Runge-Kutta Residual Distribution Schemes

by

Andrzej Warzyński

Submitted in accordance with the requirements
for the degree of Doctor of Philosophy.

UNIVERSITY OF LEEDS
The University of Leeds
School of Computing

May 2013

The candidate confirms that the work submitted is his/her own, except where work which has formed part of jointly authored publications has been included. The contribution of the candidate and the other authors to this work has been explicitly indicated below. The candidate confirms that appropriate credit has been given within the thesis where reference has been made to the work of others.

Some parts of the work presented in this thesis in Chapter 5 have been published in the following article:

A.Warzyński, M.E.Hubbard, M.Ricchiuto, "Discontinuous residual distribution schemes for time-dependent problems", in *Proceedings of the 8th International Conference on Scientific Computing and Applications, J.Li, H.Yang (Eds.)*, 2012, Contemporary Mathematics, Volume: 586

The derivation of the algorithm presented in the above paper was carried out jointly by all authors. Implementatio and experimental validation of the proposed discretisation technique was carried solely by the author of this thesis.

This copy has been supplied on the understanding that it is copyright material and that no quotation from the thesis may be published without proper acknowledgement.

© 2013 The University of Leeds and Andrzej Warzyński

Acknowledgements

First and foremost, I would like to thank Dr Matthew Hubbard for his keen supervision, guidance and continuous encouragement to review my calculations and numerical implementations whenever I was willing to accept that they are correct regardless the results I was getting. Next, I want to thank Dr Mark Walkley, Dr Mario Ricchiuto and Dr Domokos Sàrmány for all the enlightening discussions and their suggestions while working on this thesis. I also wish to thank all the current and past members of the Scientific Computation research group at the University of Leeds for numerous conversations on matters not necessarily related to my research project. In particular, Keeran Brabazon for his ‘do you fancy a cup of tea?’. Finally, I am very grateful for the support I received from all other persons over the last three and a half years. There were many (probably too many to include in this tiny section) and I keep all of them in my heart - thank you guys!

Abstract

The residual distribution framework and its ability to carry out genuinely multidimensional upwinding has attracted a lot of research interest in the past three decades. Although not as robust as other widely used approximate methods for solving hyperbolic partial differential equations, when residual distribution schemes do provide a plausible solution it is usually more accurate than in the case of other approaches. Extending these methods to time-dependent problems remains one of the main challenges in the field. In particular, constructing such a solution so that the resulting discretisation exhibits all the desired properties available in the steady state setting.

It is generally agreed that there is not yet an ideal generalisation of second order accurate and positive compact residual distribution schemes designed within the steady residual distribution framework to time-dependent problems. Various approaches exist, none of which is considered optimal nor completely satisfactory. In this thesis two possible extensions are constructed, analysed and verified numerically: continuous-in-space and discontinuous-in-space Runge-Kutta Residual Distribution methods. In both cases a Runge-Kutta-type time-stepping method is used to integrate the underlying PDEs in time. These are then combined with, respectively, a continuous- and discontinuous-in-space residual distribution type spatial approximation.

In this work a number of second order accurate linear continuous-in-space Runge-Kutta residual distribution methods are constructed, tested experimentally and compared with existing approaches. Additionally, one non-linear second order accurate scheme is presented and verified. This scheme is shown to perform better in the close vicinity of discontinuities (in terms of producing spurious oscillations) when compared to linear second order schemes. The experiments are carried out on a set of structured and unstructured triangular meshes for both scalar linear and non-linear equations, and for the Euler equations of fluid dynamics as an example of systems of non-linear equations.

In the case of the discontinuous-in-space Runge-Kutta residual distribution framework, the thorough analysis presented here highlights a number of shortcomings of this approach and shows that it is not as attractive as initially anticipated. Nevertheless, a rigorous overview of this approach is given. Extensive numerical results on both structured and unstructured triangular meshes confirm the analytical results. Only results for scalar (both linear and non-linear) equations are presented.

Declarations

Some parts of the work presented in this thesis have been published in the following articles:

A.Warzyński, M.E.Hubbard, M.Ricchiuto, "Discontinuous residual distribution schemes for time-dependent problems", in *Proceedings of the 8th International Conference on Scientific Computing and Applications, J.Li, H.Yang (Eds.)*, 2012, Contemporary Mathematics, Volume: 586

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Many physical and biological phenomena can be viewed and described as flows of fluids. This includes currents in oceans, atmospheric flows, lava inside the Earth, blood in veins or flow of air around space craft, to name just a few. Originally, such problems were studied with the aid of traditional laboratory experiments, i.e. wind tunnels. Partial differential equations (often abbreviated to PDEs) modelling such processes were also used, but their complexity limited practical use. It was not until the late 1950s that researchers started using computers to simulate fluid problems by solving the underlying PDEs numerically. Although not always entirely reliable, computer simulations soon became very powerful and one of the key tools in studies of fluids. These approaches eventually evolved into a separate research field - computational fluid dynamics.

In the field of mathematical modelling and computational fluid dynamics, systems of hyperbolic conservation laws are of particular interest. They often model a somewhat simplified scenario, i.e. some physical processes/forces are not taken into account, yet provide a qualitatively accurate description of real life phenomena. Such an approach reduces mathematical complexity, which then allows a significant reduction in the expense of providing numerical predictions for many flows that are of practical use. As an example, consider the Euler equations governing flow of inviscid compressible flow, which comprise three fundamental conservation laws: conservation of mass, momentum and energy. This system of equations is one of the most important systems in gas dynamics and is frequently used in aerodynamics to model flow of air around aircraft (to be more precise, in the inviscid flow regime). Although derived by neglecting various physical processes (viscous forces, thermal conductivity and turbulence), the Euler equations are considered to be a very useful mathematical model of the underlying fluid dynamics. In particular, in the case of high-speed flows. Unfortunately, they admit few analytic solutions and only for rather trivial problems. Hence the need to study alternative methods of approximating them such as numerical approximations.

With rapid growth in available computer power, numerical simulations have become one of the key research tools for studying fluid flows. In the case of hyperbolic partial differential equations, the majority of methods applied to the solution of the underlying flow problems are those developed within the Finite Volume ($\mathcal{FV}$) framework. Their popularity is largely due to their ability to mimic important physical properties like conservation, upwinding and monotonicity. In one space dimension, these methods have reached a high degree of sophistication and understanding and are considered to be very elegant and *physical*. However, $\mathcal{FV}$ methods do not extend readily to multiple dimensions. This is mainly due to the fact that the Riemann problem [101] that they heavily depend on does not extend readily to multiple dimensions. The usual workaround is to apply the one dimensional $\mathcal{FV}$ formulation along particular mesh directions (for instance, perpendicular to the edges). Consequently, the schemes are no longer quite as physical and this causes a corresponding decrease in accuracy via excessive numerical dissipation. This lack of a genuinely multidimensional approach is understood to be the main factor reducing the accuracy of finite volume schemes on unstructured grids [35]. The construction of second or higher-order methods within the $\mathcal{FV}$ framework is performed with the aid of relatively expensive (especially on unstructured meshes) reconstruction of polynomials of the proper degree. The MUSCL method method of Van Leer [104] is one example. The underlying procedure extends the stencil of the scheme making it non-compact and hence less efficient. Still, the flexibility, adaptability and applicability to flow problems in domains with complicated geometry have enabled the finite volume framework to remain the most frequent choice when simulating flows governed by hyperbolic PDEs. Finite difference methods [98], although relatively straightforward when compared to finite volumes approximations, become rather impractical when dealing with complex flow patterns for which unstructured grids are considered mandatory. The main advantage of finite difference methods when compared to finite volumes is that these methods do not introduce such a huge overhead when constructing higher than first order approximations. Nevertheless, in this thesis the focus is laid on numerical methods on unstructured meshes (even though structured triangulations are also considered) and hence finite difference discretisations are not included in the discussion.

It is generally agreed that the state of the art of numerical methods for hyperbolic partial differential equations is not entirely satisfactory. Finite difference methods are clearly not robust enough. Attempts at *ultra-high resolution* computations using finite volume methods prove that it is not only the lack of available computer power that limits the accuracy of computations, but also the schemes themselves which are not able to capture highly nonlinear physical phenomena [35]. Instead, they add superfluous waves and the reconstruction is no longer physically close to the true solution. This is largely due to their inability to perform genuinely multidimensional upwinding and thus failure to mimic all the physics described by the equations. Therefore, other alternatives have to be investigated.

1.2 Multidimensional Upwinding and the Residual Distribution Framework

Every hyperbolic conservation law contains information about propagation of some sort of physical phenomenon and, more importantly, about the preferred *trajectories* along which this phenomenon propagates. Mathematically this can be explored and investigated by the method of characteristics - see [21, 28, 48, 72, 106] for details. Unfortunately, because of its complexity, primarily in the case of multidimensional problems, it is an analytical rather than a computational tool. In the case of one-dimensional problems, the method of characteristics inspired the development of upwind schemes which are found to be very accurate, robust and efficient methods for approximating hyperbolic PDEs. Disappointingly, there is no straightforward way of applying upwinding in a genuinely multidimensional manner. This subject was thoroughly studied in a series of papers by Roe [90, 91] and Deconinck [35].

Briefly speaking, the information described by any set of hyperbolic conservation laws travels in the form of waves (see [72] to learn more about the simple wave solutions). In the case of one-dimensional problems, these waves can only move in one of two directions, i.e. positive or negative space direction, which can be easily described on a numerical level (upwinding can be viewed as the ability of an algorithm to "follow" the appropriate direction). This is no longer the case when multidimensional problems are considered. Now waves can travel in an infinite number of directions which cannot be replicated in the discrete world. Instead, a fixed number of preferred directions is chosen (usually aligned with the mesh) along which one-dimensional problems are solved. This simplification may (and often does) lead to misinterpretation of the flow and consequently an inaccurate solution. Consider for instance the two-dimensional Euler equations. Locally, a solution of this system can be represented as a sum of simple wave solutions out of which one is an entropy wave, a second is a shear wave and the remaining two are acoustic waves. As observed in [35], selecting wrong directions along which the upwinding is performed (e.g. dependent on the mesh) may lead to a decomposition of a shear wave (which does not exist in one dimension) into three one-dimensional acoustic waves travelling with speeds which do not agree with the speed of the original wave.

The desire to construct schemes able to mimic the propagation of data in a truly multidimensional manner (i.e. to perform multidimensional upwinding) led to the development of wave-decomposition schemes and ultimately the Residual-Distribution ($\mathcal{RD}$) framework was proposed [89]. The superiority of this approach over, for example, $\mathcal{FV}$ schemes becomes apparent when dealing with multidimensional problems where physical phenomena are not necessarily aligned with the computational mesh. This is the setting that currently attracts the most interest. One of the earliest comparisons of these two approaches can be found in [93]. For other promising experimental observations on this matter refer to [1,51] and [108]. It was also demonstrated (see, for instance, [6,7,66,95] and [42]) that residual distribution methods are very robust and perform well when applied to complex problems arising in engineering and other applications, e.g. shallow water flows.

1.3 Recent Developments

The discontinuous Galerkin framework [22,37] is a yet another approach to solving hyperbolic PDEs that has been challenging the dominance of finite volume methods in the past 20 years. As with the latter, upwinding is performed with the aid of the so-called numerical fluxes. In the one-dimensional setting such an approach enables very accurate prediction of the underlying fluid flow. However, extension to two and three-dimensional scenarios is done heuristically, which is not always sufficient to capture complex physical phenomena present in the flow. In this respect, discontinuous Galerkin methods are similar to finite volume approximations and are not able capable of performing *genuinely multidimensional* upwinding. The main difference between the $\mathcal{DG}$ and $\mathcal{FV}$ frameworks is that the former is derived from the Galerkin finite element framework ($\mathcal{FE}$) for which a discontinuous-in-space data representation was assumed (discontinuities in time will not be covered here). Numerical fluxes, known from finite volume methods, are then introduced in order to impose communication between cells, and, ultimately, guarantee stability and physical realism (upwinding). The discontinuous Galerkin formulation, as opposed to finite volume methods, allows detailed formal analysis and error estimation (see, for example, [53, 55]). It facilitates $h-$adaptivity and is much better suited for $p-$adaptivity [54] than finite volume methods. This comes from the fact that in the case of the most successful high order finite volume schemes, e.g. the ENO [52] or WENO [63, 71] methods, higher order approximations are achieved with the aid of expensive nonlocal reconstruction procedures. In the case of discontinuous Galerkin schemes higher order approximations are constructed by considering in every mesh cell a higher order polynomial representation of the data. This can be done in each cell separately and thus provides a natural tool for $p-$adaptivity. The main advantage of discontinuous Galerkin methods when compared with the Galerkin finite element method is the locality of the resulting discrete formulation. This is achieved by relaxing the constraint on the continuity of the underlying approximation. The discontinuous Galerkin method also exhibits much better stability than Galerkin $\mathcal{FE}$ method, which is imposed by introducing upwinding.

The discontinuous Galerkin framework was among the key inspirations that led to the inception of the discontinuous-in-space residual distribution framework. This recent development, proposed simultaneously by Hubbard [57, 58] and Abgrall [3], aims at drawing together advantages of the residual distribution (multidimensional upwinding) and discontinuous Galerkin (localised system) approaches. It is constructed by relaxing the constraint on the continuity of the data and allowing discontinuities across cell interfaces. Similar philosophy lies at the centre of the discontinuous Galerkin framework. However, discontinuous-in-space residual distribution methods employ the so-called edge residuals (i.e. flux differences) rather than numerical fluxes to introduce upwinding. It is still a very new, and neither fully developed nor understood, strand of research. Extending this framework to time-dependent problems is the first key goal of this thesis.

In the case of steady state problems the $\mathcal{RD}$ framework has reached a high level of sophistication and understanding. The most recent reviews can be found in [38] and [4]. Further research is still being carried out (e.g. on discontinuous-in-space $\mathcal{RD}$ methods), but the emphasis is now mainly laid on the development of residual distribution methods for time-dependent problems. The main challenge is to design a scheme which retains all the properties of its steady counterpart(s) (in particular positivity and linearity preservation [38]), and which is relatively efficient. The space-time framework investigated in [29] (see also [10, 34, 38, 44] and [31]) allows construction of discretization with *all* the desired properties. Unfortunately, the methods described are subject to a CFL-type restriction on the time-step, which is particularly disappointing when taking into account that they are, by construction, implicit. In the *two layer* variant, [32] one couples two space-time slabs at a time and solves the equations simultaneously in both. On one hand the resulting system to be solved at each step is larger, but on the other the construction removes from one of the layers the restriction on the time-step. In theory this means that an arbitrarily large time-step can be used. For a full discussion see [29]. Hubbard and Ricchiuto [60] proposed to drive the height of one of the space-time slabs (and hence its associated time-step) to zero so that the scheme becomes discontinuous-in-time. The resulting formulation is simpler than the original whereas all of the desired properties are retained. Recently, Sármány et al. [61] applied this approach to shallow water equations to show that it outperforms other implicit residual distribution methods. It is, however, very expensive when compared to explicit methods.

A different approach to solving time-dependent equations with the aid of the $\mathcal{RD}$ framework was proposed by R. Abgrall and M. Ricchiuto in [85]. Their explicit Runge–Kutta Residual Distribution ($\mathcal{RKRD}$) framework, being explicit, solves one of the issues mentioned above, namely the efficiency of $\mathcal{RD}$ methods for time-dependent problems. The authors conducted a very rigorous study by experimenting with various types of time-integration algorithms (second and third order TVD Runge–Kutta methods [97]), formulations of the mass matrices (four distinct definitions) and two types of lumping - the so-called global and selective lumping (see [85] for the definitions). All of the schemes the authors presented (and which fall into the framework their proposed) have similar qualitative properties - they are second order accurate, but not completely oscillation-free. The methodology proposed by the authors can be viewed as an approximation to the implicit Runge–Kutta residual distribution methods introduced in this thesis. The main difference between the two is the fact that in the case of explicit $\mathcal{RKRD}$ methods the resulting linear system is diagonal (hence its explicit nature) and in the case of implicit $\mathcal{RKRD}$ methods the resulting system of equations is not diagonal and has to be inverted before one can advance from one time level to another. Introducing the implicit $\mathcal{RKRD}$ framework and comparing it in terms of accuracy, efficiency and monotonicity with its explicit counterpart is the second main goal of this thesis.

1.4 Key Assumptions

Throughout this thesis only two-dimensional problems (i.e. with the spatial domain embedded in $\mathbb{R}^2$) will be considered. The reason for this assumption is two-fold. First of all, the potential of residual distribution methods becomes most apparent when multidimensional problems exhibiting complex physical phenomena are considered. Hence these methods are of little interest in the simplified one-dimensional scenario where the difference between particular upwind discretisations is minimal. Three-dimensional problems are beyond the scope of this thesis and will not be covered here. Nevertheless, it should be pointed out that concepts discussed in this thesis quite naturally extend to more complex scenarios in $\mathbb{R}^3$. Some examples are discussed in [6].

The discrete representation of the data that is used throughout this thesis will remain piece-wise linear. As in the case of three-dimensional computations, extension to higher order approximations, although possible (see, for example, [13]), is beyond the scope of this thesis and will not be discussed. To avoid confusion in the interpretation of this text, this assumption will be recalled in the text whenever other details regarding the discussed methods are being outlined.

1.5 The Underlying Goals

The setting outlined in the previous section can be viewed as the set of constraints within which the development of new numerical algorithms is carried out in this thesis. There are three additional design criteria that will be taken into account here. The following are essential in the development of flexible and robust numerical algorithms for hyperbolic PDEs:

- **Accuracy** As already mentioned, only piecewise linear approximations will be considered throughout this thesis. Quite naturally, such a setting should lead to second order accurate schemes (super-convergence is not taken into account). Designing a second order accurate scheme with a linear basis is one of the key aims in this thesis.

- **Stability** Conservation laws admit discontinuous solutions with piece-wise smooth profile and without strong oscillations in the vicinity of the singularities. A numerical method solving such conservation laws must be capable of producing approximate solutions free of spurious oscillations causing instabilities. Moreover, it should perform this in a parameter-free fashion, that is independently of constants specific to particular problems.

- **Efficiency** The resulting discretisation should be accurate and stable and achieve this at modest computational cost. In this thesis this is achieved by considering only explicit time-integrators. A numerical method should also be compact, i.e. it should compute the value of unknowns in a certain mesh location based on information only from the closest grid-entities. Compactness is one of key characteristics of residual distribution methods, which is further enhanced in this thesis by introducing a discontinuous-in-space data representation.

It is not always possible to combine accuracy, stability and efficiency in one scheme. As a matter of fact, it remains an open challenge to design an algorithm within the $\mathcal{RD}$ framework that for time-dependent problems is second order accurate, produces solutions free of spurious oscillations and that on top of that constitutes inexpensive discretisations. This thesis explores possible approaches to tackle shortcomings in existing schemes and to design one that would indeed be accurate, stable and efficient.

1.6 Contribution

The research presented in this thesis deals with the construction of new numerical algorithms within the residual distribution framework and applying them to both scalar and systems of non-linear hyperbolic partial differential equations, with the emphasis laid on solving time-dependent problems. The contributions of this thesis and new developments proposed can be split into three groups:

1. A thorough overview and comparison of two distinct discontinuous-in-space residual distribution frameworks, the first due to Hubbard [57] and the second proposed by Abgrall [3], is given. The main difference between the two approaches is the way edge-based residuals are treated. The discontinuous-in-space residual distribution framework is then further extended by introducing a new distribution strategy for edge residuals. Extensive numerical comparison reveals that the

approach proposed by Hubbard leads to the most robust discretisations (in terms of accuracy, stability and efficiency of the available methods). Even though previous attempts were unsuccessful [57], application to time-dependent problems and the presented numerical results show that this framework is indeed time-accurate.

