this work were coded by the author using Python [25–28] with the commonly used scientific packages Numpy [29], Scipy [30], and Matplotlib [31]. Python and these packages are considered sufficiently reliable and of high enough quality to use in the development of a new method, greatly reducing the required programming effort. Nevertheless, as far as the new elements are concerned, the codes were written from scratch to allow the most flexibility. Some fundamental parts, mostly related to the MSEM basis functions, were already created by Zhang and could be adapted from his work [14].

The choice of Python is due to its easy use, the author's proficiency, and the focus being on the algorithm as opposed to computational speed. Of course, making efficient use of the Numpy [29] and Scipy [30] packages can already lead to reasonably fast algorithms [32]. This is sufficient for the current project, especially considering the one-dimensional nature of the test problems. The codes developed for this work are not publicly available due to their rapidly changing nature, though pseudo-algorithms will be given throughout the text.

1.4. Report Outline

This introduction already mentioned the use of the MSEM, which will be introduced in detail in Chapter 2. The underlying theory will be explained, and the role of geometry will become evident. Some additional theoretical elements, in particular those related to differential forms, will also be explained. Then, in Chapter 3, the most important aspects of (Lagrangian) advection will be reiterated. Additionally, the main elements of the previously referred to literature study will be summarised. These first two chapters serve, alongside the current chapter, as an introduction to the main topics and will give a broader overview of the relevance of this project.

Starting from Chapter 4, the numerical method will be developed. As this is one of the few Lagrangian mimetic methods, and the first using the MSEM framework, Chapter 4 will start with the linear advection equation. Some fundamental ideas will be first formulated here and several important aspects of the new method will be discovered. Then, in Chapter 5, Burgers' equation will be studied. A physical interpretation of Burgers' equation will turn out to be vital to the development of the method. Additionally, the final section of this chapter will briefly investigate shocks. The last chapter concerning the main development is Chapter 6. The method will then be extended to the isentropic Euler equations of fluid dynamics. Treating compressible flows with a mimetic method is still in its infancy, so the successful discretisation of these equations may be a significant step forward.

The last three chapters will wrap up some loose ends. Firstly, in Chapter 7, some theoretical and conceptual notes will be given. This chapter is thus somewhat different from the preceding ones, as it looks more at the foundations of the method. Secondly, in Chapter 8, some conclusions will be drawn and the answer to the research question will be given. Thirdly and finally, in Chapter 9, some recommendations for future work will be given. These will be related to various aspects of this work, and will also indicate some, in the author's opinion, interesting avenues for mimetic methods to explore.

2

An Introduction to Mimetic Methods and the Mimetic Spectral Element Method

As mentioned in the introduction (Chapter 1), mimetic methods ”aim to preserve key mathematical and physical properties of continuum physics models in their finite-dimensional algebraic representations.” [15, p.1039]. To achieve this goal, mimetic methods (re-)introduce geometry into the equations and their discretisation. Unfortunately, the well-known vector calculus is not well suited for this purpose, as will be shown below. Therefore, this chapter will introduce various concepts from exterior calculus (differential forms) and (algebraic) geometry, and their discretisation. The content of this chapter is by no means new, and various authors have written introductions on (discrete) differential forms, see for example the introductions by Bossavit [33–36] and [37–41], Desbrun et al. [42], Deschamps [43], Fumeron [44], Gerritsma [12], Mattiussi [45], Perot [46, 47], Tonti [48, 49], and Warnick et al. [50, 51]. While more references will be cited in this chapter, it is not an exhaustive overview of the literature. One of the main differences between the various works in the literature is their breadth and mathematical rigour. In this chapter, most of the basics will be covered, and while some mathematical definitions will be included, many mathematical details will be left out. For more mathematically rigorous treatments, see for example Hiptmair [52], Arnold and Khesin [53, 54], or Kreeft, Palha and Gerritsma [10].

As with any type of numerical method, many different mimetic methods exist. Some examples include the mimetic finite difference (MFD)/virtual element method (VEM) [55, 56], finite element exterior calculus (FEEC) [57, 58], the cell method [59, 60], discrete exterior calculus (DEC) [61, 62], and the mimetic spectral element method (MSEM) [11–14]. Despite their differences, these methods have the same theoretical background, so the concepts introduced in this chapter will apply to most mimetic methods. In this work, the mimetic spectral element method will be used. So, if there is a difference between the methods, the MSEM implementation will be explained. This will be indicated in the relevant parts.

This chapter will start with some brief arguments for the use of mimetic methods and their associated introduction to differential forms and algebraic geometry in Section 2.1. With these potential benefits in mind, the basics of differential forms and some basic operations on them are discussed in Section 2.2. Then, in Section 2.3, the discretisation of forms and the basic building blocks of mimetic methods are introduced. This is followed by a brief introduction to some other relevant, more advanced, operators in Section 2.4. These operators will not be discretised directly, but nevertheless play an important role in the development of mimetic methods. Finally, this chapter ends, in Section 2.5, with some notes on the role of time in mimetic methods, a topic that has not received as much attention as the spatial discretisations.

2.1. Raison d’Etre of Mimetic Methods

A mimetic numerical method will be developed in this work to solve advection equations. But numerical methods exist in many different forms for a wide range of applications, so why would it be necessary to introduce elements of two branches of mathematics (differential forms and algebraic geometry)? Especially so since advection equations are well known in vector calculus? Therefore, before introducing an array of new concepts, let's look at three well-known and very important integral theorems in vector calculus; the fundamental theorem of calculus, the (Kelvin-)Stokes theorem and the (Gauss(-Ostrogradskii)) Divergence theorem. These theorems read (see for example [63])

$$\int_C \nabla f \cdot d\mathbf{r} = f(\mathbf{r}(b)) - f(\mathbf{r}(a)), \tag{2.1}$$

$$\iint_S \nabla \times \mathbf{F} \cdot d\mathbf{S} = \int_C \mathbf{F} \cdot d\mathbf{r}, \tag{2.2}$$

$$\iiint_E \nabla \cdot \mathbf{F}\, dV = \iint_S \mathbf{F} \cdot d\mathbf{S}, \tag{2.3}$$

with f a function over the line $\mathbf{r}$ with endpoints $\mathbf{r}(a)$ and $\mathbf{r}(b)$ for the first equation, $\mathbf{F}$ a vector field on the surface S with C its boundary for the second equation, and $\mathbf{F}$ a vector field on the volume E with S its boundary for the third equation. It is interesting to see that all these theorems say that the integral of some derivative of a vector field over a region equals the integral of the vector field over the boundary of that region. While vector calculus cannot explain this similarity, differential forms can [64]. It turns out that all theorems of vector calculus can also be derived using differential forms [64]. Thus, introducing differential forms allows underlying connections and similarities to be unified and explained, which is key in developing mimetic methods.

Besides the mathematical similarities just mentioned, there are also physical similarities between different physical theories. Tonti [65] was the first to systematically analyse these. This has led to his book [48] describing the reasons behind the similarities and creating a classification of the theories. One important aspect of Tonti's work is the use of geometry [48]. The geometry leads to the use of algebraic geometry, which can be used as a natural starting point for numerical methods [12, 42, 45, 66]. This geometric aspect is difficult to include (if at all possible) when using vector calculus, while it is much more natural when using differential forms, as will be shown in this chapter.

It turns out that mimetic methods can use the formalisms of differential forms and algebraic geometry to retain certain mathematical and physics properties in the discretisation [45]. This can, for example, lead to improved conservation properties [46, 67], stability [68–70], and accuracy [70, 71] of the numerical methods.

2.2. Forms and their Basic Operations

In this section, forms and some basic operators on them will be discussed. While the concepts of forms and the related operators may sound unfamiliar, they are not so different from the well-known vector calculus. In Cartesian coordinates, in particular, the difference is small. The benefit of forms is in the uniformity of the notation and operators in higher dimensions and alternative coordinate systems. Differential forms also provide the ability to include geometrical aspects, a topic that will play an important role throughout this work.

2.2.1. Differential Forms

"The objects which we shall study are called exterior differential forms. These are the things which occur under integral signs." [72, p.1]. These two sentences are the start of Flanders' [72] renowned treatment of differential forms and directly indicate the link between differential forms and integration. In fact, the dimension of a form can be determined based on the type of integral called for by the form [43, 50]. This description of forms provides an intuitive understanding. k-dimensional-forms (or k-forms) can more formally be introduced as anti-symmetric k-tensors or as a multi-linear map on k-tangent vectors [42, 73]. An alternative is the geometric description used by Warnick and collaborators [50, 51], the book by Misner, Thorne and Wheeler [74], and the book of Bachman [75]. Here, the intuitive definition of forms as 'the things which occur under the integral sign' will suffice. Note that this includes the differentials. Some example expressions will be given in a three-dimensional space throughout.

In an n-dimensional space, forms of degree $0,\ldots,n$ may exist. As an example, consider $\mathbb{R}^3$. In this three-dimensional space, forms of degrees 0, 1, 2, and 3 may exist. 0-forms are functions and can be written as

$$\alpha^0 = f(x, y, z). \tag{2.4}$$

The superscript indicates the degree of the form. It is natural to 'integrate' 0-forms over 0-dimensional geometric objects; points. In this case 'integrate' comes down to evaluating the 0-form at points

$\mathcal{P}_1, \dots, \mathcal{P}_k$ and summing the different values:

$$\int_{\mathcal{P}_0,\dots,\mathcal{P}_k} \alpha^0 = \sum_{i=1}^{k} f(\mathcal{P}_i). \tag{2.5}$$

The use of the term 'integrate' allows a universal description that holds regardless of the degree of the form, as will become clearer when discussing higher dimensional forms. The association of functions to points can be considered a duality pairing between the points and the functions:

$$\langle \alpha^0, \mathcal{P} \rangle = f(\mathcal{P}). \tag{2.6}$$

This duality pairing is what is strictly meant when talking about the associating of variables to geometric elements. It is the mathematical definition that says that a variable 'lives' on this particular geometric element.

1-forms are associated to 1-dimensional objects; lines (or curves). A 1-form can be written as

$$\beta^1 = A(x,y,z)\,dx + B(x,y,z)\,dy + C(x,y,z)\,dz. \tag{2.7}$$

Integrating 1-forms is natural when integrating over curves. For a curve $\mathcal{C}$ this becomes

$$\int_{\mathcal{C}} \beta^1 = \int_{\mathcal{C}} A(x,y,z)\,dx + B(x,y,z)\,dy + C(x,y,z)\,dz. \tag{2.8}$$

This is a standard line integral and is the reason that we also speak of 'integration of 0-forms over points'. It also shows that Flanders' [72] definition of forms as things that occur under the integral sign is very natural. Writing

$$\langle \beta^1, \mathcal{C} \rangle = \int_{\mathcal{C}} \beta^1 \tag{2.9}$$

for the duality pairing between a curve $\mathcal{C}$ and the 1-form β^1 again formalises the association of 1-forms to lines.

2-forms are associated to 2-dimensional objects; surfaces. A 2-form can be written as

$$\gamma^2 = P(x,y,z)\,dy\,dz + Q(x,y,z)\,dz\,dx + R(x,y,z)\,dx\,dy. \tag{2.10}$$

Integrating 2-forms is natural when integrating over surfaces. For a surface $\mathcal{S}$ this becomes

$$\int_{\mathcal{S}} \gamma^2 = \int_{\mathcal{S}} P(x,y,z)\,dy\,dz + Q(x,y,z)\,dz\,dx + R(x,y,z)\,dx\,dy. \tag{2.11}$$

This is a standard surface integral. Writing

$$\langle \gamma^2, \mathcal{S} \rangle = \int_{\mathcal{S}} \gamma^2 \tag{2.12}$$

for the duality pairing between a surface $\mathcal{S}$ and the 2-form γ^2 again formalises the association of 2-forms to surfaces.

Finally, 3-forms are associated to 3-dimensional objects; volumes. A 3-form can be written as

$$\zeta^3 = g(x,y,z)\,dx\,dy\,dz. \tag{2.13}$$

Integrating 3-forms is natural when integrating over volumes. For a volume $\mathcal{V}$ this becomes

$$\int_{\mathcal{V}} \zeta^3 = \int_{\mathcal{V}} g(x,y,z)\,dx\,dy\,dz. \tag{2.14}$$

This is a standard volume integral. Writing

$$\langle \zeta^3, \mathcal{V} \rangle = \int_{\mathcal{V}} \zeta^3 \tag{2.15}$$

for the duality pairing between a volume $\mathcal{V}$ and the 3-form ζ^3 again formalises the association of 3-forms to volumes.

If these forms are considered in a Euclidean space $\mathbb{R}^3$, there is a direct correspondence to vector calculus. The 0- and 3-forms would both be functions. For a 0-form, this is straightforward to see. For a 3-form, the function that one would normally work with is the proxy of the form, that is, the part in front of $dx\,dy\,dz$. This can also be seen in Equation (2.14) where $g(x, y, z)$ is the function that one would normally integrate. A similar story holds for 1- and 2-forms, though they correspond to vectors. This shows the first important difference between forms and vectors; where vector calculus has only two objects to work with (functions and vectors), differential geometry has four; one for each geometric object. This shows that the role of geometry is hidden or entirely unavailable when using vector calculus.

Each form is associated to a geometric object of the same dimension, as formalised by the duality pairing. Geometric objects can be oriented, and there exist two types of orientation; inner and outer [48]. A form associated to an oriented object will inherit this orientation. This means that there exist two different types of forms. When discussing differential forms, as done here, the two different types of forms are not always explicitly linked to geometric objects. As a result, many different names exist for the two different types; non-twisted and twisted forms [50], ordinary and twisted forms [76], even and odd forms [48], forms and pseudo-forms [73], and straight and twisted forms [37]. As the introduction given here is a precursor to mimetic methods, inner and outer-oriented forms will be used, since those terms are common in the discretised cases.

The inner or outer orientations of an object and its associated form play an important role in mimetic methods. However, during this introduction to differential forms, orientation plays a very small part only. Therefore, the only thing that will be said about this is that outer-oriented forms change sign when the space in which they are embedded changes orientation [73]. Inner-oriented forms can be oriented without regard for the embedding space. The orientations and their importance will be discussed in detail when discussing discrete forms in Section 2.3.

2.2.2. Wedge Product

The first operator on forms presented here is the wedge (or exterior) product. It is the product between k- and l-forms which produces a $(k+l)$-form ($(k+l)$ must be smaller than or equal to n, the degree of the embedding space, for this product to be non-zero) and has the following properties [12, 44]:

- (multi-)linearity: $(a\alpha + b\beta) \wedge (c\gamma + d\delta) = ac(\alpha \wedge \gamma) + ad(\alpha \wedge \delta) + bc(\beta \wedge \gamma) + bd(\beta \wedge \delta)$,
- associativity: $(\alpha \wedge \epsilon) \wedge \gamma = \alpha \wedge (\epsilon \wedge \gamma) = \alpha \wedge \epsilon \wedge \gamma$,
- supercommutativity: $\alpha \wedge \gamma = (-1)^{kl} \gamma \wedge \alpha$,

where α, β are k-forms, γ, δ are l-forms, ϵ is a p-form, and a, b, c, d are constants. The last property implies that the product of an odd form with itself is zero, that is $\alpha^k \wedge \alpha^k = 0$ with k odd.

It turns out that the wedge product has already appeared in the previous section, albeit implicitly. Take for example the 2-form $\gamma^2 = P(x,y,z)\,dy\,dz + Q(x,y,z)\,dz\,dx + R(x,y,z)\,dx\,dy$ of the previous section. The $dy\,dz$-part actually reads $dy \wedge dz$, since dy and dz can both be seen as forms. The same is true for the other terms. This is exactly why no $dx\,dx$-like terms appear and why only the $dy\,dz$-part and not the $dz\,dy$-part is needed. The supercommutativity property of the wedge product handles this. The omission of the $\wedge$-symbol is common and will be used here as well.

The properties of the wedge product also automatically take care of the orientation of a coordinate system in a rigorous way [51]. This orientation should not be confused with the inner or outer orientations of geometric elements. Rather, this takes care of choosing between right- and left-handed coordinate systems for example.

Next to the wedge product introduced here, another, possibly more important (for mimetic methods), operator is the exterior derivative. This will be introduced now.

2.2.3. Exterior Derivative

The exterior derivative, denoted d, is a very important operator when using differential forms. Formally, it transforms k-forms into $(k+1)$-forms [12, 73]. When applying it consecutively, it always returns the zero $(k+2)$-form as $d \circ d = 0^{k+2}$. This operator has the following properties (see [73], α and β are forms of an appropriate degree):

- d is additive: $d(\alpha+\beta) = d\alpha + d\beta$,
- $d\alpha^0$ is the usual differential of the function α^0,
- d satisfies a Leibniz' rule: $d(\alpha^p \wedge \beta^q) = d\alpha^p \wedge \beta^q + (-1)^p \alpha^p \wedge d\beta^q$,
- $d^2\alpha = d(d\alpha) = d \circ d\alpha = 0$.

To see how this operator works in practice, let's apply it to the example forms introduced in Section 2.2. Remember that these forms were introduced in the three-dimensional space $\mathbb{R}^3$.

Applying the exterior derivative to the 0-form $\alpha^0 = f(x,y,z)$ simply results in (second property above)

$$d\alpha^0 = df = \frac{\partial f}{\partial x}\,dx + \frac{\partial f}{\partial y}\,dy + \frac{\partial f}{\partial z}\,dz. \tag{2.16}$$

The vector proxy of the right-hand side of this equation is clearly the well know gradient operator applied to the function f. As such, the exterior derivative applied to 0-forms embodies the gradient operator of vector calculus.

Applying the exterior derivative to the 1-form $\beta^1 = A(x,y,z)\,dx + B(x,y,z)\,dy + C(x,y,z)\,dz$ gives

$$\begin{aligned} d\beta^1 = & \left(\frac{\partial A}{\partial x}\,dx + \frac{\partial A}{\partial y}\,dy + \frac{\partial A}{\partial z}\,dz\right) \wedge dx + A\,d \circ dx \\ & + \left(\frac{\partial B}{\partial x}\,dx + \frac{\partial B}{\partial y}\,dy + \frac{\partial B}{\partial z}\,dz\right) \wedge dy + B\,d \circ dy \\ & + \left(\frac{\partial C}{\partial x}\,dx + \frac{\partial C}{\partial y}\,dy + \frac{\partial C}{\partial z}\,dz\right) \wedge dz + C\,d \circ dz \\ & = \left(\frac{\partial C}{\partial y} - \frac{\partial B}{\partial z}\right) dy\,dz + \left(\frac{\partial A}{\partial z} - \frac{\partial C}{\partial x}\right) dz\,dx + \left(\frac{\partial B}{\partial x} - \frac{\partial A}{\partial y}\right) dx\,dy. \end{aligned} \tag{2.17}$$

Note that all properties of the exterior derivative were used to obtain this expression. In this case, the vector proxy of the right-hand side can be identified with the curl of the vector $[A,B,C]^T$. The exterior derivative thus embodies the curl when applied to 1-forms.

Applying the exterior derivative to the 2-form $\gamma^2 = P(x,y,z)\,dy\,dz + Q(x,y,z)\,dz\,dx + R(x,y,z)\,dx\,dy$ gives

$$d\gamma^2 = (\frac{\partial P}{\partial x} + \frac{\partial Q}{\partial y} + \frac{\partial R}{\partial z})\,dx\,dy\,dz. \tag{2.18}$$

It is again obvious which vector proxy is represented on the right-hand side; the divergence of the vector $[P,Q,R]^T$. This shows that the exterior derivative captures the grad, curl, and div in one operator. Therefore, the last property of the exterior derivative leads to the well-known identities of $div \circ curl \equiv 0$ and $curl \circ grad \equiv 0$. Additionally, while the examples have used Cartesian coordinates, the use of the exterior derivative with for example cylindrical coordinates works in the same way [73]. The exterior derivative thus unifies, extends and removes the metric from these three operators. This has led Burke to write his strongly titled "Div, Grad, Curl are dead" [77].

Finally, applying the exterior derivative to the 3-form $\zeta^3 = g(x,y,z)\,dx\,dy\,dz$ simply results in zero, as no 4-form can exist in a three-dimensional space.

This shows how general and important the exterior derivative is. This operator can be used in integral theorems as well, leading to the most important integral theorem for mimetic methods, which is given in the next section.

2.2.4. Generalised Stokes' Theorem

With the help of the exterior derivative, the so-called generalised Stokes' theorem can be written. This single theorem unites and extends the integration theorems of Equations (2.1) to (2.3). It can be written as

$$\int_{\partial\Omega} \omega^k = \int_{\Omega} d\omega^k, \tag{2.19}$$

where ω is a k-form, Ω the $(k+1)$-dimensional integration domain with $\partial\Omega$ its k-dimensional boundary. The importance of this theorem cannot be overstated.

The generalised Stokes' theorem indeed extends Equations (2.1) to (2.3), but it does something else too. It removes the metric from these statements. That is, no dot products or normal vectors are needed. This is particularly important for mimetic methods and when considering spaces that are not flat. Using such metric-free expressions where possible is what allows a proper separation between parts that need a metric and those that do not, which is at the heart of mimetic methods.

In Section 2.2 integration was used as duality pairing between a form and a geometric element. The generalised Stokes' theorem can also be written in this form as

$$\langle d\omega^k, \Omega\rangle = \langle \omega^{k-1}, \partial\Omega\rangle. \tag{2.20}$$

Mathematically, this shows that the exterior derivative is the formal adjoint of the boundary operator [12]. An important consequence of this is that the exterior derivative can be defined without requirements on differentiability or smoothness [12]. Mimetic methods use this to create incidence matrices which represent the exterior derivative on a discrete mesh and which are exact. Therefore, no approximation of (integrated) derivatives is needed. How these incidence matrices can be constructed will be shown in Section 2.3.

2.2.5. Pullback

The generalised Stokes' theorem of the previous section is of fundamental importance. Another powerful concept, but of more practical importance, is the pullback. The pullback operator appears when considering mappings. Therefore, consider the mapping $\Phi : (x, y, z) \rightarrow (u, v, w)$ with (u, v, w) another coordinate system. Suppose, then, that there is a differential k-form α^k in (u, v, w)-coordinates. The pullback Φ^* is then defined as

$$\int_{\Phi(\Omega)} \alpha^k = \int_\Omega \Phi^* \alpha^k. \tag{2.21}$$

Note that this is effectively a change of variables with differential forms [43]. This is a useful aspect in itself, but there are two more properties worth mentioning. Firstly, the pullback commutes with the wedge product, that is, $\Phi \circ \wedge = \wedge \circ \Phi$. Secondly, the pullback also commutes with the exterior derivative ($\Phi \circ d = d \circ \Phi$). In both cases, this formalises the coordinate independence and interchangeability of the wedge product and exterior derivative [12, 78].

These two properties are also of practical importance. For example, if one wants to take the exterior derivative of a k-form α^k in (u, v, w)-space and map it to a different domain afterwards, this can be inconvenient if the derivative in (u, v, w)-space is complicated. Instead, thanks to the commutativity property, it is also possible to perform the mapping first (that is, the pullback) and take the exterior derivative in the transformed space. The same can be done with the wedge product. This property will be of great importance in this work, although it may occur more implicitly than shown here.

2.2.6. Hodge Operators

The last concept to be introduced in this section is somewhat different from any of the operators discussed so far. The Hodge-$\star$ operators introduced in this section relate forms of different degrees. Not only that, they link forms of different degrees and orientations. The different orientations of forms, inner and outer, were briefly mentioned in Section 2.2.1 and will be discussed in more detail in the discrete case in Section 2.3.1.

In particular, a Hodge-$\star$ operator maps a k-form to an $(n-k)$-form (its dual, see [79] for a definition of duality), where n is the dimension of the embedding space. Mathematically, the Hodge operators are defined as [12]

$$\left((\star\eta)^{(n-k)}, \lambda^{(n-k)}\right)\omega^n = \eta^k \wedge \lambda^{(n-k)}, \tag{2.22}$$

where ω^n is the volume form of an n-dimensional space (in three dimensions, this is the common $dx\,dy\,dz$). The change in orientation is not obvious from the definition, which is why the orientation is discussed in more detail in the discrete case, where this is more evident.

It turns out that the Hodge operators play a well-known role; that of a constitutive equation [45, 48]. It is therefore no surprise that its discretisation is important. Unfortunately, however, the Hodge operators do not have a natural discrete counterpart, leading to considerable freedom in its discretisation [80]. As such, various studies into discrete Hodge operators have been performed [76, 81, 82].

2.2.7. The deRham Complexes

So far, differential forms have been introduced together with some basic operations. One of the strengths of differential forms is their ability to unify and generalise various seemingly different relations. One such example is the generalised Stokes' theorem, treated in Section 2.2.4. Another unifying structure is the deRham complex, which will be treated here. The deRham complexes presented in this section will encapsulate relations between forms of different degrees and they underlie many physical processes, though variations of such complexes may appear as well [58, 83–85].

The exterior derivative, introduced in Section 2.2.3, maps a k-form into a $(k+1)$-form. In three dimensions, this can be encapsulated by

$$\mathbb{R} \rightarrow \Lambda^0(\Omega) \xrightarrow{d} \Lambda^1(\Omega) \xrightarrow{d} \Lambda^2(\Omega) \xrightarrow{d} \Lambda^3(\Omega) \xrightarrow{d} 0, \tag{2.23}$$

where $\Lambda^k(\Omega)$ is the space of k-forms on a domain Ω. Such a sequence is known as a deRham complex. A graphical illustration can also be given, as shown in Figure 2.1.

It is important to realise, and this is clearer in Figure 2.1, that applying the exterior derivative consecutively results in zero ($d \circ d \equiv 0$). Any form, say α^k, for which $d\alpha^k = 0$ is known as a closed form. Such forms are common in physics and engineering and include irrotational and solenoidal (divergence-free) forms. Also, any form, say β^k, for which $\beta^k = d\gamma^{(k+1)}$ is known as an exact form. It turns out, and this is graphically illustrated in Figure 2.1 as well, that every exact form is closed. This is again $d \circ d \equiv 0$. The converse, however, is not always true. Only under certain regularity conditions is a closed form exact, and this is known as Poincaré's lemma [73]. In principle, all this information is also conveyed in Equation (2.23) albeit somewhat implicitly.

The single deRham complex already encapsulates some important information. However, with the help of the Hodge operators of the previous section, a double deRham complex can also be made. An example in three dimensions looks as follows.

$$\begin{tikzcd}
\mathbb{R} \arrow[r, hook] & \Lambda^0(\Omega) \arrow[r, "d"] \arrow[d, leftrightarrow, "\star"] & \Lambda^1(\Omega) \arrow[r, "d"] \arrow[d, leftrightarrow, "\star"] & \Lambda^2(\Omega) \arrow[r, "d"] \arrow[d, leftrightarrow, "\star"] & \Lambda^3(\Omega) \arrow[r, "d"] \arrow[d, leftrightarrow, "\star"] & 0 \\
0 & \tilde{\Lambda}^3(\Omega) \arrow[l, "d"'] & \tilde{\Lambda}^2(\Omega) \arrow[l, "d"'] & \tilde{\Lambda}^1(\Omega) \arrow[l, "d"'] & \tilde{\Lambda}^0(\Omega) \arrow[l, "d"'] & \mathbb{R} \arrow[l, hook']
\end{tikzcd} \tag{2.24}$$

The second deRham complex has tildes on the spaces to indicate a different orientation. The double deRham complex thus also includes the Hodge operators and shows the connection between forms of different degrees and orientations.

The exterior derivative encapsulates the grad, curl, and div operators of vector calculus. As such, the double deRham complex also encodes the adjointness property of these operators. This becomes clearer when writing the double deRham complex explicitly with grad (∇), curl ($\nabla\times$) and div ($\nabla\cdot$), which becomes the following.

$$\begin{tikzcd}
\mathbb{R} \arrow[r, hook] & \Lambda^0(\Omega) \arrow[r, "\nabla"] \arrow[d, leftrightarrow, "\star"] & \Lambda^1(\Omega) \arrow[r, "\nabla\times"] \arrow[d, leftrightarrow, "\star"] & \Lambda^2(\Omega) \arrow[r, "\nabla\cdot"] \arrow[d, leftrightarrow, "\star"] & \Lambda^3(\Omega) \arrow[r] \arrow[d, leftrightarrow, "\star"] & 0 \\
0 & \tilde{\Lambda}^3(\Omega) \arrow[l] & \tilde{\Lambda}^2(\Omega) \arrow[l, "\nabla\cdot"'] & \tilde{\Lambda}^1(\Omega) \arrow[l, "\nabla\times"'] & \tilde{\Lambda}^0(\Omega) \arrow[l, "\nabla"'] & \mathbb{R} \arrow[l, hook']
\end{tikzcd} \tag{2.25}$$

This shows that the gradient is the adjoint of the divergence[2] and that the curl is self-adjoint.

As has been shown throughout Section 2.2 differential forms are a powerful mathematical tool that allows easy extensions to multiple dimensions, a clear separation between metric and non-metric aspects, and an association to geometric elements. While these aspects are useful in their own right, the key for mimetic methods is to preserve all such properties at the discrete level. How this can be done will be explained in the next section.

---

[2] One may expect a minus in the adjointness relation of the gradient and divergence. This has to do with the choice that outward normals are considered positive. The standard orientation of a dual volume, however, is as a sink and thus with the inward-pointing normal.

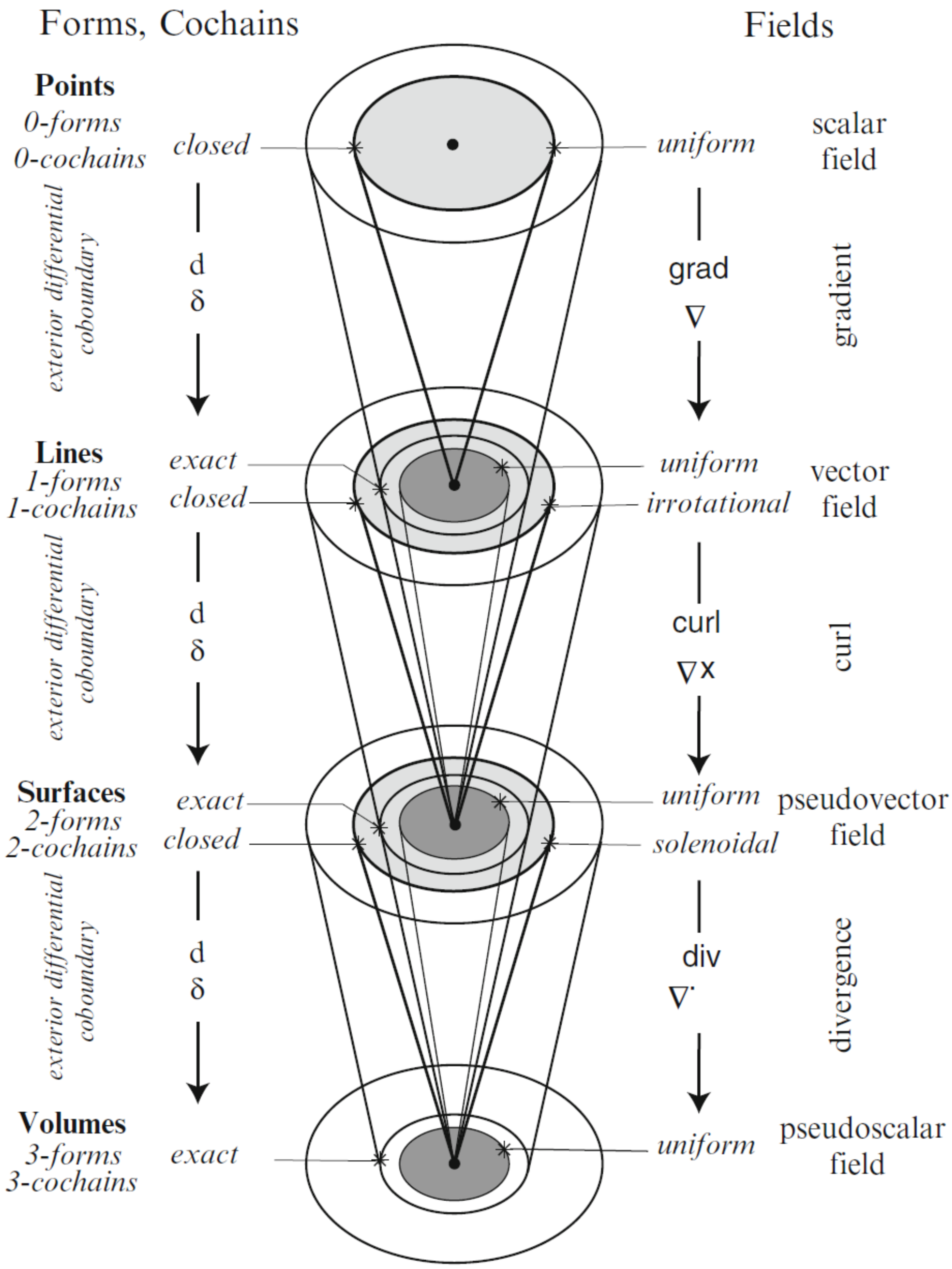


Figure 2.1: A graphical illustration of a single deRham complex, from Tonti [48, p.416].

2.3. Discrete Forms and Incidence Matrices: the Building Blocks of Mimetic Methods

So far, differential forms and some operations on them have been introduced. This was only done in the continuous setting and thus serves as the continuous counterpart of mimetic methods. However, mimetic methods are of course discrete, and to keep the continuous properties the discretisation must be done with care. This section will present the important discrete aspects of mimetic methods, with a particular focus on the MSEM.

2.3.1. Orientation of Geometric Elements

This first aspect towards mimetic methods is assigning an orientation to the various geometrical elements. This was alluded to before, but it is best explained in a discrete setting. There are two types of orientation; inner and outer orientation. These are schematically shown in Figure 2.2.

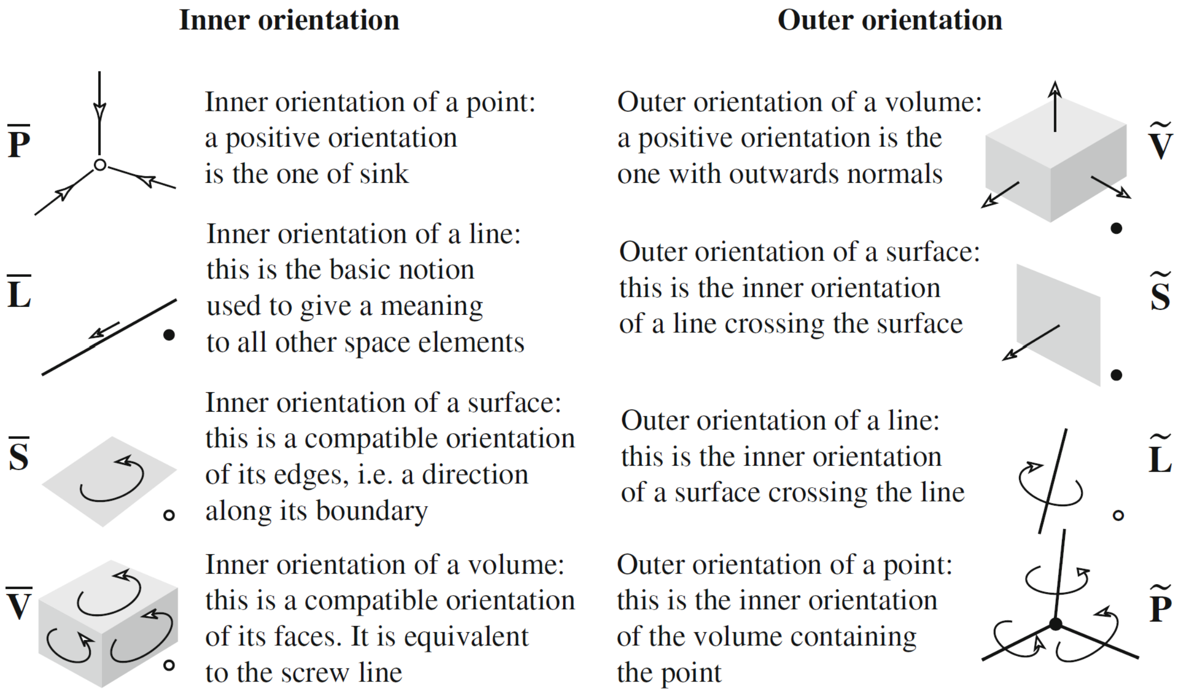


Figure 2.2: An overview of the inner and outer orientations of geometric elements in three dimensions, from Tonti [48, p.41].

The inner orientation can be defined without regard for the embedding space and remains the same independent of the dimension of that embedding space. The inner orientation can thus be defined by staying ’in’ the geometric element. Consider a point, line, surface, and volume in three dimensions as shown on the left of Figure 2.2. A point can be oriented by considering it as a sink or source. A line requires a direction to travel along the line. A surface also requires a direction, but in this case, the direction indicates the way to travel along the boundary. Finally, the volume is oriented based on the orientation of its faces.

The outer orientation does depend on the embedding space. The outer orientation in a three-dimensional space is shown on the right of Figure 2.2. The outer orientation essentially orients a geometric element from the outside. The outer orientation of a volume is well-known and is usually chosen to be the orientation with outwards normal vectors (a source, though a sink is also possible). A surface is outer-oriented by assigning a direction through the surface, while a line is outer-oriented by choosing a direction around it. Finally, a point is oriented by a rotational sense around it.

Note that for both the inner and outer orientations, one can assign a positive and negative direction by choosing one of source/sink, forwards/backwards, and clockwise/counterclockwise as positive. Which sense is chosen as positive is not relevant, though consistency is key. Note also that the outer orientation can be defined based on the inner orientation of the associated (dual) geometric element. The orientation of geometric elements is not commonly mentioned as extensively as used for mimetic methods. The orientation of points in particular is rarely if ever mentioned. Yet, the orientation of the geometric elements will play an important role in the development of mimetic methods.

The existence of two types of orientations demands the use of two related grids[3], the primal and dual grids. In this work, the primal grid will represent the inner-oriented geometric elements, while the dual grid will represent the outer-oriented ones. Additionally, the orientation of the geometric elements is a key aspect of the so-called incidence matrices, which are explained in the next section.

2.3.2. Incidence Matrices

One of the most powerful aspects of mimetic methods are the incidence matrices. These matrices ensure that derivatives can be computed without approximation. Additionally, thanks to their telescop-

---

[3] Strictly speaking, two explicit grids are not needed and only the properties that the dual grid would have, are needed. There are however some (conceptual) advantages to using an explicit dual grid. This will therefore be done in this work.

ing property, they also play the main role in the conservation properties of mimetic methods [46]. As already mentioned in the previous section, the orientation of the geometric elements plays a key role in the creation of the incidence matrices.

As such, let's consider a small inner oriented mesh as shown in Figure 2.3. The orientations for the lines and surfaces are shown in the figures, while the points are oriented as sinks. In this two-dimensional example, two different incidence matrices can be created; one connecting lines and points, and one connecting surfaces and lines. The first, which connects lines and points, represents the exterior derivative applied (Section 2.2.3) to zero-forms (associated to points), which returns a one-form (associated to lines). This corresponds to taking a gradient in vector calculus, see Equation (2.16). The second, which connects surfaces and lines, represents the exterior derivative applied to one-forms (associated to lines), which returns a two-form (associated to surfaces). This corresponds to a curl in vector calculus, see Equation (2.17).

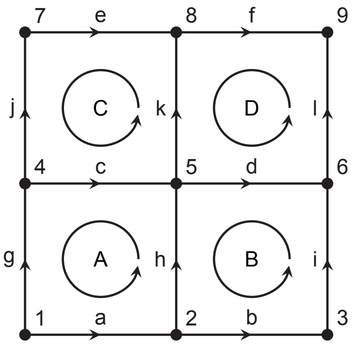


Figure 2.3: Small example mesh with inner orientation. The positive directions are indicated for the lines and surfaces. The points are oriented as sinks. The points are labelled with a number, the lines with a lowercase letter, and the surfaces with an uppercase letter.

In this example, there are twelve lines and nine points. As a result, the incidence matrix connecting lines and points, denoted $\mathbb{E}^{(1,0)}$, will be a twelve-by-nine matrix. The final matrix is given as Equation (2.26). The first row of $\mathbb{E}^{(1,0)}$ is related to the line labelled 'a' in Figure 2.3. Since the line has a positive orientation to the right, it leaves point '1', which is not compatible with the sink nature of the point and thus has a -1 in the matrix in the first column. This line arrives at point '2', which is compatible with the sink orientation so that it has a 1 in the matrix in the second column. No other point is connected to line 'a', so all the other columns have a zero value. This way of checking the compatibility of the orientations is all that is needed to create the incidence matrix, and the result is shown in Equation (2.26).

$$\mathbb{E}^{(1,0)} = \begin{array}{c|ccccccccc|}
 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\
a & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
b & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
c & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 \\
d & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 \\
e & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 \\
f & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \\
g & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
h & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
i & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 \\
j & 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 \\
k & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 \\
l & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1
\end{array} \tag{2.26}$$

There are three notes to make about Equation (2.26). Firstly, there is a clear distinction between the top half and bottom half of the matrix. The top half has the non-zero values close together, while the bottom half has two zeros between the non-zero values. This is because the top half represents the gradient in the horizontal direction, while the bottom half represents the gradient in the vertical direction. Secondly, the top half consists of three two-by-two blocks which each represent one complete horizontal line. The slight shift in the position of each block is due to moving to the next horizontal line. Thirdly and finally, with the mesh topology as used here, a point has at least two and at most four non-zero values in its column, while an edge always has two values in its row. This is because the points are shared and some are located at a boundary and some in the interior.

Next to matrix $\mathbb{E}^{(1,0)}$, which connects lines and points, there is also a matrix $\mathbb{E}^{(2,1)}$ which connects surfaces and lines. In this example, there are four surfaces and twelve lines. As a result, $\mathbb{E}^{(2,1)}$ will be a four-by-twelve matrix. The final matrix is given as Equation (2.27). The first row of $\mathbb{E}^{(2,1)}$ is related to the surface labelled ’A’ in Figure 2.3. As before, only the compatibility of the orientations needs to be checked. In the case of the counter-clockwise-oriented surface ’A’, the orientation matches with the orientation of edges ’a’ and ’h’, while it is opposite to the orientation of edges ’c’ and ’g’. None of the other edges is connected to surface ’A’, so the other entries in this row are zero. The final result is given in Equation (2.27).

$$\mathbb{E}^{(2,1)} = \begin{array}{c|cccccccccccc} & a & b & c & d & e & f & g & h & i & j & k & l \\ \hline A & 1 & 0 & -1 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 \\ B & 0 & 1 & 0 & -1 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 \\ C & 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & -1 & 1 & 0 \\ D & 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & -1 & 1 \end{array} \tag{2.27}$$

Some notes are again in order. While it appears that this matrix can be split into a right half and a left half, the curl is a two-dimensional operator so this will not result in a curl per direction. This only shows the contributions from the horizontal and vertical edges, which are conveniently ordered. It is still true that some edges are part of multiple surfaces, while others form the outer boundary.

The matrices just introduced represent the discrete gradient ($\mathbb{E}^{(1,0)}$) and curl ($\mathbb{E}^{(2,1)}$) on the small grid of Figure 2.3. They are thus both discrete versions of the exterior derivative, though which is appropriate depends on the (discrete) form to which it must be applied. Neither matrix requires any form of approximation, though to create a reconstruction some basis functions are required, which are introduced in the next section. Nevertheless, one important property is already present. The curl of a gradient is always zero, and multiplying $\mathbb{E}^{(1,0)}$ with $\mathbb{E}^{(2,1)}$ (so $\mathbb{E}^{(2,1)}\mathbb{E}^{(1,0)}$) gives a zero-matrix. This means that, independently of the basis functions, the curl always annihilates the gradient, as it should. Such a property is fundamental when one wants to preserve certain mathematical and physical properties of a system. Additionally, these matrices only depend on the connectivity of the mesh. Bending the mesh edges or heavily deforming the entire mesh thus has no effect on the incidence matrices, as long as the connectivity remains unaltered. This is the reason behind the superior performance of the mimetic operator tested in the work of Candelaresi, Pontin and Hornig [24], where a moving mesh was used. Additional examples of incidence matrices on stationary, but heavily deformed meshes can for example be found in [12, 20, 21].

As already mentioned in the previous section, there are two types of orientation resulting in two distinct meshes. Indeed, superposing a dual mesh over the one shown in Figure 2.3 results in the mesh shown in Figure 2.4. Note that it is not necessary to add new labels or new orientations. Because these two meshes are each other’s geometric dual, the number of dual surfaces corresponds to the number of primal points and vice versa, while the number of primal and dual edges also match. The orientations are also inherited, as was also indicated in Figure 2.2. One important distinction, however, is that the surfaces on the dual grid are oriented as sources, *not* as sinks. This ensures consistency with the commonly used orientation of outward normals. Furthermore, note that the dual grid is not closed. For the time being, this will cause no problems and will be left like this. In Chapter 6, when discussing the compressible Euler equations of fluid dynamics, it will become important to close the dual grid. How this is done will be explained there. Finally, note that the outer orientation of a surface in Figure 2.4 is different from that presented in Figure 2.2. This is due to the different dimensions of the embedding spaces, between two- and three-dimensional. Such a change does not occur for the inner orientation.

On the dual grid, like on the primal grid, two different incidence matrices can be made. The first, denoted $\tilde{\mathbb{E}}^{(1,0)}$, connects the dual edges to the dual points. In this example, there are twelve edges and four points, resulting in a twelve-by-four matrix, given in Equation (2.28). Here too, the only thing to determine is if the orientations match. The first row of $\tilde{\mathbb{E}}^{(1,0)}$ corresponds to the dual-edge ’a’, which is now a vertical edge. The rotation around dual point ’A’ crosses the dual edge ’a’ in the same direction

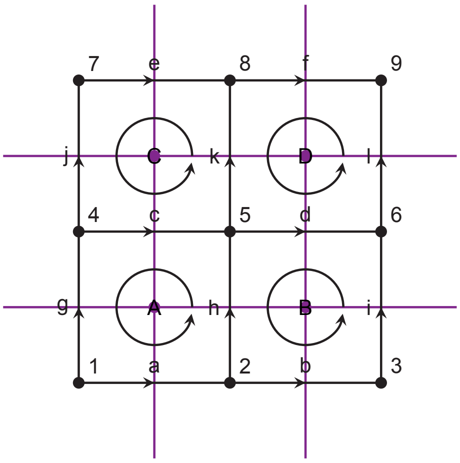


Figure 2.4: Small example mesh with outer orientation. The orientation and labels are inherited from the underlying inner-oriented grid. However, to remain consistent with the commonly used orientation of outward normals, the surfaces on the dual grid are oriented as sources, *not* as sinks.

as indicated, so this is a positive one. This edge is not associated with any other dual point, so the other entries in this row are zero. This final result is given in Equation (2.28).

$$\tilde{\mathbb{E}}^{(1,0)} = \begin{array}{c} \\ a \\ b \\ c \\ d \\ e \\ f \\ g \\ h \\ i \\ j \\ k \\ l \end{array} \begin{array}{c} \begin{array}{cccc} A & B & C & D \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \\ -1 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{array} \tag{2.28}$$

The most important part to note is that $\tilde{\mathbb{E}}^{(1,0)}$ turns out to be the transpose of $\mathbb{E}^{(2,1)}$. Implementing the dual operator can thus be done using this relation. More importantly, this relation gives $\tilde{grad} = curl^T$, which is a well-known adjoint relation[4] and a key feature of the deRham complex (Section 2.2.7).

While it is possible to create $\tilde{\mathbb{E}}^{(2,1)}$ as above, this matrix will turn out to be the negative transpose of $\mathbb{E}^{(1,0)}$, so $\tilde{\mathbb{E}}^{(2,1)} = -{\mathbb{E}^{(2,1)}}^T$. The minus sign is due to the change of orientation of the dual surfaces, which were made sources to remain consistent with the outward normals orientation. The reason for this minus is now obvious. It is also present in vector calculus, albeit for unclear reasons, as vector calculus does not explicitly define most orientations, let alone two different sets.

With the incidence matrices on both the primal and dual grid present, the next step will be to create suitable basis functions. The basis functions will allow for the reconstructions and must have certain properties, as will be explained next. Note, however, that the exactness of the derivatives, the annihilation of the gradient by the curl (or the consecutive application of the exterior derivative), and the adjoint relations between the dual operators are all independent of the basis functions. In fact, only the connectivity and geometric duality of the meshes are required for these properties to hold. Having these properties independently of the basis functions is an important part of what makes mimetic methods so powerful.