2. A study of similarities between the residual distribution and finite element frameworks is extended to the discontinuous-in-space setting. Common features of discontinuous-in-space residual distribution and the so-called strong form of the discontinuous Galerkin method are thoroughly discussed. A number of links between the two frameworks are highlighted and discussed. This investigation was motivated by the desire to construct a robust, *second* order discontinuous-in-space residual distribution method for time dependent problems. Comparing the two approaches led to an introduction of a new distribution strategy for edge-based residuals (see Point 1.).

3. The second order TVD Runge-Kutta method [97] is employed and implemented to construct a new continuous-in-space residual distribution scheme for time-dependent problems. The properties of the resulting discretisation are rigorously studied with the aid of extensive numerical experiments. An efficient way of solving the resulting linear system is also proposed. Recently, Ricchiuto and Abgrall [85] employed a modified/*shifted* TVD Runge-Kutta procedure to derive a genuinely explicit second order residual distribution scheme for which the resulting linear system is diagonal. Although the results they obtained are sound and very interesting, the comparison presented here shows that the superiority in terms of efficiency of the genuinely explicit approach is not as striking as originally assumed. A discontinuous-in-space data representation is also incorporated into this new framework and a number of numerical results are presented.

Furthermore, to investigate robustness of the discussed numerical schemes, the Euler equations of fluid dynamics were discretised and solved with the aid of the presented numerical methods.

1.7 Thesis Outline

In the following chapters different classes of residual distribution methods are derived and discussed and the corresponding mathematical problems used in the numerical experiments are introduced.

Chapter 2 focuses on introducing the residual distribution ($\mathcal{RD}$) framework for scalar steady-state problems. A continuous-in-space data representation is assumed and a review of the most successful and frequently used $\mathcal{RD}$ methods falling into this category is given. The discussion is summarised with a selection of numerical results. In Chapter 3 the assumption on the continuity of the data is relaxed and the discontinuous-in-space residual distribution framework is introduced. All available schemes falling into this framework are first presented and then compared experimentally. Additionally, a new way of distributing edge-based residuals is introduced and evaluated numerically. Residual distribution methods for time-dependent problems are dealt with in Chapter 4. In particular the Runge-Kutta residual distribution schemes are studied. As in Chapter 2, the discrete representation of the data is again assumed to be continuous. A new second order approximation is introduced and results of a thorough numerical investigation are presented to demonstrate the behaviour of this new method. Incorporating the discontinuous-in-space data representation into the new framework developed in Chapter 4 is the main goal of Chapter 5. This new technique motivated a thorough study into similarities between the discontinuous Galerkin and discontinuous residual distribution frameworks. The outcome of that research is thoroughly discussed and extensive numerical results are given. Chapter 6 is devoted to further evaluation of the numerical frameworks presented in this thesis. In particular, a detailed description of the procedure that is used to apply residual distribution methods to the Euler equations of gas dynamics is given. This is then followed by an extensive numerical study, carried out for both the steady-state and transient problems. Concluding remarks and future prospects are outlined in Chapter 7. Appendix A contains the exact solution to one of the test problems used in Chapters 4 and 5, namely the two-dimensional inviscid Burgers' equation. A brief overview of the notation employed in this thesis can be found in Appendix B. Appendix C contains the derivation of the consistent mass matrix employed in Chapters 4 and 5 and Appendix D deals with the derivation of the limit on a time-step guaranteeing positivity of one of the schemes considered in Chapter 3. Finally, in Appendix E a compact definition of a new framework introduced in Chapter 3 is given.

Chapter 2

The Continuous $\mathcal{RD}$ Framework

2.1 Introduction

Systems of nonlinear hyperbolic PDEs, such as the Euler or Shallow Water equations, are among the most interesting, but also challenging models in fluid dynamics. Desire to increase the accuracy, efficiency and robustness with which these models are approximated stimulated the inception of the Residual Distribution ($\mathcal{RD}$) framework. In practice, it is very often the case that numerical methods for this type of complex problem are based on approximate solvers for *scalar* hyperbolic equations, which are then, more or less heuristically, extended and applied to systems. This was the case when the residual distribution methods were introduced by Roe in 1982 [89]. It is thus essential, at least as far as residual distribution schemes are concerned, to understand how to tackle scalar equations before attempting to solve more realistic and complex problems governed by systems of nonlinear equations. The development of such understanding is the main purpose of this chapter. In particular, it will be shown how residual distribution methods can be used to solve scalar conservation laws:

$$\partial_t u + \nabla \cdot \mathbf{f}(u) \ = \ 0 \qquad \text{in } \Omega \times [0, T], \tag{2.1}$$

with Ω being the spatial domain and T being a given final time.

Equation (2.1) is very often considered in its integral form:

$$\int_\Omega \partial_t u \, d\Omega + \int_\Omega \nabla \cdot \mathbf{f}(u) \, d\Omega \;=\; 0 \qquad \text{in } \Omega \times [0,T], \tag{2.2}$$

or, equivalently, as:

$$\int_\Omega \partial_t u \, d\Omega + \oint_{\partial\Omega} \mathbf{f}(u) \cdot \mathbf{n} \, d\Gamma \;=\; 0 \qquad \text{in } \Omega \times [0,T], \tag{2.3}$$

in which $\mathbf{n}$ is the outward unit normal to the boundary $\partial\Omega$ of Ω. The above states that the rate of change of a given conserved quantity u in any spatial domain Ω is balanced by the flux of this quantity (denoted here by $\mathbf{f}(u)$) through the boundary of Ω. Obviously, every function u that satisfies (2.1) will also satisfy (2.2) and (2.3), but not necessarily vice-versa. However, balance laws are usually derived in the integral form first and then expressed in terms of derivatives like (2.1). In this respect, Formulations (2.2) and (2.3) are more plausible from a physical point of view and hence the focus in this thesis is laid on finding the solution to the integral formulation. In order to pose a well-defined mathematical and physical problem, Equation (2.2) has to be equipped with an initial solution:

$$u(\mathbf{x}, t = 0) = u_0(\mathbf{x}) \qquad \mathbf{x} \in \Omega,$$

and/or some boundary conditions defined on $\partial\Omega$ or a properly defined subset (see [48] for details on imposing boundary conditions for this type of mathematical problems).

The main idea underlying $\mathcal{RD}$ discretisations is incorporating as much physics into the computational model as possible. The challenge is particularly acute in fluid mechanics, where a complex continuous problem is replaced by a discrete model. In order to achieve this, Roe [89] introduced two basic concepts: *'A fluctuation is something detected in the data, indicating that it has not yet reached equilibrium, and a signal is an action performed on the data so as to bring it closer to equilibrium.'* (p. 221). To see how this is applied in practice, consider the steady state counterpart of Equation (2.2):

$$\int_\Omega \nabla \cdot \mathbf{f}(u) \, d\Omega \;=\; 0 \qquad \text{in } \Omega, \tag{2.4}$$

with inflow boundary conditions defined on $\partial\Omega$. Equation (2.4) describes an equilibrium of some physical phenomenon. In this case, reaching the state of balance is equivalent to finding the steady state solution. To test whether this has been achieved, fluctuations (also referred to as residuals) are calculated:

$$\phi^K = \int_K \nabla \cdot \mathbf{f}(u)\, d\Omega,$$

in which K is a given subset of Ω. Existence of a set $K' \subset \Omega$ such that the fluctuation $\phi^{K'}$ is non-zero indicates that the equilibrium has not yet been reached. In such a case signals, calculated as fractions of the fluctuation, are sent to mesh nodes to iterate to the steady state. This is, in short, an outline of how residual distribution methods came to life. A more formal definition of $\mathcal{RD}$ methods is given in the next section.

Originally the $\mathcal{RD}$ framework was considered only in terms of steady state solutions and only such problems are considered in this chapter. The definition of the $\mathcal{RD}$ framework is followed by a review of its key properties, particular examples of residual distribution methods and numerical experiments to report on their behaviour in practice.

2.2 The Framework

It is assumed that the spatial domain $\Omega \subset \mathbb{R}^2$ is subdivided into non-overlapping triangular elements, denoted by E, belonging to $\mathcal{T}_h$, such that

$$\bigcup_{E \in \mathcal{T}_h} E = \Omega.$$

The triangulation is assumed to be regular in the sense that there exist constants C_1 and C_2 such that

$$0 < C_1 \leq \sup_{E \in \mathcal{T}_h} \frac{h_E^2}{|E|} \leq C_2 < \infty,$$

in which h_E is the characteristic length of E (the length of its longest side) and $|E|$ is the area of E. Cell interfaces will be denoted by e and $\mathcal{D}_i$ will stand for the subset of triangles containing node $\mathbf{x}_i$. The median dual cell is obtained by joining the gravity centres of triangles in $\mathcal{D}_i$ with the midpoints of the edges meeting at $\mathbf{x}_i$. This is illustrated in Figure 2.1.

For each element $E \in \mathcal{T}_h$ and for each node $\mathbf{x}_i \in E$, ψ_i^E is defined as the linear Lagrange basis function associated with $\mathbf{x}_i$ respecting:

$$\psi_i^E(\mathbf{x}_j) \;=\; \delta_{ij} \;\; \forall i, j \in \mathcal{T}_h, \qquad \sum_{j \in E} \psi_j^E = 1 \;\; \forall E \in \mathcal{T}_h. \tag{2.5}$$

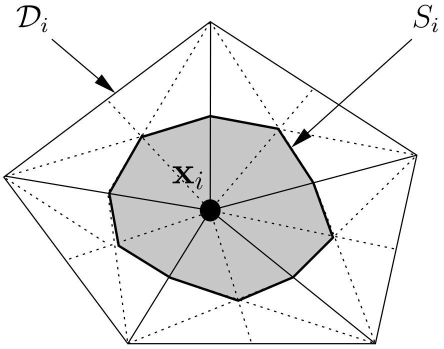


Figure 2.1: Median dual cell S_i.

As long as it does not introduce any ambiguity, the superscript ^{E} will be omitted. The approximate solution u_h is assumed to be globally continuous and linear within each element $E \in \mathcal{T}_h$, and to be of the following form:

$$u_h(\mathbf{x}) \;=\; \sum_i \psi_i(\mathbf{x})\, u_i, \tag{2.6}$$

in which $u_i = u_h(\mathbf{x}_i)$. It is worth noting that the assumption on the linearity of the underlying discrete representation can be relaxed, and indeed is when higher than second order residual distribution methods are considered [13, 19, 74]. Such generalisation is beyond the scope of this thesis and will not be considered here. Only piece-wise linear approximations will be discussed.

It is clear that in order to find u_h one has to construct a set of equations, ideally linear, to which the solution gives the nodal values of the approximate solution. In the residual distribution framework this is achieved via cell fluctuations, hereafter referred to as residuals:

$$\phi^E \;=\; \int_E \nabla \cdot \mathbf{f}(u)\, d\Omega.$$

These are computed for each cell $E \in \mathcal{T}_h$ and then, with the aid of the distribution coefficients $\beta_{i,E}$, split between its vertices as shown in Figure 2.2. These fractions will be referred to as signals and denoted as ϕ_i^E :

$$\beta_{i,E}\, \phi^E \;=\; \phi_i^E.$$

Most of the time the subscript in the distribution coefficient $\beta_{i,E}$ will be abbreviated to $_i$. The second parameter (the cell) will be clear from the context. To finish the construction of the system, for each node $\mathbf{x}_i \in \mathcal{T}_h$, assemble the signals and sum them up. For a steady state solution these sums should be equal to 0 and the resulting system of equations is given by:

$$\sum_{E \in \mathcal{D}_i} \beta_i \, \phi^E \; = \; 0 \qquad \forall i. \tag{2.7}$$

In practice, system (2.7) is solved with the aid of pseudo time-stepping:

$$u_i^{n+1} \; = \; u_i^n \; - \; \frac{\Delta t}{|S_i|} \sum_{E \in \mathcal{D}_i} \beta_i \phi^E \qquad \forall i, \tag{2.8}$$

which is used to iterate to the steady state. Constraints on Δt guaranteeing convergence of this iteration will be discussed later (see Section 2.6.2).

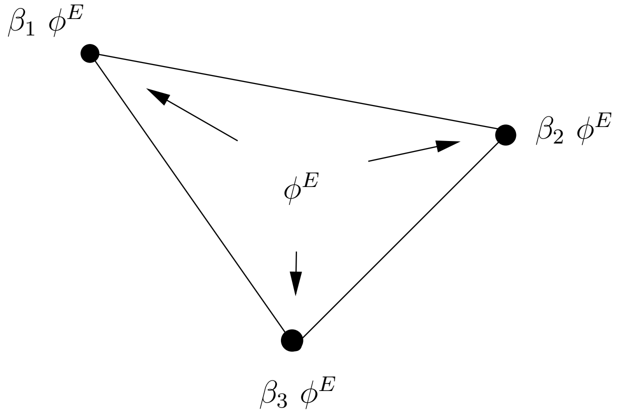


Figure 2.2: The distribution of the residual ϕ^E to the vertices of a cell.

Since the distribution coefficients remain unspecified, the above defines only a framework, not a particular scheme. It is rather intuitive that the βs ought to sum up to 1, i.e.

$$\beta_1 \, + \, \beta_2 \, + \, \beta_3 \; = \; 1 \qquad \forall_{E \in \mathcal{T}_h}.$$

If the βs do not some up to one, artificial *mass* is added to or taken from the system. Other restrictions on how the available information/residuals should be distributed will be discussed in Section 2.4. First, however, a particular example of a $\mathcal{RD}$ method will be presented. This is primarily to show a very close link between the residual distribution and finite element frameworks.

2.3 Relation to Finite Elements

The approximate solution (2.6) is assumed to be of the same form as in the case of linear Finite Element ($\mathcal{FE}$) approximations [17]. A natural question to ask is whether there exist more links between residual distribution and finite element frameworks? Interestingly enough, the latter can be rewritten in the $\mathcal{RD}$ formalism. Indeed, consider the scalar equation (2.4). The linear system resulting from discretizing it using the finite element method reads:

$$\sum_{E \in \mathcal{D}_i} \int_E \nabla \cdot \mathbf{f}(u_h)\, \psi_i \, d\Omega \;=\; 0 \quad \forall i, \tag{2.9}$$

in which, as previously, ψ_i stands for the Lagrange basis function associated with node i. It is apparent that also in this case signals are being sent to each node. These are then assembled to get the set of equations for the nodal values of the numerical solution. The definition of the signals, at least at first sight, differs from that of residual distribution methods. However, from the properties of the basis functions it follows that

$$\sum_{i \in E} \int_E \nabla \cdot \mathbf{f}(u_h)\, \psi_i \, d\Omega \;=\; \int_E \nabla \cdot \mathbf{f}(u_h)\, d\Omega \;=\; \phi^E.$$

The above expression implies that:

$$\int_E \nabla \cdot \mathbf{f}(u_h)\, \psi_i \, d\Omega \;=\; \beta_i^{\mathcal{FE}}\, \phi^E \qquad \text{and} \qquad \beta_i^{\mathcal{FE}} \;=\; \frac{\int_E \nabla \cdot \mathbf{f}(u_h)\, \psi_i \, d\Omega}{\int_E \nabla \cdot \mathbf{f}(u_h)\, d\Omega}.$$

Although the distribution coefficients $\beta_i^{\mathcal{FE}}$ are defined via a rather complicated formula, the above fits nicely into the framework outlined in the previous section. As a matter of fact, it is very often the case that the distribution coefficients are defined implicitly via the definition of the signals, ϕ_i^E. Further examples in Section 2.6 will confirm this.

The $\mathcal{FE}$ approximation becomes particularly interesting when considering the non-conservative form of (2.4):

$$\mathbf{a}(u) \cdot \nabla u = 0 \qquad \text{in } \Omega,$$

where $\mathbf{a}(u) = \frac{\partial \mathbf{f}}{\partial u}$ is the flux Jacobian (in the scalar case often referred to as the advection velocity). Denoting by $\vec{\mathbf{n}}_i$ the outward-pointing unit normal vector to edge e_i (opposite ith vertex, illustrated in Figure 2.3), and noting that:

$$\nabla \psi_i \;=\; -\frac{\vec{\mathbf{n}}_i}{2|E|}\, |e_i| \qquad \forall i \in E,$$

it follows that for *constant in space* advection velocities the signals in (2.9) can be rewritten as:

$$\begin{aligned}\int_E \nabla \cdot \mathbf{f}(u_h)\, \psi_i \, d\Omega \;=\; \int_E \mathbf{a} \cdot \nabla u_h \, \psi_i \, d\Omega \;&=\; -\sum_{j \in E} \int_E \mathbf{a} \cdot u_j \frac{\vec{\mathbf{n}}_j}{2|E|} \, |e_j| \, \psi_i \, d\Omega \\ &= -\sum_{j \in E} \left(\mathbf{a} \cdot u_j \frac{\vec{\mathbf{n}}_j}{2|E|} \, |e_j| \right) \int_E \psi_i \, d\Omega \\ &= -\sum_{j \in E} \left(\mathbf{a} \cdot u_j \frac{\vec{\mathbf{n}}_j}{2|E|} \, |e_j| \right) \frac{1}{3}|E| \;=\; \frac{1}{3} \int_E \mathbf{a} \cdot \nabla u_h \, d\Omega.\end{aligned}$$

This means that in the case of the constant advection equation, the finite element approximation of (2.2) is a $\mathcal{RD}$-type method for which:

$$\beta_i = \frac{1}{3} \qquad \forall i.$$

Defining a distribution for which $\beta_i = \frac{1}{3}$ regardless what the discretized equation is gives the FE scheme. To be more precise, for this new method β_i is *always* set to $\frac{1}{3}$. Note that the FE and $\mathcal{FE}$ methods are two distinct discretizations. $\mathcal{FE}$ is used to denote the finite element method, and FE is a particular residual distribution scheme which was derived from the $\mathcal{FE}$ method. Obviously, the FE scheme and the finite element approximation, $\mathcal{FE}$, are identical in the case of the constant advection equation. Another feature that both approaches have in common is that for all mesh cells E, both the $\mathcal{FE}$ and the FE schemes send signals to all vertices of E, no matter what the direction of the flow is. Such methods are usually referred to as central (as opposed to upwind methods discussed in the following section).

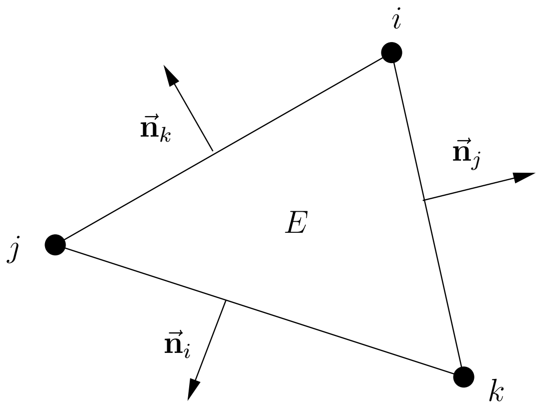


Figure 2.3: A generic cell E and unit outward pointing normal vectors associated with its sides.