---

[4] This relation holds in two dimensions. In three dimensions, the gradient and divergence are each other's (negative) adjoint, while the curl is self-adjoint. This is the reason that there are two different curls, though vector calculus does not make this explicit.

2.3.3. Basis Functions and Discrete Hodge Operators

The various different mimetic methods in existence vary most significantly in the basis functions that are used. The choice of basis functions must be made with care. While there are a variety of different options, the basis functions must be defined such that, in combination with the incidence matrices just introduced, a discrete version of the deRham complex (Section 2.2.7) holds. The basis functions can only be defined with the shape of a mesh element defined. Common choices include triangular elements (simplices) and rectangular elements, the latter is used here. The benefit of using rectangular elements is on the one hand the availability of the tensor product for easy extensions to higher dimensions, and on the other hand, a dual mesh that also consists of rectangular elements so that dual polynomials can be explicitly found [86]. This is different from the triangular elements, for which the dual elements are polygons. For some recent works in this direction, see for example [87–89].

The MSEM has two main basis functions. These are the 1-D Lagrange interpolating polynomial (nodal polynomials that represent 0-forms) on the Gauss-Lobatto-Legendre (GLL) nodes and the 1-D edge polynomial [11] (histapolant polynomials that represent 1-forms). Higher dimensional basis functions are found using a tensor product in a reference domain, which is the unit square or cube depending on the desired dimension. As is common for spectral element methods, the basis functions can be mapped to other more general domains using, for example, transfinite interpolation [90, 91], see also [11, 86] for the basis functions and mappings of MSEM, their transformation properties, and the associated dual polynomials. Here only the two 1-D polynomials and their duals are briefly reviewed. Higher dimensional basis functions follow through the straightforward use of the tensor product [11, 14, 86], so they will not be shown here. Additionally, if two-dimensional basis functions are used, the tensor product is explicitly shown, so that no confusion will arise.

One-Dimensional Primal Basis Functions

The first basis function that is used in the MSEM is the Lagrange interpolating polynomial on the GLL-nodes. Consider the reference interval $I_{\text{ref}} = [-1, 1]$ and the $p+1$ GLL-roots ξ_i of the polynomial $(1-\xi^2)\frac{dL_p(\xi)}{d\xi}$ where $L_p(\xi)$ is the Legendre polynomial of degree p with $\xi \in I_{\text{ref}}$. These roots satisfy $-1 = \xi_0 < \xi_1 < \cdots < \xi_{p-1} < \xi_p = 1$. The Lagrange polynomial of degree p is then given by

$$h_i^p(\xi) = \prod_{\substack{k=0 \\ k\neq i}}^{p} \frac{\xi - \xi_k}{\xi_i - \xi_k}. \tag{2.29}$$

The Lagrange polynomials satisfy the important property

$$h_i^p(\xi_j) = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}, \quad i, j = 0, \ldots, p. \tag{2.30}$$

The Lagrange polynomials are nodal polynomials since they interpolate a function exactly at the GLL-nodes thanks to this property. The Lagrange polynomials of degree $p = 4$ are shown in Figure 2.5a.

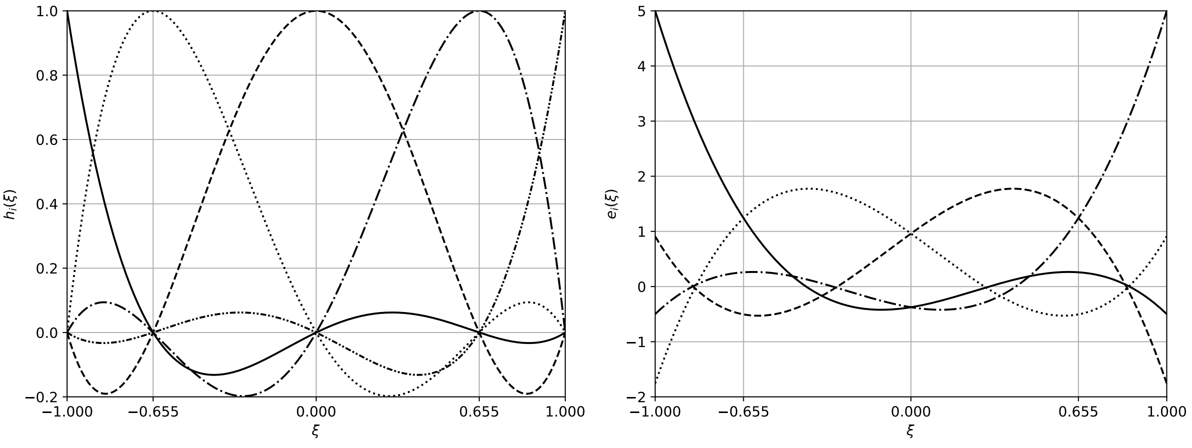
The Lagrange polynomials represent 0-forms and transform as such. In particular, consider the mapping $\Phi : I_{\text{ref}} \to I$ where $I = [a, b]$ is the more general interval to be mapped to. The Lagrange polynomial in the new interval $l_i(x)$ with $x \in I$ is then [86]

$$l_i(x) = h_i \circ \Phi^{-1}(\xi), \tag{2.31}$$

where $\Phi^{-1} : I \to I_{\text{ref}}$ is the inverse mapping. Note that this transformation can only effectively scale the original Lagrange polynomials and that any mapping that is not linear may not produce the intended result. This mapping does not change the main properties of the Lagrange polynomials [11], in particular that of Equation (2.30).

A function can be approximated using the Lagrange polynomials as follows. Consider a function $f(\xi)$ defined on I_{ref}. The function can be approximated using Lagrange polynomials up to degree p as

$$f(\xi) \approx \sum_{i=0}^{p} f_i h_i^p(\xi), \tag{2.32}$$



(a) Lagrange polynomials.

(b) Edge polynomials.

Figure 2.5: One-dimensional basis functions for $p = 4$. The vertical grid lines indicate the GLL-nodes.

with $f_i = f(\xi_i)$. If the function $f(\xi)$ is a polynomial of degree p or lower, the above expression becomes an exact equality.

The second basis function used in the MSEM is the edge polynomial [11], recently also called a histapolant polynomial [92], and is shown in Figure 2.5b. The edge polynomials are derived from the Lagrange polynomials and can be written as [11]

$$e_j^p(\xi) = -\sum_{i=0}^{j-1} dh_i^p(\xi), \quad j = 1, \dots, p, \tag{2.33}$$

with $\xi \in I_{\text{ref}}$. Note that this is a polynomial of degree $p - 1$. The edge polynomials satisfy [11]

$$\int_{\xi_{j-1}}^{\xi_j} e_i^p(\xi)\, d\xi = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}, \quad i, j = 1, \dots, p. \tag{2.34}$$

The edge polynomials of degree $p = 4$ are shown in Figure 2.5b.

The edge polynomials represent 1-forms and transform like them. Consider the same mapping as for the Lagrange polynomials and the Jacobian $J = \frac{dx}{d\xi}$ of the mapping. A reference edge function $e_i(\xi)$ then transforms to the I domain as [86]

$$e_i(x) = \frac{1}{J} e_i \circ \Phi^{-1}(\xi). \tag{2.35}$$

The edge functions transform in the same fashion whether the map is linear or not. The edge functions can be scaled but also stretched or compressed thanks to the Jacobian term. As for the Lagrange polynomials, mapping the edge functions to a different domain does not change its properties [11].

A function can be approximated using the edge polynomials as follows. Consider a function $g(\xi)$ defined on I_{ref}. The function can be approximated using edge polynomials up to degree $p - 1$ as

$$g(\xi) \approx \sum_{i=1}^{p} g_i e_i^p(\xi), \tag{2.36}$$

with $g_i = \int_{\xi_{j-1}}^{\xi_j} g(\xi)\, d\xi$. If the function $g(\xi)$ is a polynomial of degree $p-1$ or lower, the above expression becomes an exact equality.

Due to the definition of the edge functions as in Equation (2.33), the incidence matrix applied to the coefficients of a discrete 0-form produces the coefficients of a discrete 1-form. This ensures that a discrete deRham complex can be made [86]. In one dimension this becomes

$$\mathbb{R} \hookrightarrow P_h \xrightarrow{\mathbb{E}} Q_h \to 0, \tag{2.37}$$

where P_h is the space of discrete 0-forms spanned by the Lagrange polynomials and Q_h is the space of discrete 1-forms spanned by the edge polynomials. This discrete deRham complex holds in the reference domain as well as a transformed domain [11]. The recovery of a deRham complex in the discrete setting is fundamental in the creation of a mimetic method. This also ties into the properties of the incidence matrices introduced in the previous section (albeit the example incidence matrices were introduced in a 2-D setting).

Discrete Hodge Operators

Using the primal basis functions as just introduced, it is possible to create mass matrices as done for any finite/spectral element method. However, for mimetic methods, the mass matrices play the role of the Hodge matrices [81]. The Hodge operators were briefly introduced in Section 2.2.6. The discrete Hodge operators, the Hodge matrices, are thus easily found by creating the mass matrices. Unfortunately, the finite-dimensional nature of this procedure means that some approximation is involved. Indeed, this is the place where the approximations in a mimetic method are made.

In one dimension, two different types of discrete Hodge operators can be made. These so-called Hodge matrices are based on the Lagrange or edge polynomials. Again consider the reference interval I_{ref}. Using the Lagrange polynomials results in a Hodge matrix for zero-forms $\mathbb{H}^0_{ij}$ as

$$\mathbb{H}^0_{ij} = \int_{-1}^{1} w(\xi) h_i(\xi) h_j(\xi)\, d\xi, \tag{2.38}$$

with $\xi \in I_{ref}$, $h_i(\xi)$ the i-th Lagrange polynomial on I_{ref} (see Equation (2.29)) and $w(\xi)$ a weighting function. The weighting function is what really makes the difference between the functions of a conventional mass matrix and a Hodge matrix. The weighting function can be a material parameter coming from a constitutive equation, the Jacobian of a mapping, and/or a new metric, all in one go[5]. The multiple aspects of the role of a Hodge matrix were stressed by for example Mattiussi [45].

As mentioned in Section 2.2.6 the Hodge operator, and thus its discrete version as well, maps an inner-oriented k-form into an outer-oriented $n-k$-form, with n the dimension of the embedding space. By defining the Hodge matrix as done in Equation (2.38), the coefficients of an inner-oriented discrete zero-form α^0_h is related to the coefficients of an outer-oriented discrete one-form $\tilde{\beta}^1_h$ through

$$\tilde{\beta}^1_h = \mathbb{H}^0_{ij} \alpha^0_h, \tag{2.39}$$

assuming a relation of the form $\tilde{\beta}^1 = w\alpha^0$ between the continuous forms holds.

Similarly, a Hodge matrix based on the edge polynomials can be defined as

$$\mathbb{H}^1_{ij} = \int_{-1}^{1} w(\xi) e_i(\xi) e_j(\xi)\, d\xi, \tag{2.40}$$

with $e_i(\xi)$ the i-th edge polynomial on I_{ref} (see Equation (2.33)). This matrix thus transforms the coefficients of inner-oriented discrete one-forms to the coefficients of outer-oriented discrete zero-forms through

$$\tilde{\epsilon}^0_h = \mathbb{H}^1_{ij} \gamma^1_h. \tag{2.41}$$

It should be noted that, even though a geometric dual is assumed here, the Hodge matrices are not unique [76, 82]. The differences between mimetic methods are often most explicit in their choice of discrete Hodge operator. While finite/spectral element methods base their Hodge matrices on the mass matrices, a simple finite volume scheme can have a diagonal Hodge matrix with only the size of the elements on its diagonal. Even for the finite/spectral element methods, the values in the Hodge matrix differ depending on the used basis functions.

---

[5] While implementing such a matrix, one can define the weight based on other functions. For example $w(\xi) = \rho(\xi) g(\xi)$ where ρ can be a material parameter and g a new metric. The weight function w thus does not need to be defined as a single function.

One-Dimensional Dual Basis Functions

Next to the primal basis functions and using the discrete Hodge operators just discussed, the MSEM also has a set of (explicit) dual basis functions [86]. The possibility of explicitly creating the dual basis functions is uncommon and is due to the choice of quadrilaterals as geometric elements and the integral nature of the primal edge basis function. While it is possible to create mimetic methods without explicitly constructing the dual basis, it can be helpful to have them available, especially when developing new methods. As such, they will be used throughout this work.

Like the primal basis functions, there are two main dual basis functions. The basis functions that are dual to the primal Lagrange functions of degree p (h^p) are $\tilde{e}^p$ defined (on the reference interval) as

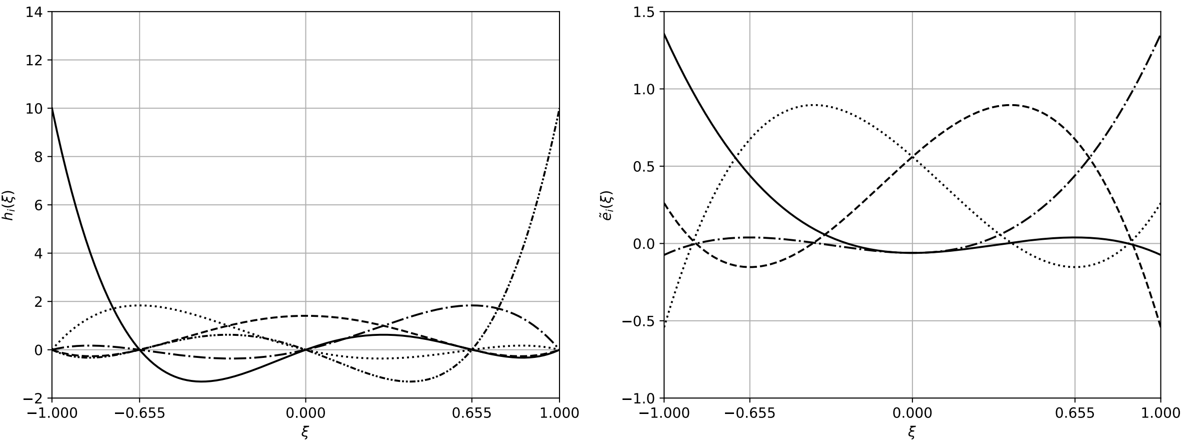
$$\tilde{e}^p = \left(\mathbb{M}^0\right)^{-1} h^p, \tag{2.42}$$

with $\mathbb{M}^0$ the Hodge matrix of Equation (2.38) with a unity weight function, that is, the conventional mass matrix. The weight function used for the dual polynomials often does include the transformation, as this is precisely how the dual basis transforms [86]. Such a transformation is not used here because the definition is given for the reference domain. The material parameters, however, are only intended for use on the coefficients and are thus not included when building a dual basis. Note that the dual basis functions are the basis for the dual one-forms and that the degree of the polynomials is inherited from the primal functions. The functions are plotted for $p = 4$ in Figure 2.6a.

Similarly, the basis functions that are dual to the primal edge functions of degree p (e^p) are $\tilde{h}^p$ defined (on the reference interval) as

$$\tilde{h}^p = \left(\mathbb{M}^1\right)^{-1} e^p, \tag{2.43}$$

with $\mathbb{M}^1$ the Hodge matrix of Equation (2.40), again with a unity weight function. The functions are plotted for $p = 4$ in Figure 2.6b.



(a) Dual to the Lagrange polynomials.

(b) Dual to the edge polynomials.

Figure 2.6: One-dimensional dual basis functions for $p = 4$. The vertical grid lines indicate the GLL-nodes.

Remark. When comparing the dual basis functions of Figure 2.6a to the corresponding dual polynomials in [86], one will notice that they are not exactly the same, while Figures 2.5a, 2.5b and 2.6b are. The reason for this lies in the quadrature rules used to compute the mass matrices needed to construct the dual basis functions. In [86] a Gauss-Lobatto quadrature rule on more nodes (than five) was used, while a Gauss-Lobatto rule with five nodes was used in this work. Due to the tensor product nature of the mass matrix, the Lobatto rule of this work cannot exactly integrate the mass matrix, while the setup of [86] can. As a result, the dual basis functions do not look exactly the same. This does not occur for the edge polynomials, as they are one degree lower. Because the Lobatto rule with five nodes is used throughout the code, the basis functions as shown in Figure 2.6a will be used.

With the definitions of the discrete Hodge operators and the dual bases, one can show that a discrete double deRham complex is present [86]. This can be written as

$$\begin{tikzcd}
\mathbb{R} \arrow[r, hook] & P_h \arrow[r, "\mathbb{E}"] \arrow[d, shift left, "\mathbb{H}^0"] & Q_h \arrow[r] \arrow[d, shift left, "\mathbb{H}^1"] & 0 \\
0 & \tilde{Q}_h \arrow[l] \arrow[u, shift left, "\mathbb{H}_0"] & \tilde{P}_h \arrow[l, "\tilde{\mathbb{E}}"'] \arrow[u, shift left, "\mathbb{H}_1"] & \mathbb{R} \arrow[l, hook']
\end{tikzcd} \tag{2.44}$$

where P_h is the space of discrete, inner-oriented zero-forms spanned by the Lagrange polynomials, Q_h the space of discrete inner-oriented one-forms spanned by the edge polynomials, $\tilde{Q}_h$ is the space of discrete, outer-oriented one-forms spanned by the polynomials dual to the Lagrange polynomials, and $\tilde{P}_h$ the space of discrete outer-oriented zero-forms spanned by the polynomials dual to the edge polynomials. Also, $\mathbb{H}_0 = \left(\mathbb{H}^0\right)^{-1}$ and $\mathbb{H}_1 = \left(\mathbb{H}^1\right)^{-1}$ for clarity.

Now that the geometric elements, incidence matrices, basis functions, and discrete Hodge matrices have been introduced, the main elements of the MSEM are now clear. While the individual aspects are important, their links in the discrete deRham complexes are what makes the whole method truly mimetic. That is, the mathematical structures can now be properly mimicked on a discrete mesh. It is with these elements that new operators and schemes can be developed, as will be done throughout this report. However, in the next section, some other and more advanced operators on differential forms are briefly explained, as they are important for advection.

2.4. Forms and some Advanced Operations

Now that the basics of differential forms and mimetic methods have been explained, a brief look into some more advanced operators can be given. The operators presented in this section have not always been discretised with success. However, the main two operators of this section, the Lie and covariant derivatives, can both serve as advection operators. While the Lie derivative has received considerable attention as an advection operator, the covariant derivative tends to appear only in theoretical works. This section serves as an introduction to these two operators and their differences. Later in this work, the developed method will be shown to be a discretisation of one of these operators.

2.4.1. Interior Product

The first operator introduced here is the interior product, which will appear again when introducing the Lie derivative. The interior product, sometime also called inner derivative [54], takes a $(k+1)$-form and returns a k-form [73]. Graphically, it removes the surfaces of the first form from the second [50]. Alternatively, it can be seen as a contraction operator with extrusion being its dual [93]. The extrusion is shown in Figure 2.7. The contraction is then the dual operator so that the extrusion is mapped to the point that is dual to the extruded surface of Figure 2.7.

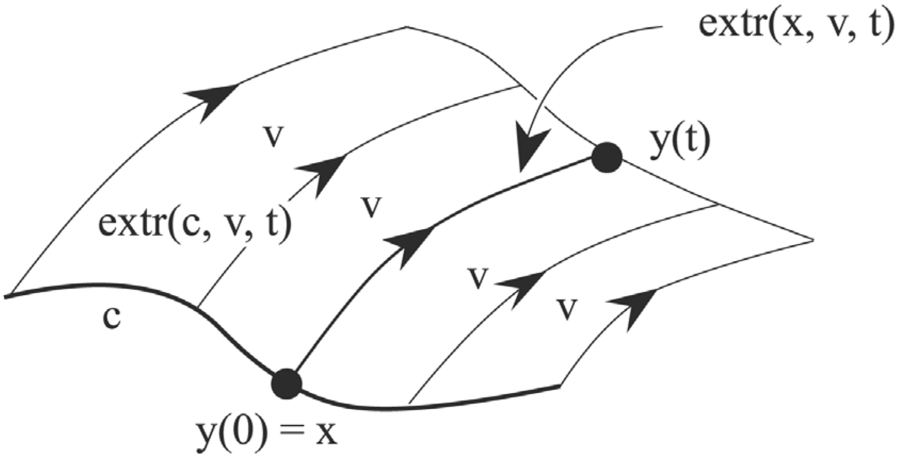


Figure 2.7: A graphical illustration of extrusion, from Bossavit [93, p.471].

2.4.2. Lie Derivative

This work is concerned with the development of a numerical method for advection problems. So how can advection be described using exterior calculus? Let's start with the definition of a flow.

Definition 2.4.1 (Time-independent flow of a vector field [73])**.** Define a 1-parameter family of maps $\phi_t : \mathbb{R}^n \to \mathbb{R}^n$ such that ϕ_t takes a point p at $t = 0$ and gives the position of that point t-seconds later. This is a group of maps if $\phi_s(\phi_t(p)) = \phi_{s+t}(p) = \phi_t\phi_s(p))$ and $\phi_{-t}(\phi_t(p)) = p$ (so $\phi_{-t} = \phi_t^{-1}$), which require time-independence. If each ϕ_t is differentiable, then so is each ϕ_t^{-1} (and each ϕ_t is a diffeomorphism), and this family of maps is a flow.

Associated with such a flow is the time-independent velocity field

$$\mathbf{v}_p := \left.\frac{d\phi_t(p)}{dt}\right|_{t=0}. \tag{2.45}$$

Using this on a function f gives

$$\mathbf{v}_p(f) = \left.\frac{df(\phi_t(p))}{dt}\right|_{t=0}, \tag{2.46}$$

which is the derivative of f along the 'streamline' through the point p [73].

It is also possible to associate to each (time-independent) vector field $\mathbf{v}$ in $\mathbb{R}^n$ a flow $\{\phi_t\}$ with $\mathbf{v}$ its velocity field. The flow $\{\phi_t\}$ is then said to be the flow generated by vector field $\mathbf{v}$. In that case, $\phi_t(p)$ can be found by solving the system

$$\frac{d\mathbf{x}}{dt} = \mathbf{v} \quad with \quad x(0) = p. \tag{2.47}$$

This will give the integral curves (orbits) of the system. $\phi_t(p)$ then says to move along the integral curve through p for time t [73].

Definition 2.4.2 (Lie derivative of a vector field [73])**.** Consider a pair of vector fields, $\mathbf{X}$ and $\mathbf{Y}$, on a manifold M^n and let $\phi(t) = \phi_t$ be the local flow generated by the field $\mathbf{X}$. The Lie derivative of $\mathbf{Y}$ with respect to $\mathbf{X}$ is then defined as

$$\mathcal{L}_{\mathbf{X}}\mathbf{Y}|_x = \lim_{t\to 0}\frac{[\mathbf{Y}_{\phi_t x} - \phi_{t*}\mathbf{Y}_x]}{t} = \lim_{t\to 0}\frac{[\phi_{-t*}\mathbf{Y}_{\phi_t x} - \mathbf{Y}_x]}{t}. \tag{2.48}$$

The above definition of the Lie derivative is that of the Lie derivative applied to a vector field. Note that vector fields are transported in the forward direction [54]. Instead, the Lie derivative can also be defined for forms, which are transported backwards [54].

Definition 2.4.3 (Lie derivative of a form [73])**.** With ϕ_t and $\mathbf{X}$ as before, consider the p-form α^p. The Lie derivative of α^p with respect to $\mathbf{X}$ is then defined as

$$\mathcal{L}_{\mathbf{X}}\alpha^p = \frac{d}{dt}[\phi_t^*\alpha^p]_{t=0} = \lim_{t\to 0}\frac{\phi_t^*\alpha_{\phi_t x} - \alpha_x}{t}. \tag{2.49}$$

Remark. Definition 2.4.3 includes the case for 0-forms (functions), which can be written as (for a function f)

$$\mathcal{L}_{\mathbf{X}}f = \frac{d}{dt}[\phi_t^* f]_{t=0} = \frac{d}{dt}f[\phi_t x]_{t=0}. \tag{2.50}$$

This says that the Lie derivative describes how f changes along the orbits of $\mathbf{X}$. Comparing this to Equation (2.46) shows that the Lie derivative is an extension of directional derivatives [42, 94, 95].

These definitions show that the Lie derivative can be used as an advection operator, because it describes how a vector field or form is transported by a velocity field $\mathbf{v}$. This is schematically shown in Figure 2.8 for a vector field. It should therefore be no surprise that the Lie derivative has received considerable attention from the mimetic community. Some of these works will be mentioned in the brief literature review of Section 3.3. In Figure 2.8 the vector field Y is transported along the vector field X to obtain $\mathcal{L}_{\mathbf{X}}Y$. This shows that the vector field Y is transported along the orbits of the vector field X by means of the flow ϕ (see Definition 2.4.1).

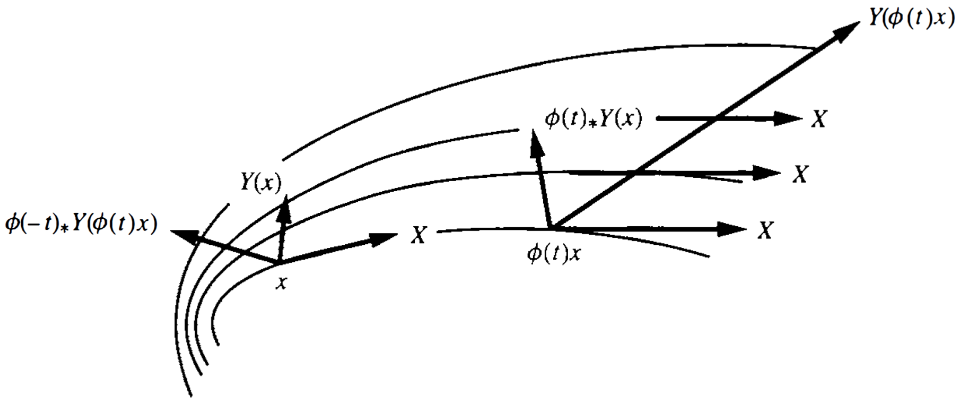


Figure 2.8: Schematic representation of the Lie derivative for a vector field. Picture from Frankel [73, p.125].

From Figure 2.8, it becomes clear that the Lie derivative requires a vector field that transports a vector or form. This transportation vector field cannot change during transport, or the comparison with the starting vector will make little sense. This is called a frozen vector field. The Lie derivative requires such a frozen vector field [54], which is an important aspect of the Lie derivative and the main reason for its use in incompressible flows only. This will be briefly reviewed in Section 3.3.

The Lie derivative can also be expressed in many different forms (see [73] for an overview), as is often the case for advection operators. Mathematically, all these forms are equivalent. However, when discretised, differences may arise leading to different properties of the advection scheme [96]. One interesting and often-used formulation of the Lie derivative is known as Cartan's homotopy (or magic) formula:

$$\mathcal{L}_v = i_v \circ d + d \circ i_v, \tag{2.51}$$

where d is the exterior derivative discussed in Section 2.2.3, and i_v the interior product of Section 2.4.1. Using a definition like this allows the discrete representations of d and i_v to be reused if such representations are available (see Section 3.3 for some works in this direction).

Lastly, it should be noted that, in a Lagrangian formulation, the Lie derivative disappears from the equations. For example, writing the advection of a function f as

$$\frac{\partial f}{\partial t} + \mathcal{L}_v f = 0, \tag{2.52}$$

in the Eulerian frame, leads to

$$\begin{cases} \dfrac{\partial f}{\partial t} = 0, & (2.53a) \\ \dfrac{dX(t)}{dt} = v, & (2.53b) \end{cases}$$

in the Lagrangian frame. The disappearance of advection operators, which may be non-linear, in the Lagrangian formulation is what makes it convenient to describe advection in this formulation. This does not change when using forms.

2.4.3. Covariant Derivative

Next to the Lie derivative, introduced in the previous section, there is another operator which can describe advection. This operator, called the covariant derivative, features in more theoretical works [54]. The covariant derivative, like the Lie derivative, is an extension of a directional derivative, though their working principles differ.

Where the Lie derivative required a vector field to transport and compare two vectors, the covariant derivative compares two vectors by introducing some extra structure; the connection. The covariant derivative of f along v is denoted $\nabla_v f$. In comparing the vector at the start and end of the transport, the covariant derivative does not use the vector field (that is the Lie derivative) but instead parallel transports one vector to the other. How one keeps a vector parallel in a general setting is what is accounted for by the connection. While an additional structure is needed to define the covariant derivative, the two vectors may vary with respect to each other. As such, a frozen vector is not required for this operator.

An interesting property of the covariant derivative, and the reason for its name, is how it transforms under a change of coordinates. Consider a vector field expressed in two different coordinates V and U, and the transition matrix functions c_{VU} and c_{UV} which describe how the coordinates V and U transform into one another. The covariant derivative then satisfies the following rule [73, p.255]

$$\nabla_V \circ c_{VU} = c_{UV} \circ \nabla_U. \tag{2.54}$$

In words, this means that one can take the covariant derivative in the coordinates V and then change to U, or first change to U and then take the covariant derivative in the V coordinates. That is, coordinate transformations and the covariant derivative commute. This property can be used to first transform to a coordinate system that has a known (and preferably simple) connection so that the derivative is simple, and then to transform the result back to the original system. That way, designing a numerical scheme for the covariant derivative should be relatively straightforward.

Given that both the Lie and covariant derivatives are extensions of a directional derivative, one may expect a relation between them. Such a relation indeed exists [54, 97]. It can be formulated as a theorem as follows (the vector field does not have to be divergence-free [54]).

Theorem 2.4.1 (Equivalence of the Lie and covariant derivatives [54])**.** *The Lie derivative of the one-form corresponding to a vector field on a Riemannian manifold differs from the one-form corresponding to the covariant derivative of the field along itself by a complete differential:*

$$L_v(v) = \nabla_v v + \frac{1}{2} d\langle v, v\rangle. \tag{2.55}$$

Here $\langle v, v\rangle$ is the function on the manifold equal at each point x to the Riemannian square of the vector $v(x)$.

Proof. See [54, pp.202-204]. ☐

This theorem gives the relation between the two derivatives. It also shows that this relation involves the differential of a metric. This is the term that essentially encapsulates how the vector v changes during transport. This is also the reason why the Lie derivative requires a frozen vector field (frozen with respect to the transporting field); it does not account for the changing vector field during transport. The result is that only the covariant derivative is suitable when describing compressible flows unless additional terms are added to the Lie derivative expression. See also [73, p.144].

Despite the suitability of the covariant derivative, it has not been used as a discrete advection operator. Some discretisation exist, see for example [98] and the references therein, but these are not used as advection operators. Like the Lie derivative, the covariant derivative disappears from the equations when using the Lagrangian formulation. As such, no a priori choice of advection operator is needed in this work. At the end of this report, in Chapter 7, some notes will be given on the relation between the method, and the Lie and covariant derivatives.

Several operations on forms have been discussed in this chapter, alongside the discretisation of some of those operations. Nevertheless, since differential forms constitute a branch of mathematics on their own, many more topics can be described using forms. These topics are not relevant to the current work, so are not treated here. The interested reader is referred to the books of Frankel [73] and Flanders [72] for a wide variety of topics and applications using differential forms. A topic not often discussed in relation to mimetic methods and differential forms is the role of (non-relativistic) time. Some notes on the role of time are given in the next and final section of this chapter.

2.5. The Role of Time (in Mimetic Methods)

Mimetic methods are generally considered to be spatial discretisation techniques [70]. This thus results in the use of the method of lines to create a semi-discretisation which can be integrated with standard integrators for ordinary differential equations (ODEs), which is common in many numerical methods [99]. However, time integrators play an important role in the conservation properties of numerical methods [96]. It is therefore important to carefully select an integrator, especially for methods that focus on the preservation of conservation laws or symmetries, such as mimetic methods.

The easiest way to deal with the temporal aspect is by circumventing it. This can for example be done by using a Fourier transform [51, 52, 100], which is occasionally used in electromagnetism. However, this approach is not very versatile, so time integration must eventually be dealt with.

There has been considerable research into time integrators. This has led to various time integrators with desirable properties, see for example [96, 101]. An interesting branch of time integration is geometric integration [71, 102] (including the Lie group integrators [70, 103]). These integrators are designed to preserve the underlying properties of the system to be integrated. This objective is very similar to that of mimetic methods. The connection between such integrators and mimetic methods, however, is not clear. Most mimetic methods have used conventional integrators, though some have used a geometric integrator as well, see for example [104, 105]. In general, however, the temporal aspect has received little attention.

While it may seem convenient to focus separately on the spatial and temporal aspects of a scheme, their interaction may cause unexpected results [96, 99]. As opposed to using a separate integrator for the time stepping, it is also possible to create a space-time method. In this case, the temporal and spatial discretisations are blended, and their discretisation is treated with equal importance. Space-time methods are rare in the mimetic community. It was indicated as an interesting area by Lipnikov, Manzini and Shashkov [56] for the MFD, and some space-time aspects are also important in the works of Vazquez-Gonzalez, Llor and Fochesato [106]. Roe [107] recently argued for the use of space-time stencils (especially for hyperbolic equations) for numerical methods in general. Space-time methods are more common outside the mimetic community and seem to gain more interest in recent years.

Despite the rarity of space-time mimetic methods, the time component was extensively treated by Tonti [48, 49] and Mattiusi [45, 108]. Mattiusi's "topological time-stepping" [45, 108] effectively indicates a natural time-stepping procedure, and shows that this time-stepping procedure is not limited to mimetic methods [45, 108]. While he does not use incidence matrices for the time component, this can be done [109] and comes down to the same approach. Lee and Palha [109] also note the similarity of their approach with some geometric integrators by Hairer et al. It is the author's opinion that this connection between (geometric) integrators and mimetic methods, and the role of time in mimetic methods in general, is worth investigating further.

In this work, a space-time approach will be used, as was hinted at by the slightly suggestive notation used in this chapter. This choice is made for three principal reasons. Firstly, advection is an inherently unsteady phenomenon (see Chapter 3), meaning that the temporal discretisation is very important. Secondly, treating space and time on an equal footing will allow staying close to the works of Tonti [48, 49] and Mattiusi [45, 108], which is especially useful for the physical and geometric interpretations. It is also in line with the suggestions of Roe [107]. Thirdly and finally, Lee and Palha [109] indicated that their time integrator, effectively an incidence matrix in time, was energy conserving. This is an important property to investigate. Applying incidence matrices in time was also done by Ferretti with the cell method [60]. It is hoped that this approach provides a clearer insight into the conservation properties of the schemes developed here.

Some caution may be needed when creating a space-time mimetic method because the underlying physics theory in this work is Newtonian. Since Newtonian gravitation does not allow for a true space-time metric [74][6], it is important to realise that the space-time aspect happens in $\mathbb{R}^{3+1}$ (for three spatial dimensions) instead of an $\mathbb{R}^4$, which is more common in electromagnetism (see for example [43]). The space-time forms are therefore a 0-, or 1-form in time and simultaneous a 0-, 1-, 2-, or 3-form in space, as also indicated in for example [110]. Nevertheless, the term space-time will be used throughout this work.

The basic principles behind differential forms and mimetic methods were briefly reviewed in this chapter. Before the new method will be developed, however, some notes on advection and the existing discretisations are in order. These will be given in the next chapter.

---

[6] A mathematical way of saying that time, in Newtonian physics, is absolute.

3

Advection and its Numerical Approximation

As mentioned in Chapter 2, mimetic methods aim to mimic the mathematical and physical properties of the problem under consideration. Therefore, before starting the design of the numerical method, some notes on the advection process and the current numerical (mimetic) methods to solve them are in order. Firstly, in Section 3.1, a brief description of advection in both the Eulerian and Lagrangian formulations will be given. This will provide a starting point for the development of the numerical method and will show the benefit of the Lagrangian formulation. Secondly, in Section 3.2, some characteristic properties of advection will be given. Thirdly and finally, a brief review of conventional and mimetic advection schemes will be given in Section 3.3.

3.1. Eulerian and Lagrangian Advection

Advection, in simple terms the act of transporting something, is ubiquitous in the physical sciences and is often one of the main building blocks of a physical process [1]. Despite 'just' being the act of transporting something, advection can greatly complicate a process. This is of course the case when considering non-linear advection.

Advection can be described using two main formulations; the Eulerian and Lagrangian formulations. These two complementary formulations, or viewpoints, are different descriptions of the same process and are thus related. The Eulerian formulation will be described first, as this is more commonly used in fluid dynamics. However, the Lagrangian formulation turns out to be more natural for advection and will be used throughout the rest of this work.

3.1.1. Eulerian Advection

The Eulerian (or spatial) description is ubiquitous in fluid dynamics [111, 112]. The main reason for using the Eulerian formulation is that it focuses on a spatial region, which leads to a more convenient formulation of the diffusion term in particular. The difficulty with the Eulerian description, however, is advection. Especially when considering fluid dynamics, the advection term will be non-linear and can cause extreme phenomena, such as shocks, to occur.

The advection operator can be written as the total (or advective) derivative $\frac{D}{Dt}$ in the Eulerian formulation as

$$\frac{D}{Dt} = \frac{\partial}{\partial t} + u \cdot \nabla, \tag{3.1}$$

where t is the temporal coordinate, u is the advection velocity, and ∇ the spatial gradient. The rightmost term in the equation is generally considered the advective part, though advection cannot take place without time. After all, it takes some time to move something from one place to the next.

The Eulerian view of advection is often compared to standing at a crossroads where cars pass. When keeping your focus on the crossroads, you will notice various cars and other traffic pass. This is the typical Eulerian view. A similar situation arises when considering fluid flow. The focus will be on one spatial region, while the mass of fluid moves through the domain.

Another way of looking at advection is by considering some simple advection equations. Take for example the linear advection equation in one-dimension

$$\frac{D\phi}{Dt} = \frac{\partial \phi}{\partial t} + u\frac{\partial \phi}{\partial x} = 0, \tag{3.2}$$

with t and u as above, x the spatial coordinate, and ϕ the property to be transported. For linear advection, ϕ is not a function of u. In the simplest case, u will be constant.

Another commonly used example is Burgers' equation. There are now two main ways in which to write this. These are the so-called advective form

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = 0, \tag{3.3}$$

and conservative form

$$\frac{\partial u}{\partial t} + \frac{1}{2}\frac{\partial u^2}{\partial x} = 0. \tag{3.4}$$

These forms are mathematically equivalent, though the non-linearity in the conservative form is more obvious. In fact, the advective form is actually quasi-linear [113].

The above-mentioned advective and conservative forms of the advection equation are only two such forms. There are many different and mathematically equivalent formulations that can be used, especially in multiple dimensions [114]. Theoretically, there is no difference between these formulations. However, when discretised, the different formulations can lead to numerical methods with very different properties [96, 114–116]. As a result, there are a variety of different numerical methods, though one is not guaranteed to be better than the other.

One can consider the Lagrangian formulation as yet another way of writing the advection term. This turns out to be very natural, as will be shown next.

3.1.2. Lagrangian Advection

The Lagrangian (or material) description is much less common in fluid dynamics, though fairly common in oceanography [117–119]. This is because describing three-dimensional viscous fluid flow in this formulation is complicated [117]. However, the one exception is advection [117]. Advection is naturally described in the Lagrangian formulation. This already becomes clear when looking at the total derivative, which now becomes

$$\frac{D}{Dt} = \frac{\partial}{\partial t}. \tag{3.5}$$

In the Lagrangian formulation, only a time derivative remains. This greatly simplifies the description of advection, in particular when considering non-linear problems.

The Lagrangian view of advection can also be understood when looking at the crossroads analogy. While the Eulerian view involved looking at a fixed region and seeing various cars pass through, the Lagrangian view focuses on individual cars. The same car is thus always in view as it moves through various spatial regions. This focus on an individual car becomes a focus on individual fluid particles[7], leading to a natural description of fluid dynamics.

The individual fluid particles are tracked by assigning a label to each particle. While any label can be used [117], the most common choice is the initial position. Since each particle is then identified by its label, the Lagrangian formulation thus has information about where a particle comes from. That is, the Lagrangian formulation includes a particle history. This is an important difference between the Lagrangian and Eulerian formulations and will have an interesting consequence later in this work.

Looking again at the linear advection equation, which can now be written as

$$\begin{cases} \dfrac{\partial \phi}{\partial t} = 0, & (3.6a) \\ \dfrac{dX(t)}{dt} = u, & (3.6b) \end{cases}$$

---

[7] Note that the term 'fluid particles' (and later also 'particle') refers to a fluid element that consists of a very large number of molecular particles and there is no influence from individual molecular particles. A fluid particle is thus considered a piece of the continuum, similar to the definition used in [112]. This term will be used throughout this work.

some important changes are visible. Firstly, the spatial coordinate is now moving, hence its time dependence. Secondly, the system has been split into two first-order equations. While this may seem somewhat extreme for the linear advection equation, it is the main reason for the advantage of the Lagrangian formulation. Additionally, it underlies the method of characteristics [113].

The position variable, indicated as $X(t)$, is known as the flow map Φ. That is, the mapping gives the position of a particle when the label and current time are given as inputs. Formally, the flow map of fluid dynamics can be considered a diffeomorphism since fluid particles cannot cavitate, superimpose, or jump [120, 121]. Note, however, that this may not hold when considering idealised flows, such as inviscid flows.

The split of the linear advection equation into two equations makes it easier to analyse. In Equation (3.6a), the advected quantity ϕ is shown to remain unaltered during transport. The second equation, Equation (3.6a), shows that the location is being changed. Both statements are obvious in this formulation. They also hold true in the Eulerian formulation, Equation (3.2), but are less obvious from the equation.

Similarly, Burgers' equation can be written as

$$\begin{cases} \dfrac{\partial u}{\partial t} = 0, & \text{(3.7a)} \\ \dfrac{dX(t)}{dt} = u. & \text{(3.7b)} \end{cases}$$

The split into two simple equations is again evident. In this case, an interesting feature of Burgers' equation becomes immediately obvious: the velocity of a particle (which is being tracked along its own path, a characteristic of the equation) does not change. This is not at all obvious in the Eulerian formulation (Equations (3.3) and (3.4)) because the Eulerian formulation does not consider the velocity of a fluid particle. Instead, it chooses a fixed point and investigates the velocity of the particle that happens to be there. This of course changes and is the reason behind the non-linearity. Such problems do not occur in the Lagrangian formulation, making it a natural starting point for advection. Following individual fluid particles also means that their history can be tracked, a feature that is absent in the Eulerian formulation.

3.1.3. Relation between Eulerian and Lagrangian Advection

As the Lagrangian and Eulerian formulations describe the same phenomenon, they are directly related. This relation can be expressed as [117]

$$u_L(a_j, s|t) = \frac{\partial X_i(a_j, s|t)}{\partial t} = u_E(X_k(a_j, s|t), t) \tag{3.8}$$

where u_L and u_E are the Lagrangian and Eulerian velocities, respectively, a_j a label given to the Lagrangian particle at time s and the equation is evaluated at time t. In words, this equation says that the Eulerian velocity is equal to the velocity of the Lagrangian particle that occupies the place where the Eulerian velocity is evaluated (in both space and time).

Using the partial differential equation (PDE), switching between the Lagrangian and Eulerian formulations is relatively easy. However, when using an integral formulation, differentiation of integrals must be used. In the general form, this is known as the Reynolds transport theorem (see for example [122]) and can be written as

$$\frac{d}{dt}\int_{\Omega(t)} \phi(t, x)\, d\Omega = \int_{\Omega(t)} \frac{\partial \phi(t, x)}{\partial t}\, d\Omega + \oint_{\Gamma(t)} \phi(t, x)(\hat{n} \cdot v)\, d\Gamma, \tag{3.9}$$

where $\Omega(t)$ is a moving domain with $\Gamma(t)$ its boundary, $\hat{n}$ an outward pointing normal vector, and v the velocity of the boundary. Differentiation under the integral sign was also expressed using differential forms by Flanders [123] (see also [73]).

Next to the differential and integral relations, one can also wonder about the equivalence of the weak form. This form is fundamental for finite element methods. Unfortunately, however, only in the one-dimensional case have the two formulations been proven to be equivalent [124] (away from vacuums at least). Nevertheless, the wealth of numerical work on multi-dimensional Lagrangian/Eulerian formulations seems to indicate that the two formulations are equivalent in this case as well.

3.2. Some Characteristics of Advection

Now that the Eulerian and Lagrangian formulations of advection have been explained, a brief look into the characteristics of advection is warranted. This will help clarify what advection alone can do and what phenomena one can expect while studying advection.

3.2.1. Hyperbolicity and Shocks

Advection equations are well known to have a hyperbolic character. This means that the solution at any future time is completely determined by the starting conditions [125]. This is clear in the Lagrangian formulation of Burgers' equation (Equation (3.7)). The velocity is constant in time according to Equation (3.7a), so that the velocity (of a particle) is immediately known at all time. The solution can thus be easily obtained directly from the starting conditions.

The hyperbolic character also has a physical consequence; there is no interaction between the different particles. That is, the path taken by a given particle is completely independent of all the other particles. This can be used as a significant advantage when investigating pure advection equations, as will be done in Chapter 5.

Another important consequence of the hyperbolic character is the tendency to generate shocks [125]. Take again Burgers' equation. Since the velocities are fixed from the beginning, a particle may overtake its neighbour at some time, depending on the starting velocities. Strictly speaking, given the non-interacting character of the equation, the particles can (mathematically) pass the same point at the same time and continue unhindered. This would create a multi-valued solution [125]. Physically, such solutions make no sense, so different solutions are sought. Such 'weak solutions' (more on this in Chapter 5) have a discontinuous profile. That is, they include shocks. The tendency to create shocks is unique to hyperbolic equations and requires the absence of viscosity. The numerical simulation of shocks is very difficult, and only a brief initial look into this will be given in Chapter 5.

3.2.2. Frozen Flows

In his essay [118], Price distinguishes three main modes of advection. For the cases considered here, the frozen flow mode is the most interesting. The term frozen flow has indeed already appeared in Section 2.4.2 when discussing the Lie derivative. Here, this term will be briefly investigated.

The term frozen flow (or sometimes frozen field or wave) appears throughout the fluid dynamics literature in a wide array of applications. Let's consider some examples. The term appears in chemically reacting flows [126]. In this case, a frozen flow is a flow in which the reaction rate is zero. The term frozen refers to the chemical composition of the flow; the chemical composition does not change in space and time (that is, it is frozen). The chemical composition of the fluid may still vary via diffusion if present. The term frozen flow also arises in the study of flow instabilities [127]. The frozen wave instability is a variation of the Kelvin-Helmholtz instability that, instead of rolling up into a series of vortices and breaking down into turbulence, produces more regular standing (that is, frozen) waves that do not roll up [127]. Another example comes from the field of atmospheric turbulence, where the frozen flow hypothesis is used to describe atmospheric turbulence on short time scales [128]. The turbulent pattern is said to be frozen in each atmospheric layer [128]. These examples show that, despite the different applications, the meaning of the term frozen field is very similar; a frozen field is a field that is transported by a fluid without alteration.

Looking back at the Lie derivative (Section 2.4.2), it now becomes clear that this operator is suitable for use in cases such as those just mentioned. However, frozen flows are only one aspect of advection, so the Lie derivative may not always be the appropriate operator.

In particular, the general case of compressible fluid dynamics has a non-linear term in the momentum equation. The momentum is by no means frozen with respect to the velocity so the Lie derivative cannot be used in this case.

3.2.3. Conservation Properties

The last aspect of advection to briefly consider is its role in conservation properties. As mentioned before, mimetic methods try to preserve conservation laws in a discrete setting. Knowing which conservation laws are expected to hold is then an important aspect.

The conservation laws are most commonly expressed in the Eulerian formulation, though it is also possible to consider them using Lagrangian variables, see for example [129]. The most important conservation laws are the primary conservation laws, or the first integrals. These are essentially the governing equations. For Burgers' equation, the conservation properties were extensively studied by Hopf [130]. Hopf showed that there is one first integral for the viscous Burgers' equation, which is the governing integral. This integral remains conserved in the inviscid case, though a second first integral appears. This will be explained in more detail in Chapter 5, as it is related to the computation of shocks.

Next to the primary conservation laws are the secondary laws, or second integrals. These properties are derived from the main equations. For fluid dynamics, important secondary conservation properties are for example related to kinetic energy. Other quantities are higher powers of the transported quantity, the so-called Casimir invariants [131], which are also conserved.