It should be pointed out that the residual distribution framework was not derived from the $\mathcal{FE}$ approach and the above discussion should be treated as an observation, rather than an overview of the history of the $\mathcal{RD}$ framework. As a matter of fact, it was not until 1995 [20] that this close link between both frameworks was discovered.

The invention of the $\mathcal{RD}$ framework was driven by the desire to construct a scheme with all the properties that an optimal method for hyperbolic problems should have. These properties and ways of imposing them are the subject of the next section.

2.4 Design Principles

The procedure outlined in Section 2.2 defines a framework rather than a particular scheme. To construct a particular method within that framework, the distribution coefficients β_i have to be specified. These should be designed with care as otherwise the resulting scheme may exhibit poor stability, give inaccurate solutions or not converge to the solution at all. This section is concerned with the properties ideally every scheme solving hyperbolic problems should satisfy and which are to guarantee efficiency, accuracy and robustness. Alongside, restrictions on the distribution coefficients to impose these properties are given.

Conservation guarantees that the discrete Rankine-Hugoniot condition [48] is satisfied. It can be imposed by choosing the distribution coefficients so that:

$$\sum_{i \in E} \beta_i^E \;=\; 1 \qquad \forall E \in \mathcal{T}_h, \tag{2.10}$$

which was briefly discussed in Section 2.2. It guarantees that:

$$\sum_{E \in \mathcal{T}_h} \sum_{i \in E} \beta_i \, \phi^E \;=\; \sum_{E \in \mathcal{T}_h} \phi^E \;=\; \int_\Omega \nabla \cdot \mathbf{f}(u_h) \, d\Omega \;=\; \oint_{\partial\Omega} \mathbf{f}(u_h) \cdot \vec{\mathbf{n}} \, d\Gamma. \tag{2.11}$$

The above means that the information/mass within the discrete system can only appear/disappear through the boundary terms. In practical computations it ensures that discontinuities are captured correctly. This is crucial as, in general, hyperbolic PDEs do exhibit discontinuous solutions. In particular, non-linear equations with shocks.

Positivity means that every new value u_i^{n+1} can be written as a convex combination of old values, i.e.

$$u_i^{n+1} = \sum_k c_k \, u_k^n \tag{2.12}$$

with

$$c_k \geq 0 \quad \text{and} \quad \sum_k c_k \,=\, 1. \tag{2.13}$$

This guarantees that the scheme satisfies a maximum principle which prohibits the occurrence of new extrema in the solution (see [83] and references therein for a thorough discussion). In particular, the resulting numerical approximations are free of unphysical oscillations even in the vicinity of sharp changes in the solution. Positive scheme are also referred to as non-oscillatory.

Linearity preserving schemes are characterized by the ability to preserve exactly steady state solutions whenever these are linear functions in space. This condition is satisfied if and only if (cf. Lemma 1.6.1 in [41]) there exists a constant $C \in \mathbb{R}$ such that

$$\beta_{i,E} \;\leq\; C \qquad \forall E \in \mathcal{T}_h \quad \forall i \in E \tag{2.14}$$

for ϕ^E tending to zero. It can be shown that for residuals calculated from piece-wise linear polynomials, a linearity preserving scheme is second order accurate [1, 13], it is thus an accuracy requirement.

Continuity of the distribution coefficients with respect to both the numerical solution and the advection velocity is also desirable as otherwise the scheme may exhibit limit cycling and not converge to the solution. Nonlinear schemes are particularly sensitive in this respect.

Multidimensional upwinding not only facilitates construction of positive schemes but is also used for physical realism. A scheme is considered to be multidimensional upwind if no signals are sent to the upstream nodes of the cell. In one dimension it is a rather obvious restriction as there are only two directions and only one of them can be upstream. However, in the multidimensional setting the information can travel in infinitely many directions and imposing upwinding becomes very tricky and challenging. Schemes which are not multidimensional upwind, such as the FE scheme, are referred to as central schemes. Multidimensional upwind schemes will also be referred to as upwind schemes.

Note that construction of multidimensional upwind algorithms is somehow simplified. As illustrated in Figure 2.4, each mesh triangle E can have only one (the *one-target case*) or two (the *two-target case*) downstream vertices. In the one target case an upwind scheme will send all the information to the only downstream node, i.e. (notation as in Figure 2.4):

$$\beta_i \,=\, 1, \qquad \beta_j \,=\, 0, \qquad \beta_k \,=\, 0.$$

The two-target case is somewhat more involved as one needs to decide what fraction of the cell residual to send to each of the two downstream nodes. This will be addressed in Section 2.6, where examples of multidimensional upwind methods are presented.

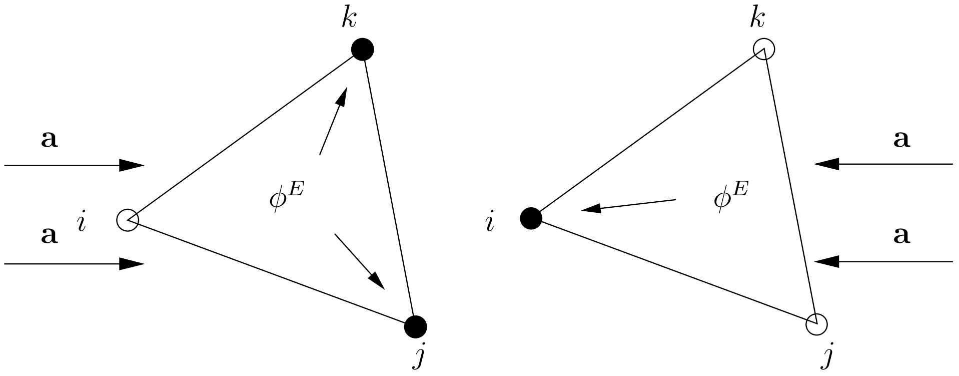


Figure 2.4: A triangle with two inflow sides (left) and one with one inflow side (right.)

Further distinction between particular residual distribution methods can be drawn by considering a slightly modified general framework (already considered while discussing positivity, cf. Formulation (2.12)):

$$u_i^{n+1} = \sum_k c_k \, u_k^n. \tag{2.15}$$

A scheme of this form is said to be *linear* if in the case of the linear advection equation all coefficients c_j are independent of the solution u_i^n. It will become clear in Section 2.6 that all $\mathcal{RD}$ methods can be rewritten in the general form (2.15) (not necessarily as a linear scheme, though). It will also turn out that from linearity of the distribution coefficients β_i (i.e. their independence from u) follows linearity of the scheme. Clearly, a linear scheme will be, in general, cheaper then a non-linear one. However, according to Godunov's theorem [49] (see also Theorem 3.15 in [38] for a similar result regarding residual distribution methods), a linear scheme cannot be non-oscillatory and second order accurate at the same time. Hence it is necessary to consider non-linear schemes to combine these two properties. Nevertheless, linear schemes are still of interest as in practice they are used as building blocks for non-linear methods which exhibit all the desired properties.

2.5 Non-linear Equations

Thus far, it has been assumed that the cell residual ϕ^E is computed exactly. It is a rather natural requirement but one has to realize that this may not be easy to achieve in practice. In particular, when the flux $\mathbf{f}(u_h)$ is a highly nonlinear function, not to mention systems of nonlinear equations. However, if the flux Jacobian $\mathbf{a}(u_h)$ is linear then the following holds:

$$\begin{aligned}\phi^E &= \int_E \nabla \cdot \mathbf{f}(u_h)\, d\Omega = \int_E \mathbf{a}(u_h) \cdot \nabla u_h \, d\Omega \\ &= \left(\int_E \mathbf{a}(u_h)\, d\Omega \right) \cdot \nabla u_h|_E = \overbrace{\frac{|E|}{3} \sum_{j \in E} \mathbf{a}(u_j)}^{\text{exact!}} \cdot \nabla u_h|_E.\end{aligned} \tag{2.16}$$

It gives a very straightforward and exact recipe to calculate the residuals. Moreover, it shows that in the case of a linear flux Jacobian the advection velocity can be *assumed to be constant* within each cell. Indeed, defining $\bar{u} = \frac{u_1+u_2+u_3}{3}$ one can write

$$\frac{|E|}{3} \sum_{j \in E} \mathbf{a}(u_j) = |E|\, \mathbf{a}(\bar{u}) \tag{2.17}$$

in which u_1, u_2 and u_3 are the nodal values of u_h in E. Formulation (2.17), together with (2.16) show that one can substitute $\mathbf{a}(\bar{u})$ instead of $\mathbf{a}(u_h)$ and that cell residuals will still be calculated exactly. More importantly, conservation of the scheme will also be preserved as:

$$|E|\, \mathbf{a}(\bar{u}) \cdot \nabla u_h|_E = \oint_{\partial E} \mathbf{f}(u_h) \cdot \vec{\mathbf{n}}\, d\Gamma.$$

Equipped with the above observation, one can proceed assuming that $\mathbf{a}(u_h)$ is constant within each cell.

Although the case of linear flux Jacobian may seem an oversimplified scenario, it covers two very important examples of scalar hyperbolic equations, namely the advection and Burgers' equations. Since these are the only scalar equations that will be considered in this thesis, no further discussion with regard to calculating the residuals will be carried out. More general equations and ways of computing cell residuals were investigated in [5] and [33]. For a brief overview consult [38]. Systems of equations will be treated separately in Chapter 6.

2.6 Examples of $\mathcal{RD}$ Schemes

In this section the most successful and frequently used linear and non-linear $\mathcal{RD}$ schemes for steady state problems are introduced. A brief discussion of each scheme with regard to the properties discussed in Section 2.4 is also given. In order to make the presentation more compact, extra notation will be now introduced.

The so-called *flow sensors* have been part of the $\mathcal{RD}$ nomenclature almost since the inception of the framework. They are used to define various methods within the framework and to determine the local behaviour of the flow. For each cell $E \in \mathcal{T}_h$ and vertex $i \in E$ these are defined as:

$$k_i \;=\; -\frac{\mathbf{a}(\bar{u}) \cdot \vec{\mathbf{n}}_i}{2}|e_i|, \qquad k_i^+ \;=\; \max(0, k_i), \qquad k_i^- \;=\; \min(0, k_i), \tag{2.18}$$

in which $\vec{\mathbf{n}}_i$, as in Figure 2.3, is the outward pointing unit normal vector to edge e_i. Note that, from the properties of the linear Lagrange basis functions and the form of the numerical solutions, the cell residual can be calculated exactly using:

$$\phi^E \;=\; \sum_{i \in E} k_i \, u_i. \tag{2.19}$$

This is true provided that the flux Jacobian is linear. No other scenario will be considered in this work.

Since the flow sensors (2.18) are linear with respect to the advection velocity and independent of the solution, one concludes from Formulae (2.19) and (2.8) that linearity of the distribution coefficients implies linearity of the scheme (cf. Formulation (2.15)).

In what follows, six distinct residual distribution methods are presented.

2.6.1 The Low Diffusion A (LDA) Scheme

The design process for *multidimensional upwind* schemes is simplified as only the two-target case has to be considered. A straightforward strategy can be derived by looking at a generic triangle with two downstream vertices. As drawn in Figure 2.5, the advection velocity $\mathbf{a}$ divides the cell into two sub-triangles: E_{124} and E_{143}. Defining the distribution coefficients as

$$\beta_3^{LDA} \;=\; \frac{|E_{124}|}{|E|}, \qquad \beta_2^{LDA} \;=\; \frac{|E_{143}|}{|E|}, \qquad \beta_1^{LDA} \;=\; 0,$$

gives the Low Diffusion A scheme of Roe [92], more often referred to as the LDA scheme. Quite naturally, the closer the advection vector gets to a particular node the bigger fraction of the cell residual that node receives. It can be deduced from basic trigonometric identities that [41]:

$$\beta_i^{LDA} = \frac{k_i^+}{\sum_{j \in E} k_j^+} \geq 0. \tag{2.20}$$

The distribution coefficients do not depend on the solution and hence the scheme is both linear and continuous. It is upwind by definition and linearity preserving as

$$\beta_i \leq 1 \qquad \text{for} \qquad i = 1, 2, 3.$$

Conservation follows immediately from (2.20). As a linear linearity preserving scheme it cannot be positive. On the other hand, it produces very low cross-diffusion which, as reported in [77], vanishes on regular grids.

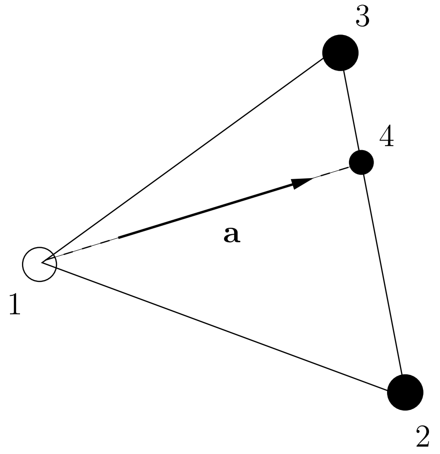


Figure 2.5: In the two-target case the advection velocity **a** divides the cell into two sub-triangles. Here cell E_{123} is split into triangles E_{143} and E_{124}.

2.6.2 The Narrow Scheme

Another very successful upwind scheme is the N scheme (N for narrow), also due to Roe [92]. As in the case of the LDA scheme, it can be derived based on purely geometrical considerations. First, observe that the cell residual, ϕ^E, can be decomposed as:

$$\phi^E(\mathbf{a}) = \int_E \mathbf{a} \cdot \nabla u_h \, d\Omega = \phi^E(\mathbf{a}_2) + \phi^E(\mathbf{a}_3)$$

for any vectors $\mathbf{a}_2$ and $\mathbf{a}_3$ such that

$$\mathbf{a}_2 + \mathbf{a}_3 \;=\; \mathbf{a}.$$

Taking $\mathbf{a}_2$ and $\mathbf{a}_3$ as in Figure 2.6 gives a distribution strategy defined by:

$$\beta_1^N \phi^E \;=\; 0, \qquad \beta_2^N \phi^E \;=\; \phi^E(\mathbf{a}_2), \qquad \beta_3^N \phi^E \;=\; \phi^E(\mathbf{a}_3). \tag{2.21}$$

No signal is sent to the upstream node 1 and hence this scheme is upwind. The distribution coefficients sum up to one and hence:

$$\beta_1^N \phi^E \;+\; \beta_2^N \phi^E \;+\; \beta_3^N \phi^E \;=\; \phi^E$$

which guarantees that the scheme is conservative. There is no explicit formula for the distribution coefficients, but since the decomposition of $\mathbf{a}$ into its components $\mathbf{a}_2$ and $\mathbf{a}_3$ is linear and continuous (with respect to the advection velocity and the solution), so is the N scheme. It is positive under a CFL-type restriction [38]:

$$\Delta t \leq \frac{|S_i|}{\sum_{E \in \mathcal{D}_i} k_i^+}, \qquad \forall i \in \mathcal{T}_h. \tag{2.22}$$

As a linear positive scheme it cannot be linearity preserving, but as far as first-order schemes are concerned the N scheme is one of the most successful ones. This was discussed in more detail in reference [100] where the authors show that among linear positive schemes the N scheme allows the largest time-step and has the smallest cross diffusion.

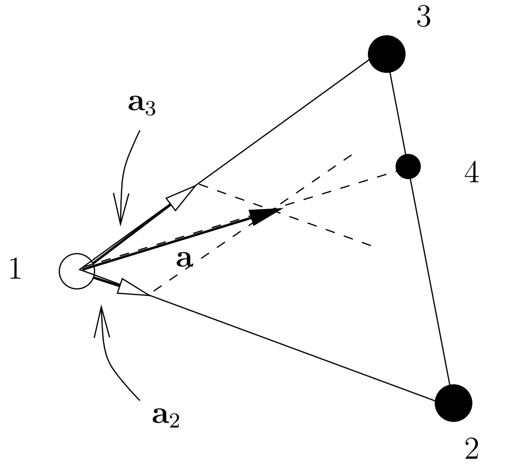


Figure 2.6: The advection velocity $\mathbf{a}$ can be decomposed into vectors parallel with the sides of the triangle pointing from upstream to downstream vertices. Above, $\mathbf{a}$ is decomposed into $\mathbf{a_2}$ and $\mathbf{a_3}$.

2.6.3 The BLEND Scheme

Desire to construct methods which are simultaneously linearity preserving and positive brings the need to consider non-linear distributions. A very robust scheme can be obtained by *blending* the two linear schemes presented so far, namely the N and the LDA schemes. Defining signals as:

$$\phi_i^E \;=\; (1-\theta(u_h))\,\phi_i^{LDA} \;+\; \theta(u_h)\phi_i^N$$

in which $\theta(u_h)$ is a blending coefficient, gives rise to the so called Blended scheme, hereafter referred to as the BLEND scheme.

Even though the idea is quite simple, specifying $\theta(u_h)$ rigorously is not obvious at all. Fortunately, the heuristic definition of Deconinck and collaborators [7]:

$$\theta(u_h) \;=\; \frac{|\phi^E|}{\sum_{j\in E}|\phi_j^N|} \in [0,1] \tag{2.23}$$

proved to give good results in a number of applications (see [11, 30, 96] or [38] and references therein). Numerical results show that the resulting scheme is nearly positive (small or very small overshoots and undershoots are usually present) and exhibits accuracy of order 2. However, as reported in [51] and [83], from theoretical point of view this scheme is not sound. Its heuristic construction complicates formal analysis and positivity has yet to be ensured. Since both the N and the LDA schemes are multidimensional upwind, conservative and continuous, so is the BLEND scheme.

One should bear in mind that the blending parameter is yet another degree of freedom that has to be taken into account when implementing the BLEND scheme. Definition (2.23) gave good results when applied to model problems, but may need tuning when used in practical computations.

2.6.4 The PSI Scheme

The most successful non-linear scheme is the PSI scheme of Struijs [99]. It is often referred to as the limited N scheme as its distribution coefficients are constructed by limiting those of the N scheme:

$$\beta_i^{PSI} \;=\; \frac{(\beta_i^N)^+}{\sum_{j\in E}(\beta_j^N)^+},$$

in which β_i^N, $i = 1, 2, 3$, are computed using (2.21). It is immediate to see that:

$$\beta_i^{PSI} \geq 0 \qquad \text{and} \qquad \sum_{i \in E} \beta_i^{PSI} \; = \; 1 \qquad \text{for} \quad i = 1, 2, 3.$$

The scheme can therefore be claimed to be linearity preserving and conservative. Being derived from the positive N scheme it is guaranteed to produce numerical approximations free of spurious oscillations (see Section 3.6.7 in [38] for a thorough mathematical justification). Multidimensional upwinding and continuity follow immediately as well.

In a number of references, see for example [12, 77, 93, 100], it was reported that for the steady scalar advection equation, especially on unstructured meshes, the PSI scheme performs better than standard second order limited finite volume methods. Its disadvantages when compared to the linear schemes are the difficulty with which it can be generalised to time-dependent problems and nonlinear systems of equations and the slower convergence to the steady state it exhibits [2]. However, being completely parameter free, it is a potential alternative to finite element methods with stabilizing terms [20, 77].