As far as advection is concerned, advection does not contribute to the global conservation of energy [114]. This important aspect is what often goes wrong in the discretised equations, leading to instabilities and inaccuracies. It can thus be a challenge to design suitable advection schemes that preserve the right conservation laws.

3.3. Numerical Methods for Advection

Now that mimetic methods, and the MSEM in particular, have been explained along with advection, the last thing to review before diving into the new method are existing advection schemes. Of course, as this work concerns a mimetic method, the focus lies on mimetic advection schemes, though some notes on conventional schemes are also in order.

3.3.1. Mimetic Methods

Mimetic methods, as was mentioned in Chapter 2, are known to have certain desirable properties related to conservation [46, 67], stability [68–70], and accuracy [70, 71]. Such properties became known over the course of several decades worth of development. While a complete historical overview is too extensive, some notes on the directions of development can place certain advances in perspective. For additional historical notes, see for example [56].

Several important realisations led to mimetic methods as they are now known. For starters, trying to retain mathematical and physical structures in the discretised equations is of fundamental importance. While this is not unique to mimetic methods, this was an important aspect in the early development of most numerical methods [9]. Then, the work of Tonti on the analogies between physical theories and the importance of geometrical associations provides a solid geometric basis [48, 49, 65]. This can be put into the more formal framework of differential forms, the use of which was advocated by various authors [43, 64, 72, 123, 132]. Together, these elements provide a theoretical basis. The link to numerical methods was made by Bossavit and collaborators [67, 81, 133]. Of particular importance are the realisations that Whitney forms (see [134] for a recent review) can serve as discrete differential forms, and the link between the mass matrices of finite element methods and the Hodge matrices of mimetic methods [81].

The first applications considered the linear and elliptic problems of electromagnetism and diffusion. With the aid of the theory of mixed finite element methods [135], various mimetic methods were developed for these model problems. With contributions from Bossavit [93, 136, 137] (even using a Lagrangian formulation [136]), Hiptmair [52, 76, 78, 82], Mattiussi [45, 80, 108], Bochev, Hyman, Shashkov and Steinberg [22, 23, 138–144], and Perot [46, 145, 146], mimetic methods gathered a prominent place in the numerical analysis literature. Despite the various approaches and frameworks, they all achieved excellent results and mimetic methods came to be known as very accurate and stable. With this knowledge on linear elliptic problems, the mimetic community started to investigate problems with advection.

As explained above, the advection equations have a hyperbolic character. This is a very different situation from the previously studied elliptic problems for electromagnetism and diffusion. When blending diffusion and advection, for example, the equations become parabolic. This is yet another type of equation. In short, treating advection, whether on its own or in combination with different terms, changes the character of the equations. This requires a different approach to the development of numerical methods.

Among the earliest mimetic works on fluid dynamics (and thus advection) is the work of a group at Caltech [94, 104, 105, 147]. This work focuses on incompressible flow and use the Lie derivative (see Section 2.4.2) as an advection operator. In particular, their advection scheme uses Bossavit's extrusion-contraction idea [93]. Nevertheless, the use of the Lie derivative makes the schemes limited to incompressible flows and the discretisation introduces errors in non-metric terms.

The Lie derivative has received attention from various authors, such as Heumann and Hiptmair [148, 149], Palha and Gerritsma [150–152], Cotter and Thuburn [121], and Mohamed, Hirani and Samtaney [153], to name a few. The use of Cartan's homotopy formula is particularly popular in these works. Unfortunately, the contraction operator (see Section 2.4.1) does not have a natural and metric-free discretisation. As such, all these works run into similar problems, leading to sub-optimal advection schemes.

Other works that do involve the Lie derivative but in a less prominent role, are those by Heumann and Hiptmair [154–157]. These works also include a semi-Lagrangian scheme. While compressible velocity fields do appear in these works, they are assumed to be known and the advected quantities are only passively transported by the scheme.

More generally applicable advection schemes were also made. These are often based on a more 'engineering'-type approach to the equations, by using a term-by-term discretisation strategy. This led to various schemes applied to fluid dynamics by, for example, Thuburn and Cotter [121, 158–160], and Palha, Gerristma, Lee and Zhang [92, 109, 161–164]. While these schemes were slightly easier to extend, the advection elements are still treated sub-optimally and extensions to compressible flows are scarce due to the difficulties with advection.

Another indication that mimetic advection is not fully understood is the use of conventional advection schemes in combination with mimetic operators [24, 159, 160, 165, 166]. This has the benefit of allowing extension to more complicated phenomena, but such schemes suffer from similar problems as their non-mimetic counterparts.

A final note on some compatible schemes for Lagrangian hydrodynamics in a finite difference setting [167–170] is in order. These appear to be the only mimetic(-like) schemes considering Lagrangian hydrodynamics. Unfortunately, these schemes were so early in the development of mimetic methods that they only considered point values. An important strength of mimetic methods is the use of degrees of freedom associated to other geometric objects than just points, as explained in Chapter 2.

All-in-all, mimetic methods for advection exist but are not what was hoped for, especially given the superior performance in the cases of electromagnetism and diffusion. There is thus room for a new advection scheme, and the lack of space-time Lagrangian methods makes this formulation particularly interesting.

3.3.2. Non-Mimetic Methods

Mimetic methods barely have a decade worth of experience with advection equations. Conventional numerical methods, however, have been applied to advection equations and the equations of fluid dynamics for several decades. It is therefore no surprise that the body of literature concerning non-mimetic methods is considerably larger than that concerning mimetic methods. Due to the large number of papers, a full review is impossible, so only some relevant notes are given. For more extensive reviews, especially concerning high-order methods, see for example [5, 6, 171–173].

It is well-known that conventional numerical methods often suffer from dispersion and dissipation errors, and this is part of the motivation of this work. For the workhorse methods of CFD, finite volume methods [174], the choice is often made to use dissipative schemes. The dissipation error helps stabilise the scheme, especially near discontinuities. This is the reason that artificial viscosity is often used as a stabilisation term. While extra stability sounds beneficial, overly dissipative schemes are unsuitable for use in sensitive flows and long-time simulations, as they remove vital information from the solution. This can, even with the relatively advanced grad-div stabilisation, lead to negative effects on the computation of lift and drag [175]. Schemes with a dispersive error do not lose information like that but can lead to instabilities and blow-ups, especially near discontinuities. Being able to prevent both errors is thus a formidable task.

Artificial viscosity was originally introduced to be able to compute shocks [176] (see also [177–179] for some historical notes and alternative interpretations of artificial viscosity). The idea was straightforward; add enough dissipation to prevent a true discontinuity from occurring. Since then, the form of shock capturing has been the standard [172]. Unfortunately, artificial viscosity will eventually affect other parts of the solution as well. Additionally, when investigating sensitive problems or problems with very different phenomena, artificial viscosity may become problematic [180]. For example, the class of (weighted) essentially non-oscillatory schemes have received a lot of attention since their introduction [171, 181, 182], especially in combination with shocks. However, the (implicit) addition of artificial viscosity causes such schemes to be too dissipative for use in shock-turbulence simulations [180], and can negatively affect the computations of contact discontinuities [183]. This is a clear example that current high-order schemes are by no means perfect, especially when shocks are involved.

One way of reducing the need for artificial viscosity, at least away from shocks, is choosing the most convenient form of the advection term. It was briefly mentioned before that the discretised forms of the advection equation may lead to different properties [96, 114–116]. In the case of non-mimetic methods, the Lagrangian formulation has received a reasonable amount of attention as well [184]. The Lagrangian formulation can indeed be used to create a very accurate advection scheme [131]. However, mesh distortion and difficulty with other terms have led to few applications in CFD. More interesting for CFD are blended schemes, which perform part of the calculations using the Lagrangian formulation and the other part using the Eulerian formulation. A particular version of this is the arbitrary Lagrangian-Eulerian (ALE) formulation [185, 186], which has received a reasonable amount of attention in recent years. For the time being, ALE schemes will remain part of the non-mimetic literature only, as the geometric content of this formulation is currently unknown.

It is interesting to note that some existing non-mimetic methods can be rewritten as a mimetic method. One example of such a method is the finite difference time domain scheme developed by Yee [187]. It was shown by for example Bossavit [188, 189] and Mattiussi [45, 108] that this scheme is indeed a mimetic method in disguise. It was not until this was fully realised that the success of Yee's scheme was fully understood. Another example is the space-time conservation element and solution element method developed by Chang [190–192]. This method was developed for fluid dynamics and has since been applied to a variety of challenging problems [193]. Given the effort put into respecting the conservation laws and ensuring that the method inherits most of the properties of the equation that is to be solved, it is no surprise that this method is essentially a mimetic method too. Other examples can be found in the literature, see for example [80, 83]. Whatever the link between the methods is, these relations indicate that one can always learn from other interpretations and previous work, even if the underlying basis was not the same.

Now that some aspects of existing schemes have been highlighted, it is time, at last, to start developing the new method. The starting point will be the linear advection equation, which is treated in the next chapter.

4

Linear Advection

This chapter is the first in which the development of the method is discussed. As is common when developing a numerical method for advection, the starting point is advection in its simplest form; linear advection with a constant velocity. The next step is then the non-linear (inviscid) Burgers' equation. However, this results in a relatively big step in complexity. Therefore, an additional step is made in this work. After discussing constant velocity linear advection in Section 4.1, some spatially varying velocity linear advection problems are discussed in Section 4.2, followed by some brief temporally varying velocity linear advection problems in Section 4.3. The last two cases add some complexity due to potential stretching or compression of the flow without introducing non-linearities.

4.1. Constant Velocity Linear Advection

The well-known constant velocity linear advection equation is the starting point for the development of many numerical methods, and this work is no exception. This equation is used to set up the first steps of the method and the corresponding code. There is no physical interpretation behind the constant velocity linear advection equation. In fact, it shares more similarities with a body in the vacuum of space that has been given an initial push with nothing to change its velocity or path. The simplicity of the equation allows the focus to be on the setup of the numerical method instead of the complexities of the equation.

The constant velocity linear advection equation in the Eulerian frame can be written as

$$\frac{\partial \phi}{\partial t} + \frac{\partial u\phi}{\partial x} = 0, \tag{4.1}$$

$$\frac{\partial \phi}{\partial t} + u\frac{\partial \phi}{\partial x} = 0, \tag{4.2}$$

where Equation (4.2) follows from Equation (4.1) because the advection velocity u is constant. In both equations, ϕ is the transported quantity and t and x are the temporal and spatial coordinates, respectively. These equations are to be solved on the space-time domain $\Omega : [t_{\text{start}}, t_{\text{end}}] \times (-\infty, \infty)$ subject to the initial condition $\phi(0, x) = \phi_0(x)$. The general solution to this equation is well known and can be written as $\phi(t, x) = \phi_0(x - ut)$.

In the Lagrangian frame, the advection equation becomes

$$\begin{cases} \dfrac{\partial \phi}{\partial t} = 0, & \text{(4.3a)} \\ \dfrac{dX(t)}{dt} = u, & \text{(4.3b)} \end{cases}$$

where $X(t)$ is the position. For this set of equations, the initial condition has two parts. The first is $\phi(0, X) = \phi_0(X)$, and the second is $X(0) = X_0$. In the Lagrangian frame, it is immediately clear that the value of ϕ remains constant for all time, since, from Equation (4.3a), $\phi(t, X) = \phi(0, X)$. This value of ϕ is transported according to Equation (4.3b) via $X(t) = X_0 + ut$. This equation also defines the flow map $\Phi_f = X(t)$ that will be used in the development of the method.

Solving the linear advection equation with a constant velocity in the Lagrangian frame with the MSEM will require some preparation. As explained in Section 2.3.3, the basis functions are generally expressed in a reference domain and mapped to the physical domain using transfinite interpolation [90, 91]. For the cases considered in this chapter, the mappings will be known analytically and this will be used in the numerical method. In the case of Lagrangian advection, the physical domain will move and the domain could potentially deform. In the case of constant velocity linear advection, the physical domain can only move. Since the reference domain can never deform (it would not be much of a reference if it did), the mapping from the reference to the physical domain will be done in two steps. The first will be from the reference domain to the starting domain in the physical space. The second map will be the advection or flow map, which will move the original domain to the new location. The total mapping is the composition of the two. This is schematically shown in Figure 4.1.

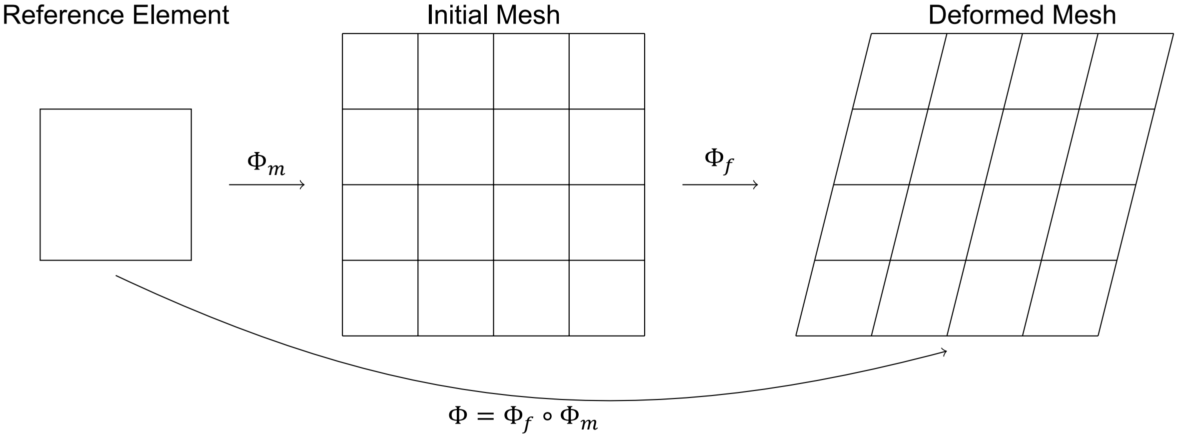


Figure 4.1: Schematic overview of the composition of the mesh map and the flow map.

Equation (4.3a) will give, after discretisation, the coefficients of the transported quantity in the reference domain. There are two options for the discretisation of ϕ in space; nodal polynomials and edge polynomials. It turns out that, for this specific case, both options will work. These will be discussed in order, starting with the use of a nodal expansion. Using ϕ_h to indicate the discretised ϕ, ϕ_h is then defined using a degree-p expansion as

$$\phi_h(\tau,\xi) = \sum_{i=0}^{p}\sum_{j=0}^{p}\phi_{ij}h_i(\tau)h_j(\xi), \tag{4.4}$$

where ϕ_{ij} are the coefficients of the expansion and are defined as $\phi_{ij} = \phi(\tau_i,\xi_j)$. Note that ϕ is expanded in both the time and space directions. However, for the time part, only an expansion in nodal polynomials makes sense because a time derivative is taken meaning that an edge polynomial in time will result, see [11] for the same principle but in space. Since no operation on the spatial part is required there are indeed two options[8]. Using an expansion in time allows the solution to be constructed at any moment in time. This can be advantageous in general but is even more so for a Lagrangian method since no Courant–Friedrichs–Lewy (CFL) restriction is present for these methods. The time step can thus be made larger, while the solution can still be reconstructed at intermediate moments.

Equation (4.3a) can be written using an incidence matrix that represents the time derivative, denoted $\mathbb{E}_t^{1,0}$ (see Section 2.3.2 for the construction of the incidence matrices). The discrete version of Equation (4.3a) can then be written as

$$\mathbb{E}_t^{1,0}\phi_{ij} = 0, \tag{4.5}$$

or, by separating the initial condition, as

$$\mathbb{E}_t^{1,0}\phi_{ij} = \mathbb{E}_{t,ic}^{1,0}\phi_j, \tag{4.6}$$

---

[8] When solving more complicated problems, the nature of the variable will dictate the expansion that is called for [48].

where $\mathbb{E}^{1,0}_{t,ic}$ is the initial condition part of the original incidence matrix, and ϕ_j are the coefficients of the initial condition. This equation is for one element. The degrees of freedom for the Lagrange polynomials on the GLL-nodes include the boundary nodes of an element. For the constant velocity linear advection equation, the continuity of two adjacent elements need not be enforced because the elements move independently of each other. However, the Lagrange polynomials allow this (if there are no discontinuities present) and the continuity of the boundaries will be enforced. If this method is extended to more complicated equations, continuity must be enforced. Enforcing element continuity is standard for finite/spectral element methods and there are two main options. The first assembles the element matrices into a global matrix which can be done in various ways, see for example [32] where mass matrices are used. The second is by using a continuity matrix which looks like an incidence matrix. The continuity matrix links the overlapping points in the element matrices. Both approaches are standard for finite/spectral element methods so these approaches are not further discussed, but can for example be found in [194].

Since Equation (4.3b) is solved analytically[9], no further solution step is required. The analytical flow map can be written as

$$X(t) = \Phi^t_{t^n}(t) = X^n + u(t - t^n), \tag{4.7}$$

and the mesh map to a space-time mesh as shown in the middle of Figure 4.1 is, using $\tau \in [-1, 1]$ for the reference time coordinate and $\xi \in [-1, 1]$ for the reference spatial coordinate,

$$t = \Delta t(\tau + 1)/2 + t^n, \tag{4.8a}$$

$$X^n = \Delta x^n(\xi + 1)/2 + x^n_L, \tag{4.8b}$$

where $\Delta t = t^{n+1} - t^n$ is the time step size and $\Delta x = x^n_R - x^n_L$ is the size of the element in the physical space.

The next steps are the reconstruction of the solution on the moved domain and a fixed set of points (the stationary original domain). The reconstruction on the moved domain is straightforward. The solution is reconstructed using Equation (4.4) for the solution values and Equation (4.7) for the new spatial locations. The reconstruction on the original domain is more involved. The x-locations are known, but each time step these points may lie in a different element. Therefore, the first step is to find which points lie in which element. This is done by computing the location of the element boundaries (always corresponding to $\xi = \pm 1$) and checking if a point lies between these boundaries, which is a trivial task in one spatial dimension. Then, the corresponding τ and ξ are determined from the inverse map. Using these reference values, the solution values can again be found using Equation (4.4).

The full procedure is given in Algorithm 4.1.

Besides expanding ϕ using the nodal polynomials, the edge polynomials (see Section 2.3.3) can also be used. In that case the expansion becomes

$$\phi_h(\tau, \xi) = \sum_{i=0}^{p} \sum_{j=1}^{p} \phi_{ij} h_i(\tau) e_j(\xi), \tag{4.9}$$

where $\phi_{ij} = \int_{\xi_{j-1}}^{\xi_j} \phi(\tau_i, \hat{\xi})\, d\hat{\xi}$ with $\hat{\xi}$ a dummy integration variable and ξ_j the j-th GLL-node. The procedure of discretising Equation (4.3a) using an incidence matrix and removing the boundary, as shown in Equations (4.5) and (4.6) remains the same.

Remark. The incidence matrices that represent the time derivative ($\mathbb{E}^{1,0}_t$) in Equations (4.5) and (4.6) will be slightly different in size when using the expansion with edge polynomials. In the nodal spatial expansion case, there are $p + 1$ degrees of freedom per element while there are only p in the edge spatial expansion case.

The degrees of freedom for the edge polynomials are integrals over line elements, so element continuity cannot be enforced. This is different from the nodal expansion case, where enforcing continuity was a choice. For pure advection in the Lagrangian frame, not enforcing element continuity is natural;

---

[9] In this work, the flow map is indeed computed from the analytical expression. A numerical solution of the flow, as long as it is equally exact, would produce the same results.

Algorithm 4.1: 1D linear advection with a constant velocity field using nodal polynomials.

```
Input: $K_t$, $K_x$                                  // Number of time and space mesh elements
Input: $p_t$, $p_x$                                  // Polynomial orders in time and space
Input: $f_{ic}$                                      // Initial condition

/* Step 0: Create incidence matrices and basis functions.                   */
$h_i(\tau)$, $h_j(\xi)$                              // Lagrange polynomials in $\tau$ and $\xi$
$\mathbb{E}_t^{1,0}$, $\mathbb{E}_{t,ic}^{1,0}$      // Incidence matrix without ic and the ic part

/* Step 1: Discretise the initial condition.                                */
$\phi_0(\xi) = \sum_j \phi_j h_j(\xi)$ with $\phi_j = f_{ic}(\xi)$

for time step in $K_t + 1$ do
    /* Step 2: Assemble and solve the first equation.                       */
    for $k$ in $K_x$ do
        $\mathbb{E}_t^{1,0}\phi_{ij} = \mathbb{E}_{t,ic}^{1,0}\phi_j \implies \phi_{ij}$
        $s.t.\ \phi_{i,0}^k = \phi_{i,p+1}^{k-1}$
    end

    /* Step 3: Reconstruction on moved domain.                              */
    for $k$ in $K_x$ do
        $X(t) = \Phi(\tau, \xi)$
        $\phi(\tau, \xi) = \sum_{i,j} \phi_{ij} h_i(\tau) h_j(\xi)$
        $\phi(t, X(t)) = \phi(\Phi^{-1}(t, X(t))) = \phi(\tau, \xi)$
    end

    /* Step 4: Reconstruction on fixed points $x$.                          */
    for $k$ in $K_x$ do
        $x_{\text{bounds}} = \Phi(\tau, \pm 1)$
        $\hat{x} = x \in x_{\text{bounds}}$
        $\hat{\tau}, \hat{\xi} = \Phi^{-1}(t, \hat{x})$
        $\phi(t, x) = \phi(\hat{\tau}, \hat{\xi}) = \sum_{i,j} \phi_{ij} h_i(\hat{\tau}) h_j(\hat{\xi})$
    end
end
```

the elements all trace out their own domain of dependence in one time-step. Since no information crosses the boundary of a domain of dependence, there is no need to enforce continuity. This also allows the solution in each element to be computed in parallel, though the problems considered here are so small that the effect will be negligible and this is thus not implemented.

The mapping used in the edge expansion case is identical to the nodal expansion case and given in Equations (4.7) and (4.8). The procedure to find the τ- and ξ-coordinates for a given set of spatial points is also unaltered. The reconstruction, however, is different. In the reference domain, Equation (4.9) is used instead of Equation (4.4) when expanding in edge polynomials, but the biggest difference is in the transformation to the physical domain. The nodal and edge polynomials transform differently (see Section 2.3.3) since the edge polynomials transform as 1-forms while the nodal polynomials transform as 0-forms. The edge polynomial expansion of ϕ in the reference domain can be written as

$$\phi(\tau, \xi) = \sum_{i,j} \phi_{ij} h_i(\tau) e_j(\xi)\, d\xi, \tag{4.10}$$

emphasising the 1-form nature. Transforming this to the physical domain gives

$$\phi(t, X) = \phi(\Phi(\tau, \xi)) = \phi(\tau, \xi)\left[\frac{\partial \xi}{\partial X}\, dX + \frac{\partial \xi}{\partial t}\, dt\right] = \phi(\tau, \xi)\frac{\partial \xi}{\partial X}\, dX. \tag{4.11}$$

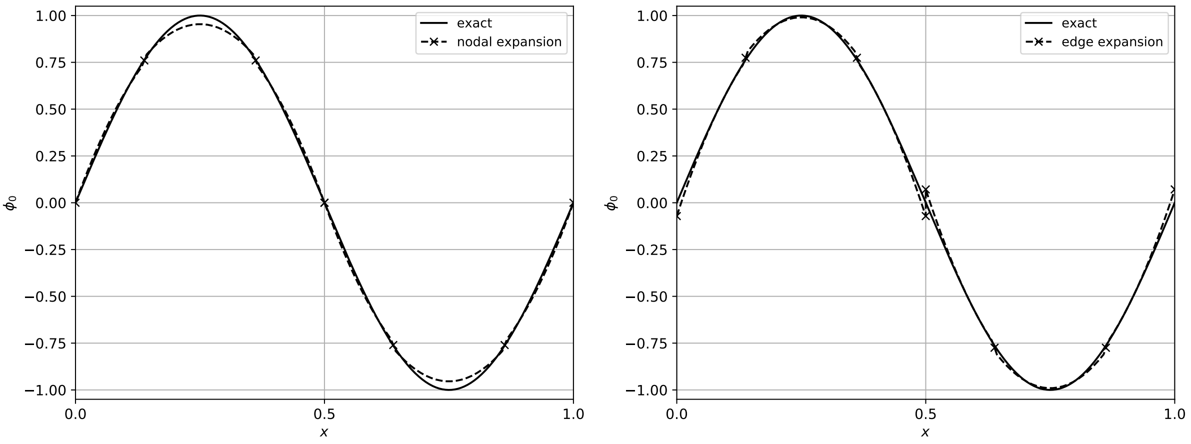
The third and fourth terms show the main difference; there is a Jacobian term in the transformation. The fourth step follows from the third because the ξ-coordinate of a point is independent of time. Because the Jacobian depends on the mapping, the Jacobian can be found analytically and this analytical expression will be used in the method. The final algorithm using edge polynomials is given as Algorithm 4.2.

Remark. Independently of the chosen expansion, the ξ-coordinate together with the element number constitute the Lagrangian labels, similar to an address with a street name and number.

Algorithm 4.2: 1D linear advection with a constant velocity field using edge polynomials.

```
Input: $K_t$, $K_x$                           // Number of time and space mesh elements
Input: $p_t$, $p_x$                              // Polynomial orders in time and space
Input: $f_{ic}$                                                   // Initial condition

/* Step 0: Create incidence matrices and basis functions.                      */
$h_i(\tau)$, $e_j(\xi)$                   // Lagrange and edge polynomials in $\tau$ and $\xi$
$\mathbb{E}_t^{1,0}$, $\mathbb{E}_{t,ic}^{1,0}$                  // Incidence matrix without ic and the ic part

/* Step 1: Discretise the initial condition.                                   */
$\phi_0(\xi) = \sum_j \phi_j e_j(\xi)$ with $\phi_j = \int_{\xi_{j-1}}^{\xi_j} f_{ic}(\xi)\,d\xi$

for time step in $K_t$ do
    /* Step 2: Solve the first equation.                                       */
    for $k$ in $K_x$ do
        $\mathbb{E}_t^{1,0}\phi_{ij} = \mathbb{E}_{t,ic}^{1,0}\phi_j \implies \phi_{ij}$
    end

    /* Step 3: Reconstruction on moved domain.                                 */
    for $k$ in $K_x$ do
        $X(t) = \Phi(\tau,\xi)$
        $\phi(\tau,\xi) = \sum_{i,j} \phi_{ij} h_i(\tau) e_j(\xi)\,d\xi$
        $\phi(t,X) = \phi(\Phi(\tau,\xi)) = \phi(\tau,\xi)\left[\frac{\partial\xi}{\partial X}\,dX + \frac{\partial\xi}{\partial t}\,dt\right] = \phi(\tau,\xi)\frac{\partial\xi}{\partial X}\,dX$
    end

    /* Step 4: Reconstruction on fixed points $x$.                             */
    for $k$ in $K_x$ do
        $x_{\text{bounds}} = \Phi(\tau,\pm 1)$
        $\hat{x} = x \in x_{\text{bounds}}$
        $\hat{\tau},\hat{\xi} = \Phi^{-1}(t,\hat{x})$
        $\phi(t,x) = \phi(\hat{\tau},\hat{\xi}) = \sum_{i,j}\phi_{ij} h_i(\hat{\tau}) e_j(\hat{\xi})\frac{\partial\xi}{\partial X}\,dX$
    end
end
```

Now that the two algorithms have been explained, let's have a look at the results produced for the same simple problem. The initial profile $f_{ic} = \sin(2\pi x)$ is advected with a constant velocity $u = 1$ up to $t_{\text{final}} = 0.55$. Since the main difference between the two algorithms is the reconstruction procedure, the reconstruction of the initial condition using Algorithm 4.1 is shown in Figure 4.2a and with Algorithm 4.2 in Figure 4.2b. These plots are made using two elements and $p = 3$ in both cases, resulting in a 3rd-degree nodal expansion and a 2nd-degree edge expansion. The reconstructions are evaluated at 50 equally spaced points plus the GLL-nodes per element (without duplicates) and the vertical grid line shows the element boundary. This way of generating the reconstruction points will be used for all results in this chapter. In the figures, the GLL-nodes are indicated with a marker. The differences in the reconstructions are obvious. The nodal case, where continuity is explicitly enforced, shows a smooth reconstruction, though not perfectly overlapping the exact profile. The edge expansion case shows a jump on the element boundary. Increasing the polynomial order and/or the number of elements reduces the jump. This jump is there because the edge polynomials are not guaranteed to be continuous over an element boundary. One can also see that the edge polynomial approximation has the same integrated value between the GLL-nodes, whereas the nodal approximation coincides with the exact profile at these points.

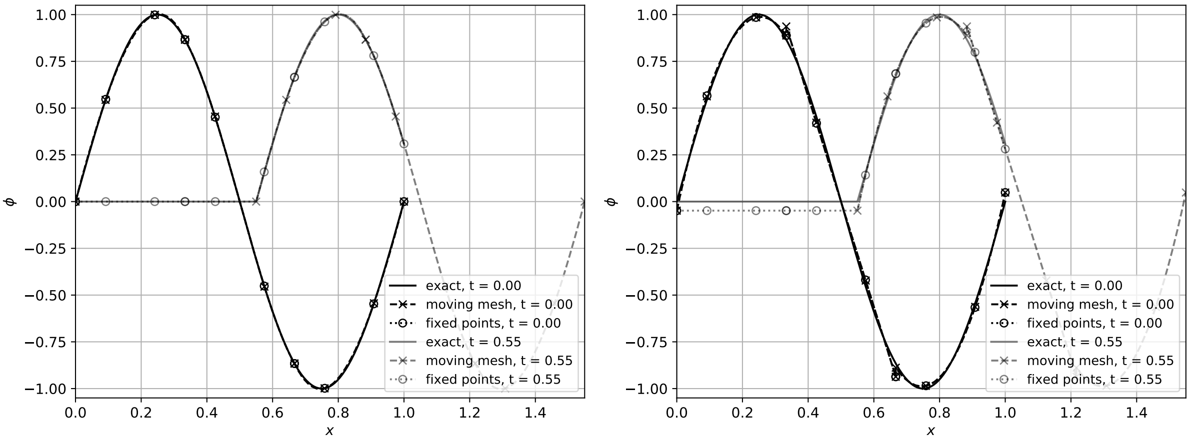


(a) Lagrange polynomials.

(b) Edge polynomials.

Figure 4.2: Reconstruction of the initial condition $\sin(2\pi x)$ using Lagrange and edge polynomials with $p = 3$ on two elements. The vertical grid line shows the element boundary. The solution/interpolation values at the GLL-nodes are indicated with a marker.

The difference in the reconstruction of the initial condition is the main difference between the two algorithms. Using $p = 3$ and three elements, the solutions at $t = 0$ and $t = 0.55$ using the Lagrange polynomials are shown in Figure 4.3a and using the edge polynomials in Figure 4.3b. In both cases, the approximations of the sine shape remain unaltered during the simulation and only the location changes. As in Figure 4.2, the edge polynomial reconstruction shows a small jump on the element boundary. This does not affect the simulation, nor does the jump change in size over time.

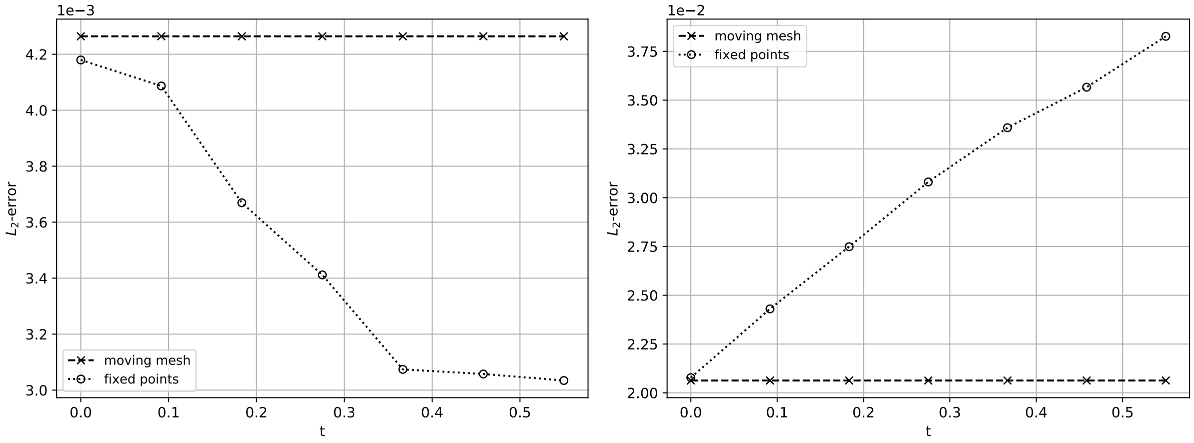


(a) Lagrange polynomials.

(b) Edge polynomials.

Figure 4.3: Solutions at $t = 0$ and $t = 0.55$ for constant velocity linear advection with a sinusoidal initial profile using $p = 3$ on three elements.

To more clearly show that the solutions remain unaltered, the L_2-errors with respect to the exact solution are shown in Figure 4.4a for the Lagrange polynomials and in Figure 4.4b using the edge polynomials. The errors computed for the moving mesh use the new locations, so the boundary treatment is not taken into account. For the fixed points, if part of the domain is no longer covered by the moving mesh, the value of the reconstruction closest to this region is used. The same procedure is used for the exact solution. The figures show that, on the moving mesh, the solutions do not change at all. On the fixed points, the error for the Lagrange polynomials decreases, while the error for the edge polynomials increases. The reason for this is the boundary treatment. As one can see from Figure 4.3, the value on the left boundary (where a GLL-node lies) is exact for the Lagrange polynomials, while it is too low for the edge polynomials. During the simulation, the profile moves out of the domain, making this point the most dominant, explaining the different error behaviours on the fixed points. This is the consequence of a moving mesh method. In this work, the results on the moving mesh are the more important ones. Note also that the value of the errors (on the moving mesh in particular) is now an order of magnitude smaller for the Lagrange polynomials than for the edge polynomials. This is caused by the use of $p = 3$, which is a third-degree reconstruction for the Lagrange polynomials, but only a second-degree reconstruction for the edge polynomials.



(a) Lagrange polynomials.

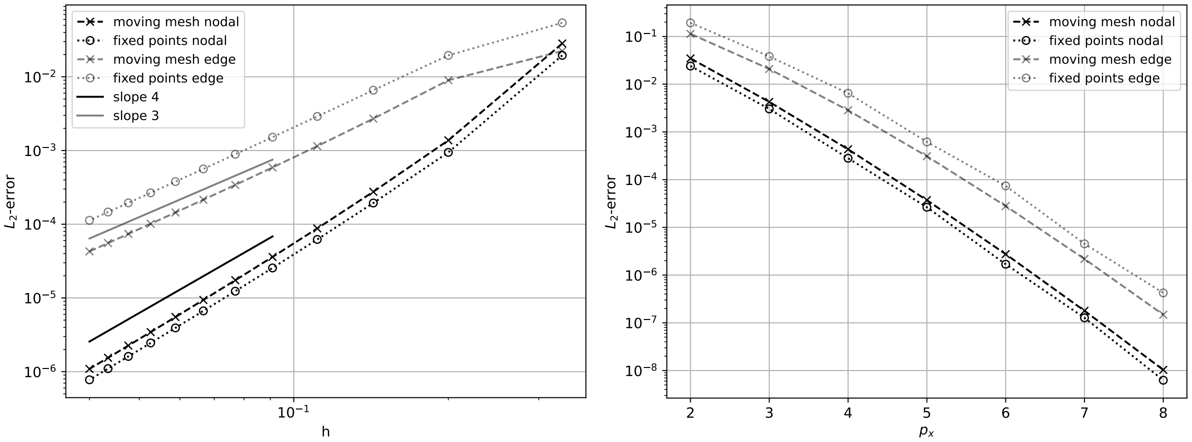
(b) Edge polynomials.

Figure 4.4: L_2-errors over the time interval $t = [0, 0.55]$ for constant velocity linear advection with a sinusoidal initial profile.

The value of the errors can be decreased by increasing the polynomial degree and/or the number of elements (hp-refinement). The L_2-error for this case at $t = 0.55$ is shown in Figure 4.5a for h-refinement and in Figure 4.5b for p-refinement. For both the Lagrange expansion as well as the edge expansion the convergence using both h- and p-refinement follows similar curves on a moving mesh and a fixed set of points. Note that the axes of Figure 4.5a have a log-log scale and that the horizontal axis has smaller values of the mesh size as one goes to the left. The h-refinement shows a clear linear convergence rate as the element size decreases, with slopes of about 4 for the nodal expansions and 3 for the edge expansions. These convergence rates are optimal for polynomials of degree 3 and 2, respectively. The p-refinement shows spectral behaviour, as indicated by the straight line in the semi-log plot of Figure 4.5b. It is interesting to note that in the p-refinement case, the edge polynomial curve (on the moving mesh) appears to be shifted to one degree higher, but the values are similar. The shift by one polynomial degree is easily explained by the fact that the edge polynomial creates a $p-1$-degree interpolation, whereas the Lagrange polynomials create a p-degree interpolation. The similarity of the error values is expected to be related to the similarity of basis functions under an affine mapping (which the flow map for linear advection is). A similar observation, albeit for different basis functions, was made by Cotter and Thuburn [121], and this affine-equivalence property is more general [52].

Remark. The errors just discussed can only come from two sources because the mapping and its Jacobian are computed analytically; errors in the expansion coefficients (ϕ_{ij}) and errors in the interpolation. When discretising the initial condition, the expansion coefficients for the nodal expansion are found by evaluating the initial condition at the GLL-nodes. This gives the exact coefficients (thanks to the Kronecker property, Equation (2.30)). During the simulation, the expansion coefficients are never altered, thanks to the structure of the incidence matrix which only consists of the values $-1, 0, 1$. From Equation (4.5) this results in $\phi_{ij}^n = \phi_{ij}^{n+1}$, so that the expansion coefficients always remain exact.

The situation is similar for the edge expansion. The expansion coefficients of the initial condition are now found by integrating between the GLL-nodes. While this is done using numerical quadrature



(a) h-refined with $p = 3$.

(b) p-refined with three elements ($h = 1/3$).

Figure 4.5: L_2-errors at $t = 0.55$ for constant velocity linear advection with a sinusoidal initial profile with h- and p-refinement.

in the implementation, it can also be computed exactly (the initial condition is analytically known). More importantly, Equation (4.5) again indicates that $\phi_{ij}^{t^n} = \phi_{ij}^{t^{n+1}}$ due to the incidence matrix, though the number of degrees of freedom will be slightly different. Nevertheless, the expansion coefficients remain unaltered during the simulation.

The fact that the expansion coefficients remain unaltered is essentially what is shown in Figure 4.4 (the moving mesh cases). The other error term, the interpolation error, can be made arbitrarily small by using hp-refinement. This is what is shown in Figure 4.5. This leads to the important result that the method described so far gives a topological advection scheme; it is independent of the actual basis functions (they are only used for the reconstruction) and it introduces no error during the simulation.

With this excellent result, the algorithm can now be extended to other cases. No other examples of the constant velocity linear advection problems are shown since the results will be similar to the above. It can also be concluded that the most important aspect to control the error behaviour is an appropriate discretisation of the initial condition. This can be achieved by choosing a suitable grid and polynomial order. At this stage, both expansions can be used and give good results. In the next section, the velocity field will be made dependent on the spatial coordinate and only one expansion will produce the correct result.

4.2. Spatially Varying Velocity Linear Advection

The case of a spatially varying velocity field for linear advection is not commonly used, but was first studied by Zoppou and Knight [195]. Their method was recently extended by Corrigan, Fullard and Lynch [196] to include more variations of the velocity field and thus test cases. Both show the use of the spatially varying velocity field for testing numerical methods for advection.

The linear advection equation with a spatially varying velocity field in the Eulerian formulation is

$$\frac{\partial \phi}{\partial t} + \frac{\partial u(x)\phi}{\partial x} = 0, \tag{4.12}$$

$$\frac{\partial \phi}{\partial t} + u(x)\frac{\partial \phi}{\partial x} + \phi\frac{\partial u(x)}{\partial x} = 0, \tag{4.13}$$

and in the Lagrangian formulation

$$\begin{cases} \dfrac{\partial \phi}{\partial t} = 0, & \text{(4.14a)} \\ \dfrac{dX(t)}{dt} = u(X(t)). & \text{(4.14b)} \end{cases}$$

The velocity field is always compressible (never divergence-free) in one spatial dimension since the only way for $\frac{\partial u(x)}{\partial x}$ to equal zero is to be constant, making the velocity field spatially invariant.

Some of the eight 1D cases with four different velocity fields presented in [196] will be used here. Table 4.1 gives an overview of the cases presented in [196]. The explicit solutions can be found in [196, Table 1]. The first case uses a simple linear (in x) velocity field and is discussed next.

Table 4.1: Overview of the test cases presented in [196]. Note that a is a constant and H is the Heaviside step function.

case	initial profile f_{ic}	velocity field $u(x)$
1	1	ax
2	1	ax^2
3	1	$0.5\left(H(2x-1)+1\right)$
4	1	$\operatorname{sech}\left(a(2x-1\right)$
5	x	x^2
6	$0.5\left(H(1-2x)+1\right)$	x
7	$0.5\left(H(1-2x)+1\right)$	x^2
8	$0.5\left(H(ax-1)+1\right)$	$0.5\left(H(2x-1)+1\right)$

4.2.1. Linear Velocity Fields

The first case in [196] (and Table 4.1) uses $u(x) = ax$ with a constant (the constant is indicated with k in [196]). The initial profile is a constant function. However, because the velocity now varies in the domain, the initial profile will be stretched and subsequently decrease to satisfy the conservative form of the advection equation. The exact solution is $u(t,x) = e^{-at}$ [196]. Note that, because the initial condition is constant, the final solution does not depend on x while the velocity does.

This case can essentially be solved by applying the methods as developed in Section 4.1. The mapping, as obtained by using the method of separation of variables (see for example [113]) to solve Equation (4.14b), can be written as

$$\Phi^t_{t^n} = X^n e^{a(t-t^n)}. \tag{4.15}$$

There is, however, one very important difference; *only the expansion using edge polynomials produces the correct solution*. Using a nodal spatial expansion will move the nodes to the correct location (the maps are still analytical), but it will not capture the decay of the solution values. The edge expansion does capture this decay due to the Jacobian term in the expansion, as shown in step 3 of Algorithm 4.2.

To see this more clearly, let's compare the two reconstructions on the moving domain. These can be written as

$$\phi(t,X(t)) = \phi(\Phi^{-1}(t,X(t))) = \phi(\tau,\xi) \tag{4.16}$$

for the Lagrange polynomials and

$$\phi(t,X) = \phi(\Phi(\tau,\xi)) = \phi(\tau,\xi)\left[\frac{\partial\xi}{\partial X}\,dX + \frac{\partial\xi}{\partial t}\,dt\right] = \phi(\tau,\xi)\frac{\partial\xi}{\partial X}\,dX \tag{4.17}$$

for the edge polynomials. These are only the reconstructed solution values. The corresponding locations are given by the analytical mapping, which is the same in both cases. Note that in Equation (4.17) the dX-term simply indicates that ϕ is a 1-form. The main difference between the two reconstructions is thus $\frac{\partial\xi}{\partial X}$, the Jacobian of the mapping. Note that the Jacobian for the mapping of Equation (4.15) can be written as $\frac{\partial\xi}{\partial X} = \frac{2}{\Delta x}e^{-a\Delta t}$. The Δx-part of the Jacobian accounts for the fact that ϕ is expanded as a 1-form, while the e-part ensures the decay of the function values in the physical domain. This explains the difference between the two reconstructions. Because the correct solution can only be obtained by using the edge polynomials, only results for the edge polynomials will be shown throughout Section 4.2.

Remark. The solution using 0-forms is considered wrong. However, the solution is exactly what one would expect for 0-forms. A 0-form (for example the temperature) is associated to a point. The concept of stretching or compression is entirely foreign when talking about a point. If a temperature field were to be transported by this velocity field, the temperature would not be affected by the divergence of the velocity. Conversely, a 1-form, such as the density in 1D, *is* affected by the divergence and one would indeed expect the value to vary if an element is stretched or compressed. Note that the solutions given by Corrigan, Fullard and Lynch [196] are for their variable ρ, which indicates a tracer density.

The mathematical counterpart to this physical reasoning can be easily shown using the Lie derivative[10], in particular using Cartan's homotopy formula (see Equation (2.51)). When expanding ϕ as a 0-form, the following equation results

$$\frac{\partial \phi^0}{\partial t} + \mathcal{L}_u \phi^0 = \frac{\partial \phi^0}{\partial t} + i_u d\phi^0 + d i_u \phi^0 = 0, \tag{4.18}$$

where i_u is the contraction operator (see Section 2.4.1) and d is the exterior derivative (see Section 2.2.3). For a 0-form, the contraction operator returns zero, hence

$$\frac{\partial \phi^0}{\partial t} + i_u d\phi^0 = 0. \tag{4.19}$$

This equation corresponds to

$$\frac{\partial \phi}{\partial t} + u\frac{\partial \phi}{\partial x} = 0, \tag{4.20}$$

which is not correct for a velocity field that depends on x. However, when expanding ϕ as a 1-form the equation becomes

$$\frac{\partial \phi^1}{\partial t} + \mathcal{L}_u \phi^1 = \frac{\partial \phi^1}{\partial t} + i_u d\phi^1 + d i_u \phi^1 = 0. \tag{4.21}$$

Noting that the exterior derivative in one dimension returns zero for a one form gives,

$$\frac{\partial \phi^1}{\partial t} + d i_u \phi^1 = 0. \tag{4.22}$$

This expression now corresponds to

$$\frac{\partial \phi}{\partial t} + \frac{\partial u\phi}{\partial x}, \tag{4.23}$$

which is the sought-after equation. This clearly shows, from a mathematical point of view, why only the 1-form expansion produces the correct results for this case.

The results for this case can thus only be found using Algorithm 4.2. Using an initial condition of $f_{ic} = 1$, $p = 3$, three elements, and choosing the reconstruction points as before, the results up to $t_{\text{final}} = 2.0$ become as shown in Figure 4.6. The result clearly shows the decay of the solution value over time.

The L_2-error with respect to the exact solution is shown in Figure 4.7. At first glance, it may appear as if the error decreases over time. However, all the errors are already on the level of machine precision (1×10^{-16}). This makes sense since the initial condition (a constant) can be seen as a zeroth-order polynomial so that the reconstruction is always exact and no interpolation error will be made. This is in line with the remark at the end of the previous section. Since the error will always be on the order of machine precision, there is no point in investigating the convergence behaviour. Instead, the next section will discuss velocity fields that vary non-linearly in x.

4.2.2. Non-Linear Velocity Fields

The other three velocity fields used by Corrigan, Fullard and Lynch are non-linear in x [196], see Table 4.1. The options are $u(x) = ax^2$, $u(x) = \operatorname{sech} a(2x-1)$, and $u(x) = 0.5(H(2x-1)+1)$. The first field will be discussed in detail, while the others are only treated briefly because the idea behind the algorithms is the same for all non-linear fields.

Unfortunately, the method developed in Section 4.1 can no longer be directly applied. This is because the computation of X^n for the maps no longer holds. In all cases discussed so far, X^n was computed directly from the reference domain by using Equation (4.8b). However, the non-linearity introduced in this section prevents this from working. Instead, a composition of multiple flow maps must be used, on top of the composition of the mesh and one flow map. The composition of two flow maps is schematically shown in Figure 4.8.

---

[10] Note that this is still linear advection, that is, the advected quantity is independent of the velocity field. The use of the Lie derivative is thus appropriate.

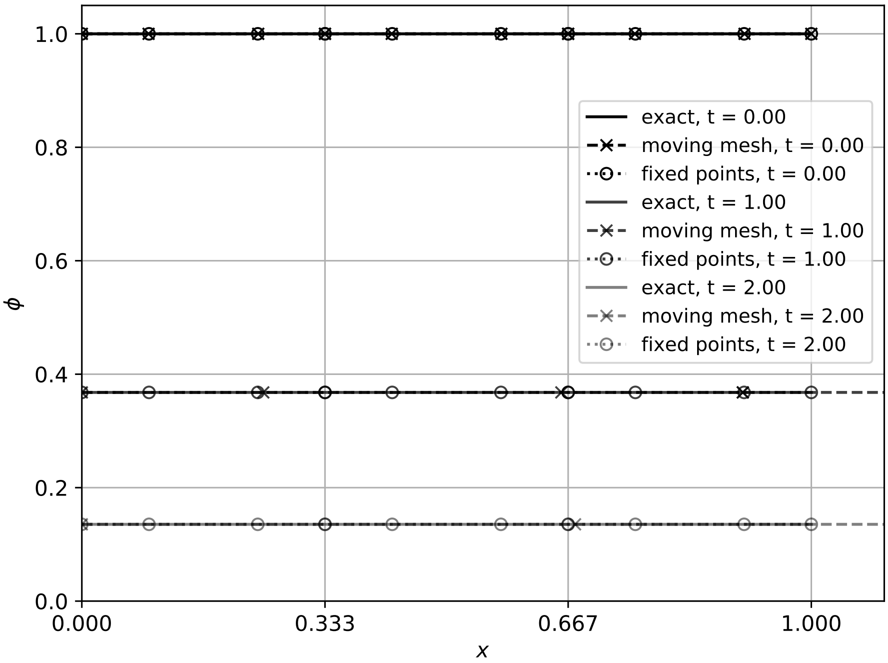


Figure 4.6: Solutions at $t = 0, 1, 2$ for the linear-in-x-velocity advection problem with $f_{ic} = 1$ and using $p = 3$ on three elements. The grid lines indicate the edge boundaries at $t = 0$ and the solution values at the GLL-nodes are indicated with markers.

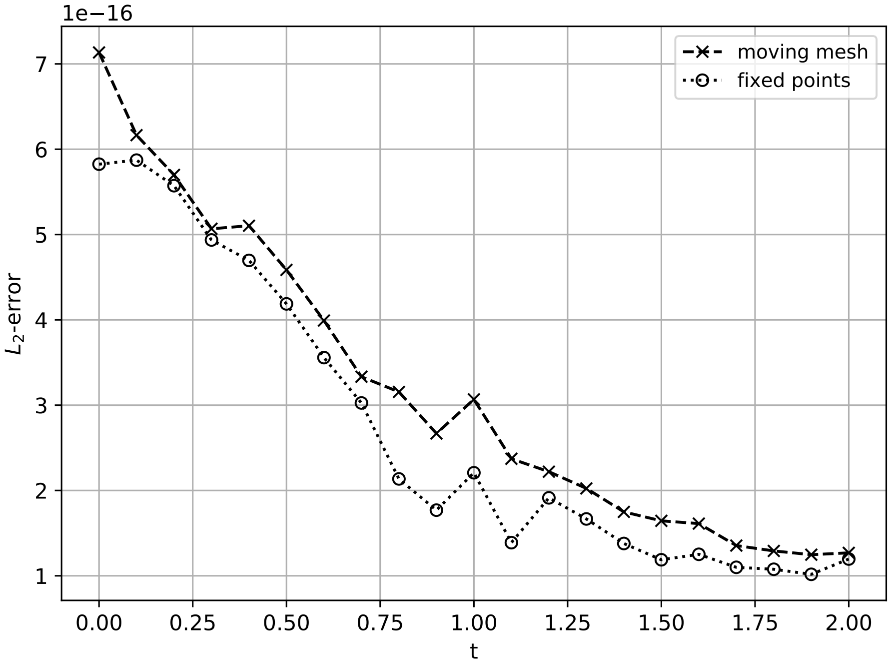


Figure 4.7: The L_2-error over time for the linear-in-x-velocity advection problem with $f_{ic} = 1$ and using $p = 3$ on three elements.

In equation form, the composition becomes

$$\Phi_{ref}^{t} = \Phi_{t^n}^{t} \circ \Phi_{t^{n-1}}^{t^n} \circ \dots \circ \Phi_{t^0}^{t^1} \circ \Phi_{ref}^{t^0}, \tag{4.24}$$

with the Jacobian becoming

$$J_{ref}^{t} = J_{t^n}^{t} J_{t^{n-1}}^{t^n} \dots J_{t^0}^{t^1} J_{ref}^{t^0}. \tag{4.25}$$

The last term in both equations is the mapping (respectively the Jacobian) from the reference domain to the initial mesh in the physical domain. For this map only, Equation (4.8) can still be used. The Δt

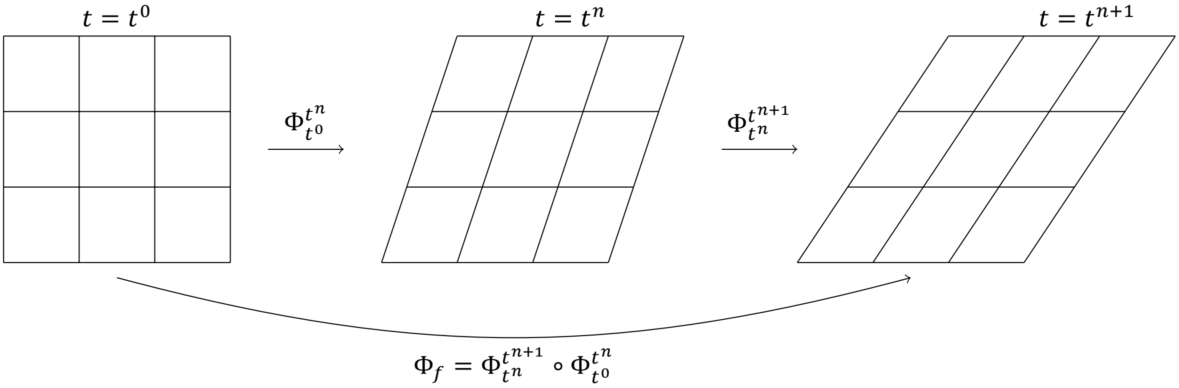


Figure 4.8: Schematic overview of the composition of multiple flow maps.

and Δx are user inputs and indicate the time step and cell size at the starting time. Neither need to be constant and the mesh size will vary over time.

It turns out that this flow map composition technique had been previously used. This started with the work of Brunton and Rowley [197], who used flow map composition as a method to speed up the computations of the finite time Lyapunov exponent (FTLE)[11]. This work also used a Lagrangian formulation and the composition is like that used here, though no advection scheme was developed. Through the subsequent works of Leung and coworkers [201–204], who focused on creating an Eulerian method based on this idea, Kulkarni and Lermusiaux [205] eventually came up with an advection scheme using the flow map composition technique. However, that advection scheme was only designed to passively advect tracers with a given velocity field [205]. Nevertheless, some interesting properties may carry over to the framework used here, which goes in a different direction.

Other than the composition of the flow maps, the algorithm can stay largely unaltered. The only other change is in the reconstruction of the solution on a fixed set of points. While still possible, the inverse mapping, which is now a composition of all previous steps, must be used to trace the fixed points back to the reference domain, where the reference coordinates can be found and the solution can be reconstructed. This result must then be mapped back to the current time step again. This can be computationally expensive, especially if many time steps are used.

Like before, the flow map and Jacobian are still computed analytically. While these change if the velocity changes, using a different initial condition for the same velocity does not require a change of flow map and Jacobian. As a result, looking at Table 4.1, only four unique flow maps are required. These maps are given in Table 4.2. For completeness, the mapping for the constant velocity case of Section 4.1 is also included. In the lower right entry of Table 4.2, A is

$$A = \sqrt{(\sinh\left(a(2X(t) - 1) - 2a(t - t^n)\right)^2 + 1}, \tag{4.26}$$

for brevity.

The full procedure is given in Algorithm 4.3. This algorithm only uses the edge polynomials, as was explained in Section 4.2.1. The creation of the incidence matrices and basis functions remains unaltered, so are not included for brevity.

With the algorithm and mappings defined as above, the test cases of Table 4.1 can be investigated. Discussing all cases is too lengthy and will not always lead to new insights. Therefore, only the results of cases 4 and 5 are discussed.

Case 4 is chosen because it was labelled as challenging in [196]. The reason for this is the secant velocity. The secant function creates a very sharp peak in the profile, as can be seen in the results shown in Figures 4.9a and 4.9b, at $t = 5$ and $t_{\text{final}} = 10$, respectively. As the initial condition is simply a constant, it is not shown for clarity of the final results. For some simple schemes tested in [196],

---

[11] For a brief introduction to Lagrangian coherent structures (LCS) and the FTLE in particular, see for example the note by Peacock and Haller [198], the review by Haller [199], the related review on chaotic advection by Aref et al. [1], or the computational introduction by Onu et al. [200].

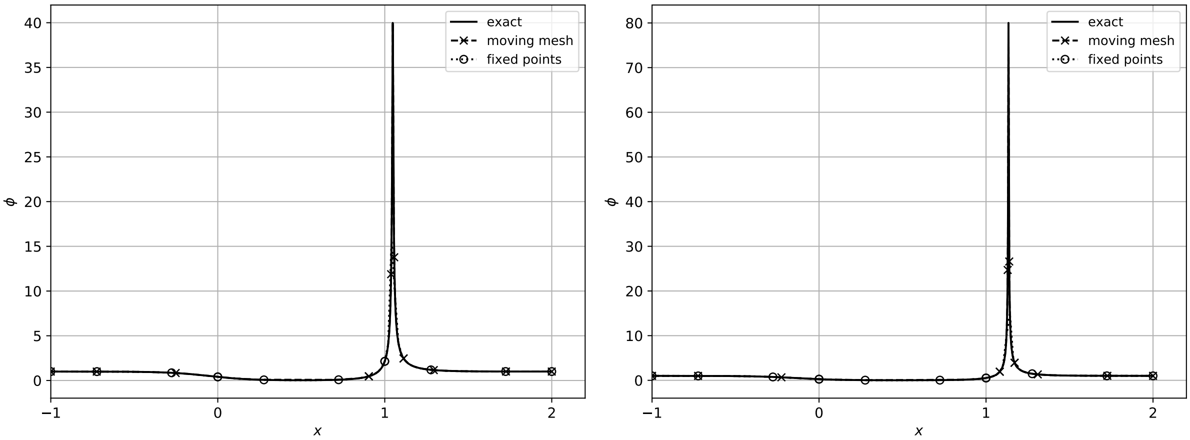
Table 4.2: Overview of the mappings and Jacobians used throughout this chapter. As before, a is a constant and H is the Heaviside step function. Note that $J_t^{t^n}$ is given and that $J_0^{ref} = \frac{2}{\Delta x_0}$.

velocity field $u(x)$	flow map $\Phi_{t^n}^{t}$	inverse Jacobian $J_t^{t^n}$
a	$X^n + a(t - t^n)$	1
ax	$X^n e^{a(t-t^n)}$	$e^{-a(t-t^n)}$
ax^2	$\frac{X^n}{1-aX^n(t-t^n)}$	$\frac{1}{(1+aX(t)(t-t^n))^2}$
$0.5\,(H(2x-1)+1)$	$\begin{cases} X^n + \frac{1}{2}(t-t^n) \text{ if } X(t) < \frac{1}{2} \,\&\, X^n < \frac{1}{2} \\ 2X^n - \frac{1}{2} + (t-t^n) \text{ if } X(t) \geq \frac{1}{2} \,\&\, X^n < \frac{1}{2} \\ X^n + (t-t^n) \text{ if } X^n \geq \frac{1}{2} \end{cases}$	$\begin{cases} 1 \text{ if } X(t) < \frac{1}{2} \,\&\, X^n < \frac{1}{2} \\ \frac{1}{2} \text{ if } X(t) \geq \frac{1}{2} \,\&\, X^n < \frac{1}{2} \\ 1 \text{ if } X^n \geq \frac{1}{2} \end{cases}$
$\operatorname{sech}(a(2x-1)$	$\frac{1}{2a}\operatorname{asinh}(\sinh(a(2X^n-1)+2a(t-t^n)) + \frac{1}{2}$	$\frac{\cosh(a(2X(t)-1)}{A}$

Algorithm 4.3: 1D linear advection with a non-linear (in x) velocity field.

```
/* Define the number of mesh elements and polynomial orders in space
   and time, the initial condition, and $J_0^{ref}$.                   */
Input: $K_t$, $K_x$, $p_t$, $p_x$, $f_{ic}$, $J_0^{ref}$

/* Step 1: Discretise the initial condition.                           */
$\phi_0(\xi) = \sum_j \phi_j e_j(\xi)$ with $\phi_j = \int_{\xi_{j-1}}^{\xi_j} f_{ic}(\xi)\, d\xi$

for time step in $K_t$ do
    /* Step 2: Solve the first equation.                               */
    for $k$ in $K_x$ do
        $\mathbb{E}_t^{1,0}\phi_{ij} = \mathbb{E}_{t,ic}^{1,0}\phi_j \implies \phi_{ij}$
    end

    /* Step 3: Reconstruction on moved domain.                         */
    for $k$ in $K_x$ do
        $X(t) = \Phi(\tau, X^n)$
        $\phi(\tau,\xi) = \sum_{i,j} \phi_{ij} h_i(\tau) e_j(\xi)\, d\xi$
        $J^{-1} = J^{-1} \cdot \frac{\partial X^n}{\partial X(t)}$
        $\phi(t,X) = \phi(\Phi(\tau,\xi)) = \phi(\tau,\xi) J^{-1}\, dX$
    end

    /* Step 4: Reconstruction on fixed points $x$.                     */
    for $k$ in $K_x$ do
        $x_{\text{bounds}} = \Phi(\tau, \pm 1)$
        $\hat{x} = x \in x_{\text{bounds}}$
        $\hat{\tau}, \hat{\xi} = \Phi_t^{ref} = \Phi_t^{t^n} \circ \ldots \circ \Phi_{t^1}^{t^0} \circ \Phi_{t^0}^{ref}(t, \hat{x})$
        $J_{stat}^{-1} = J_t^{ref} = J_t^{t^n} \circ \ldots \circ J_{t^1}^{t^0} \circ J_{t^0}^{ref}(t, \hat{x})$
        $\phi(t,x) = \phi(\hat{\tau},\hat{\xi}) = \sum_{i,j} \phi_{ij} h_i(\hat{\tau}) e_j(\hat{\xi}) J_{stat}^{-1}\, dX$
    end
end
```

the large gradients made the computations difficult. This is not a problem here. However, two main aspects typical for Lagrangian schemes can be seen in the results. The first is that the resolution near the peak is fairly high. This ensures that the peak is reasonably well captured, which is a problem for the Eulerian schemes tested in [196]. On the other hand, the starting domain used for the Lagrangian calculation was slightly extended to the left. This was needed to prevent the mesh from leaving the leftmost part of the domain. This adaption to the flow features is typical for Lagrangian schemes and can be both beneficial as well as limiting. After all, while more resolution around the peak is helpful, having no points at all in the left part of the domain is not. The resolution problem is also visible in the solution plotted on the fixed points. As the same reconstruction points are used, the peak soon becomes too sharp to see on the stationary mesh. The adaptivity of the Lagrangian formulation is thus mostly beneficial in this case.



(a) Solution at $t = 5.0$.

(b) Solution at $t = 10.0$.

Figure 4.9: Solutions for the hyperbolic secant-in-x-velocity advection problem with $f_{ic} = 1$ and using $p = 3$ on three elements. The grid lines indicate the edge boundaries at $t = 0$ and the solution values at the GLL-nodes are indicated with markers.

The result for case 5 is shown in Figure 4.10 up to $t_{\text{final}} = 6.0$. In this case, the starting profile is not a constant, but linear in x. As the result is again shown using a polynomial degree of three, the representation of the initial condition is again exact. Coupled with the analytical mapping and Jacobian, the error in the simulation is again machine precision for the entire domain and duration, and on the moving mesh as well as the fixed points.

These results show that the developed method is very accurate. Additionally, an interesting property can be deduced from the results; monotonicity preservation with respect to the discretised initial condition. That is, the method does not create new minima or maxima during the simulation, other than those warranted by the divergence. This is an important property for an advection scheme, and as such is formulated as a theorem.

Theorem 4.2.1. *The method developed so far is shape-preserving (monotonicity-preserving) as defined in [206] with respect to the discretised initial condition. Thuburn [206] indicates that the shape of the advected quantity ϕ is preserved according to the advective form of the advection equation*

$$\frac{D\phi}{Dt} = 0. \tag{4.27}$$

In the Lagrangian formulation used here, this corresponds to

$$\frac{\partial\phi}{\partial t} = 0. \tag{4.28}$$

Proof. This is a direct consequence of the structure of the incidence matrix and was essentially given in the remark at the end of Section 4.1. The structure of the incidence matrix ensures that $\phi_{ij}^{n} = \phi_{ij}^{n+1}$ for any time step (the same incidence matrix is used).

Only the divergence can alter the shape of ϕ [206]. In the current method, the divergence is represented by the Jacobian of the mapping, which is known analytically. Therefore, the result also holds in the case of a non-divergence-free velocity. ☐

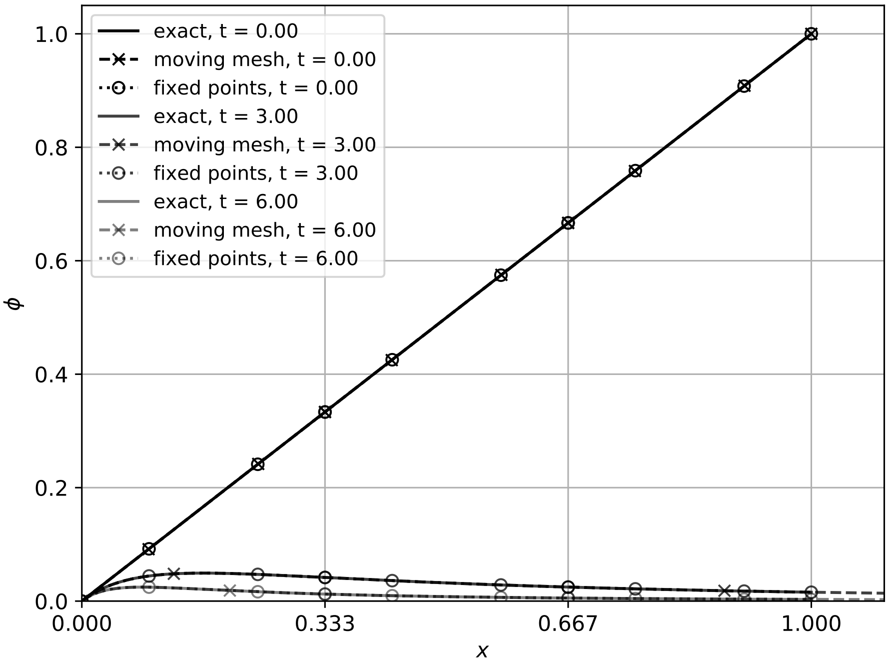


Figure 4.10: Solutions at $t = 0, 3, 6$ for the quadratic-in-x-velocity advection problem with $f_{ic} = x$ and using $p = 3$ on three elements. The grid lines indicate the edge boundaries at $t = 0$ and the solution values at the GLL-nodes are indicated with markers.

Remark. The remark at the end of Section 4.1 still holds, albeit the Lagrange polynomials can not be used for non-divergence-free velocity fields. While the total mapping is now a composition of different flow maps, its Jacobian can be found using Equation (4.25) and is still analytical. Since the expansion coefficients are not altered and the interpolation error can again be made arbitrarily small, the developed method (when using the edge polynomials) is still a topological advection scheme in the case of non-divergence-free velocity fields. From this point of view, the monotonicity property of Theorem 4.2.1 does not come as a surprise.

This section showed how the algorithms change when the velocity becomes dependent on the spatial coordinate. One may then wonder, especially considering the space-time character of the method, what happens when the velocity becomes time-dependent. This will be investigated in the next section.

4.3. Temporally Varying Velocity Linear Advection

After the work done in the previous section using velocity fields that depend only on the spatial coordinate, one may wonder what happens when the velocity field depends on the temporal coordinate instead. This will be briefly discussed in this section.

For a velocity field that depends only on the temporal direction, the advection equation can be written as

$$\frac{\partial \phi}{\partial t} + \frac{\partial u(t)\phi}{\partial x} = 0, \tag{4.29}$$

$$\frac{\partial \phi}{\partial t} + u(t)\frac{\partial \phi}{\partial x} = 0, \tag{4.30}$$

$$\begin{cases} \dfrac{\partial \phi}{\partial t} = 0, & \text{(4.31a)} \\ \dfrac{dX(t)}{dt} = u(t), & \text{(4.31b)} \end{cases}$$

where Equations (4.29) and (4.30) are in the Eulerian frame and Equation (4.31) is in the Lagrangian frame. Note that, as opposed to the spatially varying velocity fields, the temporally varying velocity fields are always divergence-free. This means that the initial profile will not deform and can only move.

The method used in [196] to find solutions for a spatially varying velocity field does not allow the velocity to depend on time, so no temporally varying velocity field is available. In order to test whether the algorithm can handle temporally varying velocity fields, a simple solution is found by using a velocity varying as $\cos at$, where a is again a constant. Note that this velocity is a non-linear function of the temporal coordinate. Using this velocity, the advection equation can be written as

$$\frac{\partial \phi}{\partial t} + \cos at \frac{\partial \phi}{\partial x} = 0. \tag{4.32}$$

The solution to Equation (4.32) can be found using the method of characteristics and gives a general solution of $\phi(t,x) = \phi_0\left(x - \frac{1}{a}\sin(at)\right)$ where $\phi_0(x)$ is the initial condition, see Appendix A.1. The solution will look like the initial condition that is being moved back and forth with a period of 2π and it will coincide with the initial condition every π seconds.

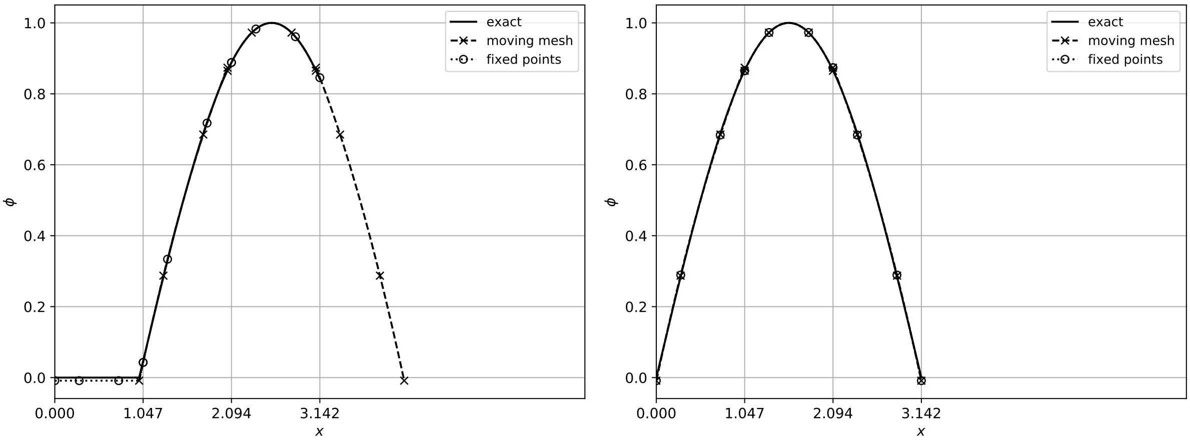
The theoretical solution is slightly different from what is used for the numerical method because the theoretical solution is from $t = 0$ while the numerical method uses the solution from $t = t^n$. Solving Equation (4.31b) with $u(t) = \cos at$ gives the flow map from t^n to t as

$$\Phi_{t^n}^{t} = x^n + \frac{1}{a}(\sin(at) - \sin(at^n)), \tag{4.33}$$

where x^n is the position of the solution at time t^n.

Contrary to the spatially varying velocity field where the solution was expanded in edge polynomials in the spatial direction, the solution does not need to be expanded using the edge polynomials in the temporal direction. This is because the mapping of the temporal coordinate remains unchanged. This, combined with the divergence-free property of the velocity field, makes this case similar to the constant velocity advection case of Section 4.1. When using the temporally varying velocity field, the solution can also be expanded using only nodal polynomials.

The temporally varying velocity field was used with an initial condition $\phi_0(x) = \sin(x)$ on a domain $[0,\pi]$. This creates one arc of a sine function that is moved back and forth and into and out of the domain. The results are shown in Figures 4.11a and 4.11b at $t = 0.5\pi$ and $t = 2\pi$, respectively. Note that the solution for $t = 2\pi$ is identical to the initial condition, as it should. Because the temporally varying velocity field does not require a new algorithm, its convergence behaviour is as before, in particular as in Section 4.1 where an initial condition with a sine-function was also used, so that it won't be shown here.



(a) Solution at $t = 0.5\pi$.

(b) Solution at $t = 2\pi$.

Figure 4.11: Solutions for the variable-in-t-velocity advection problem with $f_{ic} = \sin(x)$ and using $p = 3$ on three elements. The grid lines indicate the edge boundaries at $t = 0$ and the solution values at the GLL-nodes are indicated with markers.

Remark. The remarks at the end of Sections 4.1 and 4.2 also hold for temporally varying velocity fields, though there is no need to expand the variables in temporal edge functions.

The method developed in this chapter presents a good stepping stone for the development of a fully numerical advection scheme. Because the expansion coefficients of the advected quantities are not altered and the interpolation error can be made arbitrarily small, the method is a topological advection scheme. The importance of expanding the advected variables as 0-forms or 1-forms (in 1D) was shown by discussing non-divergence-free velocity fields, where it was found that the Jacobian of the mapping plays an important role. While the flow map and its Jacobian were found by analytical means, the integral coefficients of the edge polynomials (discrete 1-forms) allowed for an easy combination with the Jacobian. Now that linear advection can be accurately solved, Burgers' equation will be investigated in the next chapter.

5

Inviscid Burgers' Equation

The work described in the previous chapter has led to a well-working numerical method for solving the advection equation with constant, spatially or temporally varying velocity fields. This method will now be extended to Burgers' equation, which includes non-linearities in its solution.

Burgers' equation has been a standard test problem for non-linear advection for a long time, and much has been learned about this equation since [207, 208]. In this work, only the inviscid version of Burgers' equation will be discussed. The inviscid version is a hyperbolic advection equation like those discussed in Chapter 4, but the velocity is now equal to the advected quantity. This is sometimes called self-advection in the literature [131]. Burgers' equation can be written in the Eulerian frame as

$$\frac{\partial u(t,x)}{\partial t} + \frac{1}{2}\frac{\partial u^2(t,x)}{\partial x} = 0, \tag{5.1}$$

$$\frac{\partial u(t,x)}{\partial t} + u(t,x)\frac{\partial u(t,x)}{\partial x} = 0, \tag{5.2}$$

and in the Lagrangian frame as

$$\begin{cases} \dfrac{\partial u(t,X(t))}{\partial t} = 0, & \text{(5.3a)} \\ \dfrac{dX(t)}{dt} = u(t,X(t)). & \text{(5.3b)} \end{cases}$$

In both formulations, the non-linearity[12] is obvious.

One reason for the popularity of Burgers' equation is the availability of analytical solutions [130, 208–210]. For the inviscid equation, the method of characteristics can be used to obtain solutions. This works well up to the formation of the shock. It turns out that a solution found by the method of characteristics is also a solution after the shock forms, though it is no longer single-valued and does not show the shock [125, 211, 212]. Instead, it will show a self-overlapping profile where the solution will become triple-valued. While this may occasionally appear like a physical solution (for example for the height of a water wave), it is not and indicates the breakdown of the model (or its assumptions) used to derive the equation [125, 211, 212]. It is, however, the only (strong) mathematical solution [211, 212]. The nonphysical solution is often 'fixed' using a so-called weak solution, which leads to the inclusion of a shock. A discussion of such solutions will be postponed till Section 5.3.

In Section 5.1 the method developed in the previous chapter will be applied to Burgers' equation. This will be used to gain an initial understanding of the difficulties posed by the non-linearity of Burgers' equation. Additionally, the effect of expanding the advected variable as a 0- or 1-form will be investigated. Then, in Section 5.2, the association of the variables to oriented geometric elements will be used to create a completely numerical method. Many concepts introduced in Chapters 2 and 3 will be used explicitly. Lastly, in Section 5.3, the formation of shocks and their numerical treatment will be briefly investigated.

---

[12] Strictly speaking, Burgers' equation is quasi-linear [113].

5.1. Direct Application of the Current Method

Burgers' equation can be solved with the algorithm developed in the previous chapter up to the moment where the shock occurs. Note that the non-linear aspect of Burgers' equation is very similar to the non-linear dependence of the velocity in Section 4.2 on x. The non-linearity is introduced only through the reconstruction. As a result, Algorithms 4.1 and 4.2 can be used and only the mapping needs to be adapted.

An interesting problem to consider is the inviscid Burgers' equation with a sinusoidal initial profile. Here, the initial profile is $u_0(x) = \sin(x)$ with $x \in [0, 2\pi]$. The solution to this problem can be easily found using the method of characteristics and can be written as

$$\begin{cases} u(t, X(t)) = u(t = 0, X(0)), & \text{(5.4a)} \\ X(t) = X(0) + u(t, X(t))t = X(0) + \sin(X(0))t. & \text{(5.4b)} \end{cases}$$

The flow map required for Algorithms 4.1 and 4.2 is found in a similar way and can be written as

$$X(t) = X(t^n) + \sin(X(0))(t - t^n). \tag{5.5}$$

Note that the inverse Jacobian of the mapping, $\frac{\partial X(t^n)}{\partial X}$, is equal to 1 so that the flow can be considered divergence-free. This also means that the solution can be expanded in nodal and edge functions (0- and 1-forms) and that this problem is similar to that of Section 4.1. The result is shown in Figure 5.1. This figure shows that the numerical solution is again very close to the exact solution for both expansions.

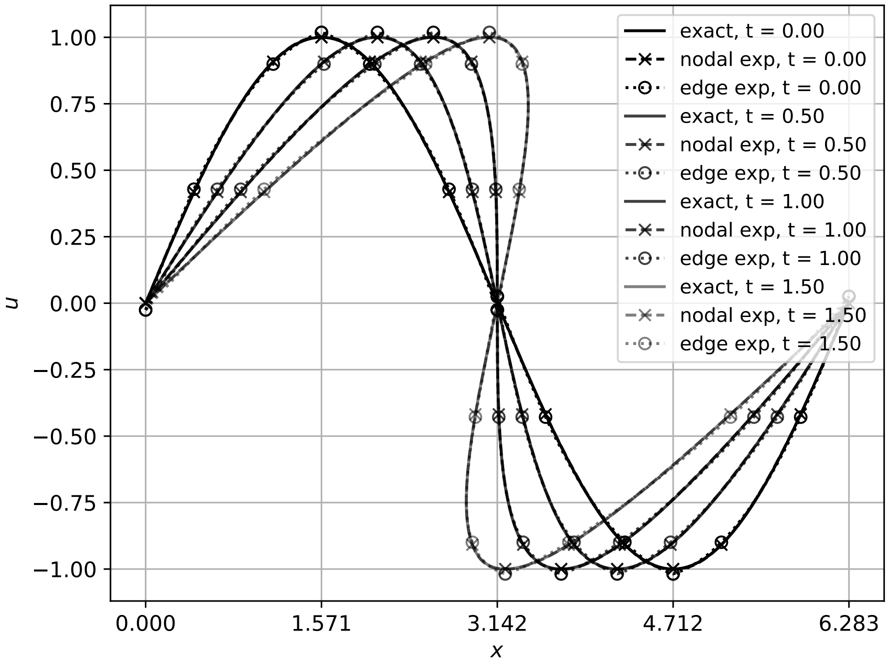


Figure 5.1: Solutions at $t = 0$, $t = 0.5$, $t = 1.0$, and $t = 1.5$ for Burgers' equation with a sinusoidal initial profile using $p_t = 2$ and $p_x = 3$ on four elements. The grid lines indicate the edge boundaries at $t = 0$ and the solution values at the GLL-nodes are indicated with markers.

The behaviour of the algorithm in terms of convergence is similar to that of the example in Section 4.1 and is therefore not shown here. This is despite the non-linearity introduced by Burgers' equation. This has two reasons. Firstly, the Lagrangian approach separates Burgers' equation into two coupled linear equations. Secondly, the use of the analytical mapping actually prevents interaction between the two equations, effectively decoupling them. Because of this, the main difference between using this algorithm for Burgers' equation and for linear advection is the analytical expression used in the mapping. This means that the algorithm constitutes a topological advection scheme, albeit with an analytical mapping.

Instead of focusing on the convergence behaviour, the next step concerns the computation of the flow maps as opposed to using the analytical solution. This is a natural step since the analytical solution for the flow map is already somewhat inconvenient to use. The flow map of Equation (5.5) requires the velocity at the starting position. In the current implementation, this was saved from the first call to the flow map function. Another option would be to compute the starting position given a current x-location, but this requires the solution to a non-linear equation. There is a much more convenient way to compute the flow map, which is detailed in the next section.

5.2. Numerical Flow Maps

So far, whether to expand the advected quantity using Lagrange polynomials (0-forms) or edge polynomials (1-forms) was either determined by the divergence of the flow map or convenience if the flow map was divergence-free. However, as was already mentioned in the footnote on page 36, the expansion as 0- or 1-form (in one dimension) should be dictated by the physics of the problem under consideration [48]. Beyond the expansion as a form, every physical variable is associated to a geometrical object in space and time, as extensively discussed and shown by Tonti [48]. The following algorithms will strongly depend on this association. So before we can continue to the new algorithms, two questions must be answered first; what does Burgers' equation physically represent? And to which geometrical objects are the variables of Burgers' equation associated?

5.2.1. Interpretation and Association of Burgers' Equation

To find the physics behind Burgers' equation, it is convenient to rewrite the equation in a slightly different formulation as

$$\begin{cases} \dfrac{\partial \pi(t,x)}{\partial t} + u(t,x)\dfrac{\partial \pi(t,x)}{\partial x} = 0, & \text{(5.6a)} \\ \pi(t,x) = u(t,x), & \text{(5.6b)} \end{cases}$$

where $\pi(t,x)$ is the momentum. Equation (5.6) more clearly shows the distinction between the advection velocity and the advected quantity (the momentum). The momentum can also be used in the Lagrangian frame as

$$\begin{cases} \dfrac{\partial \pi}{\partial t} = 0, & \text{(5.7a)} \\ \dfrac{dX(t)}{dt} = u, & \text{(5.7b)} \\ \pi = u. & \text{(5.7c)} \end{cases}$$

This shows that Burgers' equation describes momentum transport, which is a well-known interpretation, see for example Cole [209]. In fact, Burgers' equation can be obtained from the momentum equation of the Navier-Stokes equations by assuming that there are no viscous, pressure, and body forces and by assuming that the density is a constant that is equal to 1. The last assumption effectively constrains the Burgers' equation to incompressible flow. This is consistent with the observation in the previous section, where the Jacobian of the flow map was found to be equal to 1.

With the momentum transport view of Burgers' equation, the association of the variables (momentum π, velocity u and position X) to an oriented geometric object can now be determined. To do this, it is important to note that the difference between the Eulerian (Equation (5.6)) and Lagrangian (Equation (5.7)) views of Burgers' equation represents more than a mathematical change[13]. This was explicitly indicated by Tonti [48] and is part of the unique role of velocity in fluid dynamics.

The Lagrangian view considers fluid dynamics as a mechanical theory [48]. In this view, a fluid particle of fixed mass is followed and its configuration (velocity, position, etc.) is changed by the application of forces (pressure, friction, etc.). In this view, all the forces result in a (possibly zero) change in momentum (Newton's second law). Through a momentum-velocity relation, this results in the change of velocity and thus changes in the configuration of a fluid. The momentum-velocity relation encapsulates the behaviour of a specific fluid and is a constitutive relation, see again [48].

The Eulerian view, on the other hand, considers fluid dynamics as a field-based theory [48]. In this view, the concept of momentum (and force) disappears and is replaced by mass, since the view is now

---

[13] Of course, under certain continuity conditions, the Eulerian and Lagrangian views can be shown to be equivalent by means of algebraic manipulations [213] (reference indicated by [124]). The equivalence in the weak sense (including shocks) is more complicated and the 1D case was treated by Wagner [124].

fixed on a spatial region. The flow of mass (caused by different fluxes and reactions) now drives the fluid flow. A relation similar to the velocity-momentum relation exists between mass and velocity, see [48]. Through this relation, the configuration of the fluid is again changed.

The difference between a mechanical theory and field theory has consequences for the orientation of the space and time objects, as detailed by Tonti [48]. For mechanical theories, as is fluid mechanics in a Lagrangian formulation, the space and time elements have either both an inner orientation or an outer orientation. For field theories, the orientation for the space elements is opposite that of the time elements. Since fluid dynamics can be formulated as a mechanical as well as a field theory, this difference is especially important for fluid dynamics. Specifically, the association of the velocity changes when switching between the two formulations. In the field formulation (Eulerian), the velocity is associated to outer-oriented time instants and inner-oriented lines (a circulation), while in the mechanical formulation (Lagrangian), the velocity is associated to inner-oriented time intervals and inner-oriented points. Note that the latter is exactly what one should expect, considering the notion of history in Lagrangian fluid dynamics (see for example [117]). The total consequences of this difference, and the differences for other variables, are not yet fully understood.

Since a Lagrangian method is designed here, only the Lagrangian view is needed. The association to the oriented time and space variables can thus be deduced based on the diagrams for the (momentum equation of the) Euler and Navier-Stokes equations made by Tonti [48]. The diagram for Burgers' equation is shown in Figure 5.2. The main difference between the diagrams of Tonti, next to the absence of the pressure and viscosity terms, is the addition of the Lagrangian position X, the deformation gradient F and the velocity gradient L. The latter variable is added for completeness. While the associations can be found in Tonti's book [48], the relevance and use of variables like the deformation gradient and velocity gradient tensor can for example be found in the books by Chadwick [214], Reddy [122], and Astarita and Marrucci [215].

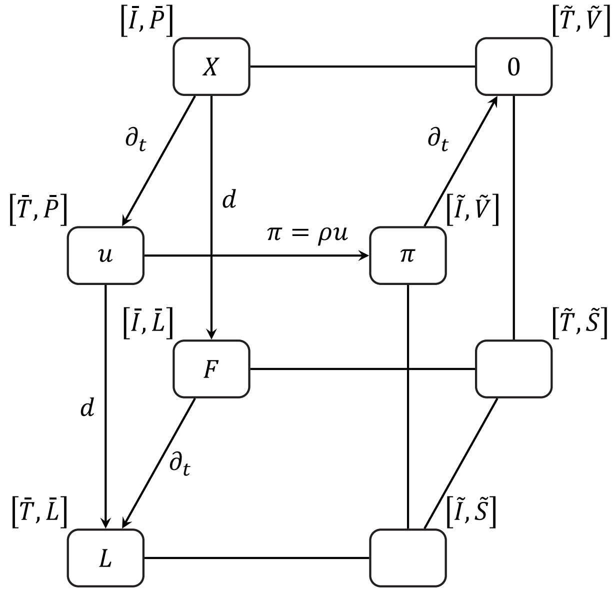


Figure 5.2: Association diagram for Burgers' equation. Comparable to diagrams FLU6 and FLU8, and table 12.4 of Tonti [48]. The left side of the diagram contains the inner-oriented elements, to which the configuration variables are associated (the kinematics). The right side of the diagram contains the outer-oriented elements, to which the source variables are associated (the dynamics).

Burgers' equation can also be written using differential forms. Instead of indicating the degree of the space-time form, a slightly different notation is used where the degrees in time and in space are indicated separately. This notation is also used in [110], although the degree in space is indicated first in [110], whereas the degree in time is indicated first here. This notation is also in line with Figure 5.2. For example, the velocity u is a 1-form in time and a 0-form in space so will be denoted $u^{(1,0)}$. To indicate its association to inner-oriented objects, a bar is added to create $\bar{u}^{(1,0)}$. Variables associated to the outer-oriented objects are indicated with a tilde. With this notation, Burgers' equation (Equation (5.7)) can be written as (in $\mathbb{R}^{3+1}$)

$$\begin{cases} \partial_t \tilde{\pi}^{(0,3)} = \tilde{0}^{(1,3)}, & (5.8a)\\ \partial_t \bar{X}^{(0,0)} = \bar{u}^{(1,0)}, & (5.8b)\\ \tilde{\pi}^{(0,3)} = \star \bar{u}^{(1,0)}. & (5.8c) \end{cases}$$

Using forms in time and space, and the association to inner- and outer-oriented geometric objects leads to the introduction of the Hodge-$\star$ operator in Equation (5.8c), which were discussed in Section 2.2.6. In this case, the Hodge operator represents the constitutive equation between velocity and momentum. It is important to note that it also encapsulates the change from inner to outer orientation. This simultaneous representation of multiple aspects by the Hodge operator was stressed by Mattiussi [45].

The relation between the momentum and the velocity is the well-known $\pi = \rho u$ with ρ the density. As indicated at the beginning of this section, the density is constant and equal to one. More importantly, the momentum-velocity relation is indicated as a constitutive equation, while some may consider the momentum to be defined by this relation (or by $\pi = mu$ with m the mass). This was clearly explained by Tonti [48, p.250], who also indicates other references where the momentum-velocity relation was indicated as a constitutive equation. One clear argument is that the relation between momentum and velocity cannot change when going to relativity if it defines momentum, yet it does. Note that the constitutive law is the only link between the two topological laws (Equations (5.8a) and (5.8b)), as is common in mimetic methods and this shows the importance of the constitutive equations [82].

Now that the interpretation and association of Burgers' equation are clear, they can be used to design a physically consistent numerical method. The numerical computation of the flow map is treated next.

5.2.2. Fully Numerical Advection

With the momentum interpretation and correct association of Burgers' equation, the flow map is easily found using a numerical method. From the diagram (Figure 5.2) it becomes clear that the flow map (now designated X for the Lagrangian position) is related to the velocity by means of a time derivative. More importantly, it shows that velocity is associated to points in space and edges in time. Therefore, expanding the flow map using nodal polynomials in time and space (a, in the notation of the previous section, (0,0)-form)

$$X_h(\tau,\xi) = \sum_{i=0}^{p}\sum_{j=0}^{p} X_{ij} h_i(\tau) h_j(\xi), \tag{5.9}$$

where $X_{ij} = X(\tau_i, \xi_j)$ with τ_i the i-th GLL-node in time, and ξ_j the j-th GLL-node in space. Expanding the velocity using nodal polynomials in space and edge polynomials in time (a (1,0)-form)

$$u_h(\tau,\xi) = \sum_{i=1}^{p}\sum_{j=0}^{p} u_{ij} e_i(\tau) h_j(\xi), \tag{5.10}$$

where $u_{ij} = \int_{\tau_{i-1}}^{\tau_i} u(\hat{\tau}, \xi_j)\, d\hat{\tau}$ with $\hat{\tau}$ a dummy integration variable. Computing the flow map from a given velocity comes down to solving[14]

$$\frac{dX}{dt} = u, \tag{5.11}$$

which, using the expansions of Equations (5.9) and (5.10), becomes

$$\mathbb{E}_t^{1,0} X_{ij} = u_{ij}, \tag{5.12}$$

with $\mathbb{E}_t^{1,0}$ the time incidence matrix. Separating the initial condition gives

$$\mathbb{E}_t^{1,0} X_{ij}^{t^{n+1}} = -\mathbb{E}_{t,ic}^{1,0} X_{ij}^{t^n} + u_{ij}. \tag{5.13}$$

For a specific point X_j this is equivalent to (thanks to the expansion of velocity on a temporal edge)

$$X_j^{t^{n+1}} = X_j^{t^n} + \int_{t^n}^{t^{n+1}} u(\hat{t}, \xi_j)\, d\hat{t}, \tag{5.14}$$

---

[14] Time t and reference time τ are still related through the linear relationship $t = \Delta t(\tau + 1)/2 + t^n$ as used throughout Chapter 4.

which is the exact solution to Equation (5.11). Equations (5.12) and (5.13) represent the numerical scheme used to compute the flow map in place of the analytical expressions used in Chapter 4.

One advantage of representing the position in the reference domain is the straightforward computation of the deformation gradient. In one spatial dimension, in particular, the deformation gradient represents the Jacobian of the mapping and various other important quantities. Because the deformation gradient computes the change in position with respect to the starting position, no product rule is needed to compute the Jacobian (as was done in Section 4.2). The representation of the position in the reference domain also makes it easier to compute the solution at different locations. Computing the solution for a given location is more involved, as it either requires interpolation or an inverse map. For the latter, the implicit function theorem (see for example [73]) may be helpful, as it guarantees the existence of such an inverse if the Jacobian is non-zero. Nevertheless, for the problems considered here, only the forward map is needed.

Another important advantage of representing all variables in the reference domain is that quadrature can be easily performed because problems associated with moving quadrature nodes are avoided. Such problems were for example observed by Heumann and Hiptmair [154] and led to the introduction of significant errors. Having the quadrature nodes as well as the solution values readily available in the reference domain, where the nodes are stationary, such problems do not occur.

The numerical method to compute the flow map, as just described, can be combined with the methods developed in Chapter 4 to create a fully numerical scheme for Burgers' equation (Equation (5.7)). To obtain a complete overview of the method, consider a space-time mesh (in the reference domain) that consists of rectangles[15], as shown in Figure 5.3. The associations of the position X, velocity u, and momentum π are shown on the right of Figure 5.3. The primal mesh with inner orientation is indicated in black and grey, while the dual mesh with outer orientation is shown in purple and light purple. While the dual mesh is explicitly shown at half-time instants in Figure 5.3, it need not be exactly in the middle, nor is it explicitly created. It does show a staggering in both space and time. While not commonly used in many mimetic methods, this treatment of time is extensively treated by Tonti [48, 49] and Mattiussi [45, 108], see also Section 2.5.

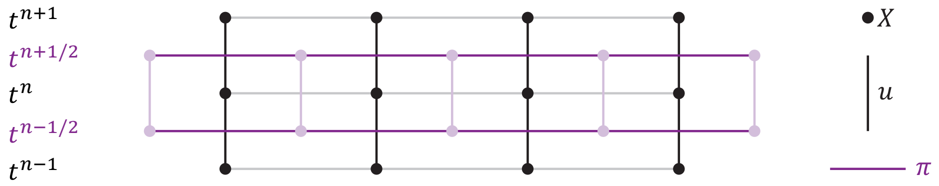


Figure 5.3: Example of a mesh (in the reference domain) for Burgers' equation and the association of the variables. The primal mesh with inner orientation is shown in black and grey, while the dual mesh with outer orientation is shown in purple and light purple. The lines and points in a light colour do not have a variable associated to them but are part of the mesh. The associations are shown on the right.

The association to the mesh elements follows from the diagram in Figure 5.2. The deformation gradient F and velocity gradient tensor L, which are also indicated in Figure 5.2, are not shown in the mesh. These can be easily found as $F = \mathbb{E}_x^{1,0} X$ and $L = \mathbb{E}_x^{1,0} u$ (or $L = \mathbb{E}_t^{1,0} F$) and are thus associated to the grey lines (for F) and the primal space-time surfaces (for L). They are not, however, required for the steps of the algorithm.

The complete algorithm is shown in Algorithm 5.1. For brevity, only the steps starting with the discretisation of the initial condition are shown. The creation of the incidence matrices and basis functions is no different from that in Chapter 4 and was also treated in Chapter 2. The reconstruction step is shown as part of the time-step loop, but if the solution is stored the reconstruction can be done after the time-step loop as well. The $2/\Delta t$-term in the reconstruction of the momentum is correct and a consequence of the transformation properties of the dual polynomials, see Section 2.3.3 or [86]. Note that both solving for and reconstructing the solution can be done element by element, reflecting the hyperbolic character of the equation. This may be used as an advantage for solving the equation in parallel, though the size of the current problem makes this unnecessary.

---

[15] Note that this is not constraining, since the reference mesh can be mapped to any starting mesh, and will be mapped to very complex meshes by the flow map.