2.6.5 The Lax-Friedrichs (LF) Scheme

To the author's best knowledge, it was Abgrall [2] who first considered the Lax-Friedrichs scheme in the context of the residual distribution framework. It is a heuristic generalization of its well-studied and popular one-dimensional counterpart and reads:

$$\phi_i^{LF} = \frac{1}{3}\left(\phi^E + \alpha_{LF} \left[\sum_{j \in E} (u_i - u_j) \right] \right), \tag{2.24}$$

where α_{LF} is the Lax-Friedrichs dissipation coefficient. The scheme can be shown to be positive provided that [3]:

$$\alpha_{LF} \geq \max_{j \in E} |k_j|.$$

Since it is linear, it can only be first order accurate. It is conservative as

$$\sum_{i \in E} \phi_i^{LF} = \phi^E,$$

but not upwind since all the vertices receive signals regardless of the direction of the flow.

A natural and hypothetically linearity preserving extension of the Lax-Friedrichs scheme can be achieved by limiting its coefficients so that the Limited Lax-Friedrichs (LLF) scheme is constructed

$$\beta_i^{LLF} = \frac{(\beta_i^{LF})^+}{\sum_{j\in E}(\beta_j^{LF})^+},$$

where $\beta_i^{LF} = \frac{\phi_i^{LF}}{\phi^E}$. A similar procedure, when applied to the N scheme, gave the very successful second order and positive PSI scheme. Unfortunately, in this case the base scheme is not a multidimensional upwind distribution and the LLF scheme exhibits some problems with iterative convergence which spoil the order of accuracy and often introduce *wiggles* into the solution. This is observed regardless the value of the CFL number. According to Abgrall [2] this is due to '*the possible existence of spurious nodes*'. To cure that a stabilizing term has to be added which in turn spoils the formal monotonicity. For a full discussion on this matter the reader is referred to [2].

In this thesis the LF scheme is considered mainly to test its performance in the discontinuous setting (introduced in Chapter 3) and to compare it against other methods. This has not yet been done in the literature.

2.6.6 The Streamline Upwind (SU) Scheme

Although the FE distribution discussed in Section 2.3 is linearity preserving and conservative, it is very unstable and hence never used in practice. As reported in [38], introducing an upwind bias helps to stabilize the scheme. Such a bias, inspired by the close link between the $\mathcal{RD}$ and $\mathcal{FE}$ frameworks (in particular the Streamline Upwind Petrov Galerkin approach [18, 62, 64]), added to the FE scheme gives the SU distribution defined as:

$$\beta_i^{SU} = \frac{1}{3} + k_i\tau, \tag{2.25}$$

in which τ is a scaling parameter, taken here as

$$\tau = \left(\sum_{j\in E}|k_j|\right)^{-1}.$$

Conservation comes from the fact that in each cell E the flow sensors k_i sum up to 0. Linearity and linearity preservation follow immediately.

The derivation of this scheme is based on the similarity between the $\mathcal{RD}$ and

SUPG-type methods for the constant advection equation, shown for instance in [38]. Heuristic extension to a general case gives (2.25). In this respect it is very similar to the LF residual distribution method (2.24) that was also inspired by other algorithms known previously from different frameworks.

2.7 Numerical Results

To illustrate the properties exhibited by each scheme described in this chapter, a brief summary of the numerical results is given. For a very thorough and extensive numerical study of the N, LDA, PSI and BLEND schemes refer to the PhD thesis of Paillere [77] or Struijs [99]. The LF scheme was very rigorously investigated by Abgrall in [2] and for the SU scheme consult [18].

To perform the experiments the semi-circular linear advection equation, given by:

$$(y, -x) \cdot \nabla u = 0 \qquad \text{on} \quad \Omega = [-1, 1] \times [0, 1],$$

was used. Two distinct inflow boundary conditions were considered, each defining a separate test case.

Test Case A: To test for positivity and see how a scheme behaves in the vicinity of sharp changes in the solution, discontinuous inflow conditions were used:

$$u(x, y) \;=\; \begin{cases} 1 & \text{for} \qquad x \in [-0.5, -0.1],\; y = 0 \\ 0 & \text{otherwise.} \end{cases}$$

The square wave profile should be advected in a circular arc without change of shape and the exact solution is given by

$$u(x, y) \;=\; \begin{cases} 1 & \text{for} \qquad r = \sqrt{x^2 + y^2} \;\in [0.1, 0.5], \\ 0 & \text{otherwise.} \end{cases}$$

Test Case B: To carry out accuracy tests and check how quickly the steady state is obtained, smooth initial conditions were prescribed:

$$u(x, y) = \begin{cases} G(x) & \text{for} \qquad x \in [-0.75, -0.25],\; y = 0 \\ 0 & \text{otherwise.} \end{cases}$$

in which

$$G(x) = \begin{cases} g(4x+3) & \text{for} \quad x \in [-0.75, -0.5], \\ g(-4x-1) & \text{for} \quad x \in (-0.5, -0.25] \end{cases}$$

where

$$g(s) = s^5(70s^4 - 315s^3 + 540s^2 - 420s + 126). \tag{2.26}$$

The exact solution to this problem is

$$u(x,y) = \begin{cases} G(r) & \text{for} \qquad r = \sqrt{x^2+y^2} \ \in [0.25, -0.75], \\ 0 & \text{otherwise.} \end{cases}$$

No boundary conditions were imposed on the outflow boundaries. In each case the initial conditions used in the interior and on the outflow boundary were $u \equiv 0$. The time-step in (2.8) was computed as (cf. the positivity restriction for (2.22)):

$$\Delta t_i = \text{CFL} \frac{|S_i|}{\sum_{E \in \mathcal{D}_i} k_i^+} \qquad \forall i \in \mathcal{T}_h,$$

i.e. local time-stepping was implemented. The CFL number was set to 0.9 for most of the schemes except for the LF method for which it was decreased to 0.6. This was necessary as otherwise the method did not converge and the numerical solution *exploded.* The topology of the used meshes is shown in Figure 2.7.

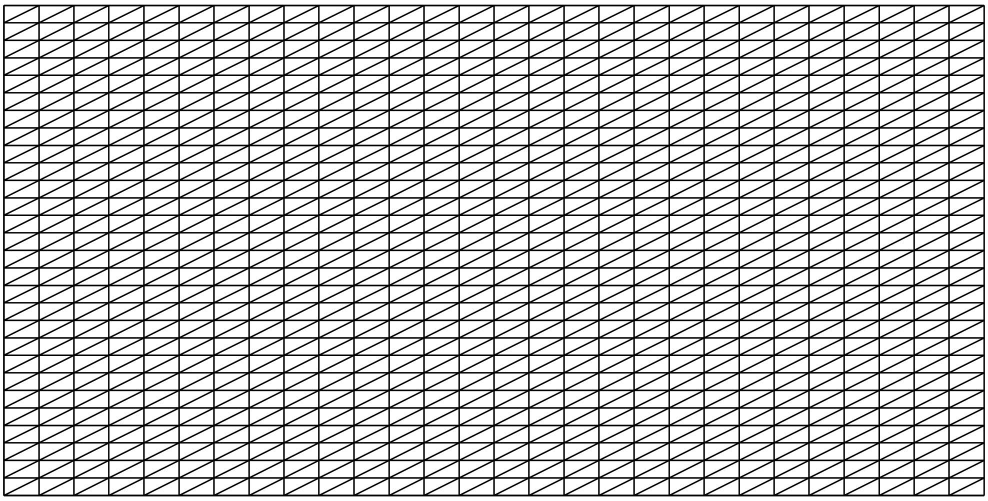


Figure 2.7: Topology of the meshes used in the numerical tests carried out in this chapter.

Figures 2.8- 2.13 show the steady state solutions for Test Case B using six schemes described earlier in this chapter. As expected, the N and the LF schemes give the most diffusive results since neither is linearity preserving. The LDA and the SU schemes are the least diffusive schemes, but at the expense of large oscillations. Finally, the BLEND and the PSI schemes gave best results with very little diffusion and no spurious oscillations. A regular triangulation of 57 × 57 grid and topology as in Figure 2.7 was used.

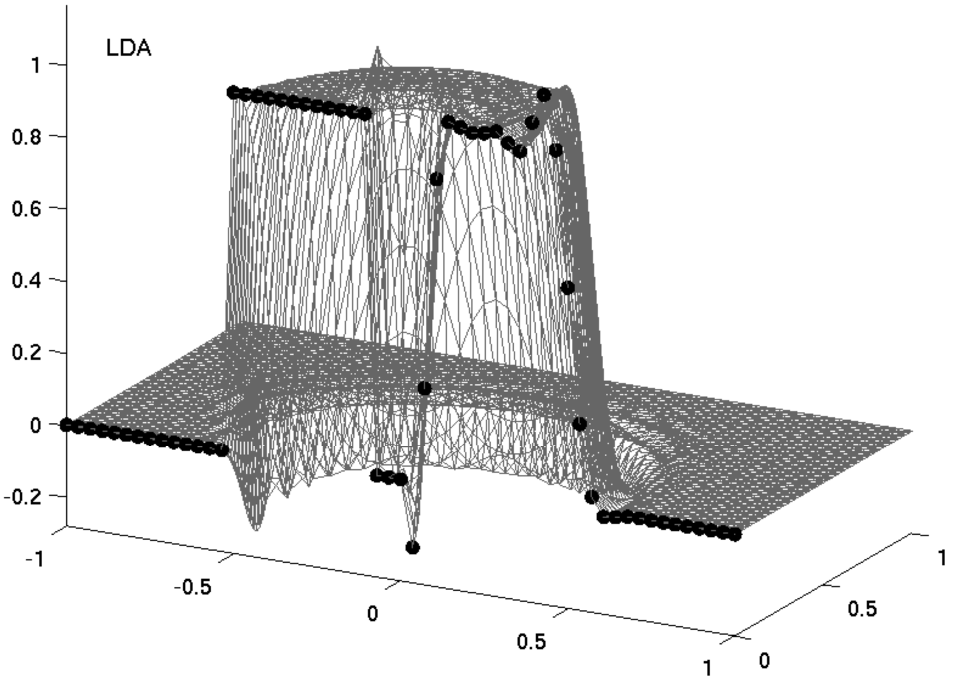


Figure 2.8: Solution for the LDA scheme for the Test Case A.

The convergence histories for the N, SU and the LF schemes are plotted on the left in Figure 2.14. Corresponding results for the LDA, PSI and BLEND methods are plotted on the right in the same figure. The convergence monitor which has been used is the root mean square (RMS) of the residual, at each time-step given as

$$\mathrm{RMS} = \sqrt{\frac{\sum_{i=1}^{N_n}(\phi_i)^2}{N_n}},$$

in which N_n is the total number of degrees of freedom (nodes). It can be seen that all schemes, apart from the LF method, converged rapidly. The CFL number for the LF scheme was lower than the one for other schemes and the scheme was expected to take longer to converge to the steady state. The scheme converged (the root mean square of the residuals reached machine precision) in roughly 6700 iterations,

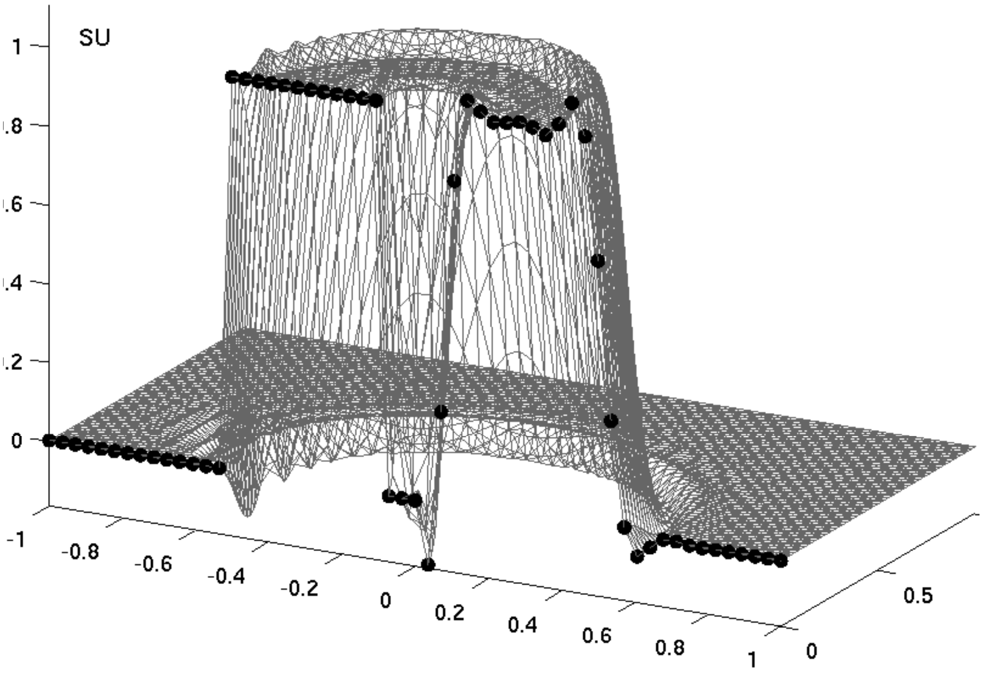


Figure 2.9: Solution for the SU scheme for the Test Case A.

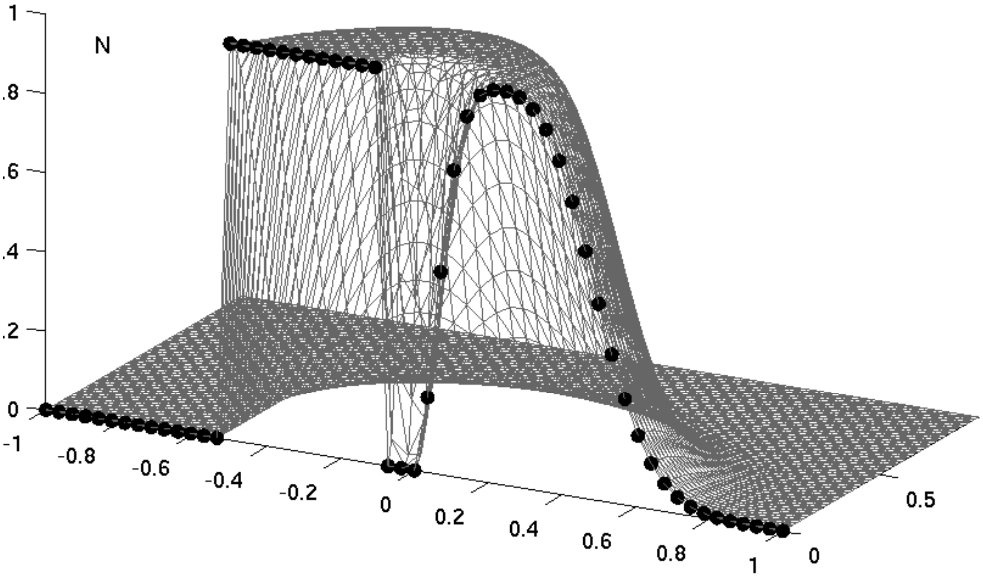


Figure 2.10: Solution for the N scheme for the Test Case A.

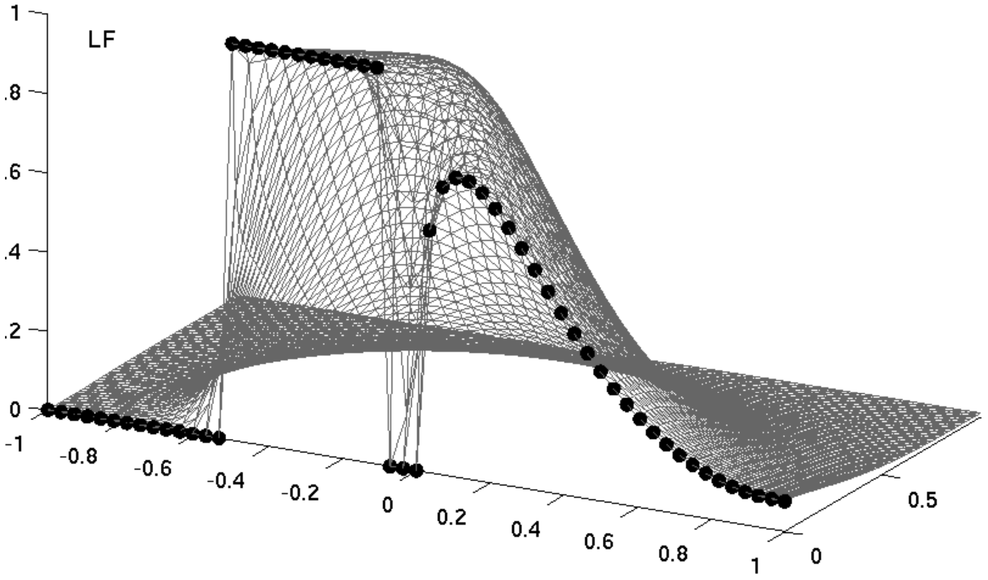


Figure 2.11: Solution for the LF scheme for the Test Case A.

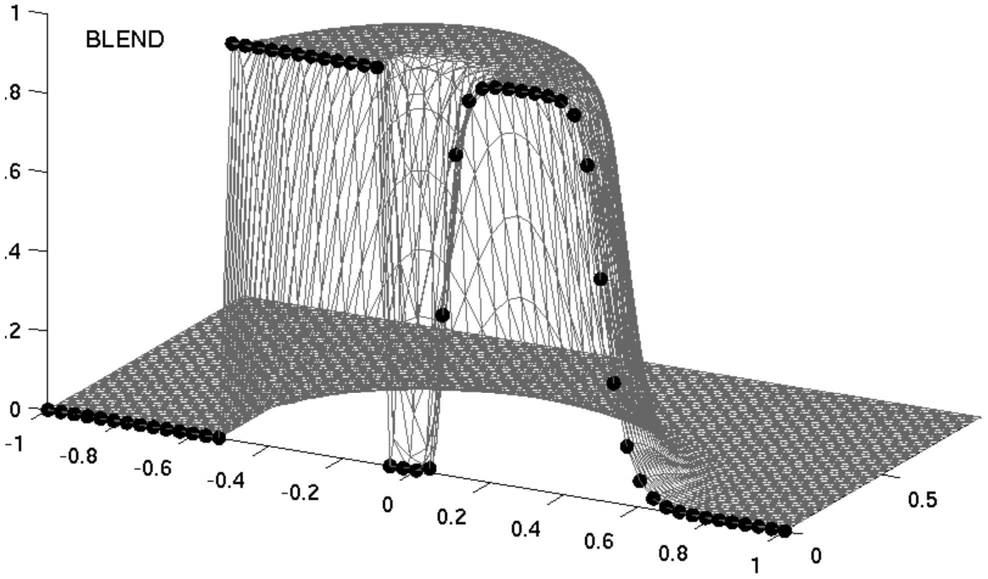


Figure 2.12: Solution for the BLEND scheme for the Test Case A.