Algorithm 5.1: Fully numerical algorithm for Burgers' equation without shocks.

```
/* Step 1: Discretise the initial conditions.                          */
$X_0(\xi) = \sum_j X_j h_j(\xi)$ with $X_j = X_{ic}(x)$
$u_0(\tau) = \sum_j u_j e_j(\tau)$ with $u_j = \int_{\tau_{j-1}}^{\tau_j} u_{ic}(\tau)\, d\tau$
$\pi_0(\xi) = \sum_j \pi_j e_j(\xi)$ with $\pi_j = \star^{-1} u_j$

for time-step in $K_t$ do
    /* Step 2: Solve Equation (5.8a).                                  */
    for $k$ in $K_x$ do
        $\tilde{\mathbb{E}}_t^{1,0} \pi_{ij} = \tilde{\mathbb{E}}_{t,ic}^{1,0} \pi_j \implies \pi_{ij}$
    end

    /* Step 3: Solve Equation (5.8c).                                  */
    for $k$ in $K_x$ do
        $u_{ij} = \star^{-1} \pi_{ij}$
    end

    /* Step 4: Solve Equation (5.8b).                                  */
    for $k$ in $K_x$ do
        $\bar{\mathbb{E}}_t^{1,0} X_{ij} = \bar{\mathbb{E}}_{t,ic}^{1,0} X_j + u_{ij} \implies X_{ij}$
    end

    /* Step 5: Reconstruct the variables in the physical domain.      */
    for $k$ in $K_x$ do
        $X(t) = \sum_{i,j} X_{ij} h_i(\tau) h_j(\xi)$
        $u(t,X) = \frac{2}{\Delta t} \sum_{i,j} u_{ij} e_i(\tau) h_j(\xi)\, d\tau$
        $\pi(t,X) = \frac{2}{\Delta t} \sum_{i,j} \pi_{ij} \tilde{h}_i(\tau) \tilde{e}_j(\xi)\, d\tilde{\xi}$
    end
end
```

5.2.3. Example Problem: Sinusoidal Initial Profile without Shocks

Now that the fully numerical method has been described, it can be tested on an example problem. The problem that will be used is the same as used in Section 5.1. The problem concerns the domain $x \in [0, 2\pi]$ with the initial condition $u_0(x) = \sin(x)$. The solutions at four different time instants ($t = 0$, $t = 0.5$, $t = 1.0$, and $t = 1.5$) are shown in Figure 5.4. Note that at $t = 1.0$ the profile becomes infinitely steep and this would normally be the time at which a shock forms if the shocked solution is of interest. In this section, only the smooth solution will be considered which was already presented in Equation (5.4). For the reconstruction of the numerically obtained solution, 75 equally spaced reconstruction points were used per element to which the GLL-nodes were added and duplicates were removed. In this case, nine equally sized time steps were used, though the number of time steps does not appreciably change the results, nor does changing the polynomial order in time. After all, from the point of view of a particle, nothing changes in time.

Figure 5.4 shows that the approximation is again close to the exact solution, even though the flow map is now part of the computation. The result remains equally good at and after $t = 1.0$, despite the triple-valuedness of the solution. This is one of the benefits of the reference domain in which the solution is computed. This is shown in Figures 5.5 and 5.6, where the same solution as presented in Figure 5.4 is now shown in the reference domain. The solution is not multi-valued in the reference domain and the position, which is the flow map, is part of the solution. Representing the solution in the physical domain, as done in Figure 5.4, is thus straightforward as the position and velocity of a particle are tied to its Lagrangian label (element number and ξ-coordinate). The fact that the velocity of a given particle stays constant is particularly obvious from Figure 5.5. Since the velocity of a particle is linked to its Lagrangian label, the velocity can also easily be used in the computations and saving

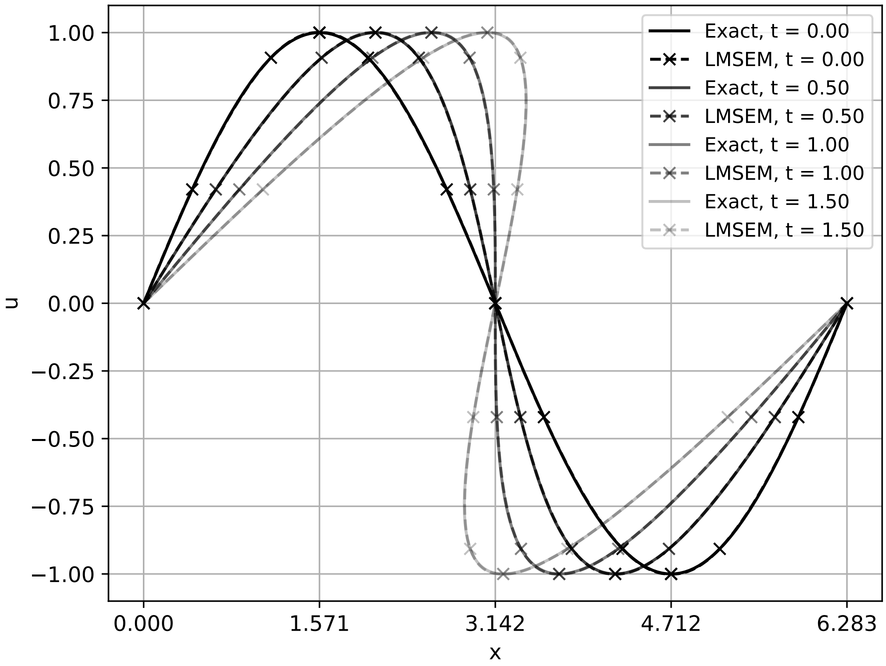
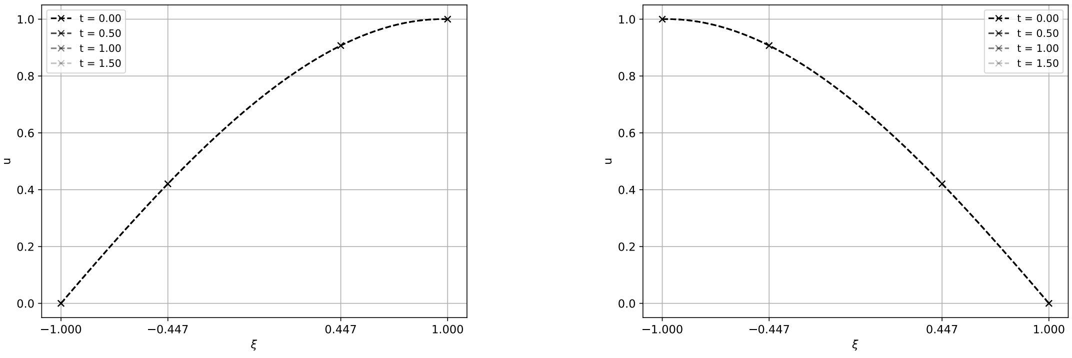


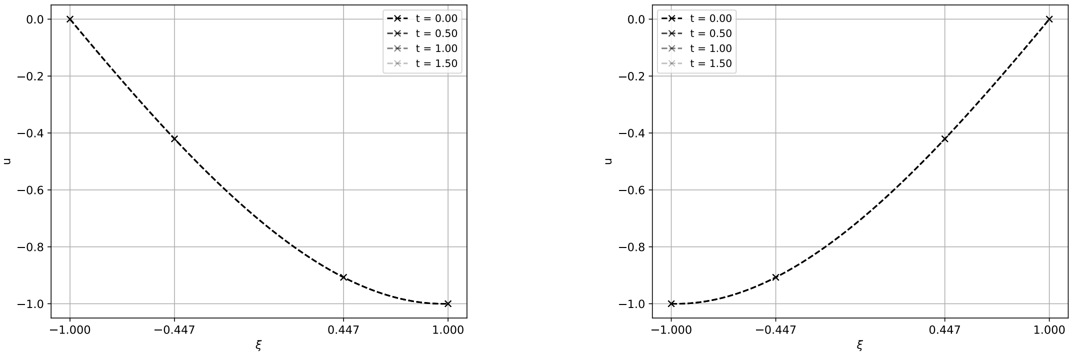
Figure 5.4: Solutions at $t = 0$, $t = 0.5$, $t = 1.0$, and $t = 1.5$ for Burgers' equation with a sinusoidal initial profile using $p_t = 2$ and $p_x = 3$ on four elements. The grid lines indicate the edge boundaries at $t = 0$ and the solution values at the GLL-nodes are indicated with markers.

the velocity from the starting time, as done in Section 5.1, is not needed. The overlapping regions are automatically handled by the mapping since no interaction between the particles is present nor is any continuity enforced.



(a) First element.

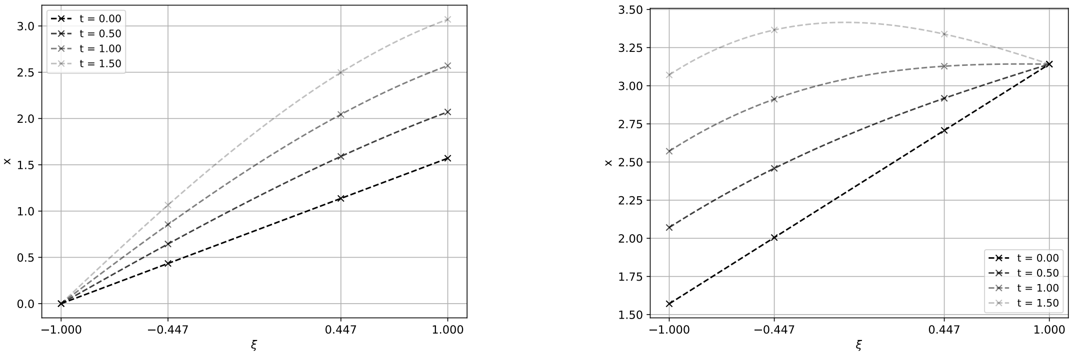
(b) Second element.



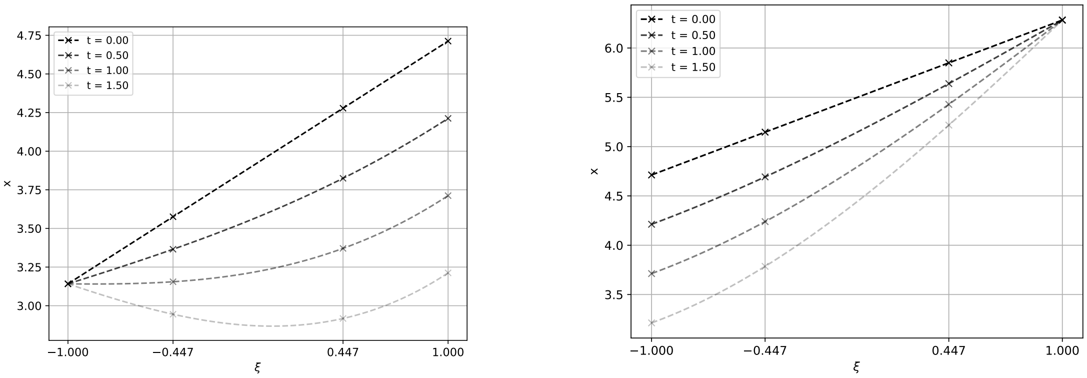
(c) Third element.

(d) Fourth element.

Figure 5.5: Solution in the reference domain for the setup of Figure 5.4.



(a) First element. (b) Second element.



(c) Third element. (d) Fourth element.

Figure 5.6: Flow map solution in the reference domain for the setup of Figure 5.4.

To quantify how close the solution is, the L_2-error norm is shown in Figure 5.7. The error norms for the errors in the position x and velocity u are shown separately. Their sum is also shown.

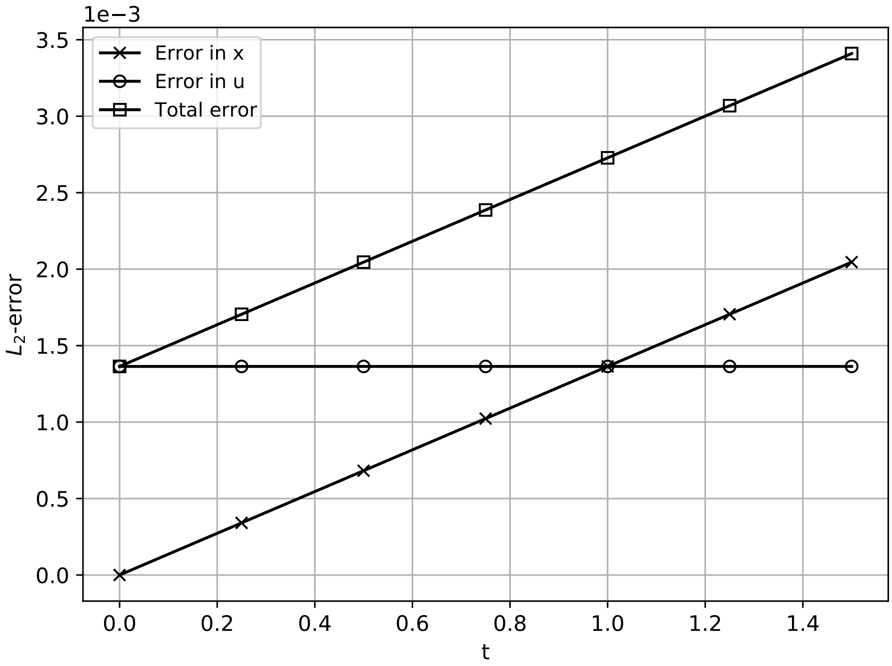


Figure 5.7: L_2-error over time for Burgers' equation with a sinusoidal initial profile using $p_t = 2$ and $p_x = 3$ on four elements. Error contributions for the velocity and position are shown separately.

Figure 5.7 clearly shows that the error of the velocity remains unchanged in time. This means that the error for the velocity is introduced when discretising the initial condition and is not caused by the algorithm. The error in the position is a simple consequence of the error in the velocity. Since the velocity is not exact, the reconstructed position will be slightly off as well. This happens at every time step, so the error will increase over time. However, no additional error is introduced in the calculation of the position, which is why the error in the position linearly increases in Figure 5.7 and crosses the error in the velocity at $t = 1.0$.

That the error of the velocity remains constant and that no additional errors are introduced when computing the position has a more fundamental origin. In Figure 5.8 the L_∞-error norm of the conservation laws is shown. These are conservation of momentum, Equation (5.8a), satisfied on the dual grid, and the kinematics, Equation (5.8b), satisfied on the primal grid. While the latter is not a true conservation law, the kinematic constraint is what is normally referred to as advection. Adhering to it is thus of fundamental importance for an advection scheme. Figure 5.8 shows that both balance laws are adhered to exactly (that is, to machine precision) during the simulation. This is possible thanks to the incidence matrices. It is also the cause of the non-aggravating errors in Figure 5.7.

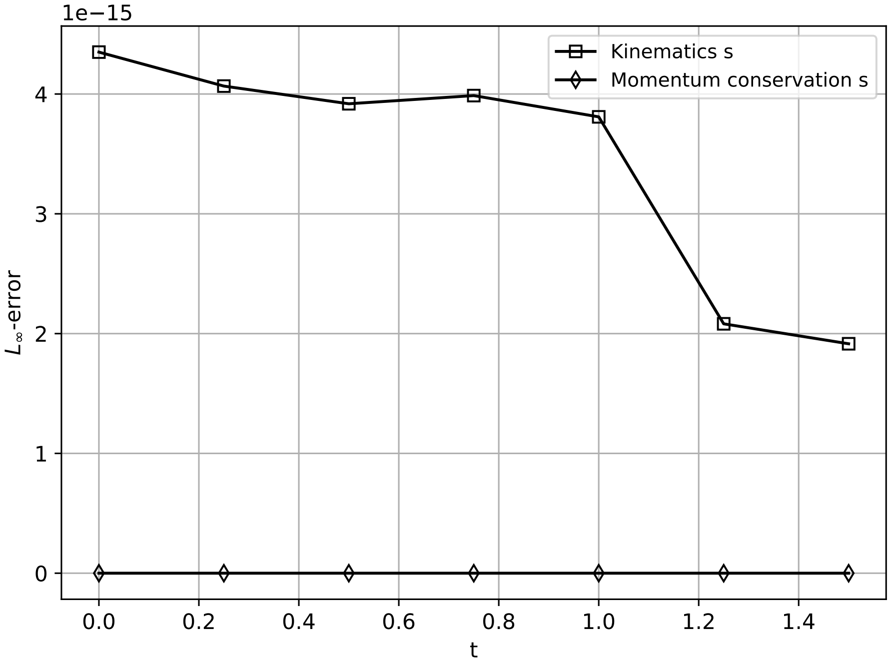


Figure 5.8: L_∞-error over time for Burgers' equation with a sinusoidal initial profile using $p_t = 2$ and $p_x = 3$ on four elements. The errors for conservation of momentum are computed on the dual grid, while the errors in the kinematic constraint are computed on the primal grid.

As shown in Section 4.1, the discretisation error of the initial condition can be made arbitrarily small through hp-refinement. The same holds true here. As a result, the algorithm developed here can again be considered a topological advection scheme. In particular, adhering to the kinematic constraints, which include advection, shows that advection can be handled without approximation. The Lagrangian formulation and the association of the velocity to a temporal edge are vital to this result. The results of this section have shown that the non-linear aspect, at least for Burgers' equation, can be handled without problems. The Lagrangian formulation is again fundamental for this result, as it turns the non-linear equation into a set of linear equations.

Remark. The results presented here were obtained using time steps of equal size. While it is possible to use time steps of varying size, this requires a scaling of the coefficients of the velocity (and thus momentum). The reason for this is straightforward. Every particle is given an initial velocity, but the coefficients used here are integrals in time. As a result, the integrated values represent a total velocity/momentum for the duration of the integration interval. If the size of a time step changes, the integration interval should also change. This can be easily done by scaling the values of the coefficients since the velocity remains constant. For more complex cases, the integral may need to be recomputed. The start-up procedure will be reevaluated in Chapter 6.

5.3. Shocks and Burgers' Equation

So far, the numerical solution has been shown to agree (arbitrarily) closely with the exact mathematical solution. However, when the inviscid form of Burgers' equation is used, as done here, the overlapping exact mathematical solution is not physical and thus of little practical interest. This problem can be solved by mathematical means via the so-called weak solutions. The idea of the weak solutions is entirely mathematical and requires multiplying the equation with a smooth test function, and the subsequent integration (by parts) over the domain [125]. This technique moves the derivatives to the test function, which is smooth by choice, allowing less smooth solutions to also satisfy the original equation. This allows discontinuous solutions (solutions with shocks) to be found. Unfortunately, weak solutions are often not unique, so an additional constraint is required; the entropy condition [125]. Next to this, using weak solutions also obscures the physical interpretation of the equations, which is particularly important for the methods in this work. Note that the weak solutions only have to allow the shock to form, transporting a discontinuity can still be done by the PDE.

As said before, the overlapping solutions of Burgers' equation are non-physical. Simply put, something is missing in the model that is used to describe the flow of fluids. Remembering the momentum transport interpretation of Burgers' equation of Section 5.2.1, and the realisation of Section 4.1 that the simple advection equations (including Burgers' equation) share more resemblance with a rigid body in space, the missing aspect of the model is easily found; interaction. In order to describe flows with shocks, the particles must be able to interact and exchange their momentum. For Burgers' equation, this interaction is instantaneous and can be visualised as the collision between two particles. From this point of view, the formation of a shock is very similar to (fully plastic) rigid body collisions [216][16].

Mathematically the weak solutions allow Burgers' equation to be solved. Alternatively, jump relations (or Rankine-Hugoniot relations as they are known in fluid dynamics) can be derived which can be used across the shock [113], while the PDE still holds away from the shock. Unfortunately, neither option can be conveniently utilised in a numerical method[17]. To remedy this, Von Neumann and Richtmyer [176] (see also [178] for other contributors and an alternative interpretation) introduced the concept of artificial viscosity; adding a small viscosity term to the governing equations aimed at smoothing out the discontinuity so that no special treatment for the shocks is needed. This fixes the missing term by introducing a term that is similar to the actual viscosity mechanism that would smooth a shock in reality. Since its introduction, artificial viscosity has been used in many different simulations and countless implementations exist. Nevertheless, introducing artificial viscosity to a hyperbolic advection equation effectively turns it into a parabolic advection-diffusion equation. This results in different behaviour of the solution, with particularly detrimental effects on simulations over longer periods of time, since the viscosity will (eventually) affect the solution in the entire domain [96, 172, 180]. Additionally, a diffusion term causes energy to be dissipated, and thus affects the conservation laws as well.

The method developed in this work is aimed at strictly adhering to the balance laws and no artificial viscosity or limiters were introduced for stability purposes, nor were they needed. To stay in line with this objective, an alternative method to compute shocks was found. This method is described in the next section.

5.3.1. Computing Shocks

A very interesting method to compute shocks in Burgers' equation was found in a paper by Marchiano, Coulouvrat, and Grenon [219], who added this technique to a previously developed method by Auger and Coulouvrat [220]. These papers focus on sonic boom modelling, see for example the short review by Plotkin [221] for some basic terminology and ideas of this research area. The method to compute shocks, however, was originally developed by Hayes, Haefeli, and Kulsrud [222] (see also Hayes and Runyan [223]), again for sonic boom modelling. When reading the work in which the shock computation method was developed [222], it should be realised that this was done in the sixties, with punch-card computers in mind. It was therefore originally intended as a graphical method. The more recent work of

---

[16] Note that even in the case of shocks, Burgers' equation still represents incompressible flow (save for the shock, where variables like density are not well defined). Normally, when shocks are discussed, the fluid particles are said to be pushed into each other with their volumes becoming smaller and smaller. Without viscosity to smooth things out, the particles will be pushed into a shock of infinitesimal thickness. In the case of Burgers' equation, the fluid particles do not push each other. There is no compression, and the fluid particles will only 'feel' the shock once they collide with it.

[17] It is possible to use the jump relations using a shock fitting method [217, 218]. However, this is limited to simple problems with weak shocks as these methods require an initial solution without shock [172].

Marchiano, Coulouvrat, and Grenon [219] adapts the method to the use of modern computers. Before directly applying the method, it should be noted that the Burgers' equations found in [219–223] have no physical meaning [219, 222]. Therefore, in what follows, the method will be adapted and interpreted in a way that is consistent with the ideas laid out in this work.

The method to compute shocks is an effective way of using the equal-area rule [222] (see also [125] for the equal-area concept). It seems that, when [222] was written, there was still some ambiguity regarding the underlying conservation principle of the equal-area rule. It was noted that conservation of mass was involved [222], although Hopf [130] had already proven the rule using a limiting process. Later publications, such as that by Aifantis and Serrin [224], indicate that both conservation of momentum as well as conservation of mass are involved in the generic case of multiple fluids. In the case of a single fluid, only conservation of momentum is sufficient [224]. This is consistent with the above interpretation of Burgers' equation as a momentum equation. On top of that, Marchiano, Coulouvrat, and Grenon [219] note that the shock method is also related to the second law of thermodynamics, so that it can be seen as incorporating an entropy condition as well.

So far, and this will be continued for the rest of this section, this method has not been called shock-capturing or shock-fitting. While it seems to more closely resemble a shock-fitting approach, especially in that it locates and places the shock after first computing a solution, the term shock-fitting often implies the use of jump relations. Such relations, however, are not used, as will be shown next.

The theoretical foundations to compute the shock is now briefly discussed. The method is based around a velocity (or really a momentum) potential, denoted here as S[18]. The potential S is defined as the integration of the velocity (momentum) over the current domain, or as the (Stieltjes[19]) integral of the velocity over the initial domain by a transformation [222], as

$$S(t, X(t)) = \int_{\infty} u(t, X(t))\, dX(t) = \int_{\infty} u(t, X(t)) \frac{dX(t)}{dx_0}\, dx_0. \tag{5.15}$$

In principle, this integral definition is enough for the method to work. (Note also that the right-most part of Equation (5.15) is essentially the transformation of a spatial 1-form, which is another argument for the momentum interpretation because momentum is a (dual) spatial 1-form.)

However, as noted and used in [219, 222], the integral need not be explicitly computed as it can be rewritten in terms of the velocity. This requires the trajectory equation, Equation (5.7b), which is repeated here for convenience

$$\frac{dX(t)}{dt} = u(t, X(t)) = u(t, x_0), \tag{5.16}$$

with x_0 the starting position of a fluid particle (remember that this is directly related to the fluid label ξ). From this equation, the position $X(t)$ can be written as (like the method of characteristics)

$$X(t) = x_0 + t u(t, X(t)). \tag{5.17}$$

Then, to rewrite Equation (5.15) the Jacobian term can be found, from Equation (5.17), as

$$\frac{dX(t)}{dx_0} = 1 + t \frac{du(t, X(t))}{dx_0}. \tag{5.18}$$

Plugging this into the last part of Equation (5.15) gives

$$\int_{\infty} u(t, X(t)) \left(1 + t \frac{du(t, X(t))}{dx_0}\right) dx_0 = \int_{\infty} u(t, X(t))\, dx_0 + \int_{\infty} u(t, X(t))\, du(t, X(t)). \tag{5.19}$$

Denoting $\int_{\infty} u(t, X(t))\, dx_0$ as $S_0(x_0)$, which can be computed once at the start, then gives a simple equation for the potential

$$S(t, X(t)) = S_0(x_0) + \frac{1}{2} t u(t, x_0)^2. \tag{5.20}$$

---

[18] The interpretation and notation used here are not related to those of [219, 222, 223], but are instead made to be consistent with the notation used throughout this work.

[19] While Stieltjes integrals, and their differences with Riemann integrals, are not needed for the discussion here, it is interesting to note that they also briefly appeared in a work by Mattiussi [80]. In fact, Mattiussi [80] (while partially referring to Lebesgue) noted that such integrals have a more profound geometric meaning and that they are thus more appropriate when considering variables associated to geometric objects. This thus seems like an interesting detail to investigate in future work, so that shocks may more easily be dealt with in mimetic methods.

This formulation is used in both [219, 222].

The potential S can be used as a type of indicator function that locates the shock. This was used very explicitly by [222] by plotting the potential function below the multi-valued solution of the velocity. An example of this is shown in Figure 5.9. The crossing points of the potential function correspond to the locations of a shock, and could thus be easily drawn. In the more recent application by [219], this was used numerically by finding the maximum potential for each spatial location and taking a finite difference derivative of the potential to get the new velocity. Nevertheless, the underlying idea is the same. It is worth noting that interacting shocks are automatically dealt with by this method [222].

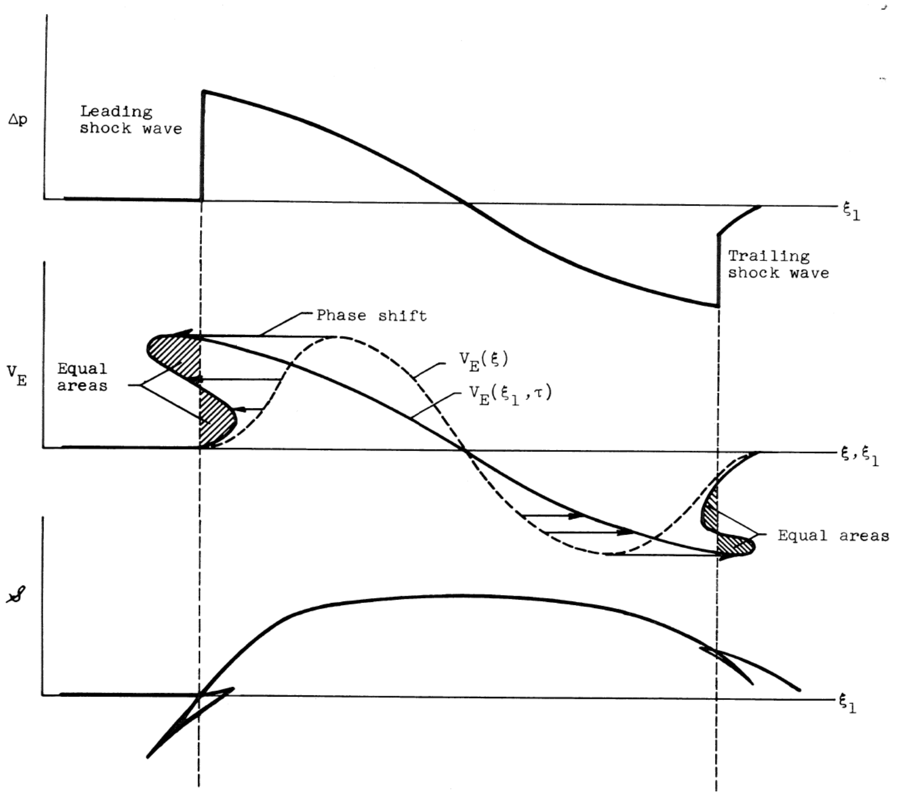


Figure 5.9: A conceptual example of the shock computation method. Picture from Hayes, Haefeli, and Kulsrud [222, p.62]. The symbols in the figure are from [222] and do not correspond to those used in the text. In the figure, Δp corresponds to the pressure changes, V_E is what is considered a velocity here (with opposite sign), ξ and ξ_1 correspond to x_0 and $X(t)$, respectively, and the $\mathcal{S}$-like symbol corresponds to the potential S.

The reason that the minimum (in [219, 222] the velocity is opposite of that defined here so that it will be a minimum now) of the potential could be used is as follows. Burgers' equation, written as

$$\frac{\partial u(t,x)}{\partial t} + u(t,x)\frac{\partial u(t,x)}{\partial x} = 0, \tag{5.21}$$

is well known to have the following first integral

$$\int_{-\infty}^{\infty} u\,dx. \tag{5.22}$$

This integral is always conserved, see for example Hopf [130]. In the interpretation used here, this should be considered conservation of momentum. Hopf [130] noted that, in the inviscid case, a second first integral appears. This integral can, with some abuse of notation, be written as

$$\min_{x} \int_{-\infty}^{x} u\,dx. \tag{5.23}$$

Note that Hopf used the same convention of positive velocity as used here. This second integral corresponds to the potential shown in the lower part of Figure 5.9 with the overlapping parts cut off (as the sign of the velocity is flipped, so is the potential). This is the mathematical reason behind the use of the potential. This was also noted in [222]. In [219] the second law of thermodynamics[20] and the fact that only compression shocks are possible for Burgers' equation were thought to be the underlying causes.

With the momentum interpretation used here, more straightforward reasoning can be used. The entire method can be seen as deriving the conservation of momentum. The parts of the potential that are cut off are simply the fluid particles that interact and exchange their momentum. The resulting potential is that of the solution with a shock. This view can be used to compute shocks with the mimetic method developed here. Next to that, it is also consistent with the previous interpretation that shocks (in Burgers' equation) are essentially two particles colliding with each other. From this point of view, the shock method adds the missing term previously mentioned by allowing particles to interact and exchange their momentum. This is very different from artificial viscosity methods that explicitly include a viscous term that effectively prevents the particles from colliding.

Due to constraints in time and scope, the above-presented interpretation is not fully used in the numerical method. Instead, the interpolation method of [219] is used in the next section to create an example solution with a shock. The steps are shown in Algorithm 5.2. It is assumed that the variables in steps two and four are array-like variables so that the formulas are valid for the complete reconstructed profile (in space).

Algorithm 5.2: Shock interpolation procedure. Can be used after computing the solution using Algorithm 5.1.

```
/* Step 1: Compute the starting potential.                                  */
$S(t) = \int_\infty u(t, X(t))\, dx_0$ at $t = 0$

for $t$ in reconstructed time steps do
    /* Step 2: Update the total potential.                                  */
    $S(t) \mathrel{+}= \frac{1}{2} t u^2$

    /* Step 3: Find the lowest value of the potential at a given
       (fixed) point $x_f$. This requires interpolation.                    */
    for $i$ in $N_{x_f}$ do
        for $x_f$ in $x_{all}$ do
            if $x_{mv}[i] <= x_f <= x_{mv}[i+1]$ then
                $S^{\text{interp}} = \frac{(x_{mv}[i+1] - x_f) \cdot S[i] + (x_f - x_{mv}[i]) \cdot S[i+1]}{x_{mv}[i+1] - x_{mv}[i]}$
        end
    end

    /* Step 4: Compute the new velocity value based on the found
       potential.                                                           */
    $u_{\text{shock}} = (S^{\text{interp}}_{i+1} - S^{\text{interp}}_{i}) / (x_{i+1} - x_i)$
end
```

5.3.2. Example Problem: Sinusoidal Initial Profile with Shocks

The same problem as used in Section 5.2.3 can be used again, but using the solution with a shock. Due to the symmetry of the problem the shock arises in the middle of the domain. A simple analytical method to find the exact solution with the shock is for example described by Whitham [211].

The result of using Algorithm 5.2 is shown in Figure 5.10. The setup is the same as before, though the GLL-nodes are not indicated with a marker. This is because 75 equally spaced points (in ξ) are used

---

[20] Not that entropy played a role in any of the calculations. If one prefers an energy-like interpretation, the closest option is an exchange of kinetic energy, given Equation (5.20) has a kinetic energy-like term. Though 'exchange' is perhaps not the right term. Note also that energy has not played any role in the preceding interpretation. The momentum exchange interpretation thus seems the most logical and consistent option.

for the reconstruction, without the addition of the GLL-nodes as reconstruction points. This was done because Algorithm 5.2 was found to experience some difficulties with unequally spaced points. Nevertheless, and despite the finite difference procedure used to compute the velocity from the potential, the result looks good.

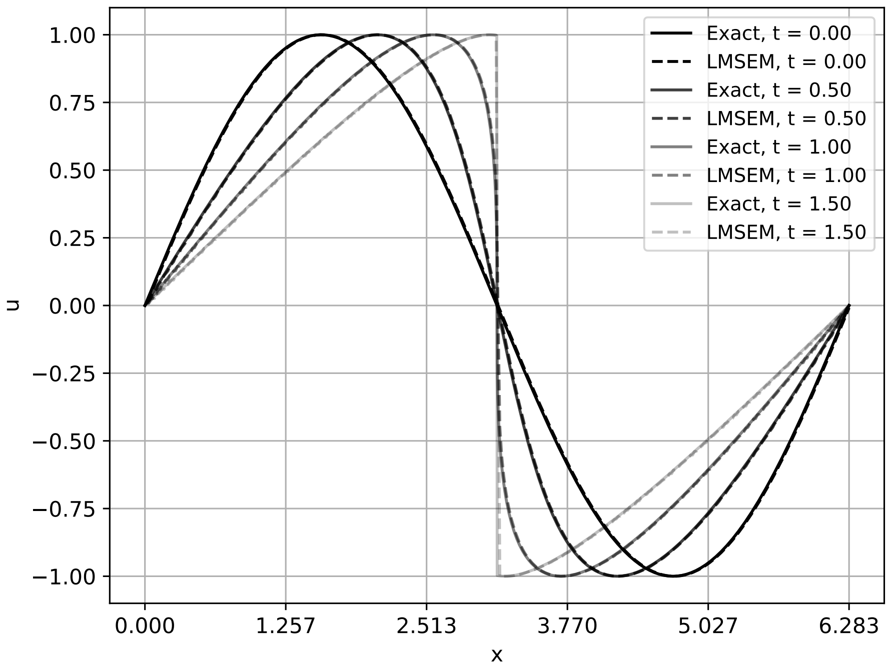


Figure 5.10: Solutions at $t = 0$, $t = 0.5$, $t = 1.0$, and $t = 1.5$ for Burgers' equation with a sinusoidal initial profile using $p_t = 2$ and $p_x = 3$ on four elements. The grid lines indicate the edge boundaries at $t = 0$. The solution with shock is computed using Algorithm 5.2.

An important aspect of the effectiveness of this simple algorithm is the fact that it only works in the post-processing step. For a spectral method, in particular, the reconstructed profile can be made with as many points as desired, creating a very high resolution around the shock. This means that, despite using only four elements and four nodes per element in the solution, as done above, the shock is still very sharp.

As a result of the used procedure, the error analysis can be relatively short. After all, the method still computes the exact same solution as computed in Section 5.2.3. The only difference is in the reconstruction. In particular, it was already mentioned by Marchiano, Coulouvrat and Grenon [219] that the procedure to compute shocks only adds interpolation error. Therefore, while the interpolation error increases, the other aspects of the error analysis of Section 5.2.3 remain valid.

Despite the good-looking result shown in Figure 5.10, the algorithm still needs to be extended. Additionally, a closer link between the algorithm and the new interpretation may be desired, but such aspects are left for future work.

Overall, the results in this chapter have again shown the remarkable accuracy and conservation properties of the method. In particular, the numerical method can indeed be seen as a topological advection scheme. Even in the case of a simple shock, the results are already good while using a very simple algorithm. Therefore, the method can now be extended to the Euler equations (without shocks), which will be treated in the next chapter.

6

Isentropic Euler Equations

In the previous chapter, a Lagrangian mimetic method was developed for Burgers' equation. This method is a fully numerical and topological advection scheme that is consistent with the physical structure as treated by Tonti [48]. The non-linearity of Burgers' equation presented little difficulty due to the natural representation of advection in a Lagrangian formulation. It was also stated in the previous chapter that Burgers' equation represents constant density flow. In this chapter, a proof of concept method will be developed that will be able to numerically solve the compressible Euler equations for an isentropic flow. This will show that the method developed here can also handle compressible flows, while the isentropic aspect prevents the need for a deep dive into thermodynamics.

This chapter follows a similar structure to that used in Section 5.2. The interpretation and association of the isentropic Euler equations are treated in Section 6.1. While the interpretation of the Euler equations is inherently physical as opposed to that of Burgers' equation, using a Lagrangian frame and adding geometry may lead to unexpected associations and interpretations. When the interpretations and associations are clarified, the numerical method can be explained. This is done in Section 6.2. This method will use the algorithm of Section 5.2 as a basis. To end the chapter, an example problem is solved in Section 6.3 to showcase some properties of the method.

6.1. Interpretation and Association of the Isentropic Euler Equations

In the previous chapters, the equations under consideration were given and a physical interpretation was found later. In this chapter, the focus lies on the isentropic Euler equations for which the interpretation is well-known; they describe the motion of a compressible and inviscid fluid. The Euler equations are often used to study fluids so that their formulation can be found throughout the literature [111, 112, 117]. In this work, the following expression is used (in the Lagrangian formulation, see [117])

$$\begin{cases} \dfrac{\partial \pi}{\partial t} = -\dfrac{\partial p}{\partial x}, & \text{(6.1a)} \\ \dfrac{dX(t)}{dt} = u, & \text{(6.1b)} \\ \pi = \rho_0 u, & \text{(6.1c)} \\ \gamma p \Delta V + V \Delta p = 0. & \text{(6.1d)} \end{cases}$$

In these equations, π is the momentum, p the pressure, X the position, u the velocity, γ the ratio of specific heats, and V the volume. Note that Equation (6.1c) is the momentum-velocity constitutive equation (see Section 5.2.1) which only requires the *initial* density [117, 118]. This is a consequence of conservation of mass (see below) and the fact that the Lagrangian formulation follows a fluid particle [117, 118]. Also, Equation (6.1d) is an expression of the conservation of energy which can be derived from the first law of thermodynamics using the assumptions of a calorically perfect gas and an isentropic flow, see Appendix A.2.

The equation for conservation of mass is not explicitly shown but is trivial (see for example [117]), and can be written as

$$\frac{\partial \rho J}{\partial t} = 0, \tag{6.2}$$

where ρ is the mass density and $J = \det(F)$ with F the deformation gradient. This gives the following result for the mass density (note that $J = 1$ at $t = 0$)

$$\rho J|_{t>0} = \rho J|_{t=0} = \rho|_{t=0} \rightarrow \rho|_{t>0} = \rho|_{t=0} J^{-1}|_{t>0}. \tag{6.3}$$

Since the new mass density follows from the starting density and the deformation gradient depends only on the position, this equation does not need to be explicitly solved by a numerical method.

The energy equation, Equation (6.1d), expresses the well-known relation $PV^{\gamma} = const.$ This relates the pressure to the volume. It should be noted that the changes in volume and pressure, ΔV and Δp, respectively, are with respect to a reference or initial value. This equation can be used to link the pressure and volume. Since the volume is closely related to the deformation gradient and position, a closed system can thus be obtained. Other variants of this equation also exist and can, in principle, be used. One example is $p = K_s\Theta$, where K_s is the isentropic bulk modulus[21] (see for example [225]) and Θ the dilatation ($\Theta = F - 1$). This expression would lead to a system similar to that used by Ramsey and Baty [226] who also use differential forms. Another well-known expression is $p = \rho^{\gamma}$, which is used in [227] from which the test case of Section 6.3 is taken. Theoretically, there is no difference at this point which would favour one expression over the other. In the subsequent sections, their differences will become clearer and a choice can be made based on which constitutive equation is easier to implement.

The association of the variables of Euler's equations in the Lagrangian frame were already discovered by Tonti [48]. With only some minor modifications already made in Section 5.2.1, the associations can be put together in a diagram as done in Figure 6.1. Tonti's Euler diagram (diagram FLU8 in [48]) is largely similar on the right side of the diagram, save for the different notation for momentum and the zero body forces assumed here. The left side of the diagram has more changes, but these were already discussed in Section 5.2.1. Some comments are, however, in order for the mass density ρ, the deformation gradient F, the dilatation Θ, and the role of the energy equation.

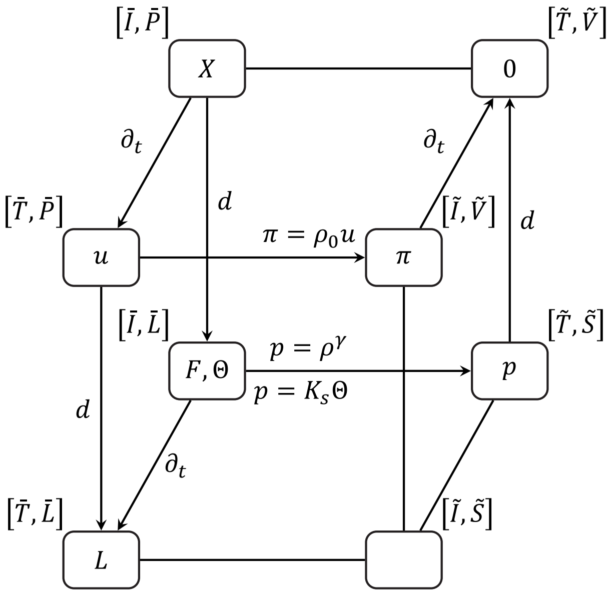


Figure 6.1: Association diagram for the Euler equations. Comparable to diagrams FLU6, FLU8 and SOL12, and table 12.4 of [48]. The left side of the diagram contains the inner-oriented elements, to which the kinematics are associated. The right side of the diagram contains the outer oriented elements, to which the dynamics are associated.

---

[21] The bulk modulus is not commonly used in fluid dynamics and the value for a gas depends on the thermodynamic process (hence isentropic bulk modulus). In fluid dynamics, the compressibility is more commonly used and is the reciprocal of the bulk modulus [112].

The deformation gradient F and the dilatation Θ are both spatial one-forms, as indicated in Figure 6.1. While this is straightforward in one dimension, it also holds true in three dimensions (or any dimension for that matter). This was also pointed out by Tonti [48, p.334] regarding the dilatation. The reason for their association as spatial one-forms is clearest when looking at the deformation gradient. The deformation gradient measures the change in length in one direction, so will have a component in each principal direction. This component will only measure the change in that specific direction. Adding possible cross terms turns the quantity into a vector-valued one-form. The same comments can be made about the dilatation.

The mass density plays a somewhat different role. In one dimension, it can be considered a 1-form and will behave similarly to the deformation gradient and dilatation. However, when moving to multiple dimensions, the mass density should not be considered a 1-form anymore. This is clearest when looking at Equation (6.3). The mass density can be found by multiplying its starting value with the determinant of the (inverse) deformation gradient. The determinant is of no consequence in one dimension but is important in multiple dimensions. In that case, the density should probably be considered as an n-form. A detailed investigation of such matters is left for future work. Note also that the density only plays a role as a material parameter and that its role in the Lagrangian formulation is much less important than in an Eulerian formulation.

With the well-known interpretation and correct association of the Euler equations, the governing equations can also be written using differential forms (using the same notation as used in Section 5.2.1). This will also aid in describing the role of conservation of energy. Euler's equations in a Lagrangian formulation then become

$$\begin{cases} \partial_t \tilde{\pi}^{(0,3)} = -\, d\tilde{p}^{(1,2)}, & \text{(6.4a)} \\ \partial_t \bar{X}^{(0,0)} = \bar{u}^{(1,0)}, & \text{(6.4b)} \\ \tilde{\pi}^{(0,3)} = \star_{\rho_0} \bar{u}^{(1,0)}, & \text{(6.4c)} \\ \tilde{p}^{(1,2)} = \star_{\rho^\gamma} \bar{1}^{(0,1)}. & \text{(6.4d)} \end{cases}$$

As already mentioned, the material parameter for the momentum-velocity relation is the initial mass density ρ_0. For the pressure-density relation, used here as an example, ρ^γ is the material parameter. When using $\tilde{p}^{(1,2)} = \star_{K_s} \bar{\Theta}^{(0,1)}$ instead, the isentropic bulk modulus K_s is the material parameter.

The particular choice of the constitutive equation for the pressure is not relevant at this moment, but it shows how the equation for conservation of energy has been reduced to a constitutive equation. This is a considerably different role compared to conservation of momentum (dynamics) and kinematics. Tonti [48, p.148] already indicated that energy can be seen as a potential of the constitutive equations. Indeed, looking at Figure 6.1, kinetic energy can serve as a potential for the momentum-velocity relation, while work can serve as a potential for the pressure-dilatation relation. This observation may be extended. Conservation of energy can be seen as connecting the equations that use the inner orientation to those that use the outer orientation. This puts certain constraints in place that must be adhered to for such a connection to be consistent with conservation of energy. Additionally, it appears that satisfying conservation of energy is a matter of selecting the correct and compatible constitutive relations while solving the energy equation as another balance law may be unnecessary. The exact consequences and formulation of this idea are left for future work. However, it seems that recent papers on thermodynamic compatibility describe something similar [228–230], albeit not for mimetic methods and with additional considerations regarding entropy. See Section 7.3 for some additional comments.

The Lagrangian formulation thus allows for a clear separation between the mechanical momentum and pressure relations, the kinematic position/advection and mass equations, and the thermodynamic energy equation. This split is particularly advantageous when using a mimetic method since balance laws can be adhered to exactly. In the next section, the LMSEM is applied to the Euler equations.

6.2. Numerical Method

The numerical method for the isentropic Euler equations builds on the algorithm developed in Section 5.2. With the associations as shown in Figure 6.1, the variables are expanded as follows (tildes indicate dual variables). The position X is expanded as

$$X_h(\tau, \xi) = \sum_{i=0}^{p_t} \sum_{j=0}^{p_x} X_{ij} h_i(\tau) h_j(\xi), \tag{6.5}$$

where $X_{ij} = X(\tau_i, \xi_j)$ with τ_i the i-th GLL-node in time, and ξ_j the j-th GLL-node in space. The velocity u is expanded as

$$u_h(\tau, \xi) = \sum_{i=1}^{p_t} \sum_{j=0}^{p_x} u_{ij} e_i(\tau) h_j(\xi), \tag{6.6}$$

where $u_{ij} = \int_{\tau_{i-1}}^{\tau_i} u(\hat{\tau}, \xi_j)\, d\hat{\tau}$ with $\hat{\tau}$ a dummy integration variable. The deformation gradient F and dilatation Θ have the same type of expansion. As an example, the deformation gradient is expanded as

$$F_h(\tau, \xi) = \sum_{i=0}^{p_t} \sum_{j=1}^{p_x} F_{ij} h_i(\tau) e_j(\xi), \tag{6.7}$$

where $F_{ij} = \int_{\xi_{j-1}}^{\xi_j} F(\tau_i, \hat{\xi})\, d\hat{\xi}$ with $\hat{\xi}$ a dummy integration variable. The momentum π is expanded as

$$\tilde{\pi}_h(\tau, \xi) = \sum_{i=0}^{p_t} \sum_{j=1}^{p_x} \tilde{\pi}_{ij} \tilde{h}_i(\tilde{\tau}) \tilde{e}_j(\tilde{\xi}), \tag{6.8}$$

where $\tilde{\pi}_{ij} = \mathbb{H}_{\rho_0} u_{ij}$. The pressure p is expanded as

$$\tilde{p}_h(\tau, \xi) = \sum_{i=1}^{p_t} \sum_{j=0}^{p_x} \tilde{p}_{ij} \tilde{e}_i(\tilde{\tau}) \tilde{h}_j(\tilde{\xi}), \tag{6.9}$$

where $\tilde{p}_{ij} = \mathbb{H}_F F_{ij}$ (for example, depending on the choice of constitutive equation, this choice is explained below). Each variable is associated to a mesh element which is schematically shown in Figure 6.2. Note that almost all geometric elements are now used, as opposed to the mesh for Burgers' equation (Figure 5.3). The mesh clearly shows that the pressure acts over a time period (a pressure impulse) in which it affects the momentum. This is different from most conventional numerical methods, where pressures are often discretised in points and the coupling with the velocity becomes difficult. Even most mimetic methods, see [231] as an example, do not consider the temporal aspect of the pressure.

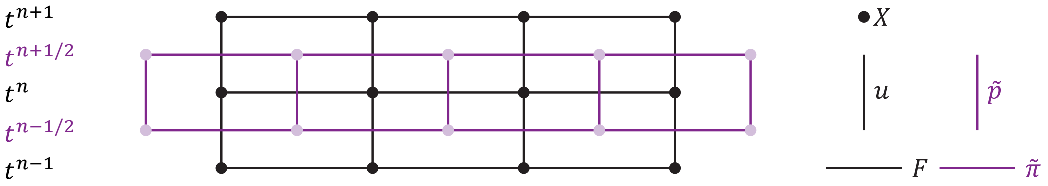


Figure 6.2: Example of a mesh (in the reference domain) for the Euler equations and the association of the variables. The primal mesh with inner orientation is shown in black, while the dual mesh with outer orientation is shown in purple. The associations are shown on the right. The treatment of the initial and boundary conditions is not shown.

Remark. The dual mesh has again been drawn as if it crosses the element boundary. Nevertheless, the dual mesh also only covers the spatial region of the same size as the primal mesh. This is clearest when considering the Hodge (or mass) matrices that are always defined based on the information inside the domain. The dual mesh thus does not stick out. Nevertheless, it was drawn like this to show that a horizontal line is always dual to a vertical one. Additionally, drawing a shorter line for the dual mesh might imply that the value near the edge is only half of one piece, which is not the case either.

With the expansions as indicated above, a system of equations valid on each element can be assembled. This is new for the Euler equations and stems from the dependence of the pressure on the volume (and thus new location). The system can be written as

$$\begin{bmatrix} \tilde{\mathbb{E}}_x^{(1,0)} \mathbb{H}_F \mathbb{E}_x^{(1,0)} & \tilde{\mathbb{E}}_t^{(1,0)} \\ \mathbb{E}_t^{(1,0)} & -\mathbb{H}_{\rho_0}^{-1} \end{bmatrix} \begin{bmatrix} x \\ \tilde{\pi} \end{bmatrix} \tag{6.10}$$

where $\mathbb{H}_F$ is the pressure-Hodge matrix for the constitutive equation of the form

$$p = \frac{\rho_0^{\gamma}}{F^{(\gamma+1)}} F \tag{6.11}$$

which will be used subsequently. The reason for this choice will be explained in the next section (note that this is a more implicit version of $p = \rho^{\gamma}$). Note also that there is no right-hand side in Equation (6.10) because the right-hand side will only become non-zero once the initial and boundary conditions are specified, which will also be done in the next section. The full[22] matrix indicates that the two equations are coupled. In this assembled matrix, the top row represents conservation of momentum (Equations (6.1a) and (6.4a)) while the bottom row represents the kinematics/advection (Equations (6.1b) and (6.4b)). The ordering of the two equations is important, as it represents a finite element type of weak formulation through

$$\begin{bmatrix} x & \tilde{\pi} \end{bmatrix} \begin{bmatrix} \tilde{\mathbb{E}}_x^{(1,0)} \mathbb{H}_F \mathbb{E}_x^{(1,0)} & \tilde{\mathbb{E}}_t^{(1,0)} \\ \mathbb{E}_t^{(1,0)} & -\mathbb{H}_{\rho_0}^{-1} \end{bmatrix} \begin{bmatrix} x \\ \tilde{\pi} \end{bmatrix} \tag{6.12}$$

where x and $\tilde{\pi}$ now also act as finite element weighting functions. Note, however, that no variational formulation is present in this work. Such theoretical considerations are left for future work. Nevertheless, the weighting is more than a mathematical trick. Consider one of the black dots in Figure 6.2 (in the centre, the boundaries are ignored for now). This black dot represents a fluid particle at a given time instant. This dot is surrounded by a purple square of the dual grid. The conservation of momentum around this particle represents the weak formulation and a graphical way of looking at the weighting. This view on the weak formulation was already advocated by Bossavit [67].

Remark. The weak aspect of the formulation enters through the finite (spectral) element approximation of the Hodge matrices. The kinematic and dynamic constraints (governing equations) are still strongly enforced on the primal and dual grids, respectively. It is the relation between them (representing energy according to the above interpretation) that is weakly enforced/approximated.

Remark. In Figure 6.2 (and throughout this report for that matter) the dual grid has been explicitly drawn with the edges crossing the middle of the primal edges. While this is a convenient way of representing the dual grid, the exact locations of the dual degrees of freedom are not known [86]. It is perhaps best to only consider a 'fuzzy' representation of the dual grid, at least when interpreting the physics of the problem (not unlike Mattiussi's attempt at explaining the finite element interpretation of mimetic methods [45]).

The 'fuzzy' interpretation of the dual grid in the above remark means that one should consider the primal points a fluid particle with its dual surface representing a form of a sphere (or square) of influence. The exact extent of the influence is unknown (though, thanks to the finite/spectral element view, it stays within an element), but will be handled by the Hodge matrices. This interaction with neighbouring particles extends over the entire element, though tends to be the strongest with the direct neighbour. Note that this non-locality also holds in time. The temporal aspect leads to the inclusion of the history of the particles and their interactions. These aspects are also the reason behind the implicit nature of the equations, both in space as well as in time.

So far, the development of the Euler solver has been similar to that of the Burgers' equation solver, albeit with some additional variables and thus the assembly of an interdependent system. However, where the method of Chapter 5 was effectively complete at this stage, two important aspects must be carefully dealt with for the Euler solver; the initial and boundary (BC) conditions, and the non-linearity of the pressure.

6.2.1. Incorporating the Initial and Boundary Conditions

As mentioned in the previous chapter, the incorporation of the initial and boundary conditions for the Euler solver works differently compared to that of the Burgers' equation solver. In the case of Burgers' equation, no boundary conditions or inter-element continuity was needed. The elements could be solved and moved independently of one another. This no longer holds for the Euler equations.

---

[22] Full in the sense that all four blocks in the assembled matrix are filled with a matrix. The individual blocks, however, have a much sparser structure.

The reason for this lies in the character of the equations. Mathematically, both Burgers' and Euler's equations are hyperbolic. However, when discretised, the situation changes. Burgers' equation remains completely hyperbolic, as the elements did not interact. In the case of Euler's equations, however, the elements interact through pressure waves. This is particularly clear in the Lagrangian formulation (Equations (6.1) and (6.4)) where the two balance equations are the kinematics of advection and the dynamics of conservation of momentum that represents a pressure wave. This wave equation for the pressure, which requires a gradient in the spatial direction, is what couples the elements. This split between advection and acoustic (that is, pressure) waves is also the reason behind the active flux method developed by Eymann, Roe and collaborators [107, 232–241], albeit they do not use a Lagrangian formulation. As such, the combination of a pressure wave and advection makes the discrete Euler equations a combination of a few hyperbolic equations.

This change in hyperbolicity makes these equations somewhat more difficult to solve, as the elements are now interconnected. The fact that this requires treatment of the boundaries, both between elements as well as the edges of the domain, can be visualised by looking at Figure 6.2. As can be seen from the figure, the right-most purple edge has a pressure associated to it. However, the domain ends at the right-most black edge. This also holds for the other elements. Some treatment of the boundary is thus natural. Imagine then that, somewhat heuristically, a 'limit' is taken by moving the purple edge closer and closer to the black one. In the limit case when the purple edge coincides with the black edge, all boundary information is available. This is a simple explanation of the need to specify boundary conditions due to the exchange of pressure waves between fluid particles.

Enforcing the initial and boundary conditions can be graphically represented by closing the dual grid. So far, the meshes have been drawn like in Figures 2.4 and 6.2. In Figure 2.4 the dual mesh is visibly open, while in Figure 6.2 it appears to be closed but overlaps. In the previous chapters, this was of no consequence. Here, however, it is important to accurately treat the IC and BCs.

To correctly interpret the meaning of the additional variables that are added, it is important to understand the meaning of the variables at the boundaries. Consider, therefore, a piece of the global domain (the boundaries of the global domain are ignored here). This piece will be divided into two elements. This is shown in Figure 6.3. The (black) nodes on the split line (vertical line through the 3rd horizontal nodes) are represented twice; once on the left element and once on the right element. Despite this double representation, it is the *same global node*. The same thing holds for the vertical black line connecting two black dots (to which a velocity is associated). Since the velocity has momentum as its dual, the momenta on the edges are also the same global momenta.

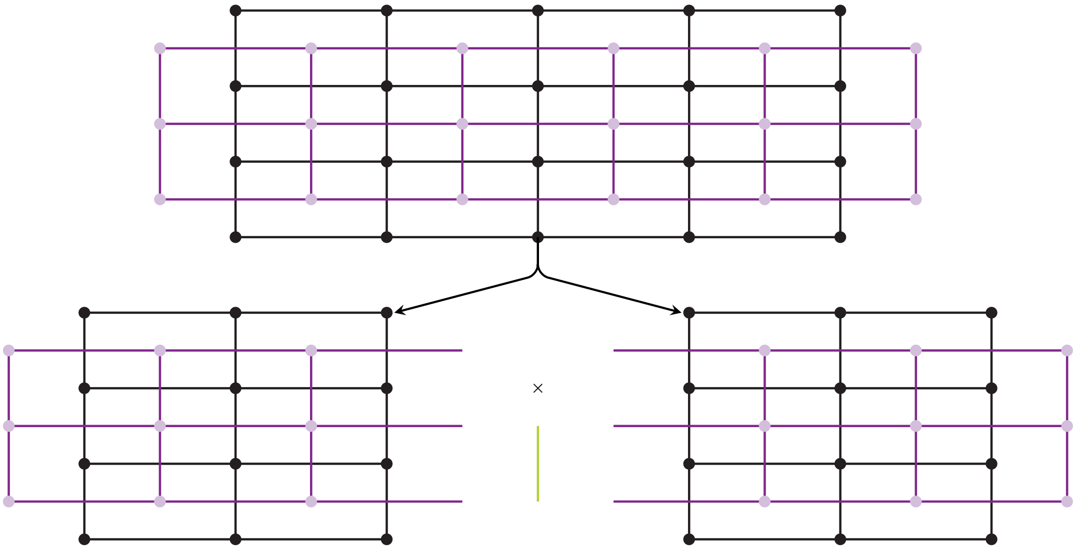


Figure 6.3: Example of a piece of the global mesh (top) as it is split (in the middle of the black grid) into two elements (bottom). The cross indicates the location of the enforcement of the continuity constraint and the green line is the associated Lagrange multiplier.

With this interpretation in mind, enforcing continuity conditions is straightforward. Note that only such conditions are needed, as only periodic boundary conditions will be used[23]. Enforcing continuity boils down to enforcing that the two representations of the global black dots end up at the same coordinate. This is achieved by using Lagrange multipliers. While continuity can also be added using a gathering matrix approach, Lagrange multipliers are easier to use in a space-time formulation, have a physical interpretation, and can be used to create a hybridisation (in space and time) which is potentially more efficient to solve [14, 242, 243]. The hybrid part will automatically result, though the more efficient method to solve will be left for future implementations. The details of this approach can be found in [14, 242, 243]. Note that the continuity matrix (see below) used here plays the role of the trace matrices used in [14, 242, 243].

In particular, when enforcing continuity between two nodes (the location is indicated by the cross in Figure 6.3), a Lagrange multiplier (indicated by the green line in Figure 6.3) will be added that enforces this constraint. This Lagrange multiplier acts as a pressure(-like) force in the indicated equation. Note that the Lagrange multiplier ensures that the momentum below the cross is adapted so that the velocity to which it is dual will be the same on both elements. Then, since the starting position is already equal, this will result in the desired equality of the position on both elements. Remember, however, that the figure is only a representation of the dual variables because the exact location is unknown. The Lagrange multipliers ensure that the influences from one element are transferred to the other since the elements have no other way of communicating.

The resulting system, including Lagrange multipliers and continuity constraints, looks like

$$\begin{bmatrix} A & C^T \\ C & 0 \end{bmatrix} \begin{bmatrix} x \\ \tilde{\pi} \\ \lambda \end{bmatrix}, \tag{6.13}$$

where A is a block diagonal matrix with each block being the block matrix of Equation (6.10), C the continuity matrix and C^T its transpose[24], and λ represents the Lagrange multipliers.

Next to the periodicity (and continuity) constraints just discussed, there needs to be a way to enforce the initial conditions. To achieve this, the dual grid must be closed, as shown in Figure 6.4. The orange momenta at the top and bottom close the dual grid in the temporal direction. The Lagrange multipliers for the continuity enforcement are also shown for completeness. Note that the orange momenta are drawn slightly above and blow the grid for clarity. Furthermore, the momenta at the top are, at the ends, not pure momenta but include part of the pressure. It is important to note that these momenta at the top serve only to ensure proper continuity in time when moving to the next time step. Neither the momenta at the top nor at the bottom take part in the weighing (like Equation (6.12)) and reconstruction, and are not affected by the Hodge matrices.

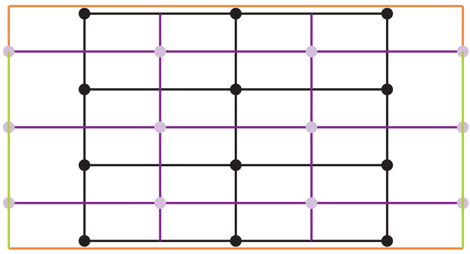


Figure 6.4: Example of a closed dual mesh in the temporal direction.

Adding these momenta at the top and bottom is easy to implement and does not change the structure of Equation (6.13). It only adds some extra momentum variables. The incidence matrices are straightforwardly extended, and the Hodge matrices are padded with zeros to ensure the system has the correct size.

Enforcing the initial conditions can then be done by removing the respective columns from the equations and moving them to the right-hand side of the equation. This can be done for both the

---

[23] The only difference between enforcing continuity and periodicity is the right-hand side. For continuity, the location of two particles must be the same, while there should be a difference equal to the domain length in the case of periodicity.

[24] In this work, there is no underlying variational principle that dictates the use of a transpose. Nevertheless, the transpose matrix does result.

momenta at the bottom as well as the positions at the bottom. Note that this does not change the geometric representation of Figure 6.4 since the specification of the initial conditions is only a numerical consideration, not a geometric one.

Remark. The closing of the dual grid and thus the inclusion of the boundary terms for the momentum equation is natural on the dual grid and represents integration by parts. This can be written as

$$\int_{t^0}^{t^1} \pi \frac{\partial x}{\partial t}\, dt = \pi x|_{t^0}^{t^1} - \int_{t^0}^{t^1} \frac{\partial \pi}{\partial t} x\, dt. \tag{6.14}$$

The first term on the right-hand side, with the boundaries, is what is handled by the orange momenta in Figure 6.4. The second term on the right-hand side is handled by the incidence matrix. Note, however, that this only weakly enforces the initial conditions for the momentum, as it is enforced on the dual grid.

Remark. As can be seen in Figure 6.4, the additional momenta are drawn directly at the edge of the element. This is different from the representation in the horizontal direction where the horizontal lines stick out of the element. The two procedures have slightly different interpretations as described above. Nevertheless, if one would prefer to draw the horizontal lines such that they too end at the edge of the element, then this can be done. This will not change anything in the implementation. Either way, the exact locations of the dual variables are unknown, so both representations are just that, representations.

The above discussion involved the initial conditions for the position and the momentum. However, to start the problem, an initial condition for the density (or pressure) is also required. These initial conditions are used in the Hodge matrices. For the momentum-velocity Hodge, only the initial density is needed and this will not change during the simulation. For the pressure-deformation gradient Hodge, the density to the power gamma (the pressure) goes into the Hodge matrix.

At this stage, the method functions and will produce results. While these results appear to look good initially, the results deviate from the exact solution as time progresses. This has a simple origin; the method does not account for the non-linear relation between the pressure and the deformation gradient. How this can be done will be described next.

6.2.2. Numerical Treatment of the Pressure Non-Linearity

So far, a complete and solvable system has been set up and solving this system will produce a result that looks good initially, but then deviates from the exact solution. The reason for this is simple; the relation between the pressure and deformation gradient (or density) is non-linear, while only linear effects are accounted for in the system.

There is some difference at this stage between the various constitutive equations for the pressure. Compare, in particular, the formulations $p = \rho^\gamma$ and $p = K_s\Theta$. When creating a Hodge matrix using the first equation, the material parameter will be ρ_0^γ and there will be no variation in the pressure during the solve. For the second equation, the material parameter K_s will move into the Hodge matrix, but $\Theta = F - 1$ so that it depends on x and will vary during the solve. Unfortunately, this variation is only linear in x (or F) as opposed to F^γ. An additional step must thus be taken to find the correct pressure.

In the current implementation, the constitutive equation of the form $p = \frac{\rho_0^\gamma}{F^{(\gamma+1)}}F$ is used. This formulation resembles $p = \rho^\gamma$ but takes the first variation into account. While $p = K_s\Theta$ would also do that, the formulation $p = \frac{\rho_0^\gamma}{F^{(\gamma+1)}}F$ is easier to implement, making it more suitable for a proof-of-concept method.

Accounting for the pressure non-linearity is most easily done through the use of iterations. In this work, a very simple iteration scheme will be used, namely fixed-point iteration. This means that, after having solved the linear system once, the pressure Hodge matrix is updated based on the new x-values, and the system is solved again. This can be repeated various times to further improve the solution.

While it may be possible to design more sophisticated iteration schemes, this simple iterator will suffice for the proof-of-concept methods developed here. More extensive iterators, or altogether different strategies for dealing with the non-linearities, are left for future work. It is, however, already expected that enforcing certain constraints (energy in particular) while iterating could provide the most benefits.

The complete LMSEM method for the isentropic Euler equations is shown in Algorithm 6.1. Note that only the main steps are shown, as the reconstructions and creation of the incidence matrices are no different from the algorithm for Burgers' equation (see Algorithm 5.1).

Algorithm 6.1: LMSEM for the isentropic Euler equations.

```
/* Step 1: Discretise the initial conditions. */
Use Equations (6.5) to (6.9).

for time-step in $K_t$ do
    for iteration in maximum-iterations do
        /* Step 2: Element matrices. */
        for $k$ in $K_x$ do
            Build the matrix of Equation (6.10).
        end

        /* Step 3: Assemble and solve the global system. */
        Assemble the matrix as in Equation (6.13) and solve.

        /* Step 4: Update the pressure Hodge matrix. */
        for $k$ in $K_x$ do
            Use Equation (6.7) for the reconstruction of the deformation gradient and
              Equation (6.11) for the pressure.
        end
    end

    /* Step 5: Reconstruct the variables in the physical domain. */
    for $k$ in $K_x$ do
        Use Equations (6.5) to (6.9).
    end
end
```

6.3. Example Problem: Smooth Isentropic Euler Flow

The method can now be tested using an example problem. To avoid potential difficulties due to shocks, a test case with a smooth solution is used. The test case was developed by Cheng and Shu [227] (variations of this problem can be found in for example [244, 245]) and was specifically designed to show the positivity-preserving property of the limiters designed in [227]. As such, conventional Lagrangian schemes without limiters will produce non-physical negative values for the density and other variables [227]. This is thus a challenging test case. The initial conditions are

$$\rho_0(x) = 1 + 0.9999995\sin(\pi x), \quad u_0(x) = 0, \quad p_0(x) = \rho_0^\gamma(x), \tag{6.15}$$

for $x \in [-1, 1]$, with periodic boundary conditions, and the special value of $\gamma = 3$. In such formulas, π simply represents the number. The solution is computed on the time interval $t \in [0, 0.1]$. Using $\gamma = 3$ reduces the Euler equations to two independent Burgers' equations for the Riemann invariants. The solution can then be found as

$$\rho(t, x(t)) = \frac{1}{2}(\rho_0(x_{0,-}) + \rho_0(x_{0,+})) \quad u(t, x(t)) = \sqrt{3}(\rho(t, x(t)) - \rho_0(x_{0,-})), \tag{6.16}$$

where $x_{0,-}$ and $x_{0,+}$ are the positions of the left and right running characteristics at $t = 0$ which reach $x(t)$ at time t, respectively. These positions can be found by backtracking the characteristic equations, which is done by solving

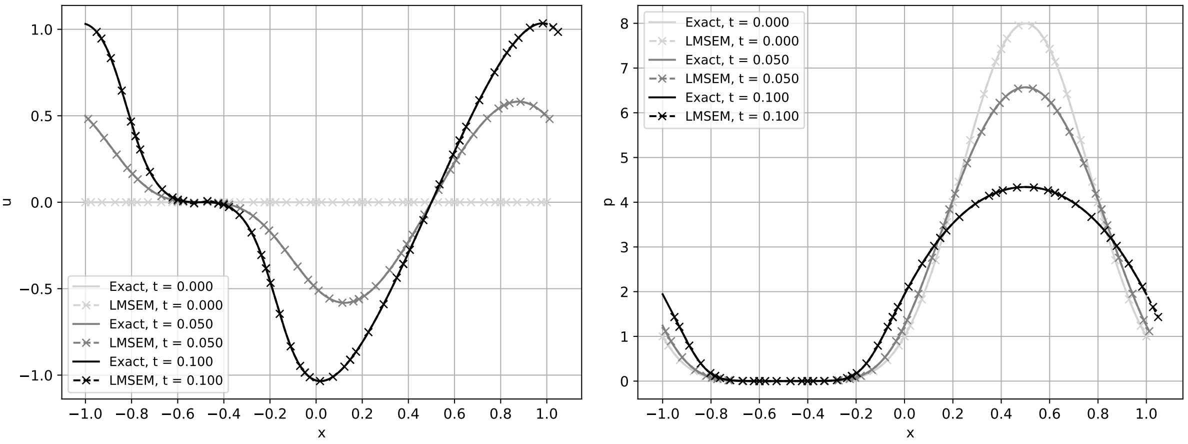
$$x_{0,-} - \sqrt{3}\rho_0(x_{0,-})t - x(t) = 0, \tag{6.17}$$

$$x_{0,+} + \sqrt{3}\rho_0(x_{0,+})t - x(t) = 0, \tag{6.18}$$

for $x_{0,-}$ and $x_{0,+}$. See [246], or Appendix A.3 for some additional details (note that the solution is not explicitly given in [227]). The periodic boundary conditions are automatically incorporated into the analytical solution due to the oddness of the sin-function in the initial condition, see the remark in

Appendix A.3. However, for the numerical method, the boundary conditions must be explicitly enforced. How this is done was already explained in Section 6.2.

Some example results are shown in Figure 6.5 together with the exact solution. The solution is indistinguishable from the exact solution in these graphs for the velocity and pressure variables, indicating that the LMSEM can, at least at first glance, generate results that look good for this challenging test problem. Note also that this setup with higher degree polynomials in space and time does not suffer from problems in the vacuum region (around $x = -0.5$), which is where the challenging aspect of this test case comes from [227]. It should be noted that the reconstructions of the solutions in this section use 75 equally spaced points plus the quadrature nodes per element. The duplicates are removed.

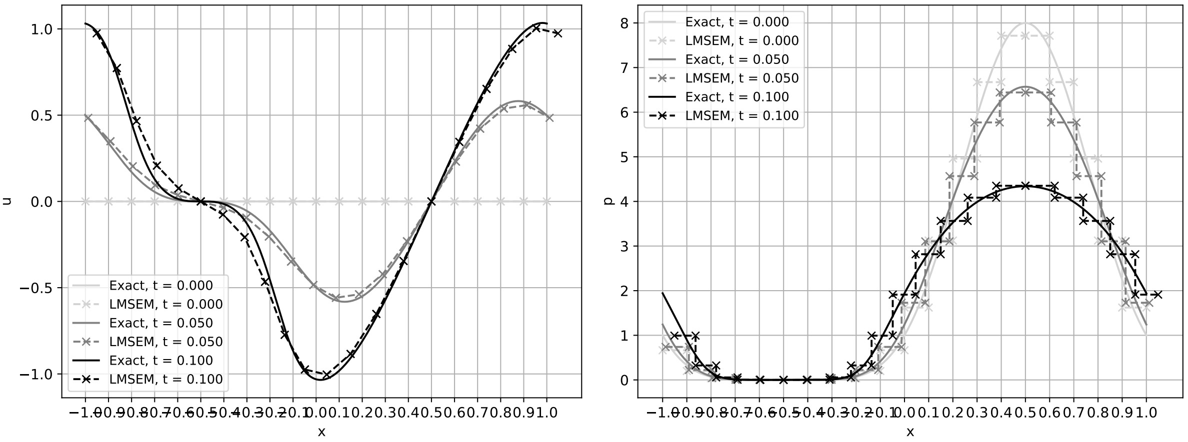


(a) Solution for the velocity u.

(b) Solution for the pressure p.

Figure 6.5: Solutions for the velocity and pressure variables during the simulation with $p_t = 4$, $p_x = 5$, thirty time-steps, ten spatial elements, and three pressure iterations per time-step. The grid lines indicate the element boundaries of the starting grid and the markers the GLL-nodes.

Another interesting result is obtained when using $p_x = 1$. These results are shown in Figure 6.6. While the number of elements in space is a little low for the solution to be close to the exact solution, the solution does start to resemble the correct shape. Note that the horizontal axis is somewhat difficult to read due to the use of twenty elements. The axis labels range from -1.0 to 1.0 in steps of 0.1.

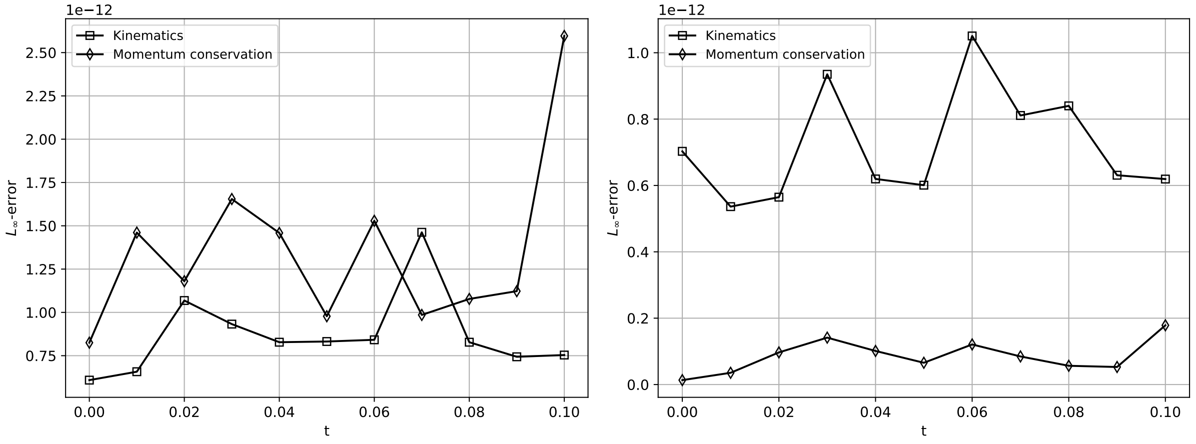


(a) Solution for the velocity u.

(b) Solution for the pressure p.

Figure 6.6: Solutions for the velocity and pressure variables during the simulation with $p_t = 4$, $p_x = 1$, thirty time-steps, twenty spatial elements, and three pressure iterations per time-step. The grid lines indicate the element boundaries of the starting grid and the markers the GLL-nodes.

Next to the example results, it is also important to confirm that the balance laws (kinematic and dynamic constraints) are indeed correctly satisfied. This is shown in Figure 6.7. Note that these errors are reconstructed over the domain so that this shows the adherence to the balance laws in the strong sense. This is also the reason why the errors are slightly higher than machine precision. Nevertheless, for both cases ($p_x = 1$ and $p_x = 5$) the same result is obtained. This is one of the main strengths of a mimetic method; it satisfies the conservation laws independently of the basis functions. This means that the obtained solution is always physically admissible in the sense that the physical balance laws are adhered to.



(a) Conservation errors with $p_x = 5$ on ten elements.

(b) Conservation errors with $p_x = 1$ on twenty elements.

Figure 6.7: L_∞-error over time for the isentropic Euler equations with $p_t = 4$, thirty time-steps, and three pressure iterations per time-step. The errors for conservation of momentum are computed on the dual grid, while the errors in the kinematic constraint are computed on the primal grid.

It should, however, be noted that there are cases in which the results are not good. This occurs when two nodes overtake each other leading to negative pressures and other unphysical phenomena. This, however, tends to occur in the region where the pressure is largest (around $x = 0.5$ in Figure 6.5b) *not* in the near-vacuum region. The reason lies in the iteration procedure for the pressure. The system that is solved only accounts for a linear change in pressure. In particular, when the distance between two nodes halves, the pressure will only (approximately) double, while its constitutive relation would give an increase by a factor of about eight. This large difference would normally prevent particles from overtaking each other (especially since this problem is shock-free), but the linear approximation does not enforce such an effect. Since the new material parameter (Hodge matrix) is computed based on the linear values, unphysical phenomena can occur in this case. Such effects are more likely to happen when using large timesteps and/or large p_x with many elements in space. Additionally, like the algorithm for Burgers' equation, only equal-sized time steps can be used due to the use of integral values in time. Nevertheless, there are plenty of settings that can be successfully used and produce good results. Improving the iteration procedure is therefore left for future work.

The example results show that the method is successful in generating solutions that resemble the exact solution. It is also interesting to investigate its convergence behaviour by means of hp-refinement (p-refinement in both time and space). This is investigated next.

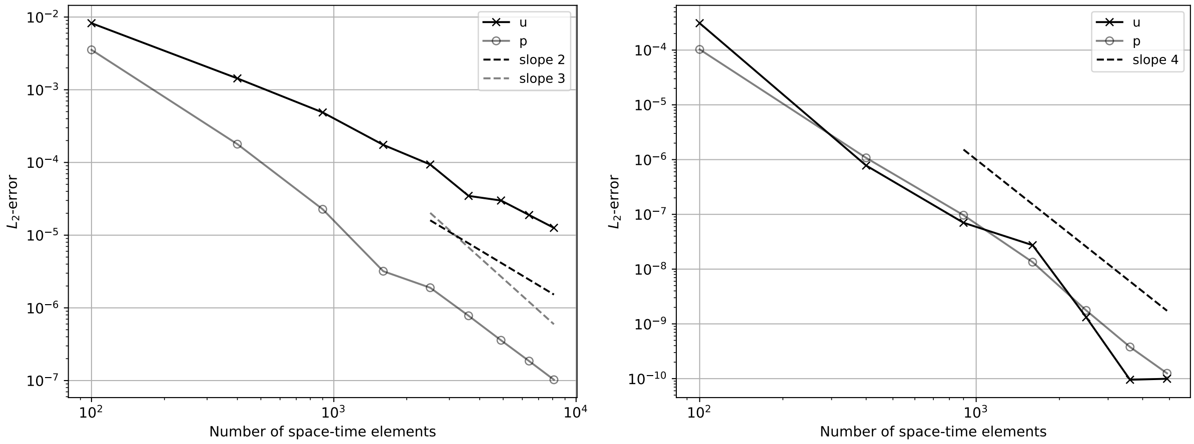
Starting with h-refinement, it should be noted that h-refinement means that both the sizes of the time-step and spatial element are reduced so that the ratio stays the same. To avoid some ambiguities in defining a size for h, the number of elements is used on the horizontal axis instead. The number of elements in space and the number of time steps for the total number of space-time elements are shown in Table 6.1. The starting number of elements was chosen such that it produced a reasonable-looking solution and allowed the number of elements to be easily increased.

The resulting graphs are shown in Figure 6.8. Firstly, in Figure 6.8a, the convergence study was performed using $p_t = 4$ and $p_x = 5$. Nevertheless, the convergence rates appear to be limited to roughly second order. This is much less than optimal. A hint at its cause can be found in Figure 6.8b, where the convergence study has been repeated using $p_t = 4$ and $p_x = 8$. The convergence rates are now of about order four, which is optimal for the velocity and nearly so for the pressure. Note that the convergence plot in Figure 6.8b has two refinements less than that in Figure 6.8a because the

Table 6.1: Number of elements in space and time used for the Euler grid-refinement study.

Number of space-time elements	Number of elements in space	Number of timesteps
100	5	20
400	10	40
900	15	60
1600	20	80
2500	25	100
3600	30	120
4900	35	140
6400	40	160
8100	45	180

higher polynomial degree on that many elements becomes unstable. These two plots indicate that the spatial error is dominant. To understand why this causes the h-refinement graph to show suboptimal convergence rates, the behaviour using p_t-refinement must be investigated first.



(a) h-refined with $p_x = 5$.

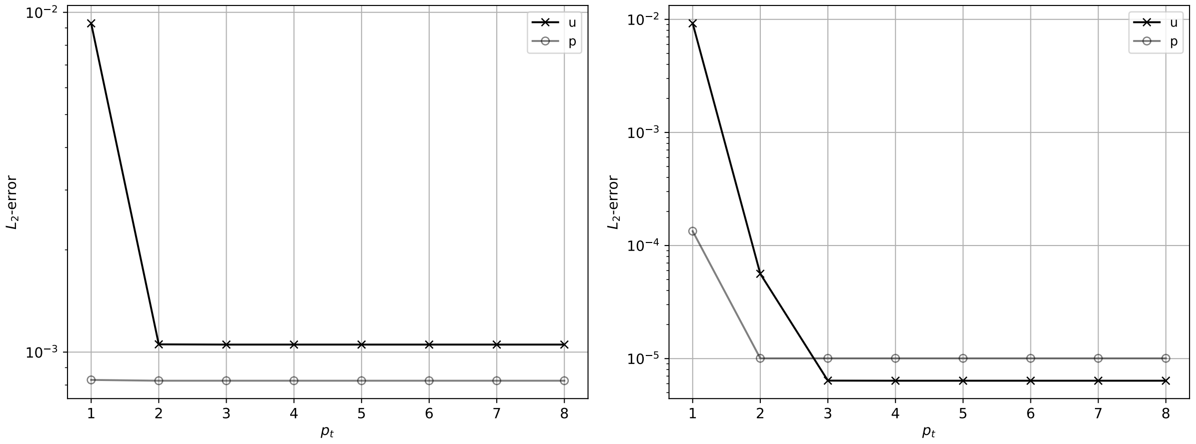
(b) h-refined with $p_x = 8$.

Figure 6.8: L_2-errors at $t = 0.1$ for the Euler test case in the velocity variable u and pressure variable p using h-refinement with $p_t = 4$, and four pressure iterations per time-step.

The p_t refinement graphs are shown in Figure 6.9. The graph in Figure 6.9a shows an improvement in the error for the velocity variable when moving from $p_t = 1$ to $p_t = 2$, but no more. The pressure graph shows the same error for all polynomial orders in time. It thus seems that the convergence rates are limited by the temporal polynomial degree. Also, it may seem from this graph as if the variation in pressure is perhaps exactly captured. This, however, should result in an h-refinement study that is limited by the spatial polynomial degree. Looking instead at Figure 6.9b, there is now a decrease in the error for both variables when going from $p_t = 1$ to $p_t = 2$. Additionally, the velocity error further decreases when going to $p_t = 3$. This is another indication that the temporal variation is not exactly captured. Nevertheless, the spatial error dominates preventing the polynomial in time to decrease the error further.

The variation in time of this test problem appears to be relatively slow. Therefore, a low polynomial degree in time ($p_t = 2$) gives a good approximation to the temporal variation in the solution. This causes the temporal error to be smaller than the spatial error, hence the dependence on the spatial polynomial degree. It seems, then, that the behaviour in time closely resembles a second-degree polynomial in time, though not exactly. In such a case, using higher polynomial degrees would result (in a monomial basis) in coefficients for the higher degree terms that are (very close to) zero. The resulting polynomial is then essentially still a second-degree polynomial, explaining the limited convergence behaviour using h-refinement. Future work, especially on test problems with larger variations in time, has to indicate in how far this explanation holds.

Moving on to the p_x-refinement graph (Figure 6.10) the behaviour is essentially what was expected.



(a) p_t-refined with $p_x = 4$.

(b) p_t-refined with $p_x = 7$.

Figure 6.9: L_2-errors at $t = 0.1$ for the Euler test case with p_t-refinement in the velocity variable u and pressure variable p with ten spatial elements, thirty time steps, and four pressure iterations per time-step.

The graph shows an essentially linear line in the semi-log plot for both variables. This hints at spectral convergence, as expected, though for the velocity variable there is one polynomial order which results in a slightly higher error than expected. This is caused by the difficulty of discretising the initial condition (the pressure in particular) by using equally spaced elements and higher-degree polynomials. Remember that the pressure initial condition, the pressure being a dual variable, follows from the discretisation of the initial density. This initial condition is used in the pressure-deformation gradient Hodge.

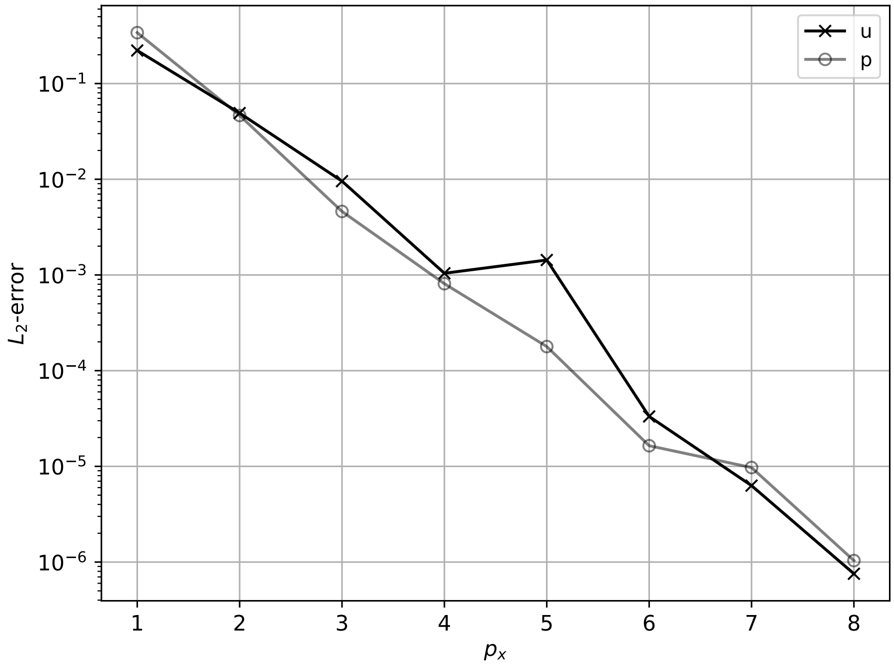
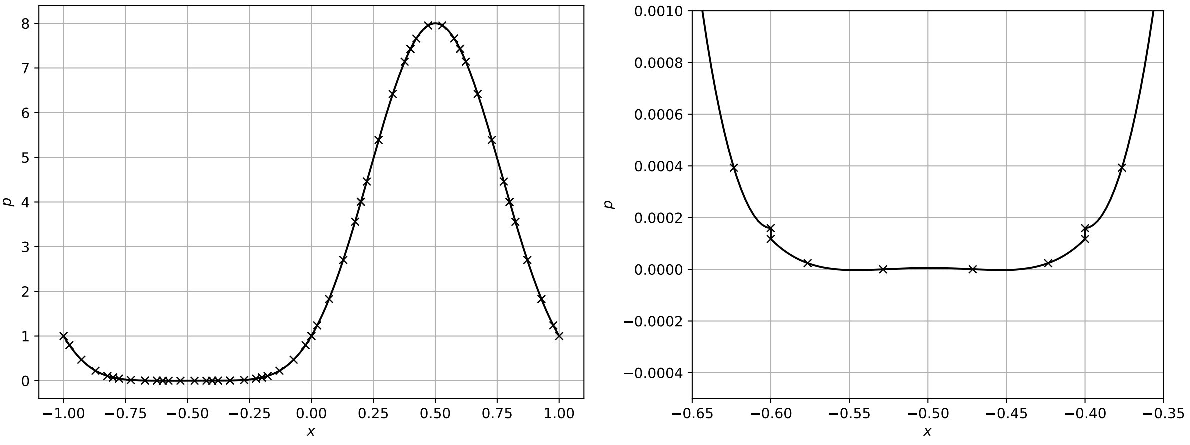


Figure 6.10: L_2-errors at $t = 0.1$ for the Euler test case with p_x-refinement in the velocity variable u and pressure variable p with ten spatial elements, thirty time-steps, $p_t = 4$, and four pressure iterations per time-step.

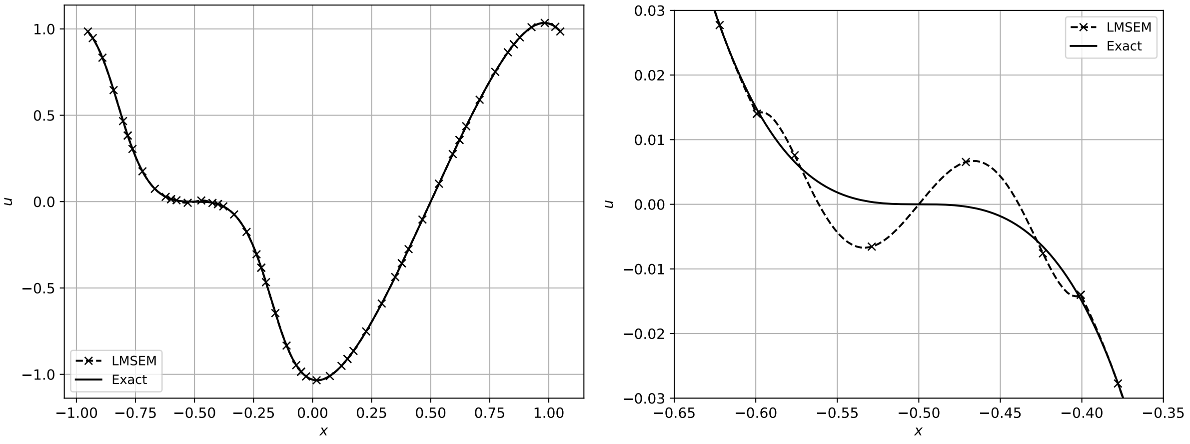
In this specific case, for $p_x = 5$, the discretisation happens to be suboptimal in the vacuum region (around $x = -0.5$), as shown in Figure 6.11. More carefully discretising the initial condition, that is adapting the starting grid to the features of the initial condition, can prevent such inaccuracies. Note that it is not the jump between the elements that causes this inaccuracy, but rather the changes in slopes at $x = -0.40$ and $x = -0.60$. After all, the pressure gradient in this region should be nearly zero, but the different slopes create non-zero pressure gradients instead, causing a motion that should not be there. The velocity solution for this setup is shown in Figure 6.12, which overall looks good, though the inaccuracy in the near-vacuum region can also be seen. This indeed shows a small additional velocity. Coupled with the exaggerating effect of the L_2-error norm on outliers [125], the resulting error is higher than one would expect for spectral convergence.



(a) Complete profile.

(b) Zoom-in of the near-vacuum region.

Figure 6.11: Discretisation of the pressure initial condition for the Euler test case with $p_x = 5$ on ten elements.



(a) Complete profile.

(b) Zoom-in of the near-vacuum region.

Figure 6.12: Final solution of the velocity for the Euler test case with $p_x = 5$ on ten elements.

This concludes the analysis of the LMSEM for the isentropic Euler equations. The accuracy of the results is encouraging and asks for future development of the scheme. Nevertheless, there are still some aspects that can be improved. This is of course left for future work. Before this thesis is concluded, however, the next chapter will present some additional theoretical/conceptual notes related to various aspects of this report.

7

Theoretical/Conceptual Notes

The previous chapters showed the development of a numerical method for advection equations and included some example problems. However, most of the mathematics that appeared were part of vector calculus, despite the introduction to differential forms of Chapter 2. In this chapter the method developed so far will be briefly investigated from a theoretical point of view. In particular, the relation to the Lie and covariant derivatives will be explained in Section 7.1. Then, in Section 7.2, a simple idea to show the solution on an Eulerian mesh is given. Additionally, some remarks regarding a more general framework of geometric fluid dynamics will be given in Section 7.3. It should be noted that the discussion in this chapter is not very formal, and merely serves as an indication of possible theoretical questions and extensions.

7.1. Lie and Covariant Derivatives: What has been Discretised?

As was already mentioned in Section 2.4.3 there is a close correspondence between the Lie and covariant derivatives. Both operators, however, disappear when using the Lagrangian formulation of fluid dynamics. It was therefore unnecessary to choose one of the two operators in advance. But now that the method developed in this work has been applied to Burgers' and Euler's equations, some indication can be given on which advection operator was (implicitly) discretised.

Given the properties pointed out in Sections 2.4.2 and 2.4.3 it appears that the method developed in this work is closely related to the covariant derivative. This has two main reasons. Firstly, being able to treat compressible flow, as done in Chapter 6, without requiring additional terms would not be possible if the Lie derivative had been discretised. Secondly, the mappings between the reference and physical domain are similar to using Equation (2.54), which showed the commuting property between the covariant derivative and coordinate transformations (mappings).

Of course, since a Lagrangian formulation was used, the link to the covariant derivative is not exact. After all, the covariant derivative disappears in the Lagrangian formulation. Nevertheless, given the close correspondence of the method with this operator, it seems worth investigating the covariant derivative further, also in a discrete setting. Such a comment was made before by for example Kreeft [247].

7.2. Extended Mappings for an Eulerian Method?

Throughout this work, the solution in the reference domain was mapped to the physical domain. In Chapter 4, this mapping, and the Jacobian in particular, included the flow map. However, in Chapters 5 and 6, the flow map was not needed and only the linear mapping to the starting domain was needed. This may make one wonder, can the flow map be included as well?

While no study was performed in this direction, some conceptual thoughts in this direction are written here. Including the flow map in the transformation would make the transformation seemingly more complicated, though its values (including the Jacobian) are all known numerically. To start with, including the flow map does not remove the non-linear aspects treated in Chapter 6. This is because the pressure term depends on the physical density, which can only be computed in the physical domain. The iterations will thus be over the mapping in this case.

Of course, the more extensive mapping will also mean that the Jacobian is more extensive. The Jacobian matrix can be written as

$$J = \begin{bmatrix} \frac{\partial t}{\partial \tau} & \frac{\partial t}{\partial \xi} \\ \frac{\partial x}{\partial \tau} & \frac{\partial x}{\partial \xi} \end{bmatrix}, \tag{7.1}$$

which, for the cases treated in Chapters 5 and 6 becomes

$$J = \begin{bmatrix} \frac{\Delta t}{2} & 0 \\ 0 & \frac{\Delta x_0}{2} \end{bmatrix}. \tag{7.2}$$

The diagonal structure makes the transformation particularly simple. This structure may disappear in multiple dimensions, as the zeros are only guaranteed for the time dependence of the coordinates. When including the flow map, however, the coordinates become time-dependent. In one dimension, the Jacobian is expected to look like

$$J = \begin{bmatrix} \frac{\Delta t}{2} & 0 \\ u(\tau, \xi) & \frac{\Delta x_0}{2} F(\tau, \xi) \end{bmatrix}. \tag{7.3}$$

The diagonal term for the spatial part is now extended by the deformation gradient (in multiple dimensions, this will be the determinant of the deformation gradient). The most important change, however, is the appearance of the velocity $u = \frac{\partial x}{\partial \tau}$ in the Jacobian. Using this mapping puts the advection part inside the mapping, which is more akin to the situation of Chapter 4. This may be a useful starting point for mapping towards stationary, or Eulerian, meshes, perhaps even in a mixed (Lagrangian/Eulerian) formulation. This, however, is left for future work.

As a final remark, note that the main change in this formulation and that used in this work is a matter of the coordinates that are mapped to. In this work, the physical domain was moving, so the velocity term was cancelled by the moving coordinates and did not appear in the mapping. In the above formulation, the mapping will be to stationary coordinates, so that the velocity does appear. The effectiveness of this approach is of course unknown at the time of writing.

7.3. Towards a Geometric Framework of Fluid Dynamics?

This work was concerned with the development of a numerical method. While some numerical methods have an extensive mathematical foundation, this was only true for the mimetic aspect of this work. As far as a guiding mathematical theorem or principle (such as a variational formulation), this work used none. While a more conventional underlying framework may be found, another direction can be to extend some of the conceptual ideas to a geometric framework in itself.

Such a framework may have value well beyond the development of numerical methods. There have been several frameworks describing fluid dynamics, such as the Port-Hamiltonian approach of Rashad et al. [248, 249], the multi-vector formalism of Panakkal, Parameswaran and Vedan [250], or the discrete framework of Caltagirone [251–254], to name a few. A framework borne of discrete geometric considerations is considerably different from the conventional mathematical theorems, though this may be an additional strength [179, 254]. Such a formulation would have similar geometric elements as used by Caltagirone [251–254], though his framework is almost a complete rewrite of the equations of physics and includes relativistic principles from the start. As a result, the main variables under consideration and the focus of the theory are different than envisioned here. Here, the theory is envisioned to provide a discrete, geometric formulation of classical fluid dynamics using the standard variables of interest.