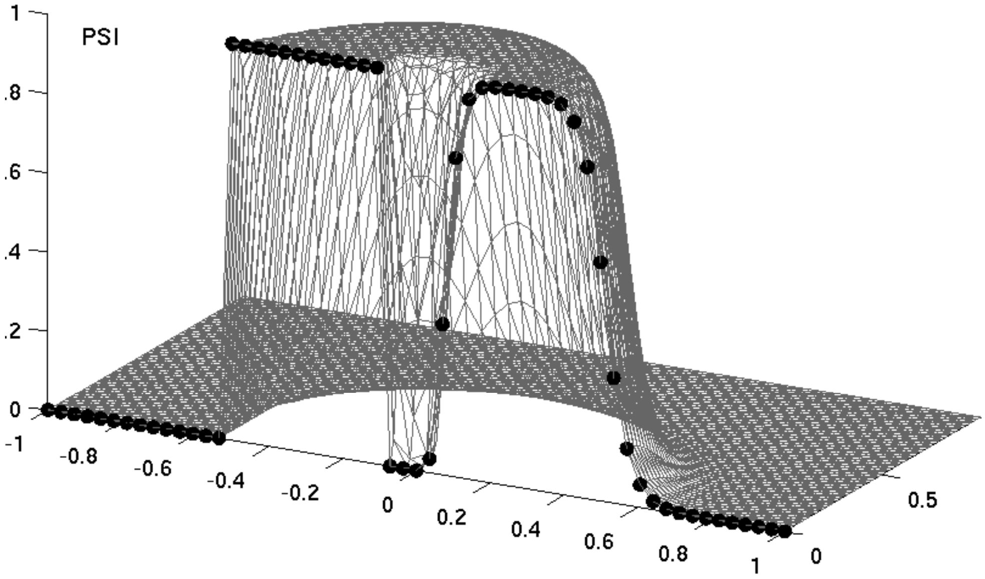


Figure 2.13: Solution for the PSI scheme for the Test Case A.

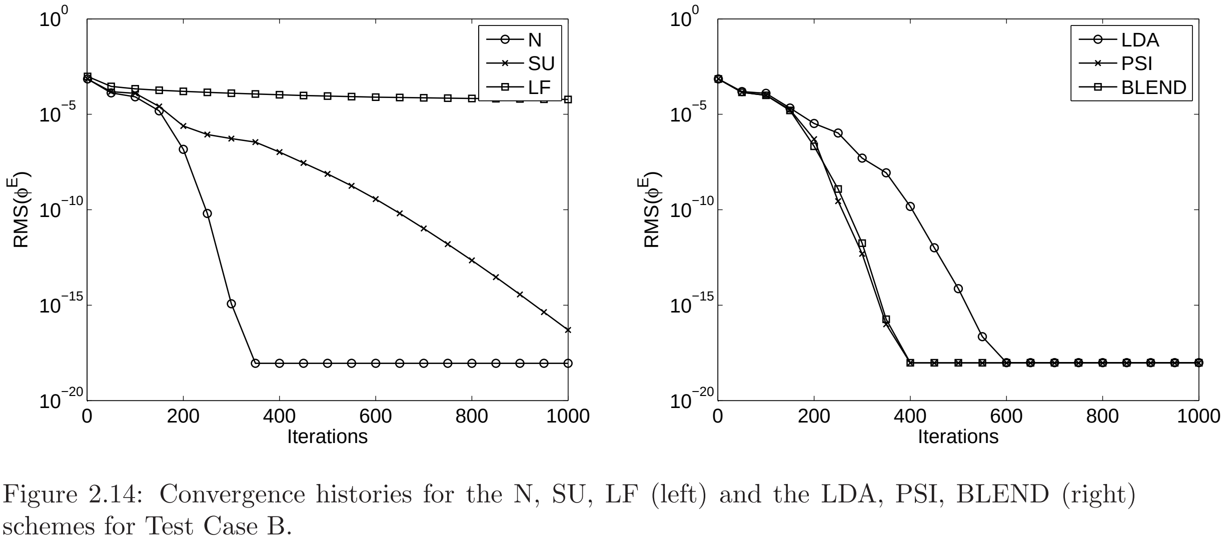


Figure 2.14: Convergence histories for the N, SU, LF (left) and the LDA, PSI, BLEND (right) schemes for Test Case B.

which is a rather poor result.

The mesh convergence analysis, results of which are plotted in Figure 2.15 (left for the N and the LF schemes, right for the LDA, PSI, BLEND and the SU schemes), was carried out on a set of regular triangular meshes with the coarsest mesh of a 14×14 regular grid refined 6 times by a factor 2 in each direction. The experiments confirmed that the LDA, SU, PSI and the BLEND schemes are second order accurate and that the N and LF schemes exhibit only first order convergence. As previously, the LF method gave the poorest results. The error was calculated using the root mean square of the nodal values of the difference between the exact and the numerical solution:

$$\text{L2 error} = \sqrt{\frac{\sum_{i=1}^{N_n}(u_i^{exact} - u_i^{approx})^2}{N_n}}. \tag{2.27}$$

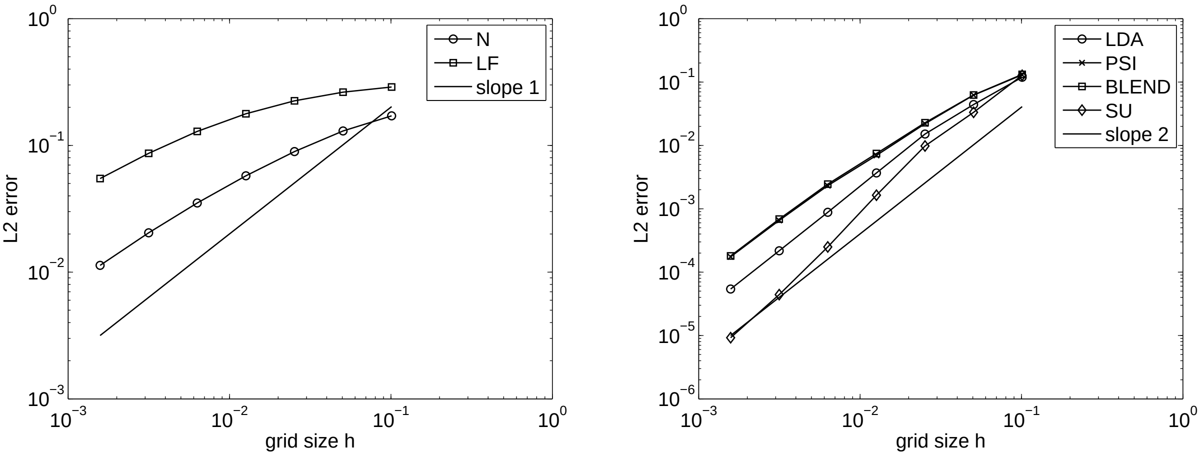


Figure 2.15: Mesh convergence for the N, LF (left) and the LDA, PSI, BLEND, SU (right) schemes for Test Case B. The PSI and BLEND schemes gave similar results which is reflected by the fact that the corresponding plots overlap each other.

2.8 Summary

In this chapter the continuous residual distribution framework was defined and 6 examples of schemes fitting into it were given. Properties of each scheme are dictated by the distribution coefficients and these, ideally, should be constructed following the design criteria discussed in Section 2.4. In order to show how these methods fit in between other widely used discretizations, a link between the residual distribution and finite element approximations was discussed.

The numerical results presented in the previous section confirmed that all of the schemes presented in Section 2.6 exhibit their theoretical properties. These properties are summarized in Table 2.1. As expected, the PSI is currently the best available residual distribution scheme as far as scalar hyperbolic PDEs are concerned. Although the BLEND scheme gave similar results, contrary to the PSI method it is not completely parameter-free and therefore a slightly less attractive alternative. The LF scheme, even though it is positive, demonstrated very poor convergence and accuracy and should not be considered in practice. It is, however, one of the few known $\mathcal{RD}$ schemes that has been extended to the discontinuous setting, and has never been compared with other choices. This will be addressed in the next chapter.

	Conservative	Upwind	Continuous	Linear	Positive	Linearity Preserving
LDA	✓	✓	✓	✓	×	✓
N	✓	✓	✓	✓	✓	×
PSI	✓	✓	✓	×	✓	✓
BLEND	✓	✓	✓	×	✓	✓
SU	✓	×	✓	✓	×	✓
LF	✓	×	✓	✓	✓	×

Table 2.1: Summary of the properties of the schemes presented in this chapter. A ✓represents success, while × indicates a short-coming in the method. Positivity of the BLEND scheme has not been proved formally yet.

Extension to non-linear equations was only briefly discussed. It will be covered in more detail in chapters on non-linear systems of equations and time-dependent problems where more challenging cases are considered.

Chapter 3

The Discontinuous $\mathcal{RD}$ Framework

3.1 Introduction

A continuous representation of u was assumed throughout the discussion in Chapter 2. Relaxing this constraint leads to a very active and promising strand of research within the community, i.e. the discontinuous residual distribution framework. Proposed simultaneously by Hubbard [57, 58] and Abgrall [3], this new concept aims at drawing together advantages of both residual distribution and Discontinuous Galerkin ($\mathcal{DG}$) approaches [22,37]. The numerical solution is now assumed to be only piecewise continuous and some sort of a 'numerical entity' has to be introduced to enable communication between the cells. In the case of $\mathcal{DG}$-type schemes such communication is imposed by introducing the numerical flux, whereas in the $\mathcal{RD}$ setting it is the edge residual that enables it. Formal definitions will be given in the following sections. This new approach, as in the case of discontinuous Galerkin approximations, facilitates construction of a localised system and a simple framework within which $h-$ and $p-$ adaptivity can be incorporated, features that are present neither in the $\mathcal{FV}$ or continuous $\mathcal{RD}$ frameworks. The concept of discontinuous residual distribution methods is relatively recent and unexplored and this chapter aims not only at introducing it, but also at reviewing, comparing and summarizing available results. As in the case of classical $\mathcal{RD}$ schemes, it was originally introduced for steady state problems and only such are considered in this chapter.

It is worth noting that also Abgrall and Shu considered discontinuous-in-space residual distribution schemes [14]. Their approach, however, is different from the one employed in this thesis. In their work the degrees of freedom are located at midpoints of the edges that connect the centroid of each element with its vertices. That choice, motivated by orthogonality of the resulting basis functions, enabled them to rewrite some $\mathcal{DG}$ methods in the $\mathcal{RD}$ framework and to apply stabilization techniques known from the latter to enforce an L^∞ stability of the former. Although interesting, their approach is fundamentally different from the one implemented in this work and will not be discussed here.

This chapter is structured as follows. First, extra notation and the framework of discontinuous residual distribution schemes is introduced. Next, its close relation with the discontinuous Galerkin approach is discussed, in particular it is shown that every $\mathcal{DG}$ method can be viewed as a special case of a discontinuous $\mathcal{RD}$ scheme. Section 3.4 outlines key properties, and the way of imposing them, ideally every discontinuous $\mathcal{RD}$ method should have. Nonlinear equations are briefly discussed in Section 3.5 and particular discontinuous $\mathcal{RD}$ methods are introduced in Section 3.6. Results of numerical experiments are presented and discussed in Section 3.7.

3.2 The Framework

The notation introduced in Chapter 2 remains mostly unchanged, only the numerical solution takes a slightly more general form now:

$$u_h(\mathbf{x})|_E \;=\; \sum_{i\in E} \psi_i^E(\mathbf{x}) u_i^E \qquad \forall \mathbf{x} \in E \quad \forall E \in \mathcal{T}_h. \tag{3.1}$$

u_i^E is the value of u_h at $\mathbf{x}_i$ taken in cell E and ψ_i^E is the linear Lagrange basis function associated with $\mathbf{x}_i$ (defined in Chapter 2, cf. Equation (2.5)).This definition reflects the fact that u_h is no longer assumed to be globally continuous and thus has to be considered separately in every cell. The superscript ^{E} will be omitted whenever the cell being considered is clear from the context.

So far only cell residuals have been used to construct the linear system. In the discontinuous setting edge residuals, denoted by ϕ^e, play an equally important role. These are defined by:

$$\phi^e(u_h) \;=\; -\int_e [\mathbf{f}(u_h)\cdot\mathbf{n}]\; d\Gamma, \tag{3.2}$$

in which $[\mathbf{f}(u_h)\cdot\mathbf{n}]$ represents the jump of the function $\mathbf{f}(u_h)\cdot\mathbf{n}$ across the edge, the sign of the difference being dictated by the direction chosen for $\mathbf{n}$, the unit normal vector to e. To be more precise:

$$\begin{aligned} -\phi^e(u_h) &= \int_e [\mathbf{f}\cdot\mathbf{n}]\,(u_h)\,d\Gamma = \int_e (\mathbf{f}_L\cdot\mathbf{n}_{E_L,e} + \mathbf{f}_R\cdot\mathbf{n}_{E_R,e})\ d\Gamma \\ &= \int_e (\mathbf{f}_L - \mathbf{f}_R)\cdot\mathbf{n}_{E_L,e}\,d\Gamma \ = \ \int_e (\mathbf{f}_R - \mathbf{f}_L)\cdot\mathbf{n}_{E_R,e}\,d\Gamma. \end{aligned}$$

The subscripts $_L$ and $_R$ mean that the value of a quantity was taken from E_L and E_R, respectively, the cells associated with edge e (see Figure 3.1). The normal vectors $\mathbf{n}_{E_L,e}$ and $\mathbf{n}_{E_R,e}$ are chosen to be unit length and pointing outward from the cell they are associated with. Obviously, ϕ^e is zero when u_h is assumed to be continuous across edge e.

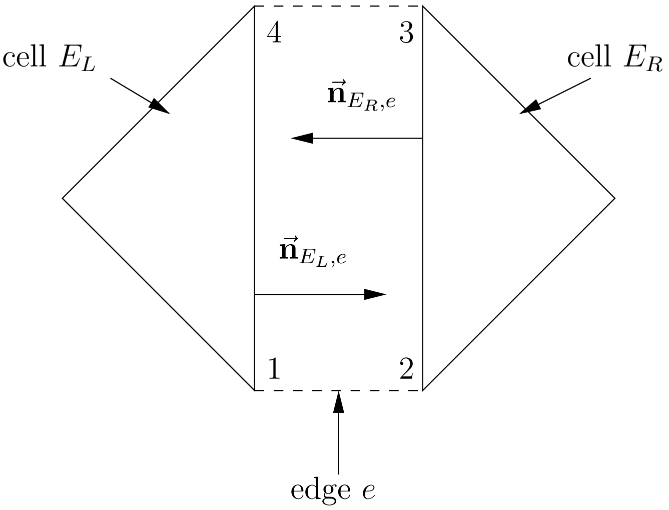


Figure 3.1: Edge e and the two cells associated with it: E_L and E_R.

Similar to continuous residual distribution methods, to find the numerical solution u_h one first assembles signals sent to each degree of freedom i and then solves the resulting linear system with the aid of pseudo time-stepping:

$$u_i^{n+1} \ = \ u_i^n \ - \ \frac{3\Delta t}{|E|}\left(\beta_i^E\phi^E \ + \ \alpha_i^{e_1}\phi^{e_1} \ + \ \alpha_i^{e_2}\phi^{e_2}\right) \qquad \forall i. \tag{3.3}$$

In analogy to cell residuals and the corresponding distribution coefficients, $\alpha_i^{e_1}$ and $\alpha_i^{e_2}$ are the distribution coefficients for the degree of freedom $i \in E$ corresponding to the edges $e_1 \in E$ and $e_2 \in E$, respectively, adjacent to vertex i. Note that in the discontinuous setting each degree of freedom belongs to only one cell and two of its edges and it seems natural to assume that it can receive signals only from the corresponding residuals. However, there is no clear reason why degree of freedom $i \in E$ should not receive signals from all of the edges of E :

$$u_i^{n+1} \;=\; u_i^n \;-\; \frac{3\Delta t}{|E|}\left(\beta_i\phi^E \;+\; \sum_{e\in E}\alpha_i\phi^e\right) \qquad \forall i. \tag{3.4}$$

This slight generalization of (3.3), and what Abgrall [3] and Hubbard [57] originally proposed, has not yet been considered in the literature. All the distribution strategies investigated so far are based on the simpler formulation (3.3). One possible method based on the form (3.4) is presented in Section 3.6. Additionally, in Appendix E a more compact version of the above general Framework (3.4) is presented.

As in the continuous case, the distribution coefficients determine properties of the scheme. Strategies for cell residuals were covered in Chapter 2 and no further examples will be considered here. Edge residuals are specific to the framework of discontinuous methods. All available techniques of distributing them are discussed in Section 3.6. First, however, one particular example will be discussed. As in the case of continuous residual distribution methods, the motivation for this is to show how the discontinuous $\mathcal{RD}$ framework fits in between other more frequently used methods.

3.3 Relation to Discontinuous Galerkin methods

Popularised by Cockburn and Shu in their series of papers [23–27], discontinuous Galerkin methods are among the most successful and popular ways of discretising hyperbolic equations. In this section the steady state variant of these methods is first introduced and then rewritten in the $\mathcal{RD}$ framework.

To construct an equation for u_i^E, multiply the steady state counterpart of (2.1) by the basis function ψ_i and integrate over E. Next, apply the Gauss-Green theorem to get:

$$-\int_E \mathbf{f}\cdot\nabla\psi_i\, d\Omega \;+\; \sum_{e\in E}\int_e \mathbf{f}\,\psi_i\cdot\mathbf{n}_{E,e}\, d\Gamma \;=\; 0$$

in which $\mathbf{n}_{E,e}$ is the outward pointing unit normal vector to edge e. Replacing u with u_h (and consequently $\mathbf{f} = \mathbf{f}(u)$ with $\mathbf{f}(u_h) = \mathbf{f}_h$) leads to a discrete formulation:

$$-\int_E \mathbf{f}_h\cdot\nabla\psi_i\, d\Omega \;+\; \sum_{e\in E}\int_e \mathbf{f}_h\,\psi_i\cdot\mathbf{n}_{E,e}\, d\Gamma \;=\; 0.$$

The desired equation is obtained by modifying the boundary integral by substituting the numerical flux $\hat{\mathbf{f}}_{E,e}$ (described in detail in the next paragraph) instead of $\mathbf{f}$. Repeating the procedure for all the degrees of freedom leads to the **weak form** of the $\mathcal{DG}$ discretization [54]:

$$-\int_E \mathbf{f}_h \cdot \nabla \psi_i \, d\Omega + \sum_{e \in E} \int_e \hat{\mathbf{f}}_{E,e} \, \psi_i \cdot \mathbf{n}_{E,e} \, d\Gamma \;=\; 0 \qquad \forall i \in \mathcal{T}_h. \tag{3.5}$$

Even though this is the most frequently used formulation, it is not of direct relevance here. This is primarily because it is not obvious how to fit it into the discontinuous residual distribution framework (cf. Scheme (3.4)). Instead, take each equation in (3.5) and once more integrate it by parts. This leads to the **strong form** of the $\mathcal{DG}$ discretization [54]:

$$\int_E \nabla \cdot \mathbf{f}_h \psi_i \, d\Omega + \sum_{e \in E} \int_e (\hat{\mathbf{f}}_{E,e} - \mathbf{f}_h) \, \psi_i \cdot \mathbf{n}_{E,e} \, d\Gamma \;=\; 0 \qquad \forall i \in \mathcal{T}_h. \tag{3.6}$$

Bear in mind that in order to enforce communication between cells, the numerical flux must remain in the discrete system and hence only the first term in (3.5) was integrated. The two formulations are mathematically equivalent.

The numerical flux is introduced to couple adjacent cells. For each cell E and edge $e \in E$ it is defined as a function of $u_h^{int(E)}$ and $u_h^{ext(E)}$ (see Figure 3.2 for the notation):

$$\hat{\mathbf{f}}_{E,e} = \hat{\mathbf{f}}(u_h^{int(E)}, u_h^{ext(E)})$$

in which superscripts $^{int(E)}$ and $^{ext(E)}$ mean that the value of the solution is taken from the interior and exterior, respectively, of cell E. The concept of the numerical flux comes originally from the finite volume framework. Three standard properties are assumed to be satisfied (discussed in detail in [48]):

A1. $\hat{\mathbf{f}}_{E,e} \;=\; \hat{\mathbf{f}}_{E',e}$ (conservation),

A2. $\hat{\mathbf{f}}(u, u) \;=\; \mathbf{f}_h(u)$ (consistency),

A3. $\hat{\mathbf{f}}$ is (globally) Lipschitz continuous (provided that $\mathbf{f}$ is Lipschitz continuous).