The basis of this theory would consist of a diagram like that presented in Figure 6.1 for the Euler equations in the Lagrangian formulation. In particular, the clear separation of the configuration variables on the inner-oriented elements and the source variables on the outer-oriented elements is important. This allows two fundamental aspects, the kinematics (including advection and conservation of mass) and dynamics (conservation of momentum), to be in a geometric duality, like a weak formulation. It also seems that the outer orientation drives the system since this captures the directions of exchange. The inner-oriented variables then follow from the new sources, and, in the case of fluid dynamics, the new configuration feeds back to the driving variables creating the non-linearities. The connections between the inner and outer orientations may then be provided by energies (this was briefly alluded to in Chapter 6 and Tonti [48] has a similar view). Ensuring the various connections are consistent would then result in the conservation of energy. A similar view on thermodynamic consistency, as opposed to forcing energy conservation as another balance law, has recently appeared in various works in the literature, see for example Llor and Vazquez-Gonzalez [228, 255], Abgrall [229], or Busto and Dumbser [230, 256]. See also the earlier work of Caramana et al. [167]. Additionally, it may be interesting to compare the role of conservation of energy as pictured here with the existing geometric frameworks of thermodynamics, such as those of Weinhold [257–260], or Quevedo's 'Geometrothermodynamics' [261, 262]. After all, Weinhold clearly shows that the thermodynamic relations come from purely geometric and algebraic relations in his framework [259]. Such theories consider the geometry of thermodynamics only in thermodynamic phase space, so some connecting to the physical geometric elements used in this framework will be needed (most likely through extensive variables).

This work only considered fluid dynamics in the Lagrangian formulation. However, the Eulerian formulation is more common. An important difference, as it turns out, is that the two formulations have different geometric associations. This was explicitly pointed out by Tonti [48] and he expected this to have a role to play in computational methods. This work has shown that it indeed does. Furthermore, when considering discrete geometric aspects as a basis for a geometrical theory, this difference may have a more extensive role to play. The difference in geometrical association seems to hint at a duality between the two formulations, though this has to be studied in detail. The different underlying geometric associations also raise the question of the equivalence of the two formulations. Pairing this to the weak formulation inherent to mimetic methods, as clearly explained by Bossavit [67], and the fact that only the one-dimensional case was proven in the weak sense [124], such differences may perhaps pave the way towards geometric arguments for the equivalence of the two formulations. In particular, if the two formulations are indeed dual, a more general duality relation (see for example [76, 82]) may exist between the two formulations. This can perhaps be used as a starting point to investigate their equivalence.

The comments and ideas presented in this chapter are only in their infancy and require much more work. It does, however, show that some of the considerations and ideas used in this work can be much more general and fundamental than apparent at first glance. Of course, this work has its limits, so further extensions are left for future work. The next two chapters will wrap up the report by drawing some conclusions and presenting various recommendations for future work.

8

Conclusion

One of the most difficult aspects of computational fluid dynamics is the treatment of (non-linear) advection. This has led to the use of numerical methods that introduce dispersion and dissipation errors. With the advent of highly accurate mimetic methods, methods that aim to preserve key mathematical and physical properties in the discrete setting, the question of whether these new methods can solve advection problems arose. Several attempts could already be found in the literature, but none was as effective as hoped for. None of these schemes used a Lagrangian formulation of advection, nor were space-time methods created in a mimetic framework. With this in mind, this work aimed to answer the following research question.

> Can the Lagrangian view of advection be used to create a feasible, space-time numerical method for advection in the mimetic spectral element framework?

In support of the main question, three sub-questions were formulated. These asked whether the method produces the correct solution, whether it converges, and whether it can handle shock waves.

The first part of this work, after introducing the mimetic spectral element method and reviewing advection schemes, answered these questions while considering linear advection problems. The Lagrangian formulation allowed the advection problem to be split into two simple equations. The first equation describes what happens to the advected quantity (nothing), while the second gives the new location. In the case of linear advection, the two equations are decoupled. As a result, the equation for the position, the flow map, was analytically known and used, as was its Jacobian. This allowed the method to be developed while only considering the basic aspects. Considering spatially varying velocity fields allowed the investigation of non-divergence-free velocity fields. All in all, the method could positively answer the main and sub-questions (save of course the one involving shocks), while showing spectral convergence and exact adherence to the governing equation.

The second part built on this success by considering Burgers' equation. This was very briefly done with an analytically known flow map as well, though the dependence of the two equations already made this inconvenient. It was in this phase that the effectiveness of the Lagrangian, space-time and mimetic method became apparent. Burgers' equation was interpreted as the conservation of momentum for an incompressible fluid. With this interpretation and the space-time geometric associations of the variables, the strengths of the method could be extended to this case as well. The most important association is that of velocity, which was seen as a temporal one-form. This allowed the advective (kinematic to be complete) aspects to be handled exactly. With the help of an interpolation method from the literature, here reinterpreted solely in terms of momentum conservation, simple shocks could also be handled. This thus led to an affirmative answer to the research question once more, and again while showing spectral convergence and exact adherence to the balance laws.

The third and final part went on to investigate the isentropic Euler equations. This required the enforcement of boundary conditions next to the initial conditions, and the need to solve a non-linear equation. The results showed that the kinematic (advection and conservation of mass) and dynamic (conservation of momentum) constraints could again be adhered to exactly. A challenging test case was used as an example, though the challenges for the LMSEM are related to the non-linear constitutive equation as opposed to a near vacuum region in the test case. While spectral convergence was still achieved in the spatial direction, the convergence behaviour through h- and p_t-refinement was not optimal. This appears to be caused by the dominance of the spatial errors and the relatively slow variation in time of the test problem. As such, the base objective was achieved, though more work is needed to get the Euler solver as effective as the Burgers solver.

The work was concluded with some brief remarks on possible extensions and some theoretical background. After all, this work serves as a proof-of-concept for a Lagrangian mimetic method for advection. It was indeed shown, and this answers the main research question, that a Lagrangian, space-time and mimetic method is ideal for the numerical treatment of advection equations. Beyond that, the method can even be extended to compressible Euler flows. It is not just possible to use the Lagrangian formulation for advection, but it is almost mandatory to do so, at least if one desires to have a topological advection scheme. As such, the result of this work is a topological advection scheme with excellent conservation properties and mostly optimal convergence behaviour.

9

Recommendations

This work aimed to develop a proof-of-concept Lagrangian, space-time and mimetic method for advection equations. As was concluded in the previous chapter, this objective was achieved and the developed method has some desirable properties. Nevertheless, no method, and in particular a proof-of-concept version, is perfect so various aspects can be improved or extended, as will be detailed here.

Starting with the recommendations that are directly related to the implementation and ideas of the new method and not its extensions, the following can be said. This work used a primal mesh with an inner orientation and a dual mesh with an outer orientation. This choice, however, is arbitrary. Using a primal mesh with an outer orientation and a dual mesh with an inner orientation is possible as well. Given that enforcing continuity conditions is more naturally done on the primal grid and that such conditions are related to exchange terms, which have an outer orientation, using a primal mesh with an outer orientation may be more convenient. This may then simplify the process of enforcing continuity as used in Chapter 6.

Another aspect that may be useful to investigate before considering extensions is the performance of the method when investigating wave equations. In this work, the choice was made to move from Burgers' equation directly to the Euler equations. However, this adds considerable complexity to the equations. In particular, the Euler equations are effectively the advective part of Burgers' equation with a pressure wave superimposed on it. While the advective aspect was investigated separately through the investigation of Burgers' equation, studying a wave equation separately as well may lead to additional insights.

Of course, given the sub-optimal convergence behaviour in time for the Euler solver of Chapter 6, some additional work in investigating this behaviour is advised. While the non-linear aspects do not appear to be the limiting factor, it is also interesting to investigate their effects on the simulations. The importance of accurately resolving non-linear effects, and how this can be done efficiently, is particularly important for future work. There may also be a role for the conservation of energy in this regard.

Of lesser importance for the current work, but of increasing importance when considering extensions, are computational considerations. In this work, considerations of the computational complexity or efficiency were neglected, save perhaps for some basic vectorisation in Numpy [29]. This was possible because the focus was on the development of the method and the test problems were small enough to solve without advanced computational strategies. When considering more extensive and complex problems, such aspects become increasingly important. Implementation-wise, the barycentric form of the Lagrange polynomials should be considered to increase computational stability and evaluation speed for the nodal polynomials [263, 264]. This can also be extended to tensor products and derivatives [265], which may also be helpful for edge polynomials. In the developed method, the evaluation of the basis functions is performed often and at varying positions, so that this may improve the computational speed [265]. Another option may be the use of efficient solvers or preconditioners, such as [266]. When using non-direct (non-exact) solvers for mimetic methods, it is important to investigate the effect on the conservation error, which may not be on the machine-precision level anymore. This also ties in with the need to find suitable strategies for dealing with non-linear problems, which were only briefly considered in this work.

Next to the above-mentioned options for tweaking the current method, the creation of a proof-of-concept method of course asks for extensions.

To begin with, the computation of shocks was only briefly treated and the implemented algorithm does not fully represent the new interpretation. Yet, the new interpretation seems to be more in line with the rest of the method and may thus serve as a starting point for further investigations. Also extending this approach to the Euler equations is an interesting direction to pursue. After all, fluid dynamics shocks are the shocks of interest.

Next to extending the method to shocks, an extension to multiple (space) dimensions is also a natural step. Note that most of the conceptual thoughts in this work were already made in three space dimensions, but the implementation has only considered one space dimension. Extensions to multiple dimensions will thus focus more on the implementation than on the conceptualisation.

Another option for an extension is blending Lagrangian and Eulerian methods in a mimetic framework. While the method of Heumann and Hiptmair [149, 154–156] indeed considers a semi-Lagrangian method, they never considered the changes in geometry in the two formulations. This difference is of fundamental importance, and should thus have a significant effect on the development of such methods. It is, perhaps, helpful to first consider the Lagrangian and Eulerian versions separately in detail, before trying to mix them.

Finally, though for the more distant future, extensions to other physical theories, especially when mixed with fluid dynamics, can also be considered. In particular, extensions to magnetohydrodynamics may be interesting, given the Lagrangian formulation and accuracy (also over long periods) of the developed method. However, before such extensions are considered, a more extensive understanding of the method for fluid dynamics is advised.

Besides the numerically oriented recommendations already described, several notes throughout this work have indicated areas for further research from a conceptual or theoretical viewpoint.

Already in Chapter 2, it was mentioned that time is not commonly treated on the same footing as space in a mimetic method. The current work has shown the importance of the temporal aspect as well. As such, further investigations on how to take this aspect into account are warranted. Additionally, given the similarities between some existing (geometric) time integrators and mimetic methods, their connection should be further explored.

Finally, the possibility to extend the discrete framework used here to a more complete theory of fluid dynamics may also have merit. Such an endeavour was briefly mentioned in Chapter 7 and requires a considerable amount of work. Nevertheless, it provides a different, yet very consistent, view on fluid dynamics that is currently not available. Such a theory may be beneficial for numerical methods as well as the general field of fluid dynamics.

A

Derivations

This appendix details some derivations that were omitted in the main text for readability reasons. It includes the derivation of the solution for the linear advection equation with a temporally varying velocity field in Appendix A.1, the specific form of the first law of thermodynamics for isentropic flows in Appendix A.2, and the solution to the Euler flow test case of Chapter 6 in Appendix A.3.

A.1. Linear Advection with a Temporally Varying Velocity Field

The 1D linear advection equation with a temporally varying velocity field $u(t)$ defined for time $t >= 0$ on the open interval $-\infty < x < \infty$ can be written as

$$\frac{\partial \phi(t,x)}{\partial t} + u(t)\frac{\partial \phi(t,x)}{\partial x} = 0, \tag{A.1}$$

with the initial condition $\phi(t=0,x) = \phi_0(x_0)$ and t and x the temporal and spatial coordinates, respectively. This equation can be solved using the method of characteristics (see any book on partial differential equations, for example [113]). Using s for the characteristic direction, the following system of ordinary differential equations is found:

$$\frac{dt}{ds} = 1, \tag{A.2}$$

$$\frac{dx}{ds} = u(s), \tag{A.3}$$

$$\frac{d\phi(t(s),x(s))}{ds} = 0. \tag{A.4}$$

These equations can be solved directly (from $s=0$, where $t(s=0) = t_0 = 0$, $x(s=0) = x_0$) to get

$$t = s, \tag{A.5}$$

$$x = \int_0^s u(s^*)\,ds^* + x_0 = \int_0^t u(t^*)\,dt^* + x_0, \tag{A.6}$$

$$\phi(t(s),x(s)) = \phi_0(x_0). \tag{A.7}$$

Putting all these equations together gives (note that ϕ_0 has x_0 as argument and not x)

$$\phi(t,x) = \phi_0\left(x - \int_0^t u(t^*)\,dt^*\right). \tag{A.8}$$

For the velocity field of Section 4.3, $u(t) = \cos(at)$ with a a constant, this can be written as

$$\phi(t,x) = \phi_0\left(x - \int_0^t \cos(at^*)\,dt^*\right) = \phi_0\left(x - \frac{1}{a}\sin(at)\right). \tag{A.9}$$

A.2. The First Law of Thermodynamics for a Calorically Perfect Gas in an Isentropic Flow

The first law of thermodynamics describes the conservation of total internal energy and, assuming a reversible process, can be written as (see for example [112])

$$dE = Tds - pdV, \tag{A.10}$$

with E the total internal energy, T the temperature, s the entropy, p the pressure, and V de volume. For an isentropic flow (adiabatic and reversible), the entropy is constant, so $ds = 0$. The first law then becomes

$$dE = -pdV. \tag{A.11}$$

For a calorically perfect gas, the change in internal energy can be related to the change in temperature using the specific heat at constant volume c_v and the number of moles of the fluid n as

$$dE = nc_v dT, \tag{A.12}$$

leading to

$$nc_v dT = -pdV \tag{A.13}$$

as expression for the first law. The change in temperature can be related to changes in pressure and volume. Starting with the ideal gas law

$$pV = nRT, \tag{A.14}$$

with R the universal gas constant, and differentiating gives

$$pdV + Vdp = nRdT. \tag{A.15}$$

The temperature difference is thus

$$dT = \frac{pdV + Vdp}{nR}. \tag{A.16}$$

Combining this with Equation (A.13) gives

$$c_v \frac{pdV + Vdp}{R} = -pdV, \tag{A.17}$$

from which follows

$$(c_v + R)pdV + c_v Vdp = 0. \tag{A.18}$$

Recognising $c_v + R$ as the specific heat at constant pressure c_p leads to

$$c_p pdV + c_v Vdp = 0. \tag{A.19}$$

Dividing by c_v and denoting the ratio of specific heats as $\gamma = \frac{c_p}{c_v}$ gives

$$\gamma pdV + Vdp = 0, \tag{A.20}$$

the form of the first law as used in Chapter 6.

A.3. Solution to the Smooth Isentropic Euler Test Case

The smooth isentropic Euler test case introduced by Cheng and Shu [227] uses the initial conditions

$$\rho(0,x) = 1 + 0.9999995 \sin(\pi x), \quad u(0,x) = 0, \quad p(0,x) = \rho^\gamma(0,x), \tag{A.21}$$

with $x \in [-1, 1]$, periodic boundary conditions, and the special value of $\gamma = 3$. Using $\gamma = 3$ reduces the Euler equations to two independent Burgers' equations for the Riemann invariants. The Riemann invariants for the 1D isentropic (actually homentropic, since, $p(0,x) = \rho^\gamma(0,x)$) Euler equations with $\gamma = 3$ are $J^\pm = u \pm a$, with a the speed of sound. The two independent Burgers' equations are then

$$\frac{\partial J^\pm}{\partial t} + J^\pm \frac{\partial J^\pm}{\partial x} = 0. \tag{A.22}$$

The Burgers' equations can be solved using the method of characteristics, as was done in Appendix A.1. Again using s to denote the characteristic direction, the following system of ODEs is found:

$$\frac{dt}{ds} = 1, \tag{A.23}$$

$$\frac{dx}{ds} = J^{\pm}, \tag{A.24}$$

$$\frac{dJ^{\pm}(t(s), x(s))}{ds} = 0. \tag{A.25}$$

These equations can be solved directly (from $s = 0$, where $t(s = 0) = t_0 = 0$, $x(s = 0) = x_0$) to get

$$t = s, \tag{A.26}$$

$$x = \int_0^s J^{\pm}(t(s^*), x(s^*))\, ds^* + x_0 = J^{\pm}(t, x(t)) \int_0^t dt^* + x_0 = J^{\pm} t + x_0, \tag{A.27}$$

$$J^{\pm}(t(s), x(s)) = J^{\pm}(0, x_0). \tag{A.28}$$

To get to the solutions, a distinction between the left and right running characteristics is useful. The additional $+$ and $-$ subscripts indicate the characteristics related to J^+ and J^-, respectively. With this notation, the left running characteristic ($-$) gives

$$x_-(t) = J^- t + x_{0,-} = J^-(0, x_{0,-})t + x_{0,-} = -a(0, x_{0,-})t + x_{0,-}. \tag{A.29}$$

(Note that $u(t = 0) = 0$ from the initial conditions.) The speed of sound for an isentropic flow with $\gamma = 3$ becomes

$$a^2 = \frac{\partial p}{\partial \rho} = \gamma \rho^{\gamma - 1} = 3\rho^2 \rightarrow a = \sqrt{3}\rho. \tag{A.30}$$

The position of the left-running characteristic is thus

$$x(t) = x_{0,-} - \sqrt{3}\rho(0, x_{0,-})t. \tag{A.31}$$

Similarly, for the right-running characteristic, one obtains

$$x(t) = x_{0,+} + \sqrt{3}\rho(0, x_{0,+})t. \tag{A.32}$$

Note that $x_{0,-}$ is indicated as x_1 and $x_{0,+}$ as x_2 in [246]. $x_{0,-}$ and $x_{0,+}$ are both the starting locations, but when given an $x(t)$ the starting positions of the left and right running characteristics that reach $x(t)$ are different. Solving for one of $x_{0,-}$ or $x_{0,+}$ thus requires a non-linear equation to be solved (Equations (A.31) and (A.32), respectively).

The solution for the density and velocity can now be found. Note that the pressure can always be obtained from the density using $p = \rho^{\gamma}$. The velocity can be found from the Riemann invariant $J^- = u(t, x(t)) - a(t, x(t))$, which, using Equations (A.28) and (A.30), becomes

$$u(t, x(t)) - a(t, x(t)) = -a(0, x_{0,-}) \tag{A.33}$$

$$u(t, x(t)) - \sqrt{3}\rho(t, x(t)) = -\sqrt{3}\rho(0, x_{0,-}) \tag{A.34}$$

$$u(t, x(t)) = \sqrt{3}(\rho(t, x(t)) - \rho(0, x_{0,-})). \tag{A.35}$$

The density can be obtained by averaging the contributions from each characteristic (the two equations are independent) as

$$\rho(t, x(t)) = \frac{1}{2}(\rho(0, x_{0,-}) + \rho(0, x_{0,+})). \tag{A.36}$$

Remark. The solution was found without using the periodic boundary conditions. However, the equations solved here are hyperbolic. It is therefore possible to extend the initial condition beyond the original domain, leading to the same solution [113]. In particular, the extension of the initial condition must be odd, which the $\sin$-function already is. Therefore, without explicitly mentioning them, the periodic boundary conditions were automatically incorporated.

Summary

Recent advances in computational fluid dynamics have ensured its widespread use and reliability. Nevertheless, the accuracy is well-known to be somewhat lacking, as is its ability to treat non-linear flows without introducing dispersion or dissipation errors. Yet, the non-linearity of fluid dynamics is the source of most of its unique features and is thus of fundamental importance.

Since the nineties, the so-called mimetic methods have been on the rise. Mimetic methods try to 'mimic' the mathematical and physical properties of the continuous system in a discrete setting. This ensures the preservation of certain fundamental properties. In particular, most balance laws can be adhered to exactly (to machine precision), resulting in various benefits in terms of accuracy and stability.

Initially developed for electromagnetism and diffusion, recent applications also consider fluid dynamics. While mimetic methods achieved tremendous success for elliptic problems, the hyperbolic and non-linear character of advection and fluid dynamics have limited this success. While there have been various attempts at solving this problem, no work to date has considered a Lagrangian and space-time mimetic method. Therefore, this work aimed to answer the following research question.

> Can the Lagrangian view of advection be used to create a feasible, space-time numerical method for advection in the mimetic spectral element framework?

This question was investigated while considering three main problems; linear advection, Burgers' equation, and the isentropic Euler equations. The resulting scheme is termed the Lagrangian Mimetic Spectral Element Method (LMSEM).

The first part of this work concerned linear advection. This allowed the Mimetic Spectral Element Method (MSEM) to be adapted to advection equations in a Lagrangian formulation. In the Lagrangian formulation, the resulting equations can be written as

$$\begin{cases} \dfrac{\partial \phi}{\partial t} = 0, & \text{(A.37a)} \\ \dfrac{dX(t)}{dt} = u, & \text{(A.37b)} \end{cases}$$

with ϕ the advected quantity, t the temporal coordinate, $X(t)$ the moving position, and u the velocity. Due to the simplicity of the equation, the exact flow map $X(t)$ was used. This way, the focus of the development could be on the advected quantity and its transformation properties.

It was found that choosing the advected quantity to be a 0-form or 1-form could have a significant effect on the result. While irrelevant for divergence-free flows, the choice is important for non-divergence-free flows. This choice is normally dictated by the underlying physics of the problem, though such an interpretation is not available for linear advection. Yet, in both cases, the equations could be solved to machine precision if the initial condition was a polynomial. Otherwise, only the interpolation error of the initial condition would be present.

The second part of this work investigated Burgers' equation. The exact mapping was no longer used so a fully numerical method was created in this phase. In order to do this, a physical interpretation of Burgers' equation was given, which saw Burgers' equation as an expression of conservation of momentum for a fluid with mass density ρ equal to one. This then led to the association diagram as shown in Figure A.1.

The most important aspect of the diagram is the velocity, which is seen as a temporal 1-form. This diagram also shows the separation between the kinematics on the left side of the diagram and the conservation of momentum (dynamics) on the right side of the diagram. Using these associations, Burgers' equation can be written as

$$\begin{cases} \partial_t \tilde{\pi}^{(0,3)} = \tilde{0}^{(1,3)}, & \text{(A.38a)} \\ \partial_t \bar{X}^{(0,0)} = \bar{u}^{(1,0)}, & \text{(A.38b)} \\ \tilde{\pi}^{(0,3)} = \star \bar{u}^{(1,0)}. & \text{(A.38c)} \end{cases}$$

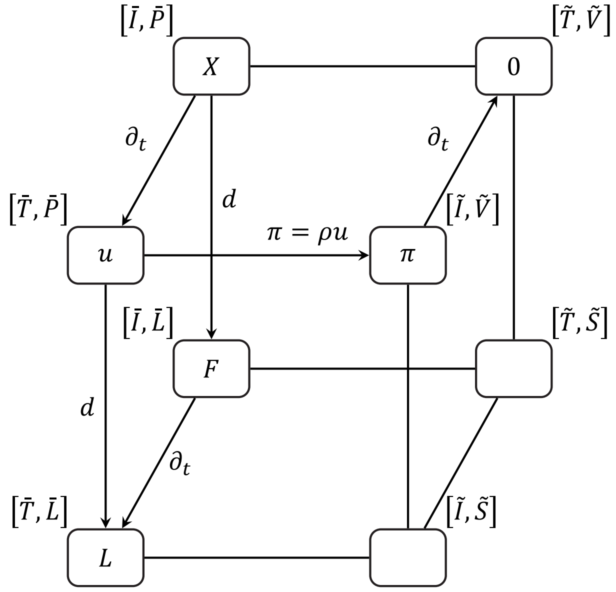


Figure A.1: Association diagram for Burgers' equation. Reprint of Figure 5.2 of this work.

with π the momentum and the other variables as above. This association of the velocity to temporal edges is what allows advection to be treated exactly and requires the Lagrangian formulation to be used.

The correct interpretation and associations allowed the kinematics (advection) and conservation of momentum (dynamics) to be adhered to exactly. The error in the solution could be made arbitrarily small by using hp-refinement for the discretisation of the initial condition. Using a simple interpolation method, even simple shocks could be dealt with. As this only affected the reconstruction step, no conservation errors were introduced with this addition.

The third and final part of this work investigated the isentropic Euler equations. These can be written as

$$\begin{cases} \partial_t \tilde{\pi}^{(0,3)} = -\, d\tilde{p}^{(1,2)}, & \text{(A.39a)} \\ \partial_t \bar{X}^{(0,0)} = \bar{u}^{(1,0)}, & \text{(A.39b)} \\ \tilde{\pi}^{(0,3)} = \star_{\rho_0} \bar{u}^{(1,0)}, & \text{(A.39c)} \\ \tilde{p}^{(1,2)} = \star_{F(\gamma+1)} \bar{F}^{(0,1)}, & \text{(A.39d)} \end{cases}$$

with p the pressure and γ the ratio of specific heats. The associations are similar to those shown in Figure A.1, though the pressure should be added to the $[\tilde{T},\tilde{S}]$ box. This shows that the pressure acts over a time period in which it affects the momentum.

The Euler solver also required the specification of inter-element continuity and (periodic) boundary conditions, which was achieved through the use of Lagrange multipliers. Additionally, the non-linear dependence of the pressure on the position (through the deformation gradient F) required iterations to resolve the non-linearity.

The Euler solver can still be improved in terms of efficiency and convergence behaviour in time, though it already shows spectral convergence in space and the kinematic (advection and conservation of mass) and dynamic (conservation of momentum) constraints are adhered to exactly. This again shows the suitability of the Lagrangian formulation and the temporal association of the velocity when investigating advection problems.

All in all, the Lagrangian formulation in combination with a space-time mimetic method is highly effective in treating advection. So much so that the LMSEM can be considered to be a topological advection scheme. Successfully handling compressible fluid dynamics on top of this makes the LMSEM a very powerful numerical method.

Of course, this work has its limits. Therefore, several avenues of interest for future work can be briefly outlined. Starting with some recommendations that are directly related to the LMSEM as developed in this work, one should first consider the MSEM performance when considering wave equations, and it would be interesting to investigate the usefulness of using an outer-oriented primal grid as opposed to an inner-oriented one. These two options were not considered in the current work but may provide a more detailed understanding of the LMSEM and a version that is more easily implemented.

As only a proof of concept method was developed in this work, it is natural to try to extend to method to more complicated problems. The initial extensions that should be considered are those pertaining to shocks and two-dimensional simulations. The starting points for such extensions were already provided in this work. When considering further extensions, it is important to maintain the code and ensure that its use of computational resources remains efficient, especially in the case of non-linear problems.

Lastly, the ideological ideas used in this work served as a foundation for the numerical method. However, they may be of more fundamental importance and can perhaps be used as a geometric basis for fluid dynamics. This would create a discrete framework for fluid dynamics, which is then easily discretised with the LMSEM.

References

[1] H. Aref *et al.*, “Frontiers of chaotic advection,” *Reviews of Modern Physics*, vol. 89, no. 2, pp. 1–66, 2017. DOI: `10.1103/RevModPhys.89.025007`.

[2] D. Drikakis, D. Kwak, and C. C. Kiris, “Computational aerodynamics: Advances and challenges,” *Aeronautical Journal*, vol. 120, no. 1223, pp. 13–36, 2016. DOI: `10.1017/aer.2015.2`.

[3] P. R. Spalart and V. Venkatakrishnan, “On the role and challenges of CFD in the aerospace industry,” *Aeronautical Journal*, vol. 120, no. 1223, pp. 209–232, 2016. DOI: `10.1017/aer.2015.10`.

[4] A. Viviani, A. Aprovitola, G. Pezzella, and C. Rainone, “CFD design capabilities for next generation high-speed aircraft,” *Acta Astronautica*, vol. 178, pp. 143–158, 2021. DOI: `10.1016/j.actaastro.2020.09.006`.

[5] S. Marras *et al.*, “A Review of Element-Based Galerkin Methods for Numerical Weather Prediction: Finite Elements, Spectral Elements, and Discontinuous Galerkin,” *Archives of Computational Methods in Engineering*, vol. 23, pp. 673–722, 2016. DOI: `10.1007/s11831-015-9152-1`.

[6] Z. J. Wang *et al.*, “High-order CFD methods: current status and perspective,” *International Journal for Numerical Methods in Fluids*, vol. 72, no. 8, pp. 811–845, 2013. DOI: `10.1002/fld.3767`.

[7] A. J. Baker, “On mathematics of physics of fluids maturation,” *Physics of Fluids*, vol. 33, no. 8, p. 081 301, 2021. DOI: `10.1063/5.0057306`.

[8] J. Slotnick *et al.*, “CFD Vision 2030 Study: A Path to Revolutionary Computational Aerosciences,” NASA Langley Research Center, NASA Contractor Report NASA/CR-2014-218178, 2014.

[9] P. L. Roe, “Computational fluid dynamics - retrospective and prospective,” *International Journal of Computational Fluid Dynamics*, vol. 19, no. 8, pp. 581–594, 2005. DOI: `10.1080/10618560600585315`.

[10] J. Kreeft, A. Palha, and M. Gerritsma, “Mimetic framework on curvilinear quadrilaterals of arbitrary order,” *ArXiv Preprint*, pp. 1–69, 2011. DOI: `10.48550/arXiv.1111.4304`.

[11] M. Gerritsma, “Edge Functions for Spectral Element Methods,” in *Spectral and High Order Methods for Partial Differential Equations*, ser. Lecture Notes in Computational Science and Engineering, J. S. Hesthaven and E. M. Ronquist, Eds., vol. 76, Berlin, Germany: Springer, 2011, pp. 199–207. DOI: `10.1007/978-3-642-15337-2_17`.

[12] M. Gerritsma, “An Introduction to a Compatible Spectral Discretization Method,” *Mechanics of Advanced Materials and Structures*, vol. 19, no. 1-3, pp. 48–67, 2012. DOI: `10.1080/15376494.2011.572237`.

[13] A. Palha, P. P. Rebelo, R. Hiemstra, J. Kreeft, and M. Gerritsma, “Physics-compatible discretization techniques on single and dual grids, with application to the Poisson equation of volume forms,” *Journal of Computational Physics*, vol. 257, no. Part B, pp. 1394–1422, 2014. DOI: `10.1016/j.jcp.2013.08.005`.

[14] Y. Zhang, “Mimetic Spectral Element Method and Extensions toward Higher Computational Efficiency,” Ph.D. Thesis, TU Delft, Delft, the Netherlands, 2022. DOI: `10.4233/uuid:a87e7be3-fb08-4706-9676-a6ae43f5aa0d`.

[15] B. Koren, R. Abgrall, P. Bochev, J. Frank, and B. Perot, “Physics-compatible numerical methods,” *Journal of Computational Physics*, vol. 257, no. Part B, p. 1039, 2014. DOI: `10.1016/j.jcp.2013.10.015`.

[16] R. L. Ricca and M. A. Berger, “Topological ideas and fluid mechanics,” *Physics Today*, vol. 49, no. 12, pp. 28–34, 1996. DOI: `10.1063/1.881574`.

[17] C. Mendoza and A. M. Mancho, “Hidden geometry of ocean flows,” *Physical Review Letters*, vol. 105, p. 038 501, 2010. DOI: `10.1103/PhysRevLett.105.038501`.

[18] I. Marusic and S. Broomhall, “Leonardo da Vinci and Fluid Mechanics,” *Annual Review of Fluid Mechanics*, vol. 53, pp. 1–25, 2021. DOI: `10.1146/annurev-fluid-022620-122816`.

[19] B. He and F. L. Teixeira, “On the degrees of freedom of lattice electrodynamics,” *Physics Letters, Section A: General, Atomic and Solid State Physics*, vol. 336, no. 1, pp. 1–7, 2005. DOI: `10.1016/j.physleta.2005.01.001`.

[20] M. Bouman, A. Palha, J. Kreeft, and M. Gerritsma, “A Conservative Spectral Element Method for Curvilinear Domains,” in *Spectral and High Order Methods for Partial Differential Equations*, ser. Lecture Notes in Computational Science and Engineering, J. S. Hesthaven and E. Rønquist, Eds., vol. 76, Berlin, Germany: Springer, 2010, pp. 111–119. DOI: `10.1007/978-3-642-15337-2_8`.

[21] M. Gerritsma, A. Palha, V. Jain, and Y. Zhang, “Mimetic Spectral Element Method for Anisotropic Diffusion,” in *Numerical Methods for PDEs*, ser. SEMA SIMAI Springer Series, D. A. Di Pietro, A. Ern, and L. Formaggia, Eds., vol. 15, Cham, Switzerland: Springer, 2018, pp. 31–74. DOI: `10.1007/978-3-319-94676-4_3`.

[22] M. J. Shashkov and S. Steinberg, “Solving Diffusion Equations with Rough Coefficients in Rough Grids,” *Journal of Computational Physics*, vol. 129, no. 2, pp. 383–405, 1996. DOI: `10.1006/jcph.1996.0257`.

[23] M. J. Shashkov, B. Swartz, and B. Wendroff, “Local Reconstruction of a Vector Field from Its Normal Components on the Faces of Grid Cells,” *Journal of Computational Physics*, vol. 139, no. 2, pp. 406–409, 1998. DOI: `10.1006/jcph.1997.5877`.

[24] S. Candelaresi, D. Pontin, and G. Hornig, “Mimetic Methods for Lagrangian Relaxation of Magnetic Fields,” *SIAM Journal on Scientific Computing*, vol. 36, no. 6, B952–B968, 2014. DOI: `10.1137/140967404`.

[25] P. F. Dubois, “Python: Batteries Included,” *Computing in Science and Engineering*, vol. 9, no. 3, pp. 7–9, 2007. DOI: `10.1109/MCSE.2007.51`.

[26] T. E. Oliphant, “Python for Scientific Computing,” *Computing in Science and Engineering*, vol. 9, no. 3, pp. 10–20, 2007. DOI: `10.1109/MCSE.2007.58`.

[27] K. J. Millman and M. Aivazis, “Python for Scientists and Engineers,” *Computing in Science and Engineering*, vol. 13, no. 2, pp. 9–12, 2011. DOI: `10.1109/MCSE.2011.36`.

[28] F. Pérez, B. E. Granger, and J. D. Hunter, “Python: An Ecosystem for Scientific Computing,” *Computing in Science and Engineering*, vol. 13, no. 2, pp. 13–21, 2011. DOI: `10.1109/MCSE.2010.119`.

[29] C. R. Harris *et al.*, “Array programming with NumPy,” *Nature*, vol. 585, pp. 357–362, 2020. DOI: `10.1038/s41586-020-2649-2`.

[30] P. Virtanen *et al.*, “SciPy 1.0: fundamental algorithms for scientific computing in Python,” *Nature Methods*, vol. 17, pp. 261–272, 2020. DOI: `10.1038/s41592-019-0686-2`.

[31] J. D. Hunter, “Matplotlib: A 2D Graphics Environment,” *Computing in Science and Engineering*, vol. 9, no. 3, pp. 90–95, 2007. DOI: `10.1109/MCSE.2007.55`.

[32] F. Cuvelier, C. Japhet, and G. Scarella, “An efficient way to assemble finite element matrices in vector languages,” *BIT Numerical Mathematics*, vol. 56, pp. 833–864, 2016. DOI: `10.1007/s10543-015-0587-4`.

[33] A. Bossavit, “On the geometry of electromagnetism (1): Euclidean space,” *Journal of the Japan Society of Applied Electromagnetics and Mechanics*, vol. 6, no. 1, pp. 17–28, 1998.

[34] A. Bossavit, “On the geometry of electromagnetism (2): Geometrical objects,” *Journal of the Japan Society of Applied Electromagnetics and Mechanics*, vol. 6, no. 2, pp. 114–123, 1998.

[35] A. Bossavit, “On the geometry of electromagnetism (3): Faraday’s law,” *Journal of the Japan Society of Applied Electromagnetics and Mechanics*, vol. 6, no. 3, pp. 233–240, 1998.

[36] A. Bossavit, “On the geometry of electromagnetism (4): Maxwell’s house,” *Journal of the Japan Society of Applied Electromagnetics and Mechanics*, vol. 6, no. 4, pp. 318–326, 1998.

[37] A. Bossavit, “Computational electromagnetism and geometry: Building a finite-dimensional ’Maxwell’s house’ (1): Network equations,” *Journal of the Japan Society of Applied Electromagnetics and Mechanics*, vol. 7, no. 2, pp. 150–159, 1999.

[38] A. Bossavit, “Computational electromagnetism and geometry: (2): Network constitutive laws,” *Journal of the Japan Society of Applied Electromagnetics and Mechanics*, vol. 7, no. 3, pp. 294–301, 1999.

[39] A. Bossavit, “Computational electromagnetism and geometry: (3): Convergence,” *Journal of the Japan Society of Applied Electromagnetics and Mechanics*, vol. 7, no. 4, pp. 401–408, 1999.

[40] A. Bossavit, “Computational electromagnetism and geometry: (4): From degrees of freedom to fields,” *Journal of the Japan Society of Applied Electromagnetics and Mechanics*, vol. 8, no. 1, pp. 102–109, 2000.

[41] A. Bossavit, “Computational electromagnetism and geometry: (5): The ‘Galerkin hodge’,” *Journal of the Japan Society of Applied Electromagnetics and Mechanics*, vol. 8, no. 2, pp. 203–209, 2000.

[42] M. Desbrun, E. Kanso, and Y. Tong, “Discrete Differential Forms for Computational Modeling,” in *Discrete Differential Geometry*, ser. Oberwolfach Seminars, A. I. Bobenko, J. M. Sullivan, P. Schröder, and G. M. Ziegler, Eds., vol. 38, Birkhäuser, 2008, pp. 287–324. DOI: `10.1007/978-3-7643-8621-4_16`.

[43] G. A. Deschamps, “Electromagnetics and Differential Forms,” *Proceedings of the IEEE*, vol. 69, no. 6, pp. 676–696, 1981. DOI: `10.1109/PROC.1981.12048`.

[44] S. Fumeron, B. Berche, and F. Moraes, “Improving student understanding of electrodynamics: The case for differential forms,” *American Journal of Physics*, vol. 88, no. 12, pp. 1083–1093, 2020. DOI: `10.1119/10.0001754`.

[45] C. Mattiussi, “A Reference Discretization Strategy for the Numerical Solution of Physical Field Problems,” *Advances in Imaging and Electron Physics*, vol. 121, no. C, pp. 143–279, 2002. DOI: `10.1016/S1076-5670(02)80027-1`.

[46] J. B. Perot, “Discrete Conservation Properties of Unstructured Mesh Schemes,” *Annual Review of Fluid Mechanics*, vol. 43, pp. 299–318, 2011. DOI: `10.1146/annurev-fluid-122109-160645`.

[47] J. B. Perot and C. J. Zusi, “Differential forms for scientists and engineers,” *Journal of Computational Physics*, vol. 257, no. Part B, pp. 1373–1393, 2014. DOI: `10.1016/j.jcp.2013.08.007`.

[48] E. Tonti, *The Mathematical Structure of Classical and Relativistic Physics*, ser. Modeling and Simulation in Science, Engineering and Technology. New York, NY, USA: Birkhäuser, 2013. DOI: `10.1007/978-1-4614-7422-7`.

[49] E. Tonti, “Why starting from differential equations for computational physics?” *Journal of Computational Physics*, vol. 257, no. Part B, pp. 1260–1290, 2014. DOI: `10.1016/j.jcp.2013.08.016`.

[50] K. F. Warnick, R. H. Selfridge, and D. V. Arnold, “Teaching Electromagnetic Field Theory Using Differential Forms,” *IEEE Transactions on Education*, vol. 40, no. 1, pp. 53–68, 1997. DOI: `10.1109/13.554670`.

[51] K. F. Warnick and P. Russer, “Differential Forms and Electromagnetic Field Theory,” *Progress in Electromagnetics Research*, vol. 148, pp. 83–112, 2014. DOI: `10.2528/PIER14063009`.

[52] R. Hiptmair, “Finite elements in computational electromagnetism,” *Acta Numerica*, vol. 11, pp. 237–339, 2002. DOI: `10.1017/S0962492902000041`.

[53] V. I. Arnold and B. A. Khesin, “Topological Methods in Hydrodynamics,” *Annual Review of Fluid Mechanics*, vol. 24, pp. 146–166, 1992. DOI: `10.1146/annurev.fl.24.010192.001045`.

[54] V. I. Arnold and B. A. Khesin, *Topological Methods In Hydrodynamics*, ser. Applied Mathematical Sciences. New York, NY, USA: Springer-Verlag, 1998, vol. 125. DOI: `10.1007/b97593`.

[55] L. Beirão Da Veiga, F. Brezzi, A. Cangiani, G. Manzini, L. D. Marini, and A. Russo, "Basic principles of virtual element methods," *Mathematical Models and Methods in Applied Sciences*, vol. 23, no. 1, pp. 199–214, 2013. DOI: `10.1142/S0218202512500492`.

[56] K. Lipnikov, G. Manzini, and M. Shashkov, "Mimetic finite difference method," *Journal of Computational Physics*, vol. 257, no. Part B, pp. 1163–1227, 2014. DOI: `10.1016/j.jcp.2013.07.031`.

[57] D. N. Arnold, R. S. Falk, and R. Winther, "Finite element exterior calculus, homological techniques, and applications," *Acta Numerica*, vol. 15, pp. 1–155, 2006. DOI: `10.1017/S0962492906210018`.

[58] D. N. Arnold, R. S. Falk, and R. Winther, "Finite element exterior calculus: From Hodge theory to numerical stability," *Bulletin of the American Mathematical Society*, vol. 47, no. 2, pp. 281–354, 2010. DOI: `10.1090/S0273-0979-10-01278-4`.

[59] E. Tonti, "A direct discrete formulation of field laws: The cell method," *CMES - Computer Modeling in Engineering and Sciences*, vol. 2, no. 2, pp. 237–258, 2001. DOI: `10.3970/cmes.2001.002.237`.

[60] E. Ferretti, "The cell method: An overview on the main features," *Curved and Layered Structures*, vol. 2, no. 1, pp. 194–243, 2015. DOI: `10.1515/cls-2015-0011`.

[61] A. N. Hirani, "Discrete Exterior Calculus," Ph.D. Thesis, California Institute of Technology, Pasadena, CA, USA, 2003. DOI: `10.7907/ZHY8-V329`.

[62] M. Desbrun, A. N. Hirani, M. Leok, and J. E. Marsden, "Discrete Exterior Calculus," *ArXiv Preprint*, pp. 1–53, 2005. [Online]. Available: `http://arxiv.org/abs/math/0508341`.

[63] J. Stewart, *Calculus: Early Transcendentals*, 8th International Metric ed. USA: Cengage Learning, 2016, ISBN: 978-1-305-27237-8.

[64] N. Schleifer, "Differential forms as a basis for vector analysis—with applications to electrodynamics," *American Journal of Physics*, vol. 51, no. 12, pp. 1139–1145, 1983. DOI: `10.1119/1.13325`.

[65] E. Tonti, "The reason for analogies between physical theories," *Applied Mathematical Modelling*, vol. 1, no. 1, pp. 37–50, 1976. DOI: `10.1016/0307-904X(76)90023-8`.

[66] M. Gerritsma, R. Hiemstra, J. Kreeft, A. Palha, P. Rebelo, and D. Toshniwal, "The geometric basis of numerical methods," in *Spectral and High Order Methods for Partial Differential Equations-ICOSAHOM*, ser. Lecture Notes in Computational Science and Engineering, vol. 95, Cham, Switzerland: Springer, 2014, pp. 17–35. DOI: `10.1007/978-3-319-01601-6_2`.

[67] A. Bossavit, "How weak is the "weak solution" in finite element methods?" *IEEE Transactions on Magnetics*, vol. 34, no. 5, pp. 2429–2432, 1998. DOI: `10.1109/20.717558`.

[68] P. B. Bochev, "A discourse on variational and geometric aspects of stability of discretizations," in *Proceedings of the 33th Computational Fluid Dynamics Lecture Series*, Rhode Saint Genese, Belgium: Von Karman Institute for Fluid Dynamics, 2003, pp. 1–85.

[69] P. B. Bochev and J. M. Hyman, "Principles of Mimetic Discretizations of Differential Operators," in *Compatible Spatial Discretizations*, ser. The IMA Volumes in Mathematics and its Applications, vol. 142, New York, NY, USA: Springer, 2006, pp. 89–119. DOI: `10.1007/0-387-38034-5_5`.

[70] S. H. Christiansen, H. Z. Munthe-Kaas, and B. Owren, "Topics in structure-preserving discretization," *Acta Numerica*, vol. 20, pp. 1–119, 2011. DOI: `10.1017/S096249291100002X`.

[71] C. J. Budd and M. D. Piggott, "Geometric Integration and its Applications," in *Handbook of Numerical Analysis*, ser. Handbook of Numerical Analysis, vol. 11, Amsterdam, the Netherlands: Elsevier, 2003, pp. 35–139. DOI: `10.1016/S1570-8659(02)11002-7`.

[72] H. Flanders, *Differential Forms: with Applications to the Physical Sciences*. New York, NY, USA: Dover Publications, Inc., 1989, Reprint of the Academic Press, Inc., New York, NY, USA, 1963 edition, ISBN: 9780486661698.

[73] T. Frankel, *The Geometry of Physics: An Introduction*, 3rd ed. Cambridge, U.K.: Cambridge University Press, 2012. DOI: 10.1017/CBO9781139061377.

[74] C. W. Misner, K. S. Thorne, and J. A. Wheeler, *Gravitation*. San Francisco, CA, USA: W. H. Freeman and Company, 1973, ISBN: 0-7167-0334-3.

[75] D. Bachman, *A Geometric Approach to Differential Forms*, 2nd ed. Boston, MA, USA: Birkhäuser, 2012. DOI: 10.1007/978-0-8176-8304-7.

[76] R. Hiptmair, "Discrete Hodge operators," *Numerische Mathematik*, vol. 90, no. 2, pp. 265–289, 2001. DOI: 10.1007/s002110100295.

[77] W. L. Burke, *Div, Grad, Curl are Dead*, 1995. [Online]. Available: https://people.ucsc.edu/$%5Csim$rmont/papers/Burke_DivGradCurl.pdf.

[78] R. Hiptmair, "Canonical construction of finite elements," *Mathematics of Computation*, vol. 68, no. 228, pp. 1325–1346, 1999. DOI: 10.1090/s0025-5718-99-01166-7.

[79] S. Axler, *Linear algebra done right*, 3rd ed. Cham, Switzerland: Springer, 2015. DOI: 10.1007/978-3-319-11080-6.

[80] C. Mattiussi, "An Analysis of Finite Volume, Finite Element, and Finite Difference Methods Using Some Concepts from Algebraic Topology," *Journal of Computational Physics*, vol. 133, no. 2, pp. 289–309, 1997. DOI: 10.1006/jcph.1997.5656.

[81] T. Tarhasaari, L. Kettunen, and A. Bossavit, "Some realizations of a discrete Hodge operator: A reinterpretation of finite element techniques," *IEEE Transactions on Magnetics*, vol. 35, no. 3, pp. 1494–1497, 1999. DOI: 10.1109/20.767250.

[82] R. Hiptmair, "Discrete Hodge-operators: An algebraic perspective," *Progress in Electromagnetics Research*, vol. 32, pp. 247–269, 2001. DOI: 10.2528/PIER00080110.

[83] J. B. Perot and V. Subramanian, "A discrete calculus analysis of the Keller Box scheme and a generalization of the method to arbitrary meshes," *Journal of Computational Physics*, vol. 226, no. 1, pp. 494–508, 2007. DOI: 10.1016/j.jcp.2007.04.015.

[84] A. Angoshtari and A. Yavari, "Differential Complexes in Continuum Mechanics," *Archive for Rational Mechanics and Analysis*, vol. 216, no. 1, pp. 193–220, 2015. DOI: 10.1007/s00205-014-0806-1.

[85] D. N. Arnold and K. Hu, "Complexes from Complexes," *Foundations of Computational Mathematics*, vol. 21, pp. 1739–1774, 2021. DOI: 10.1007/s10208-021-09498-9.

[86] 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*, vol. 95, pp. 101–142, 2021. DOI: 10.1016/j.camwa.2020.09.022.

[87] F. De Goes, A. Butts, and M. Desbrun, "Discrete Differential Operators on Polygonal Meshes," *ACM Transactions on Graphics*, vol. 39, no. 4, pp. 1–14, 2020. DOI: 10.1145/3386569.3392389.

[88] D. A. Di Pietro and J. Droniou, "An Arbitrary-Order Discrete de Rham Complex on Polyhedral Meshes: Exactness, Poincaré Inequalities, and Consistency," *Foundations of Computational Mathematics*, 2021. DOI: 10.1007/s10208-021-09542-8.

[89] J. B. Perot and C. Chartrand, "A mimetic method for polygons," *Journal of Computational Physics*, vol. 424, p. 109 853, 2021. DOI: 10.1016/j.jcp.2020.109853.

[90] W. J. Gordon and C. A. Hall, "Transfinite element methods: Blending-function interpolation over arbitrary curved element domains," *Numerische Mathematik*, vol. 21, pp. 109–129, 1973. DOI: 10.1007/BF01436298.

[91] W. J. Gordon and C. A. Hall, "Construction of curvilinear co-ordinate systems and applications to mesh generation," *International Journal for Numerical Methods in Engineering*, vol. 7, no. 4, pp. 461–477, 1973. DOI: 10.1002/nme.1620070405.

[92] D. Lee, A. Palha, and M. Gerritsma, "Discrete conservation properties for shallow water flows using mixed mimetic spectral elements," *Journal of Computational Physics*, vol. 357, pp. 282–304, 2018. DOI: 10.1016/j.jcp.2017.12.022.

[93] A. Bossavit, “Extrusion, contraction: Their discretization via Whitney forms,” *COMPEL - The International Journal for Computation and Mathematics in Electrical and Electronic Engineering*, vol. 22, no. 3, pp. 470–480, 2003. DOI: `10.1108/03321640310474877`.

[94] P. Mullen *et al.*, “Discrete Lie Advection of Differential Forms,” *Foundations of Computational Mathematics*, vol. 11, no. 2, pp. 131–149, 2011. DOI: `10.1007/s10208-010-9076-y`.

[95] J. M. Lee, *Introduction to Smooth Manifolds*, 2nd ed., ser. Graduate Texts in Mathematics. New York, NY, USA: Springer, 2012, vol. 218. DOI: `10.1007/978-1-4419-9982-5`.

[96] G. Coppola, F. Capuano, and L. De Luca, “Discrete Energy-Conservation Properties in the Numerical Simulation of the Navier-Stokes Equations,” *Applied Mechanics Reviews*, vol. 71, no. 1, 2019. DOI: `10.1115/1.4042820`.

[97] B. F. Schutz, *Geometrical Methods of Mathematical Physics*. Cambridge, U.K.: Cambridge University Press, 1980. DOI: `10.1017/CBO9781139171540`.

[98] B. Liu, Y. Tong, F. Degoes, and M. Desbrun, “Discrete connection and covariant derivative for vector field analysis and design,” *ACM Transactions on Graphics*, vol. 35, no. 3, pp. 1–17, 2016. DOI: `10.1145/2870629`.

[99] Y. Du and J. A. Ekaterinaris, “Time-marching schemes for spatially high order accurate discretizations of the Euler and Navier–Stokes equations,” *Progress in Aerospace Sciences*, vol. 130, p. 100 795, 2022. DOI: `10.1016/j.paerosci.2021.100795`.

[100] B. He and F. L. Teixeira, “Geometric finite element discretization of Maxwell equations in primal and dual spaces,” *Physics Letters, Section A: General, Atomic and Solid State Physics*, vol. 349, no. 1-4, pp. 1–14, 2006. DOI: `10.1016/j.physleta.2005.09.002`.

[101] B. Sanderse, “Energy-conserving Runge-Kutta methods for the incompressible navier-stokes equations,” *Journal of Computational Physics*, vol. 233, pp. 100–131, 2013. DOI: `10.1016/j.jcp.2012.07.039`.

[102] D. Razafindralandy, A. Hamdouni, and M. Chhay, “A review of some geometric integrators,” *Advanced Modeling and Simulation in Engineering Sciences*, vol. 5, no. 16, 2018. DOI: `10.1186/s40323-018-0110-y`.

[103] E. Celledoni, H. Marthinsen, and B. Owren, “An introduction to Lie group integrators - basics, new developments and applications,” *Journal of Computational Physics*, vol. 257, no. Part B, pp. 1040–1061, 2014. DOI: `10.1016/j.jcp.2012.12.031`.

[104] D. Pavlov, P. Mullen, Y. Tong, E. Kanso, J. E. Marsden, and M. Desbrun, “Structure-preserving discretization of incompressible fluids,” *Physica D: Nonlinear Phenomena*, vol. 240, no. 6, pp. 443–458, 2011. DOI: `10.1016/j.physd.2010.10.012`.

[105] E. S. Gawlik, P. Mullen, D. Pavlov, J. E. Marsden, and M. Desbrun, “Geometric, variational discretization of continuum theories,” *Physica D: Nonlinear Phenomena*, vol. 240, no. 21, pp. 1724–1760, 2011. DOI: `10.1016/j.physd.2011.07.011`.

[106] T. Vazquez-Gonzalez, A. Llor, and C. Fochesato, “A novel GEEC (Geometry, Energy, and Entropy Compatible) procedure applied to a staggered direct-ALE scheme for hydrodynamics,” *European Journal of Mechanics, B/Fluids*, vol. 65, pp. 494–514, 2017. DOI: `10.1016/j.euromechflu.2017.05.003`.

[107] P. L. Roe, “Designing CFD methods for bandwidth—A physical approach,” *Computers and Fluids*, vol. 214, p. 104 774, 2021. DOI: `10.1016/j.compfluid.2020.104774`.

[108] C. Mattiussi, “The geometry of Time-Stepping,” *Progress in Electromagnetics Research*, vol. 32, pp. 123–149, 2001. DOI: `10.2528/PIER00080105`.

[109] D. Lee and A. Palha, “Exact spatial and temporal balance of energy exchanges within a horizontally explicit/vertically implicit non-hydrostatic atmosphere,” *Journal of Computational Physics*, vol. 440, p. 110 432, 2021. DOI: `10.1016/j.jcp.2021.110432`.

[110] C. Deimert, M. E. Potter, and M. Okoniewski, “Collocated electrodynamic FDTD schemes using overlapping Yee grids and higher-order Hodge duals,” *Journal of Computational Physics*, vol. 326, pp. 629–649, 2016. DOI: `10.1016/j.jcp.2016.08.048`.

[111] L. D. Landau and E. M. Lifshitz, *Fluid Mechanics*, 2nd ed. reprinted with corrections, ser. Course of Theoretical Physics. Oxford, U.K.: Butterworth-Heinemann, 2006, vol. 6. DOI: `10.1016/C2013-0-03799-1`.

[112] J. D. Anderson Jr., *Fundamentals of Aerodynamics*, 6th ed., ser. McGraw-Hill series in aeronautical and aerospace engineering. New York, NY, USA: McGraw-Hill Education, 2017, ISBN: 978-1-259-12991-9.

[113] R. Haberman, *Applied Partial Differential Equations: with Fourier Series and Boundary Value Problems*, Fifth Edition, ser. Pearson Modern Classics for Advanced Mathematics Series. Harlow, U.K.: Pearson Education, 2014, ISBN: 978-1-292-03985-5.

[114] G. Coppola, F. Capuano, S. Pirozzoli, and L. de Luca, "Numerically stable formulations of convective terms for turbulent compressible flows," *Journal of Computational Physics*, vol. 382, pp. 86–104, 2019. DOI: `10.1016/j.jcp.2019.01.007`.

[115] Y. Morinishi and K. Koga, "Skew-symmetric convection form and secondary conservative finite difference methods for moving grids," *Journal of Computational Physics*, vol. 257, no. Part B, pp. 1081–1112, 2014. DOI: `10.1016/j.jcp.2013.01.040`.

[116] A. E. Veldman, "A general condition for kinetic-energy preserving discretization of flow transport equations," *Journal of Computational Physics*, vol. 398, p. 108 894, 2019. DOI: `10.1016/j.jcp.2019.108894`.

[117] A. Bennett, *Lagrangian Fluid Dynamics*, ser. Cambridge Monographs on Mechanics. Cambridge, U.K.: Cambridge University Press, 2006. DOI: `10.1017/CBO9780511734939`.

[118] J. F. Price, *Lagrangian and Eulerian Representations of Fluid Flow: Kinematics and the Equations of Motion*, last accessed: 25/10/2021, 2006. [Online]. Available: `https://www.whoi.edu/science/PO/people/jprice/website/cv.html`.

[119] E. van Sebille *et al.*, "Lagrangian ocean analysis: Fundamentals and practices," *Ocean Modelling*, vol. 121, pp. 49–75, 2018. DOI: `10.1016/j.ocemod.2017.11.008`.

[120] C. J. Cotter, D. D. Holm, and P. E. Hydon, "Multisymplectic formulation of fluid dynamics using the inverse map," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 463, no. 2086, pp. 2671–2687, 2007. DOI: `10.1098/rspa.2007.1892`.

[121] C. J. Cotter and J. Thuburn, "A finite element exterior calculus framework for the rotating shallow-water equations," *Journal of Computational Physics*, vol. 257, no. Part B, pp. 1506–1526, 2014. DOI: `10.1016/j.jcp.2013.10.008`.

[122] J. N. Reddy, *Principles of Continuum Mechanics*, 2nd ed. Cambridge, U.K.: Cambridge University Press, 2018, ISBN: 9781107199200.

[123] H. Flanders, "Differentiation Under the Integral Sign," *The American Mathematical Monthly*, vol. 80, no. 6, pp. 615–627, 1973. DOI: `10.2307/2319163`.

[124] D. H. Wagner, "Equivalence of the Euler and Lagrangian Equations of Gas Dynamics for Weak Solutions," *Journal of Differential Equations*, vol. 68, no. 1, pp. 118–136, 1987. DOI: `10.1016/0022-0396(87)90188-4`.

[125] R. J. LeVeque, *Numerical Methods for Conservation Laws*, Second Edition, ser. Lectures in Mathematics ETH Zürich. Basel, Switzerland: Birkhäuser, 1992. DOI: `10.1007/978-3-0348-8629-1`.

[126] J. D. Anderson Jr., *Hypersonic and High-Temperature Gas Dynamics*, 2nd ed., ser. AIAA Education Series. Reston, VA, USA: American Institute of Aeronautics and Astronautics, 2006. DOI: `10.2514/4.861956`.

[127] J. Porter, P. Salgado Sánchez, V. Shevtsova, and V. Yasnou, "A review of fluid instabilities and control strategies with applications in microgravity," *Mathematical Modelling of Natural Phenomena*, vol. 16, no. 24, 2021. DOI: `10.1051/mmnp/2021020`.

[128] M. Schöck and E. J. Spillar, "Method for a quantitative investigation of the frozen flow hypothesis," *Journal of the Optical Society of America A*, vol. 17, no. 9, pp. 1650–1658, 2000. DOI: `10.1364/JOSAA.17.001650`.

[129] B. Després, "Lagrangian systems of conservation laws," *Numerische Mathematik*, vol. 89, pp. 99–134, 2001. DOI: `10.1007/PL00005465`.

[130] E. Hopf, "The partial differential equation $u_t + uu_x = \mu_{xx}$," *Communications on Pure and Applied Mathematics*, vol. 3, no. 3, pp. 201–230, 1950. DOI: `10.1002/cpa.3160030302`.

[131] J. C. Bowman, M. A. Yassaei, and A. Basu, "A Fully Lagrangian Advection Scheme," *Journal of Scientific Computing*, vol. 64, no. 1, pp. 151–177, 2015. DOI: `10.1007/s10915-014-9928-8`.

[132] W. L. Burke, *Applied Differential Geometry*. Cambridge, U.K.: Cambridge University Press, 1985, ISBN: 9780521269292.

[133] F. Rapetti and A. Bossavit, "Whitney Forms of Higher Degree," *SIAM Journal on Numerical Analysis*, vol. 47, no. 3, pp. 2369–2386, 2009. DOI: `10.1137/070705489`.

[134] J. Lohi and L. Kettunen, "Whitney forms and their extensions," *Journal of Computational and Applied Mathematics*, vol. 393, p. 113 520, 2021. DOI: `10.1016/j.cam.2021.113520`.

[135] F. Auricchio, L. Beirão da Veiga, F. Brezzi, and C. Lovadina, "Mixed Finite Element Methods," in *Encyclopedia of Computational Mechanics*, E. Stein, R. de Borst, and T. J. R. Hughes, Eds., 2nd ed., Hoboken, NJ, USA: John Wiley & Sons, 2017, pp. 237–276. DOI: `10.1002/9781119176817.ecm2004`.

[136] A. Bossavit, "Differential forms and the computation of fields and forces in electromagnetism," *European journal of mechanics. B, Fluids*, vol. 10, no. 5, pp. 474–488, 1991.

[137] A. Bossavit, *Computational Electromagnetism: Variational Formulations, Complementarity, Edge Elements*, ser. Electromagnetism. London, U.K.: Academic Press, 1998, vol. 2. DOI: `10.1016/B978-0-12-118710-1.X5000-4`.

[138] M. J. Shashkov and S. Steinberg, "Support-Operator Finite-Difference Algorithms for General Elliptic Problems," *Journal of Computational Physics*, vol. 118, no. 1, pp. 131–151, 1995. DOI: `10.1006/jcph.1995.1085`.

[139] J. Hyman, M. Shashkov, and S. Steinberg, "The Numerical Solution of Diffusion Problems in Strongly Heterogeneous Non-isotropic Materials," *Journal of Computational Physics*, vol. 132, no. 1, pp. 130–148, 1997. DOI: `10.1006/jcph.1996.5633`.

[140] J. M. Hyman and M. J. Shashkov, "Natural Discretizations for the Divergence, Gradient, and Curl on Logically Rectangular Grids," *Computers and Mathematics with Applications*, vol. 33, no. 4, pp. 81–104, 1997. DOI: `10.1016/S0898-1221(97)00009-6`.

[141] J. M. Hyman and M. J. Shashkov, "Adjoint operators for the natural discretizations of the divergence, gradient and curl on logically rectangular grids," *Applied Numerical Mathematics*, vol. 25, no. 4, pp. 413–442, 1997. DOI: `10.1016/S0168-9274(97)00097-4`.

[142] J. M. Hyman and M. J. Shashkov, "Mimetic Discretizations for Maxwell's Equations," *Journal of Computational Physics*, vol. 151, no. 2, pp. 881–909, 1999. DOI: `10.1006/jcph.1999.6225`.

[143] J. M. Hyman and M. J. Shashkov, "The Orthogonal Decomposition Theorems for Mimetic Finite Difference Methods," *SIAM Journal on Numerical Analysis*, vol. 36, no. 3, pp. 788–818, 1999. DOI: `10.1137/S0036142996314044`.

[144] J. M. Hyman and M. Shashkov, "Mimetic finite difference methods for Maxwell's equations and the equations of Magnetic Diffusion," *Progress in Electromagnetics Research*, vol. 32, pp. 89–121, 2001. DOI: `10.2528/PIER00080104`.

[145] V. Subramanian and J. B. Perot, "Higher-order mimetic methods for unstructured meshes," *Journal of Computational Physics*, vol. 219, no. 1, pp. 68–85, 2006. DOI: `10.1016/j.jcp.2006.03.028`.

[146] J. B. Perot, "Conservation Properties of Unstructured Staggered Mesh Schemes," *Journal of Computational Physics*, vol. 159, no. 1, pp. 58–89, 2000. DOI: `10.1006/jcph.2000.6424`.

[147] S. Elcott, Y. Tong, E. Kanso, P. Schröder, and M. Desbrun, "Stable, Circulation-Preserving, Simplicial Fluids," *ACM Transactions on Graphics*, vol. 26, no. 1, pp. 1–12, 2007. DOI: `10.1145/1186644.1186648`.

[148] H. Heumann and R. Hiptmair, “Extrusion contraction upwind schemes for convection-diffusion problems,” Seminar für Angewandte Mathematik, ETH, Zürich, Switzerland, Technical Report, 2008. [Online]. Available: `https://www.sam.math.ethz.ch/sam_reports/reports_final/reports2008/2008-30`.

[149] H. Heumann and R. Hiptmair, “Eulerian and semi-lagrangian methods for convection-diffusion for differential forms,” *Discrete and Continuous Dynamical Systems*, vol. 29, no. 4, pp. 1471–1495, 2011. DOI: `10.3934/dcds.2011.29.1471`.

[150] A. Palha, J. Kreeft, and M. Gerritsma, “Numerical Solution of Advection Equations with the Discretization of the Lie Derivative,” in *Proceedings of the V European Conference on Computational Fluid Dynamics ECOMAS CFD 2010*, J. C. F. Pereira, A. Sequeria, and J. M. C. Pereira, Eds., Lisbon, Portugal, 2010, pp. 1–23.

[151] A. Palha, P. P. Rebelo, and M. Gerritsma, “Mimetic Spectral Element Advection,” in *Spectral and High Order Methods for Partial Differential Equations-ICOSAHOM 2012*, ser. Lecture Notes in Computational Science and Engineering, M. Azaïez, H. El Fekih, and J. Hesthaven, Eds., vol. 95, Cham, Switzerland: Springer, 2014, pp. 325–335. DOI: `10.1007/978-3-319-01601-6_26`.

[152] M. Gerritsma, J. Kunnen, and B. de Heij, “Discrete Lie Derivative,” in *Numerical Mathematics and Advanced Applications ENUMATH 2015*, ser. Lecture Notes in Computational Science and Engineering, B. Karasözen, M. Manguoğlu, M. Tezer-Sezgin, S. Göktepe, and Ö. Uğur, Eds., vol. 112, Cham, Switzerland: Springer, 2016, pp. 635–643. DOI: `10.1007/978-3-319-39929-4_61`.

[153] M. S. Mohamed, A. N. Hirani, and R. Samtaney, “Discrete exterior calculus discretization of incompressible Navier-Stokes equations over surface simplicial meshes,” *Journal of Computational Physics*, vol. 312, pp. 175–191, 2016. DOI: `10.1016/j.jcp.2016.02.028`.

[154] H. Heumann, R. Hiptmair, K. Li, and J. Xu, “Fully discrete semi-Lagrangian methods for advection of differential forms,” *BIT Numerical Mathematics*, vol. 52, no. 4, pp. 981–1007, 2012. DOI: `10.1007/s10543-012-0382-4`.

[155] H. Heumann and R. Hiptmair, “Stabilized galerkin methods for magnetic advection,” *Mathematical Modelling and Numerical Analysis*, vol. 47, no. 6, pp. 1713–1732, 2013. DOI: `10.1051/m2an/2013085`.

[156] H. Heumann and R. Hiptmair, “Convergence of Lowest Order Semi-Lagrangian Schemes,” *Foundations of Computational Mathematics*, vol. 13, pp. 187–220, 2013. DOI: `10.1007/s10208-012-9139-3`.

[157] H. Heumann, R. Hiptmair, and C. Pagliantini, “Stabilized galerkin for transient advection of differential forms,” *Discrete and Continuous Dynamical Systems - Series S*, vol. 9, no. 1, pp. 185–214, 2016. DOI: `10.3934/dcdss.2016.9.185`.

[158] J. Thuburn and C. J. Cotter, “A Framework for Mimetic Discretization of the Rotating Shallow-Water Equations on Arbitrary Polygonal Grids,” *SIAM Journal on Scientific Computing*, vol. 34, no. 3, B203–B225, 2012. DOI: `10.1137/110850293`.

[159] J. Thuburn, C. J. Cotter, and T. Dubos, “A mimetic, semi-implicit, forward-in-time, finite volume shallow water model: Comparison of hexagonal-icosahedral and cubed-sphere grids,” *Geoscientific Model Development*, vol. 7, no. 3, pp. 909–929, 2014. DOI: `10.5194/gmd-7-909-2014`.

[160] J. Thuburn and C. J. Cotter, “A primal-dual mimetic finite element scheme for the rotating shallow water equations on polygonal spherical meshes,” *Journal of Computational Physics*, vol. 290, pp. 274–297, 2015. DOI: `10.1016/j.jcp.2015.02.045`.

[161] 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*, vol. 328, pp. 200–220, 2017. DOI: `10.1016/j.jcp.2016.10.009`.

[162] D. Lee and A. Palha, “A mixed mimetic spectral element model of the rotating shallow water equations on the cubed sphere,” *Journal of Computational Physics*, vol. 375, pp. 240–262, 2018. DOI: `10.1016/j.jcp.2018.08.042`.

[163] D. Lee and A. Palha, “A mixed mimetic spectral element model of the 3D compressible Euler equations on the cubed sphere,” *Journal of Computational Physics*, vol. 401, p. 108 993, 2020. DOI: `10.1016/j.jcp.2019.108993`.

[164] Y. Zhang, A. Palha, M. Gerritsma, and L. G. Rebholz, “A mass-, kinetic energy- and helicity-conserving mimetic dual-field discretization for three-dimensional incompressible Navier-Stokes equations, part I: Periodic domains,” *Journal of Computational Physics*, vol. 451, p. 110 868, 2022. DOI: `10.1016/j.jcp.2021.110868`.

[165] B. Da Veiga, J. Droniou, and G. Manzini, “A unified approach for handling convection terms in finite volumes and mimetic discretization methods for elliptic problems,” *IMA Journal of Numerical Analysis*, vol. 31, no. 4, pp. 1357–1401, 2011. DOI: `10.1093/imanum/drq018`.

[166] T. M. Bendall, C. J. Cotter, and J. Shipton, “The ‘recovered space’ advection scheme for lowest-order compatible finite element methods,” *Journal of Computational Physics*, vol. 390, pp. 342–358, 2019. DOI: `10.1016/j.jcp.2019.04.013`.

[167] E. J. Caramana, D. E. Burton, M. J. Shashkov, and P. P. Whalen, “The Construction of Compatible Hydrodynamics Algorithms Utilizing Conservation of Total Energy,” *Journal of Computational Physics*, vol. 146, no. 1, pp. 227–262, 1998. DOI: `10.1006/jcph.1998.6029`.

[168] E. J. Caramana, C. L. Rousculp, and D. E. Burton, “A Compatible, Energy and Symmetry Preserving Lagrangian Hydrodynamics Algorithm in Three-Dimensional Cartesian Geometry,” *Journal of Computational Physics*, vol. 157, no. 1, pp. 89–119, 2000. DOI: `10.1006/jcph.1999.6368`.

[169] A. L. Bauer, D. E. Burton, E. J. Caramana, R. Loubère, M. J. Shashkov, and P. P. Whalen, “The internal consistency, stability, and accuracy of the discrete, compatible formulation of Lagrangian hydrodynamics,” *Journal of Computational Physics*, vol. 218, no. 2, pp. 572–593, 2006. DOI: `10.1016/j.jcp.2006.02.024`.

[170] R. Loubère and E. J. Caramana, “The force/work differencing of exceptional points in the discrete, compatible formulation of Lagrangian hydrodynamics,” *Journal of Computational Physics*, vol. 216, no. 1, pp. 1–18, 2006. DOI: `10.1016/j.jcp.2005.11.022`.

[171] J. A. Ekaterinaris, “High-order accurate, low numerical diffusion methods for aerodynamics,” *Progress in Aerospace Sciences*, vol. 41, no. 3-4, pp. 192–300, 2005. DOI: `10.1016/j.paerosci.2005.03.003`.

[172] S. Pirozzoli, “Numerical methods for high-speed flows,” *Annual Review of Fluid Mechanics*, vol. 43, pp. 163–194, 2011. DOI: `10.1146/annurev-fluid-122109-160718`.

[173] H. Xu, C. D. Cantwell, C. Monteserin, C. Eskilsson, A. P. Engsig-Karup, and S. J. Sherwin, “Spectral/hp element methods: Recent developments, applications, and perspectives,” *Journal of Hydrodynamics*, vol. 30, pp. 1–22, 1 2018. DOI: `10.1007/s42241-018-0001-1`.

[174] K. W. Morton and T. Sonar, “Finite volume methods for hyperbolic conservation laws,” *Acta Numerica*, vol. 16, pp. 155–238, 2007. DOI: `10.1017/S0962492906300013`.

[175] Y. Batugedara and K. J. Schwiebert, “Note on the effect of grad-div stabilization on calculating drag and lift coefficients,” *Applied Mathematics and Computation*, vol. 434, p. 127 434, 2022. DOI: `10.1016/j.amc.2022.127434`.

[176] J. V. Neumann and R. D. Richtmyer, “A Method for the Numerical Calculation of Hydrodynamic Shocks,” *Journal of Applied Physics*, vol. 21, no. 3, pp. 232–237, 1950. DOI: `10.1063/1.1699639`.

[177] A. E. Mattsson and W. J. Rider, “Artificial viscosity: back to the basics,” *International Journal for Numerical Methods in Fluids*, vol. 77, no. 7, pp. 400–417, 2015. DOI: `10.1002/fld.3981`.

[178] N. R. Morgan and B. J. Archer, “On the Origins of Lagrangian Hydrodynamic Methods,” *Nuclear Technology*, vol. 207, no. sup 1, S147–S175, 2021. DOI: `10.1080/00295450.2021.1913034`.

[179] L. G. Margolin and N. M. Lloyd-Ronning, “Artificial viscosity—then and now,” *Meccanica*, 2022. DOI: `10.1007/s11012-022-01541-5`.

[180] E. Johnsen *et al.*, “Assessment of high-resolution methods for numerical simulations of compressible turbulence with shock waves,” *Journal of Computational Physics*, vol. 229, no. 4, pp. 1213–1237, 2010. DOI: `10.1016/j.jcp.2009.10.028`.

[181] C.-W. Shu, “High Order Weighted Essentially Nonoscillatory Schemes for Convection Dominated Problems,” *SIAM Review*, vol. 51, no. 1, pp. 82–126, 2009. DOI: `10.1137/070679065`.

[182] C.-W. Shu, “Essentially non-oscillatory and weighted essentially non-oscillatory schemes,” *Acta Numerica*, vol. 29, pp. 701–762, 2020. DOI: `10.1017/S0962492920000057`.

[183] E. Johnsen, “On the treatment of contact discontinuities using WENO schemes,” *Journal of Computational Physics*, vol. 230, no. 24, pp. 8665–8668, 2011. DOI: `10.1016/j.jcp.2011.08.017`.

[184] D. J. Benson, “Computational methods in Lagrangian and Eulerian hydrocodes,” *Computer Methods in Applied Mechanics and Engineering*, vol. 99, no. 2-3, pp. 235–394, 1992. DOI: `10.1016/0045-7825(92)90042-I`.

[185] J. Donea, A. Huerta, and J. P. Ponthot, “Chapter 14: Arbitrary Lagrangian–Eulerian Methods,” in *Encyclopedia of Computational Mechanics; Part 1: Fundamentals*, John Wiley & Sons, 2004, pp. 413–437. DOI: `10.1002/0470091355.ecm009`.

[186] M. M. Francois, M. J. Shashkov, T. O. Masser, and E. D. Dendy, “A comparative study of multimaterial Lagrangian and Eulerian methods with pressure relaxation,” *Computers and Fluids*, vol. 83, pp. 126–136, 2013. DOI: `10.1016/j.compfluid.2012.06.011`.

[187] K. S. Yee, “Numerical Solution of Initial Boundary Value Problems Involving Maxwell’s Equations in Isotropic Media,” *IEEE Transactions on Antennas and Propagation*, vol. 14, no. 3, pp. 302–307, 1966. DOI: `10.1109/TAP.1966.1138693`.

[188] A. Bossavit and L. Kettunen, “Yee-like schemes on staggered cellular grids: A synthesis between FIT and FEM approaches,” *IEEE Transactions on Magnetics*, vol. 36, no. 4, pp. 861–867, 2000. DOI: `10.1109/20.877580`.

[189] A. Bossavit and L. Kettunen, “Correction to ”Yee-like schemes on staggered cellular grids: a synthesis between FIT and FEM approaches”,” *IEEE Transactions on Magnetics*, vol. 36, no. 6, p. 4050, 2000. DOI: `10.1109/TMAG.2000.914361`.

[190] S.-C. Chang and W.-M. To, “A New Numerical Framework for Solving Conservation Laws - The Method of Space-Time Conversavation Element and Solution Element,” NASA Lewis Research Center, Cleveland, OH, USA, NASA Technical Memorandum NASA/TM-104495, 1991. [Online]. Available: `https://ntrs.nasa.gov/citations/19910021553`.

[191] S.-C. Chang, “The Method of Space-Time Conservation Element and Solution Element — A New Approach for Solving the Navier-Stokes and Euler Equations,” *Journal of Computational Physics*, vol. 119, no. 2, pp. 295–324, 1995. DOI: `10.1006/jcph.1995.1137`.

[192] S.-C. Chang, X. Y. Wang, and C. Y. Chow, “The Space-Time Conservation Element and Solution Element Method: A New High-Resolution and Genuinely Multidimensional Paradigm for Solving Conservation Laws,” *Journal of Computational Physics*, vol. 156, no. 1, pp. 89–136, 1999. DOI: `10.1006/jcph.1999.6354`.

[193] Y. Jiang, C. Y. Wen, and D. Zhang, “Space–Time Conservation Element and Solution Element Method and Its Applications,” *AIAA Journal*, vol. 58, no. 12, pp. 5408–5430, 2020. DOI: `10.2514/1.J058928`.

[194] D. A. Kopriva, *Implementing Spectral Methods for Partial Differential Equations: Algorithms for Scientists and Engineers*, 1st ed. Dordrecht, the Netherlands: Springer, 2009. DOI: `10.1007/978-90-481-2261-5`.

[195] C. Zoppou and J. H. Knight, “Analytical Solutions for Advection and Advection-Diffusion Equations with Spatially Variable Coefficients,” *Journal of Hydraulic Engineering*, vol. 123, no. 2, pp. 144–148, 1997. DOI: `10.1061/(ASCE)0733-9429(1997)123:2(144)`.

[196] D. Corrigan, L. Fullard, and T. Lynch, “Advection Problems with Spatially Varying Velocity Fields: 1D and 2D Analytical and Numerical Solutions,” *Journal of Hydraulic Engineering*, vol. 146, no. 8, p. 04 020 053, 2020. DOI: `10.1061/(asce)hy.1943-7900.0001782`.

[197] S. L. Brunton and C. W. Rowley, “Fast computation of finite-time Lyapunov exponent fields for unsteady flows,” *Chaos*, vol. 20, no. 1, p. 017 503, 2010. DOI: `10.1063/1.3270044`.

[198] T. Peacock and G. Haller, “Lagrangian coherent structures: The hidden skeleton of fluid flows,” *Physics Today*, vol. 66, no. 2, pp. 41–47, 2013. DOI: `10.1063/PT.3.1886`.

[199] G. Haller, “Lagrangian coherent structures,” *Annual Review of Fluid Mechanics*, vol. 47, pp. 137–162, 2015. DOI: `10.1146/annurev-fluid-010313-141322`.

[200] K. Onu, F. Huhn, and G. Haller, “LCS Tool: A computational platform for Lagrangian coherent structures,” *Journal of Computational Science*, vol. 7, pp. 26–36, 2015. DOI: `10.1016/j.jocs.2014.12.002`.

[201] S. Leung, “An Eulerian approach for computing the finite time Lyapunov exponent,” *Journal of Computational Physics*, vol. 230, no. 9, pp. 3500–3524, 2011. DOI: `10.1016/j.jcp.2011.01.046`.

[202] S. Leung, “The backward phase flow method for the Eulerian finite time Lyapunov exponent computations,” *Chaos*, vol. 23, no. 4, p. 043 132, 2013. DOI: `10.1063/1.4847175`.

[203] G. You, T. Wong, and S. Leung, “Eulerian Methods for Visualizing Continuous Dynamical Systems using Lyapunov Exponents,” *SIAM Journal on Scientific Computing*, vol. 39, no. 2, pp. 415–437, 2017. DOI: `10.1137/16M1066890`.

[204] G. You and S. Leung, “Eulerian Based Interpolation Schemes for Flow Map Construction and Line Integral Computation with Applications to Lagrangian Coherent Structures Extraction,” *Journal of Scientific Computing*, vol. 74, pp. 70–96, 2018. DOI: `10.1007/s10915-017-0424-9`.

[205] C. S. Kulkarni and P. F. J. Lermusiaux, “Advection without compounding errors through flow map composition,” *Journal of Computational Physics*, vol. 398, p. 108 859, 2019. DOI: `10.1016/j.jcp.2019.108859`.

[206] J. Thuburn, “Multidimensional Flux-Limited Advection Schemes,” *Journal of Computational Physics*, vol. 123, no. 1, pp. 74–83, 1996. DOI: `10.1006/jcph.1996.0006`.

[207] S. Dhawan, S. Kapoor, S. Kumar, and S. Rawat, “Contemporary review of techniques for the solution of nonlinear Burgers equation,” *Journal of Computational Science*, vol. 3, no. 5, pp. 405–419, 2012. DOI: `10.1016/j.jocs.2012.06.003`.

[208] M. P. Bonkile, A. Awasthi, C. Lakshmi, V. Mukundan, and V. S. Aswin, “A systematic literature review of Burgers’ equation with recent advances,” *Pramana - Journal of Physics*, vol. 90, no. 69, 2018. DOI: `10.1007/s12043-018-1559-4`.

[209] J. D. Cole, “On a quasi-linear parabolic equation occurring in aerodynamics,” *Quarterly of Applied Mathematics*, vol. 9, no. 3, pp. 225–236, 1951. DOI: `10.1090/qam/42889`.

[210] E. R. Benton and G. W. Platzman, “A table of solutions of the one-dimensional Burgers equation,” *Quarterly of Applied Mathematics*, vol. 30, no. 2, pp. 195–212, 1972. DOI: `10.1090/qam/306736`.

[211] G. B. Whitham, *Linear and Nonlinear Waves*, ser. A Wiley-Intersience Publication. New York, NY, USA: Wiley, 1974. DOI: `10.1002/9781118032954`.

[212] J. P. Boyd, “The Energy Spectrum of Fronts: Time Evolution of Shocks in Burgers’ Equation,” *Journal of the Atmospheric Sciences*, vol. 49, no. 2, pp. 128–139, 1992. DOI: `10.1175/1520-0469(1992)049<0128:TESOFT>2.0.CO;2`.

[213] R. Courant and K. O. Friedrichs, *Supersonic Flow and Shock Waves*, ser. Applied Mathematical Sciences. New York, NY, USA: Springer, 1976, vol. 21, Unchanged reprint of the Interscience 1948 edition, ISBN: 978-0-387-90232-6.

[214] P. Chadwick, *Continuum Mechanics: Concise Theory and Problems*, 2nd ed. New York, NY, USA: Dover Publications, Inc., 1999, Corrected and enlarged reprint of the George Allen & Unwin Ltd., London, U.K., 1976 edition, ISBN: 978-0-486-40180-5.

[215] G. Astarita and G. Marrucci, *Principles of Non-Newtonian Fluid Mechanics*. Maidenhead, U.K.: McGraw-Hill, 1974, ISBN: 0-07-084022-9.

[216] R. M. Brach, “Rigid Body Collisions,” *Journal of Applied Mechanics*, vol. 56, no. 1, pp. 133–138, 1989. DOI: `10.1115/1.3176033`.

[217] T. de Neef and G. Moretti, “Shock fitting for everybody,” *Computers and Fluids*, vol. 8, no. 3, pp. 327–334, 1980. DOI: `10.1016/0045-7930(80)90022-5`.

[218] M. Onofri and R. Paciori, *Shock Fitting: Classical Techniques, Recent Developments, and Memoirs of Gino Moretti*, 1st ed., ser. Shock Wave and High Pressure Phenomena. Cham, Switzerland: Springer, 2017. DOI: `10.1007/978-3-319-68427-7`.

[219] R. Marchiano, F. Coulouvrat, and R. Grenon, “Numerical simulation of shock wave focusing at fold caustics, with application to sonic boom,” *The Journal of the Acoustical Society of America*, vol. 114, no. 4, pp. 1758–1771, 2003. DOI: `10.1121/1.1610459`.

[220] T. Auger and F. Coulouvrat, “Numerical simulation of sonic boom focusing,” *AIAA Journal*, vol. 40, no. 9, pp. 1726–1734, 2002. DOI: `10.2514/2.1877`.

[221] K. J. Plotkin, “State of the art of sonic boom modeling,” *The Journal of the Acoustical Society of America*, vol. 111, no. 1, pp. 530–536, 2002. DOI: `10.1121/1.1379075`.

[222] W. D. Hayes, R. C. Haefeli, and H. E. Kulsrud, “Sonic Boom Propagation in a Stratified Atmosphere, with Computer Program,” NASA Langley Research Center, Hampton, VA, USA, NASA Contractor Report NASA-CR-1299, 1969. [Online]. Available: `https://ntrs.nasa.gov/citations/19690013184`.

[223] W. D. Hayes and H. L. Runyan, “Sonic-Boom Propagation through a Stratified Atmosphere,” *Journal of the Acoustical Society of America*, vol. 51, no. 2C, pp. 695–701, 1972. DOI: `10.1121/1.1912903`.

[224] E. C. Aifantis and J. B. Serrin, “The Mechanical Theory of Fluid Interfaces and Maxwell’s rule,” *Journal of Colloid and Interface Science*, vol. 96, no. 2, pp. 517–529, 1983. DOI: `10.1016/0021-9797(83)90053-X`.

[225] R. A. Serway and J. W. Jewett, *Physics for Scientists and Engineers*, 6th ed. Hampshire, U.K.: Thomson Brooks/Cole, 2004, ISBN: 0534408427.

[226] S. D. Ramsey and R. S. Baty, “Symmetries of the gas dynamics equations using the differential form method,” *Journal of Mathematical Physics*, vol. 58, no. 11, p. 111 506, 2017. DOI: `10.1063/1.5011723`.

[227] J. Cheng and C. W. Shu, “Positivity-preserving Lagrangian scheme for multi-material compressible flow,” *Journal of Computational Physics*, vol. 257, no. Part A, pp. 143–168, 2014. DOI: `10.1016/j.jcp.2013.09.047`.

[228] A. Llor, A. Claisse, and C. Fochesato, “Energy preservation and entropy in Lagrangian space- and time-staggered hydrodynamic schemes,” *Journal of Computational Physics*, vol. 309, pp. 324–349, 2016. DOI: `10.1016/j.jcp.2015.12.044`.

[229] R. Abgrall, “A general framework to construct schemes satisfying additional conservation relations. Application to entropy conservative and entropy dissipative schemes,” *Journal of Computational Physics*, vol. 372, pp. 640–666, 2018. DOI: `10.1016/j.jcp.2018.06.031`.

[230] S. Busto, M. Dumbser, I. Peshkov, and E. Romenski, “On Thermodynamically Compatible Finite Volume Schemes for Continuum Mechanics,” *SIAM Journal on Scientific Computing*, vol. 44, no. 3, A1723–A1751, 2022. DOI: `10.1137/21M1417508`.

[231] J. Kreeft and M. Gerritsma, “Mixed mimetic spectral element method for Stokes flow: A pointwise divergence-free solution,” *Journal of Computational Physics*, vol. 240, pp. 284–309, 2013. DOI: `10.1016/j.jcp.2012.10.043`.

[232] T. A. Eymann and P. L. Roe, “Active Flux Schemes,” in *49th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition*, Orlando, FL, USA, 2011. DOI: `10.2514/6.2011-382`.

[233] T. A. Eymann and P. L. Roe, “Active Flux Schemes for Systems,” in *20th AIAA Computational Fluid Dynamics Conference*, Honolulu, HI, USA, 2011. DOI: `10.2514/6.2011-3840`.

[234] T. A. Eymann and P. L. Roe, “Multidimensional Active Flux Schemes,” in *21st AIAA Computational Fluid Dynamics Conference*, San Diego, CA, USA, 2013. DOI: `10.2514/6.2013-2940`.

[235] D. Fan and P. L. Roe, "Investigations of a New Scheme for Wave Propagation," in *22nd AIAA Computational Fluid Dynamics Conference*, Dallas, TX, USA, 2015. DOI: `10.2514/6.2015-2449`.

[236] P. L. Roe, "Is Discontinuous Reconstruction Really a Good Idea?" *Journal of Scientific Computing*, vol. 73, no. 2-3, pp. 1094–1114, 2017. DOI: `10.1007/s10915-017-0555-z`.

[237] W. Barsukow, J. Hohm, C. Klingenberg, and P. L. Roe, "The Active Flux Scheme on Cartesian Grids and Its Low Mach Number Limit," *Journal of Scientific Computing*, vol. 81, pp. 594–622, 2019. DOI: `10.1007/s10915-019-01031-z`.

[238] F. He and P. L. Roe, "A novel numerical scheme based on Active Flux method for the hyperbolic heat equations in multidimensional space," in *AIAA Aviation 2019 Forum*, Dallas, TX, USA, 2019. DOI: `10.2514/6.2019-3638`.

[239] F. He and P. L. Roe, "The Treatment of Conservation in the Active Flux method," in *AIAA Aviation 2020 Forum*, Virtual Event, 2020. DOI: `10.2514/6.2020-3032`.

[240] W. Barsukow, "The Active Flux Scheme for Nonlinear Problems," *Journal of Scientific Computing*, vol. 86, no. 3, pp. 1–34, 2021. DOI: `10.1007/s10915-020-01381-z`.

[241] J. P. Berberich and C. Klingenberg, "On the Active Flux Scheme for hyperbolic PDEs with source terms," *SIAM Journal on Scientific Computing*, vol. 43, no. 6, A4015–A4042, 2021. DOI: `10.1137/20M1346675`.

[242] V. Jain, J. Fisser, A. Palha, and M. Gerritsma, "A Conservative Hybrid Method for Darcy Flow," in *Spectral and High Order Methods for Partial Differential Equations ICOSAHOM 2018*, ser. Lecture Notes in Computational Science and Engineering, S. J. Sherwin, D. Moxey, J. Peiró, P. E. Vincent, and C. Schwab, Eds., vol. 134, Cham, Switzerland: Springer, 2020, pp. 215–227. DOI: `10.1007/978-3-030-39647-3_16`.

[243] Y. Zhang, J. Fisser, and M. Gerritsma, "A hybrid mimetic spectral element method for three-dimensional linear elasticity problems," *Journal of Computational Physics*, vol. 433, p. 110 179, 2021. DOI: `10.1016/j.jcp.2021.110179`.

[244] F. Vilar, P. H. Maire, and R. Abgrall, "Cell-centered discontinuous Galerkin discretizations for two-dimensional scalar conservation laws on unstructured grids and for one-dimensional Lagrangian hydrodynamics," *Computers and Fluids*, vol. 46, no. 1, pp. 498–504, 2011. DOI: `10.1016/j.compfluid.2010.07.018`.

[245] R. Abgrall and S. Tokareva, "Staggered Grid Residual Distribution Scheme for Lagrangian Hydrodynamics," *SIAM Journal on Scientific Computing*, vol. 39, no. 5, A2317–A2344, 2017. DOI: `10.1137/16M1078781`.

[246] R. Abgrall, P. Bacigaluppi, and S. Tokareva, "High-order residual distribution scheme for the time-dependent Euler equations of fluid dynamics," *Computers and Mathematics with Applications*, vol. 78, no. 2, pp. 274–297, 2019. DOI: `10.1016/j.camwa.2018.05.009`.

[247] J. Kreeft, "Mimetic Spectral Element Method: A discretization of geometry and physics," Ph.D. Thesis, TU Delft, Delft, the Netherlands, 2013. DOI: `10.4233/uuid:adbca9e7-08b9-4da3-9174-85c030dd1a15`.

[248] R. Rashad, F. Califano, F. P. Schuller, and S. Stramigioli, "Port-Hamiltonian modeling of ideal fluid flow: Part I. Foundations and kinetic energy," *Journal of Geometry and Physics*, vol. 164, p. 104 201, 2021. DOI: `10.1016/j.geomphys.2021.104201`.

[249] R. Rashad, F. Califano, F. P. Schuller, and S. Stramigioli, "Port-Hamiltonian modeling of ideal fluid flow: Part II. Compressible and incompressible flow," *Journal of Geometry and Physics*, vol. 164, p. 104 199, 2021. DOI: `10.1016/j.geomphys.2021.104199`.

[250] S. M. Panakkal, R. P. Parameswaran, and M. J. Vedan, "A geometric algebraic approach to fluid dynamics," *Physics of Fluids*, vol. 32, no. 8, p. 087 111, 2020. DOI: `10.1063/5.0017344`.

[251] J.-P. Caltagirone, "On Helmholtz-Hodge decomposition of inertia on a discrete local frame of reference," *Physics of Fluids*, vol. 32, no. 8, p. 083 604, 2020. DOI: `10.1063/5.0015837`.

[252] J.-P. Caltagirone, "On a reformulation of Navier-Stokes equations based on Helmholtz-Hodge decomposition," *Physics of Fluids*, vol. 33, no. 6, p. 063 605, 2021. DOI: `10.1063/5.0053412`.

[253] J.-P. Caltagirone, “An alternative to the concept of continuous medium,” *Acta Mechanica*, vol. 232, pp. 4691–4703, 2021. DOI: `10.1007/s00707-021-03070-w`.

[254] J.-P. Caltagirone, “Application of discrete mechanics model to jump conditions in two-phase flows,” *Journal of Computational Physics*, vol. 432, p. 110 151, 2021. DOI: `10.1016/j.jcp.2021.110151`.

[255] T. Vazquez-Gonzalez, A. Llor, and C. Fochesato, “Ransom test results from various two-fluid schemes: Is enforcing hyperbolicity a thermodynamically consistent option?” *International Journal of Multiphase Flow*, vol. 81, pp. 104–112, 2016. DOI: `10.1016/j.ijmultiphaseflow.2015.12.007`.

[256] S. Busto and M. Dumbser, “A New Family of Thermodynamically Compatible Discontinuous Galerkin Methods for Continuum Mechanics and Turbulent Shallow Water Flows,” *Journal of Scientific Computing*, vol. 93, no. 56, 2022. DOI: `10.1007/s10915-022-02017-0`.

[257] F. Weinhold, “Metric geometry of equilibrium thermodynamics,” *The Journal of Chemical Physics*, vol. 63, no. 6, pp. 2479–2483, 1975. DOI: `10.1063/1.431689`.

[258] F. Weinhold, “Metric geometry of equilibrium thermodynamics. II. Scaling, homogeneity, and generalized Gibbs-Duhem relations,” *The Journal of Chemical Physics*, vol. 63, no. 6, pp. 2484–2487, 1975. DOI: `10.1063/1.431635`.

[259] F. Weinhold, “Metric geometry of equilibrium thermodynamics. III. Elementary formal structure of a vector-algebraic representation of equilibrium thermodynamics,” *The Journal of Chemical Physics*, vol. 63, no. 6, pp. 2488–2495, 1975. DOI: `10.1063/1.431636`.

[260] F. Weinhold, “Metric geometry of equilibrium thermodynamics. IV. Vector-algebraic evaluation of thermodynamic derivatives,” *The Journal of Chemical Physics*, vol. 63, no. 6, pp. 2496–2501, 1975. DOI: `10.1063/1.431637`.

[261] H. Quevedo, “Geometrothermodynamics,” *Journal of Mathematical Physics*, vol. 48, no. 1, p. 013 506, 2007. DOI: `10.1063/1.2409524`.

[262] M. Azreg-Aïnou, “Geometrothermodynamics: comments, criticisms, and support,” *European Physical Journal C*, vol. 74, no. 2930, 2014. DOI: `10.1140/epjc/s10052-014-2930-3`.

[263] J. P. Berrut and L. N. Trefethen, “Barycentric Lagrange Interpolation,” *SIAM Review*, vol. 46, no. 3, pp. 501–517, 2004. DOI: `10.1137/S0036144502417715`.

[264] N. J. Higham, “The numerical stability of barycentric Lagrange interpolation,” *IMA Journal of Numerical Analysis*, vol. 24, no. 4, pp. 547–556, 2004. DOI: `10.1093/imanum/24.4.547`.

[265] E. Laughton, V. Zala, A. Narayan, R. M. Kirby, and D. Moxey, “Fast Barycentric-Based Evaluation Over Spectral/hp Elements,” *Journal of Scientific Computing*, vol. 90, no. 78, 2022. DOI: `10.1007/s10915-021-01750-2`.

[266] C. R. Dohrmann, “Spectral Equivalence of Low-Order Discretizations for High-Order H(curl) and H(div) Spaces,” *SIAM Journal on Scientific Computing*, vol. 43, no. 6, A3992–A4014, 2021. DOI: `10.1137/21M1392115`.