The first condition is to guarantee that the Rankine-Hugoniot condition is satisfied and hence discontinuities are captured accurately. Consistency and Lipschitz continuity are required for accuracy (see, for instance, the accuracy results in [25]). To learn more about numerical fluxes, their properties or a rigorous mathematical discussion on the above assumptions consult one of the standard text books on discontinuous Galerkin methods (for example [54] or [70]) or finite volume methods (e.g. [48, 69, 101] or [68]). Only two examples of numerical fluxes will be considered in this work, namely the upwind and the Lax-Friedrichs flux.

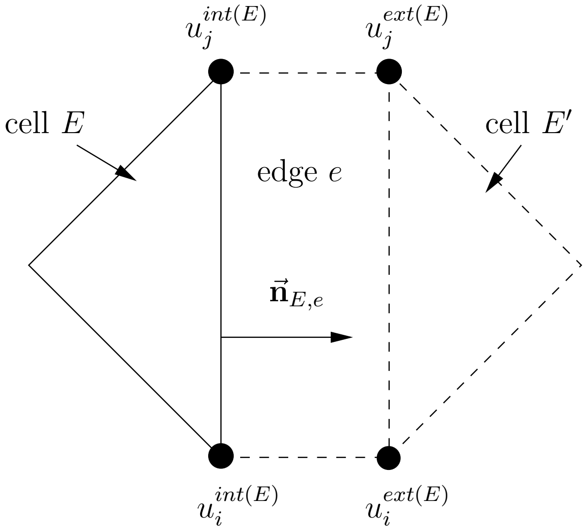


Figure 3.2: Cell E, its edge e, neighbouring cell E' and four degrees of freedom: $u_i^{int(E)}, u_j^{int(E)}, u_i^{ext(E)}$ and $u_j^{ext(E)}$, that are used to calculate the numerical flux $\hat{\mathbf{f}}_{E,e}$.

The upwind flux is a relatively simple, yet very popular and successful numerical flux. To understand how it works consider the situation from Figure 3.2 and observe that $\hat{\mathbf{f}}(u^{int(E)}, u^{ext(E)})$ depends on two values of u_h, each of which is taken from one or the other side of the edge. Only one of those values lies on the upstream side of the edge and the upwind flux is defined as the value of the analytical flux $\mathbf{f}$ at this value. Assuming that $u^{int(E)}$ is the upstream value would give:

$$\hat{\mathbf{f}}(u_h^{int(E)}, u_h^{ext(E)}) \;=\; \mathbf{f}(u_h^{int(E)}).$$

Quite clearly it is conservative as no matter which side of e is currently being considered, the upstream vertex remains the same:

$$\hat{\mathbf{f}}_{E,e} \;=\; \hat{\mathbf{f}}(u_h^{int(E)}, u_h^{ext(E)}) \;=\; \mathbf{f}(u_h^{int(E)}) \;=\; \hat{\mathbf{f}}(u_h^{ext(E)}, u_h^{int(E)}) \;=\; \hat{\mathbf{f}}_{E',e}.$$

Consistency and Lipschitz continuity (provided that $\mathbf{f}$ is Lipschitz) follow immediately.

The Lax-Friedrichs flux is defined as:

$$\hat{\mathbf{f}}_{E,e} \;=\; \frac{\mathbf{f}(u_h^{int(E)}) \,+\, \mathbf{f}(u_h^{ext(E)})}{2} \;+\; \frac{\alpha}{2}\mathbf{n}_{E,e}(u_h^{int(E)} \;-\; u_h^{ext(E)}) \tag{3.7}$$

where α is the local maximum of the directional flux Jacobian; that is,

$$\alpha \;=\; \max_{u_h \in \left[u_h^{int(E)}, u_h^{ext(E)}\right]} \left| n_x \frac{\partial f_1}{\partial u} \;+\; n_y \frac{\partial f_2}{\partial u} \right| ,$$

where $\mathbf{f} \;=\; (f_1, f_2)$ and $\mathbf{n}_{E,e} \;=\; (n_x, n_y)$. Also in this case assumptions A1-A3 are satisfied. Conservation can be shown by a direct substitution:

$$\begin{aligned}
\hat{\mathbf{f}}_{E,e} \;&=\; \frac{\mathbf{f}(u_h^{int(E)}) \,+\, \mathbf{f}(u_h^{ext(E)})}{2} \;+\; \frac{\alpha}{2}\mathbf{n}_{E,e}(u_h^{int(E)} - u_h^{ext(E)}) \\
&=\; \frac{\mathbf{f}(u_h^{ext(E)}) \,+\, \mathbf{f}(u_h^{int(E)})}{2} \;+\; \frac{\alpha}{2}\mathbf{n}_{E',e}(u_h^{ext(E)} - u_h^{int(E)}) \;=\; \hat{\mathbf{f}}_{E',e},
\end{aligned}$$

consistency is immediate and Lipschitz continuity is a consequence of $\mathbf{f}$ being Lipschitz continuous.

As in the case of continuous finite elements, it follows from the properties of the basis functions that:

$$\sum_{i \in E} \int_E \nabla \cdot \mathbf{f}(u_h)\, \psi_i \, d\Omega \;=\; \int_E \nabla \cdot \mathbf{f}(u_h)\, d\Omega \;=\; \phi^E,$$

which implies that

$$\int_E \nabla \cdot \mathbf{f}(u_h)\, \psi_i \, d\Omega \;=\; \beta_i^{\mathcal{DG}}\, \phi^E \qquad \text{and} \qquad \beta_i^{\mathcal{DG}} \;=\; \frac{\int_E \nabla \cdot \mathbf{f}(u_h)\, \psi_i \, d\Omega}{\int_E \nabla \cdot \mathbf{f}(u_h)\, d\Omega}.$$

More importantly, a similar observation can be made about the edge residuals. Indeed, assuming that edge e contains nodes i and j (cf. Figure 3.2) one shows that:

$$\begin{aligned}
&\sum_{k \in e} \int_e (\hat{\mathbf{f}} - \mathbf{f}_h)\psi_k \cdot \mathbf{n}\, d\Gamma \;= \\
&= \int_e (\hat{\mathbf{f}}_{E,e} - \mathbf{f}_h)\psi_j \cdot \mathbf{n}_{E,e}\, d\Gamma + \int_e (\hat{\mathbf{f}}_{E',e} - \mathbf{f}_h)\psi_j \cdot \mathbf{n}_{E',e}\, d\Gamma + \\
&+ \int_e (\hat{\mathbf{f}}_{E,e} - \mathbf{f}_h)\psi_i \cdot \mathbf{n}_{E,e}\, d\Gamma + \int_e (\hat{\mathbf{f}}_{E',e} - \mathbf{f}_h)\psi_i \cdot \mathbf{n}_{E',e}\, d\Gamma \\
&= \int_e [\mathbf{f}(u_h) \cdot \mathbf{n}] \;\, d\Gamma \;=\; \phi^e
\end{aligned}$$

which, again, shows that:

$$\int_e (\hat{\mathbf{f}} - \mathbf{f}_h)\psi_k \cdot \mathbf{n}\, d\Gamma \;=\; \alpha_k^{\mathcal{DG}}\, \phi^e \qquad \text{and} \qquad \alpha_k^{\mathcal{DG}} \;=\; \frac{\int_e (\hat{\mathbf{f}} - \mathbf{f}_h)\psi_k \cdot \mathbf{n}\, d\Gamma}{\int_e [\mathbf{f}(u_h) \cdot \mathbf{n}]\; d\Gamma}. \tag{3.8}$$

In other words, the method presented in this section is a discontinuous $\mathcal{RD}$ scheme. Interestingly enough, it fits more into the formulation (3.4) than the originally considered definition (3.3) as for every edge e and for every vertex $i \in E$ $(e \in E)$ the above formula specifies a signal that will be sent from edge e to vertex i (even if $i \notin e$). Note, however, that ψ_i vanishes on one of the edges of E and hence vertex i will receive signals from only 2 out of 3 edges of the cell. Nevertheless, Formulation (3.8) can accommodate more general scenarios in which different basis functions are used and in which every vertex indeed receives signals from all edges of the cell it belongs to. Such a situation will be considered in Section 3.6.

As pointed out in [79], the discontinuous Galerkin method has received relatively little attention in the steady state setting. In [80, 107] the method was combined with higher order time-stepping methods to converge to the solution. This allowed the authors to focus on subtle issues related to improving stability when applied to complex problems. Here, however, the focus is on testing and comparing various methods when applied to somewhat less involved model problems. The iterator in all cases is kept relatively simple, i.e. the forward Euler approach introduced in Section 2.2 is used, to isolate issues related to solving the linear system. The performance of $\mathcal{DG}$ discretisations in such a setting has yet to be compared against other methods, i.e. residual distribution, by running numerical experiments on a series of test problems. It is natural to stick with the same pseudo time-stepping so only the aforementioned forward Euler time-stepping will be considered in the context of steady $\mathcal{DG}$ methods. The numerical results are discussed in Section 3.7.

3.4 Design Principles

In the chapter on *continuous* residual distribution methods a guideline on how to design distribution strategies and coefficients was given. Those design criteria were dictated by the extensive theory on continuous methods. Unfortunately very little is known about *discontinuous* residual distribution methods. In [14] the authors proved a Lax-Wendroff-type theorem (convergence to the weak solution) and derived accuracy conditions, but, as mentioned in the introduction, they worked with a different discontinuous scheme. Although in [3] it is claimed that those results hold also for schemes of the form (3.3) or (3.4), no formal proofs are given to support that statement or to show that the two formulations are equivalent. Moreover, proofs in [14] utilize the underlying structure of the scheme, which, again, is different from what is investigated here. Work therefore needs to be done before one can, with full confidence, state that results obtained for one framework are automatically true for the other.

If there is no theory, what are the principles one should follow while designing the distribution coefficients? It is rather intuitive that cell residuals should be distributed following the guidelines outlined in Section 2.4. As a matter of fact, no new distribution strategies for cells have been proposed for the discontinuous schemes so far. There seems to be no demand for such. On the other hand, defining a distribution strategy for the edges remains an open problem. Both Hubbard [57] and Abgrall [3] proposed their own solutions. These are presented in Section 3.6. This section is an attempt to summarize and extend the heuristics that those algorithms are based on.

Conservation is defined as a straightforward extension of the corresponding concept for continuous schemes. For schemes in the general form (3.4) it is imposed by choosing the splitting so that, as in the continuous case, the cell distribution is conservative (cf. Condition (2.10) in Chapter 2) :

$$\sum_{i \in E} \beta_i^E \;=\; 1 \quad \forall E \in \mathcal{T}_h,$$

and, additionally, the edge coefficients satisfy the following condition:

$$\sum_{i \in e} \alpha_i^e \;=\; 1 \qquad \forall e \in \mathcal{T}_h. \tag{3.9}$$

This guarantees that:

$$\begin{aligned}\sum_{E \in \mathcal{T}_h} \left(\sum_{i \in E} \beta_i^E \, \phi^E + \sum_{e \in E \setminus \partial\Omega} \sum_{i \in e} \alpha_i^e \, \phi^e \right) &= \sum_{E \in \mathcal{T}_h} \phi^E + \sum_{e \in \mathcal{T}_h \setminus \partial\Omega} \phi^e \\ = \sum_{E \in \mathcal{T}_h} \oint_{\partial E} \mathbf{f}(u_h) \cdot \mathbf{n} \, d\Gamma \;-\; \sum_{e \in \mathcal{T}_h \setminus \partial\Omega} \int_e [\mathbf{f}(u_h) \cdot \mathbf{n}] \; d\Gamma &= \oint_{\partial\Omega} \mathbf{f}(u_h) \cdot \mathbf{n} \, d\Gamma.\end{aligned}$$

In other words, the information can only enter the domain through the boundary terms. Recall that conservation in the continuous case was necessary to assure that the discontinuities were captured accurately. It has yet to be assessed whether it is necessary and sufficient for a similar result to hold in the discontinuous setting. Splittings for edge-based residuals satisfying (3.9) are said to be conservative. Clearly, if both the decomposition for edge-based and cell-based residuals is conservative then the overall scheme is.

Positivity will also be considered in the same terms as in the continuous case. Assume that scheme (3.4) can be rewritten as:

$$u_i^{n+1} = \sum_k c_k\, u_k^n \qquad \forall i \tag{3.10}$$

in which the summation is carried out over all the degrees of freedom. The discontinuous $\mathcal{RD}$ scheme (3.4) is positive provided that

$$c_k \geq 0 \quad \text{and} \quad \sum_k c_k \,=\, 1. \tag{3.11}$$

This abstract formulation is identical to the one in the continuous case (cf. Equation (2.12)) and therefore reasoning that is usually used in the continuous case (see, for instance, Section 3.3 in [38]) also applies here. In other words, positive discontinuous $\mathcal{RD}$ schemes give solutions free of spurious oscillations. Note that the above states that if one takes a positive cell distribution and combines it with positive strategy for the edges then the resulting scheme will give oscillation-free solutions. Further remarks on positivity of particular discontinuous $\mathcal{RD}$ methods are made in Section 3.6.1 and Appendix D.

Linearity preservation in the continuous case is satisfied as long as the distribution coefficients for cells are bounded. Applying the reasoning from [41] (Lemma 1.6.1 and its proof) to the discontinuous scheme (3.4) gives exactly the same result with no additional effort. It means that a discontinuous residual distribution method is linearity preserving if and only if there exists a constant $C \in \mathbb{R}$ such that all the distribution coefficients can be uniformly bounded:

$$\beta_i^E \,\leq\, C \qquad \text{and} \qquad \alpha_i^e \leq C \qquad \forall E \in \mathcal{T}_h \qquad \forall e \in E, \tag{3.12}$$

for ϕ^E and ϕ^e tending to zero. Showing that this is sufficient for the scheme to be second order accurate is not that straightforward, but arguments from the continuous framework can be quite naturally applied to the current scenario by simply incorporating edge residuals into the original analysis from [1] and [13]. See [3] (and introduction to this section) for a further discussion. A splitting for edge-based residuals is said to be linearity preserving if it satisfies (3.12). As in the case of conservative and positive schemes, combining a linearity preserving distribution for both cell and edge-based residuals gives a linearity preserving scheme.

Continuity and **linearity** in this chapter are understood in the same sense as in the continuous case, i.e. scheme (3.4) is said to be linear/continuous if the distribution coefficients resulting from splitting the edge and cell based residuals are linear/continuous.

Upwinding has so far not been discussed in the discontinuous setting. Cell-based splitting is considered upwind if it satisfies the upwinding condition discussed in Section 2.4. Scheme (3.4) is considered upwind if the strategy employed to split cell-based residuals is upwind and on top of that the distribution strategy for the edges takes into account the direction of the flow.

To the author's best knowledge, the above is the first attempt to collect and specify design criteria one should follow while designing a distribution strategy for discontinuous residual distribution methods. As presented, they are a quite natural extension of the corresponding principles for continuous schemes. All of them fit into a general rule that the discontinuous residual distribution scheme satisfies a certain property if both the cell- and edge-based distributions do. The theory that backs those criteria up can, in many cases, be derived by a natural generalization of similar results for the framework of continuous methods. Suggestions how this can be/was done were given and no further discussion on this matter will be carried out. The main focus here is on methods for time-dependent problems and this work is by no means an attempt to gather a complete theory for the steady discontinuous $\mathcal{RD}$ framework.

3.5 Nonlinear Equations

Evaluation of the edge residual given in (3.2) is a challenge in itself. One way of tackling it is to assume that there exists a conservative linearisation for the flux difference [88]. In such a case, for any two arbitrary values u_L and u_R (for instance the left and right-hand-side values of the numerical solution across any given edge e), the following holds (see pages 360–361 in [88]):

$$\mathbf{f}(u_R) - \mathbf{f}(u_L) = \underbrace{\frac{\mathbf{a}(u_R) + \mathbf{a}(u_L)}{2}}_{\tilde{\mathbf{a}}}(u_R - u_L). \tag{3.13}$$

The edge-based residual:

$$\phi^e \;=\; -\int_e [\mathbf{f}(u_h)\cdot\vec{\mathbf{n}}]\; d\Gamma,$$

can now be evaluated exactly, giving

$$\phi^e \;=\; \sum_{l=1}^{N_q} w_l\,\tilde{\mathbf{a}}_l\cdot\mathbf{n}_{E_R,e}\,[u_l], \tag{3.14}$$

in which N_q is the number of quadrature points used in integrating (3.2), w_l are the quadrature weights and $\tilde{\mathbf{a}}_l$ is the conservative averaged flux Jocobian:

$$\tilde{\mathbf{a}} = \frac{\mathbf{a}(u_{E_R}) + \mathbf{a}(u_{E_L})}{2}$$

evaluated at $\mathbf{x}_l$. The jump $[u_l]$ is consistent with the direction chosen for the normal vector, i.e.

$$[u_l] = (u_{E_L} - u_{E_R})(\mathbf{x}_l).$$

For all the equations considered in this work (the advection equation, Burgers' equation and the Euler equations), it has been assumed that the vector of variables with respect to which the underlying equations/systems are solved vary linearly within each mesh cell (and hence along each mesh edge) and that the flux, $\mathbf{f}$, is a polynomial function of u of order no higher than 2. For example, in the case of the advection equation one has that $\mathbf{f}(u) = \mathbf{a}\,u$ ($\mathbf{a}$ being linear in space) and in the case of the Burgers' equation the flux is given by $\mathbf{f}(u) = \left(\frac{u^2}{2}, \frac{u^2}{2}\right)$. In such a case Simpson's rule is accurate enough to integrate (3.2) exactly. It has yet to be investigated how to approach equations for which conservative linearization is not known. This issue, however, will not be raised in this thesis.

3.6 Examples of Edge Distributions

Probably the most important ingredient of every scheme that fits into the framework of discontinuous $\mathcal{RD}$ methods are distribution strategies for edge-based residuals. In this chapter all available splittings are briefly presented and discussed. The first two, the mED and LF schemes, are based on ideas and concepts coming directly from the $\mathcal{RD}$ framework. Both approaches are very faithful to the residual distribution concept and until recently have been the only known strategies. The last two splittings considered in this section, the DG and m1ED distributions, are inspired by the $\mathcal{DG}$ approach and its close relation to the discontinuous $\mathcal{RD}$ framework. Introduced originally in [105], these strategies were designed to improve the accuracy of discontinuous $\mathcal{RD}$ methods when applied to unsteady problems (more on this matter can be found in Chapter 5).

3.6.1 The mED scheme

Proposed by Hubbard in [57], the **mED** scheme was designed under the assumption that there exists a conservative linearisation for the flux difference (see Equation (3.13)). To ensure that edge residuals resulting from this splitting can be used as part of a positive scheme, Hubbard [57] evaluated (3.14) using the quadrature coefficients resulting from the Simpson's rule. He arrived at the following formulation (numbering as indicated on Figure 3.1):

$$\phi^e \;=\; \frac{1}{2}\,\hat{\mathbf{a}}_{12}\cdot\mathbf{n}_{E_R,e}(u_1-u_2)|e| \;+\; \frac{1}{2}\,\hat{\mathbf{a}}_{43}\cdot\mathbf{n}_{E_R,e}(u_4-u_3)|e|. \tag{3.15}$$

The $\tilde{\mathbf{a}}_{ij}$ are averaged values of the flux Jacobian defined as:

$$\hat{\mathbf{a}}_{12} \;=\; \frac{1}{3}\left(\mathbf{a}_1 \,+\, \mathbf{a}_2 \,+\, \frac{\mathbf{a}_3 \,+\, \mathbf{a}_4}{2}\right), \quad \hat{\mathbf{a}}_{43} \;=\; \frac{1}{3}\left(\mathbf{a}_3 \,+\, \mathbf{a}_4 \,+\, \frac{\mathbf{a}_1 \,+\, \mathbf{a}_2}{2}\right) \tag{3.16}$$

in which $\mathbf{a}_i$ $(i = 1,\ldots,4)$ are the values of $\mathbf{a}$ at the vertices of e and $|e|$ is the length of the edge. The definition of the mED scheme is now clear. For a generic edge e and its vertices $1, 2, 3$ and 4 (numbering as on Figure 3.1) it is given by the following split residuals:

$$\begin{aligned}
\phi_1^{mED} &= \frac{1}{2}\left[\hat{\mathbf{a}}_{12}\cdot\mathbf{n}_{E_R,e}\right]^+(u_1-u_2)|e| \;=\; \alpha_1\,\phi^e,\\
\phi_2^{mED} &= \frac{1}{2}\left[\hat{\mathbf{a}}_{12}\cdot\mathbf{n}_{E_R,e}\right]^-(u_1-u_2)|e| \;=\; \alpha_2\,\phi^e,\\
\phi_3^{mED} &= \frac{1}{2}\left[\hat{\mathbf{a}}_{43}\cdot\mathbf{n}_{E_R,e}\right]^-(u_4-u_3)|e| \;=\; \alpha_3\,\phi^e,\\
\phi_4^{mED} &= \frac{1}{2}\left[\hat{\mathbf{a}}_{43}\cdot\mathbf{n}_{E_R,e}\right]^+(u_4-u_3)|e| \;=\; \alpha_4\,\phi^e.
\end{aligned} \tag{3.17}$$

This distribution takes into account the direction of the flow and hence it is upwind. The distribution coefficients sum up to 1, i.e. $\alpha_1+\alpha_2+\alpha_3+\alpha_4 = 1$, which means it is conservative. As noted in [57], applying (3.17) to any continuous linear steady state leads to zero contributions from the edges, since $u_1 = u_2$ and $u_3 = u_4$. In other words, any continuous steady state will always be preserved by this distribution of edge-based residuals, so the overall scheme will be linearity preserving as long as the distribution of cell-based residuals is linearity preserving. The limit on the time-step guaranteeing positivity is given by (cf. Equation (40) in [57] and the derivation in Appendix D):

$$\Delta t \ \leq \ \frac{1}{3} \frac{|E|}{(k_i^E)^+ + (k_i^{e_1})^+ + (k_i^{e_2})^+} \qquad \forall E \in \mathcal{T}_h \quad \forall i \in E, \tag{3.18}$$

in which k_i^E is a flow sensor defined in Section 2.6 and k_i^e is its edge-based counterpart related to edge e and defined as $k_i^e = \frac{1}{2}\mathbf{a}_i \cdot \mathbf{n}_{E,e}|e|$. Edges e_1 and e_2 are adjacent to cell E and such that $i \in e_1 \cap e_2$. Note that the direction of the normal vector in the definition of k_i^e is outward from the cell vertex i belongs to. Linearity and continuity of the distribution follow from the properties of linearization (3.16).

It is worth pointing that originally this distribution was proposed without any specific name (see [57]). In [105], in order to distinguish it from other distributions, it was referred to as the mED distribution. This thesis remains faithful to that convention.

3.6.2 The LF Scheme

The (local) **Lax-Friedrichs** distribution for edges was proposed by Abgrall in [3] and is based on its counterpart for cells. It is defined as

$$\alpha_i \, \phi^e = \frac{\phi^e}{4} + \alpha^e(u_i - \bar{u}), \qquad i = 1, \ldots, 4, \tag{3.19}$$

with

$$\bar{u} = \frac{u_1 + u_2 + u_3 + u_4}{4},$$

where $u_1, u_2, u_3.u_4$ are the values of u_h at the vertices of e (notation as in Figure 3.1). This distribution is positive provided that the dissipation coefficient α^e satisfies the following inequality (consult references [2] and [3] for a proof):

$$\alpha^e \geq \max_{i \in e} |k_i^e|.$$

Conservation, linearity and continuity are immediate.This scheme is not upwind as regardless of the direction of the flow all degrees of freedom will receive signals. Although no theoretical results are known, numerical experiments show that this distribution leads to only first order accurate approximations. This should come as no surprise as the cell-based LF distribution has identical properties.

As reported in [3], limiting the distribution coefficients in (3.19) (as in Section 2.6.5) gives a scheme which is *formally* second order accurate. Numerical results show that this order is never achieved in practice which is very likely related to instabilities that were discussed with regard to the continuous LF scheme in [2].

3.6.3 The DG Scheme

Signals resulting from the **DG distribution** are simply the edge integrals appearing in the strong formulation of the discontinuous Galerkin approximation (3.6):

$$\begin{aligned}
\alpha_1^{DG}\,\phi^e &= \int_e \left(\hat{\mathbf{f}}_{E_L,e} - \mathbf{f}_h\right) \cdot \mathbf{n}_{E_L,e}\,\psi_1\,d\Gamma,\\
\alpha_2^{DG}\,\phi^e &= \int_e \left(\hat{\mathbf{f}}_{E_R,e} - \mathbf{f}_h\right) \cdot \mathbf{n}_{E_R,e}\,\psi_2\,d\Gamma,\\
\alpha_3^{DG}\,\phi^e &= \int_e \left(\hat{\mathbf{f}}_{E_R,e} - \mathbf{f}_h\right) \cdot \mathbf{n}_{E_R,e}\,\psi_3\,d\Gamma,\\
\alpha_4^{DG}\,\phi^e &= \int_e \left(\hat{\mathbf{f}}_{E_L,e} - \mathbf{f}_h\right) \cdot \mathbf{n}_{E_L,e}\,\psi_4\,d\Gamma,
\end{aligned} \tag{3.20}$$

in which $\hat{\mathbf{f}}_{E,e}$ is the numerical flux discussed in Section 3.3 and ψ_i are the Lagrange basis function associated with edge vertices. Now, since $\vec{\mathbf{n}}_{E_R,e} = -\vec{\mathbf{n}}_{E_L,e}$ (and the numerical flux is assumed to be conservative) it follows that:

$$\sum_{i\in e} \alpha_i^{DG} = 1.$$

Hence the DG distribution is conservative. Of course one has to specify $\hat{\mathbf{f}}_{E,e}$ before this distribution can be implemented. Two numerical fluxes introduced in Section 3.3 will be considered here. Applying the Lax-Friedrichs flux will give the DG-LF splitting and choosing the upwind flux will lead to the DG-upwind splitting. Numerical results show that in both cases the resulting scheme is second order accurate, but not positive. This splitting is upwind as the numerical flux takes into account the direction of the flow. It is also continuous and linear as the signals defined in (3.20) are continuous and linear with respect to the approximate solution and the advection velocity (flux Jacobian).

3.6.4 The m1ED Scheme

For every cell E consider *shifted* Lagrange linear basis functions $\psi_i^{\mathcal{RD}}$ defined as:

$$\psi_i^{\mathcal{RD}} = \psi_i + \beta_i^{\mathcal{RD}} - \frac{1}{3}, \quad i \in E,$$

where $\beta_i^{\mathcal{RD}}$ is the distribution coefficient resulting from the strategy applied to distribute cell-based residual $\phi^E(u_h)$. The motivation for introducing such basis function is given in Section 5.3.2. Note that $\beta_i^{\mathcal{RD}}$ is constant. The m1ED strategy for edge $e \in E$ is defined by taking (3.20) and substituting $\psi_i^{\mathcal{RD}}$ instead of ψ_i :

$$\alpha_i^{m1ED}\, \phi^e = \int_e \left(\hat{\mathbf{f}}_{E,e} - \mathbf{f}_h\right) \cdot \mathbf{n}_{E,e}\, \psi_i^{\mathcal{RD}}\, d\Gamma.$$

In the above expression i is a vertex of E. Note that $\psi_i^{\mathcal{RD}}$, contrary to ψ_i, does not vanish on any of the edges of E (unless $\beta_i = \frac{1}{3}$). This means that now every vertex i of E will receive a signal from side $e \in E$, regardless whether i belongs to e or not. Being consistent with the strategy applied to cell-based residuals makes it a very interesting alternative. It is conservative, i.e. all signals sent from edge e sum up to ϕ^e, since $\sum_{i\in E} \psi_i^{\mathcal{RD}} = 1$. Numerical results show that it is second order accurate, but not positive. As in the case of the DG splitting, two numerical fluxes will be considered: the Lax-Friedrichs flux (the m1ED-LF distribution) and the upwind flux (the m1ED-upwind splitting). Similarly to the DG scheme, it is both continuous and linear as the formula for the signals is. More comments on positivity are made in Section 5.3.3.

More on the motivation for using the modified test function $\psi_i^{\mathcal{RD}}$ and further similarities between the $\mathcal{RD}$ and $\mathcal{DG}$ frameworks (in particular the DG and m1ED distributions) can be found in Chapter 5 in which the framework of discontinuous-in-space schemes for time-dependent problems is discussed. Since the m1ED splitting was originally designed for time-dependent problems it seems natural to postpone further discussion on its derivation till Chapter 5.

3.7 Numerical Results

The number of numerical results presenting the performance of discontinuous $\mathcal{RD}$ schemes that can be found in the literature is rather limited. Hubbard in his two papers on discontinuous $\mathcal{RD}$ schemes [57,58] considered one edge distribution, namely the mED scheme, and experimented with it on an extensive set of test cases looking at accuracy, positivity and efficiency. Abgrall [3] also implemented only one edge-based distribution, the limited LF scheme, but presented a very narrow set of results (testing only positivity). Neither of them attempted to compare different distribution strategies for the edges or to examine differences/similarities between the $\mathcal{RD}$ and $\mathcal{DG}$ approaches. Presenting such a comparison is the main goal of this section.

The test cases and meshes used in this section are identical to those introduced in Section 2.7. The cell-based residuals were distributed with the aid of the PSI scheme, and to split edge residuals distribution strategies introduced in Section 3.6 were implemented. The distribution strategy for cells was kept fixed as the focus of interest in this chapter is different strategies for edges, not cells. As such, only edge-based splittings are mentioned in the results, graphs and tables. The only exception is the $\mathcal{DG}$ scheme which was also tested. To distinguish, the DG acronym was put in front of the edge distribution whenever the discontinuous Galerkin method was used instead of the PSI scheme to distribute cell-based residuals. The time-step in (3.4) was computed as (cf. the positivity condition (3.18)):

$$\Delta t_i = CFL \frac{1}{3} \frac{|E|}{(k_i^E)^+ + (k_i^{e_1})^+ + (k_i^{e_2})^+} \qquad \forall E \in \mathcal{T}_h \quad \forall i \in E.$$

Figures 3.3-3.8 show the steady state solutions for Test Case A obtained with the aid of six schemes described in Section 3.6 (on a regular triangulation of 57 × 57 grid and with topology shown in Figure 2.7). Results of a similar experiment, but obtained with the aid of the discontinuous Galerkin scheme (i.e. with the discontinuous Galerkin method used to distribute cell-based residuals) are presented in Figures 3.9-3.10. Interestingly enough, switching from the upwind to the Lax-Friedrichs flux does not make any noticeable differences. Although DG-upwind and DG-LF gave nice results (the solutions exhibit relatively small overshoot and undershoot), they are not completely free of spurious oscillations (see Tables 3.1 and 3.2). Still, these two schemes led to better results than the m1ED-upwind and m1ED-LF splittings. Only the mED and LF schemes gave genuinely positive results, the one given by the LF method being very diffusive. The $\mathcal{DG}$ method (presented in Section 3.3) gave results very similar to those obtained with the aid of the m1ED distribution (that is the PSI scheme applied to cell residuals and the m1ED distribution for edges). This was expected as the two schemes are very similar (i.e. based on similar integrals with only the test function being different - this is discussed in more detail in Chapter 5).

The convergence histories are plotted on Figure 3.11. Recall that in most cases the PSI scheme was used to distribute cell residuals and only the name of the distribution for edge-based residuals is given. When the acronym DG is given in the front of an edge distribution, e.g. DG-DG-upwind and DG-DG-LF, then the Discontinuous Galerkin rather than the PSI scheme was used to distribute cell residuals. Most schemes, apart from the m1ED-LF, converge rather rapidly, the m1ED-upwind, DG-DG-upwind, and DG-DG-LF giving the best performance. The m1ED-LF scheme never produced residuals smaller than 10^{-15} (the machine precision being 10^{-16}). Instead, it oscillated around that value with jumps smaller than 10^{-16}. However, with discretization errors for this test case at around 10^{-4} this is still a satisfactory result. A quick comparison with the results obtained for the continuous approach (see Figure 2.14) shows that the discontinuous $\mathcal{RD}$ framework is consistently slower in terms of number of iterations than its continuous counterpart. This is related to the more restrictive constraint on the pseudo-time-step required to impose positivity on iteration, cf. Eqs. (2.22) and (3.18). The CFL number for this problem was set to 0.9 for the mED, DG-upwind, m1ED-upwind and DG-DG-upwind schemes. In all other cases a CFL number equal to 0.3 was used. The LF scheme was particularly sensitive as $CFL = 0.1$ had to be used. No stability analysis is available for this distribution and the CFL number was found experimentally. All experiments were run on a regular triangulation of 57×57 grid.

Results of the mesh convergence analysis are plotted on Figure 3.12. As in all previous cases, switching from the upwind to the Lax-Friedrichs flux does not show any noticeable differences (though the results are not identical). As expected, only the LF scheme is first order accurate, all other schemes exhibiting second order convergence. The m1ED and $\mathcal{DG}$ discretizations gave similar results. In Chapter 5 it will be shown that, in some cases, these two schemes are in fact identical.

	exact	mED	LF	m1ED-upwind	m1ED-LF	DG-upwind	DG-LF
$min(u_h)$	0.0	$1e-18$	0.0	-0.018	-0.018	-0.000047	-0.000048
$max(u_h)$	1.0	1.0	1.0	1.023	1.023	1.0	1.0

Table 3.1: Minimum and maximum values of the solutions presented on Figures 3.3-3.8.

	exact	DG-DG-upwind	DG-DG-LF
$min(u_h)$	0.0	-0.018	-0.018
$max(u_h)$	1.0	1.02	1.02

Table 3.2: Minimum and maximum values of the solutions presented on Figures 3.9-3.10.

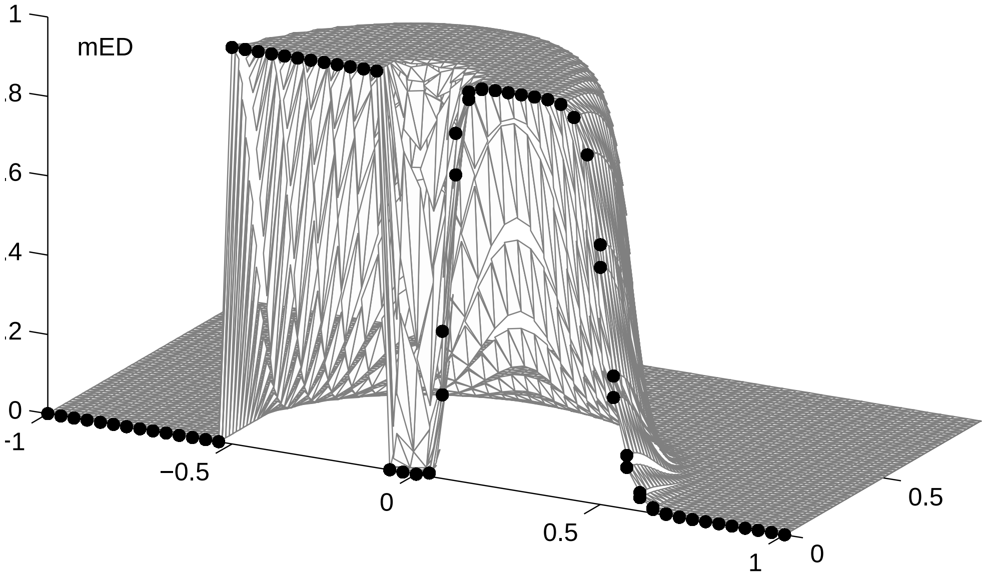


Figure 3.3: Solution for the mED scheme for the Test Case A. The PSI scheme was used to distribute cell residuals.

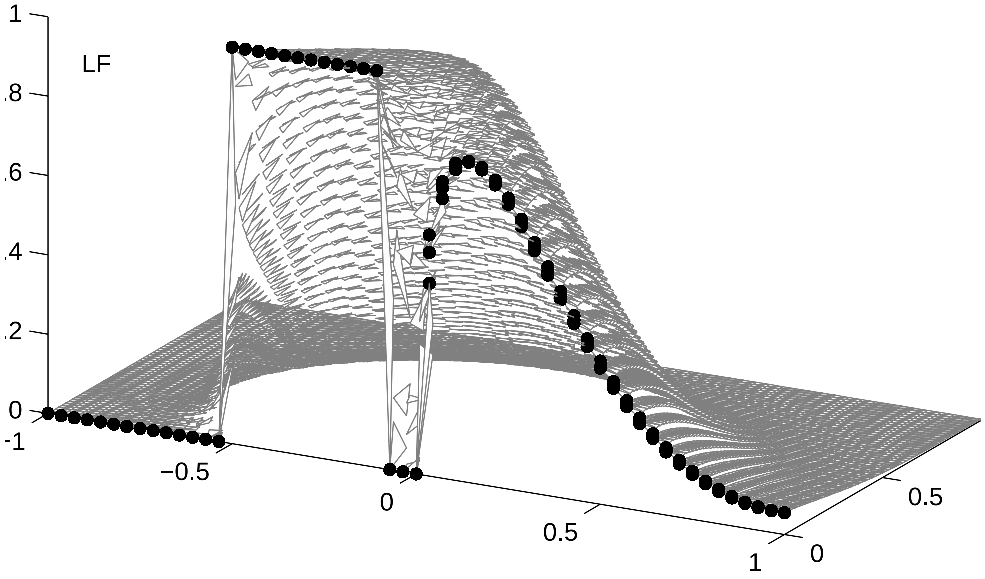


Figure 3.4: Solution for the LF scheme for the Test Case A. The PSI scheme was used to distribute cell residuals.

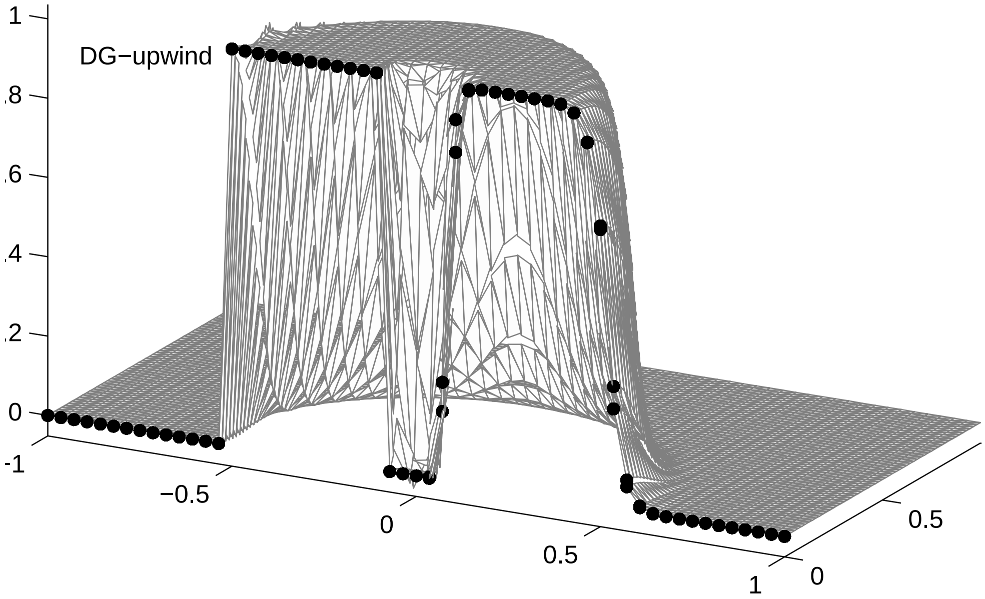


Figure 3.5: Solution for the DG-upwind scheme for the Test Case A. The PSI scheme was used to distribute cell residuals.

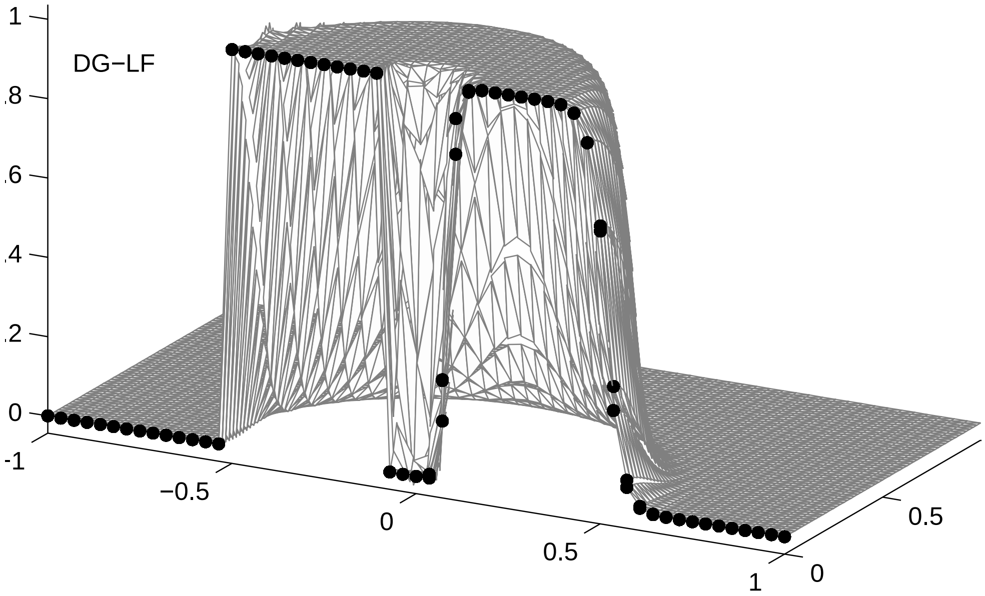


Figure 3.6: Solution for the DG-LF scheme for the Test Case A. The PSI scheme was used to distribute cell residuals.

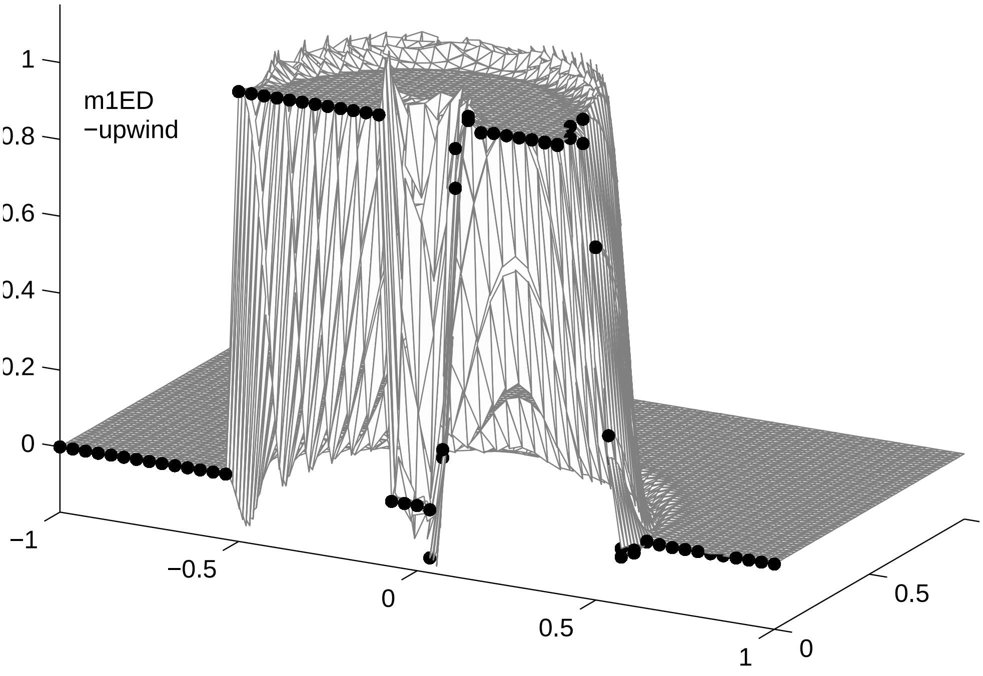


Figure 3.7: Solution for the m1ED-upwind scheme for the Test Case A. The PSI scheme was used to distribute cell residuals.

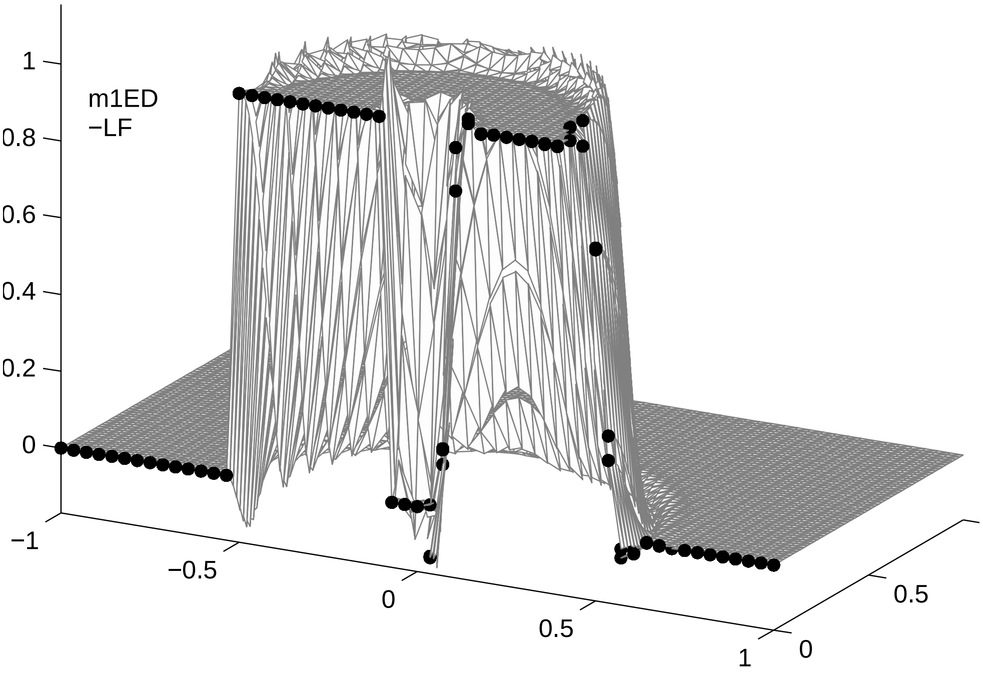


Figure 3.8: Solution for the m1ED-LF scheme for the Test Case A. The PSI scheme was used to distribute cell residuals.

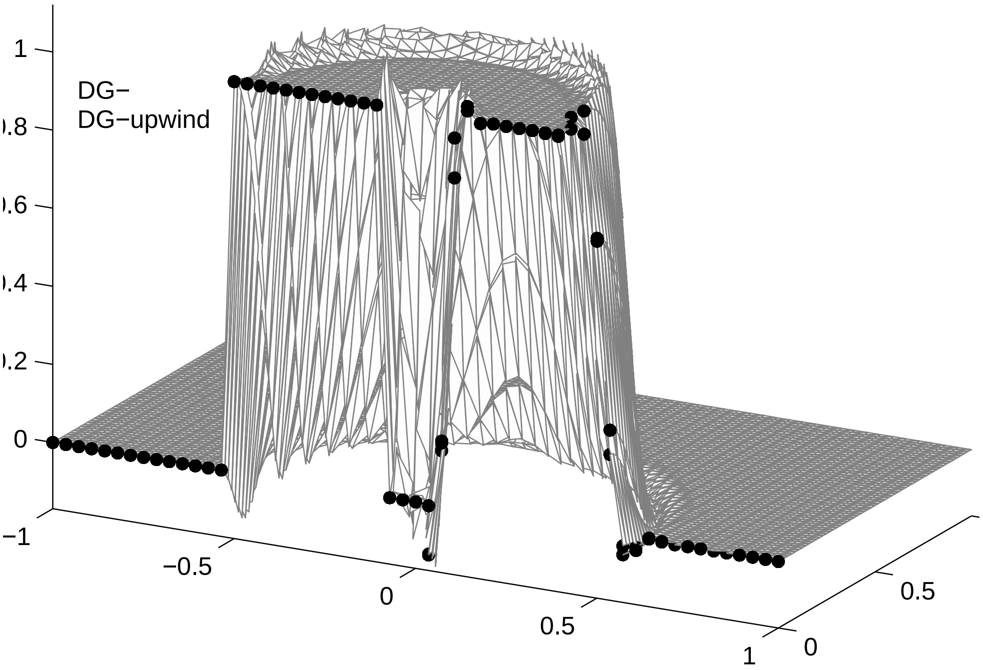


Figure 3.9: Solution for the $\mathcal{DG}$ scheme for the Test Case A using the upwind flux.

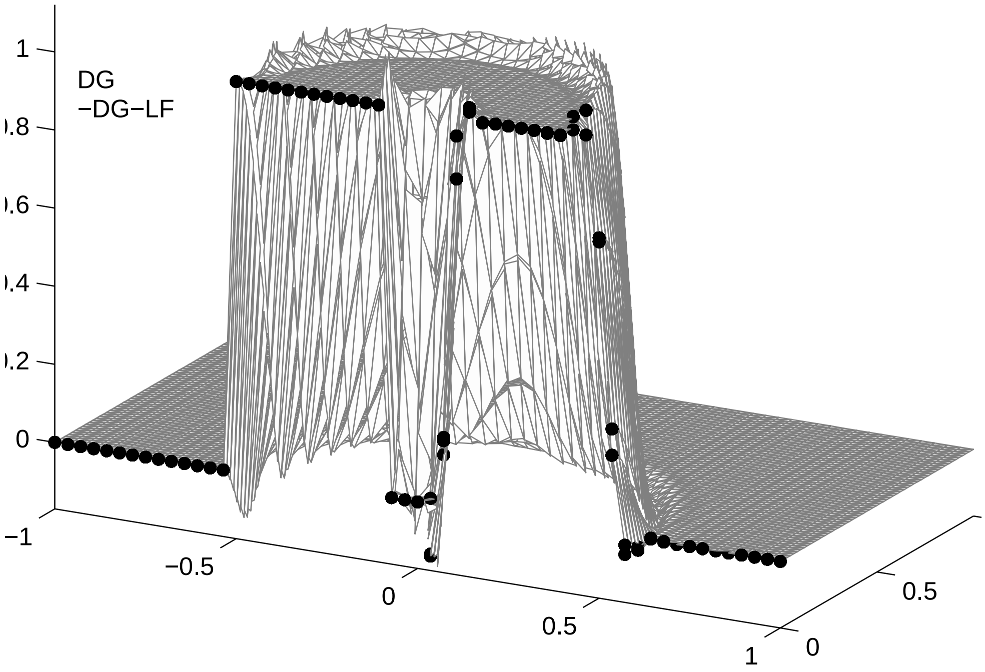


Figure 3.10: Solution for the $\mathcal{DG}$ scheme for the Test Case A using the Lax-Friedrichs flux.

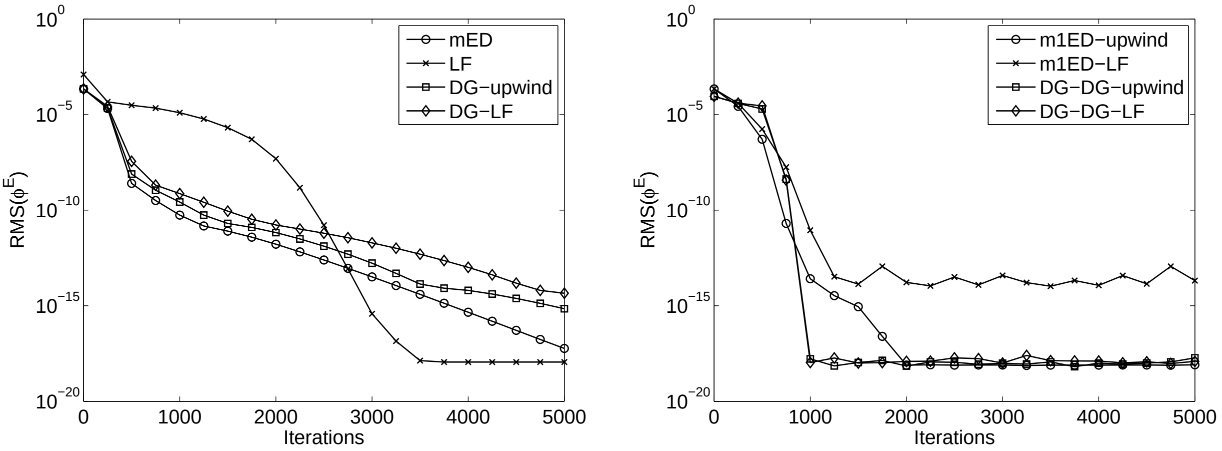


Figure 3.11: Convergence histories for the mED, LF, DG-upwind, DG-LF (left) and the m1ED-upwind, m1ED-LF, DG-DG-upwind and DG-DG-LF (right) schemes for the Test Case B.

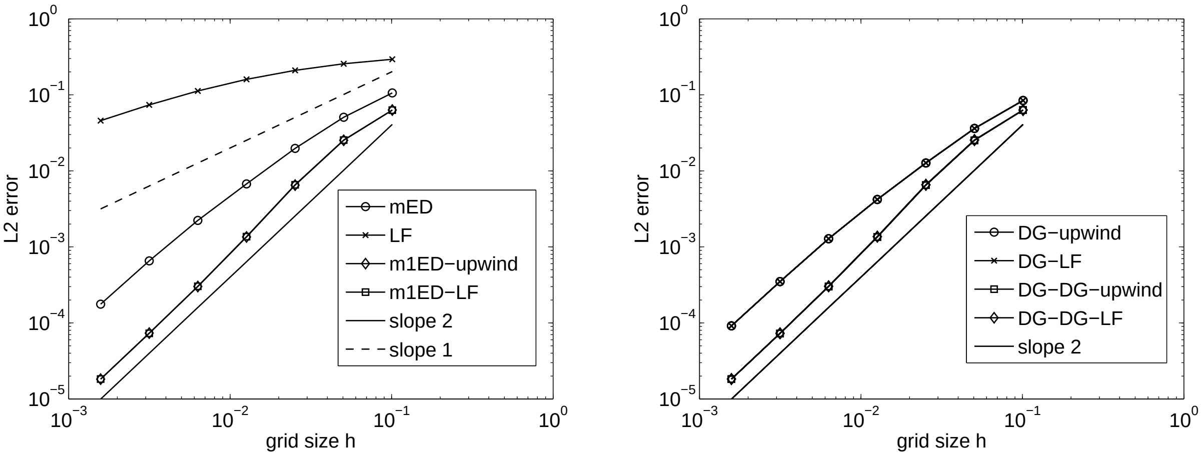


Figure 3.12: Mesh convergence for the mED, SU, LF, m1ED-upwind, m1ED-LF (left) and the DG-upwind, DG-LF, DG-DG-upwind and DG-DG-LF (right) schemes for Test Case B. In all cases switching from the upwind flux to the Lax-Friedrichs flux made very small changes and hence some plots in the above figures seem to overlap each other.

3.8 Summary

The goal of this chapter was to introduce and discuss the discontinuous residual distribution framework. Alongside the definition of the framework, an overview of its properties and key design criteria were outlined. Different schemes within this new setting are constructed by selecting a separate distribution strategy for cell- and edge-based residuals. The former were already discussed in Chapter 2. Characteristic to the discontinuous setting splitting strategies for edge residuals were presented in Section 3.6. Their main properties are outlined in Table 3.3. Finally, the resemblance between discontinuous residual distribution and discontinuous Galerkin approaches was discussed. The discussion carried out in Section 2.3 suggests that every $\mathcal{DG}$ method can be viewed as a particular discontinuous $\mathcal{RD}$ discretization.

	Conservative	Upwind	Continuous	Linear	Positive	Linearity Preserving
mED	✓	✓	✓	✓	✓	✓
LF	✓	×	✓	✓	✓	×
m1ED	✓	✓	✓	✓	×	✓
DG	✓	✓	✓	✓	×	✓

Table 3.3: Summary of the properties of the edge distributions presented in this chapter. A ✓represents success, while × indicates a short-coming in the method.

Presented numerical results showed that the mED method for edges performs the best, i.e. the resulting numerical scheme is second order accurate, the solution is free of spurious oscillations and there is very little diffusion. No other method gave results both second order accurate and positive. It should be pointed out that in the case of the discontinuous Galerkin method no limiting to avoid oscillations was used. Such approach guarantees a fairer comparison between the two discontinuous frameworks. The accuracy and convergence history results show that the discontinuous Galerkin method is a very appealing way of integrating hyperbolic PDEs.

Briefly summarizing, the main contributions of this chapter include:

- discussion of the similarities between the discontinuous residual distribution and discontinuous Galerkin discretizations;
- introduction of the design principles for the discontinuous residual distribution framework;
- development of the m1ED edge distribution;
- numerical comparison of different splittings for edge residuals.

As in the continuous case, extension to non-linear equations will be covered in more detail in chapters on time-dependent problems and non-linear systems of equations.

Chapter 4

The Continuous $\mathcal{RKRD}$ Framework

4.1 Introduction

Although very interesting theoretically and frequently used in applications, steady state models considered in Chapters 2 and 3 are not capable of describing physical phenomena that evolve in time. Instead, time-dependent models have to be employed. This has an immediate consequence in that the success of each numerical framework for solving hyperbolic PDEs is determined, among others, by its ability to tackle not only steady-state, but also transient problems. Bear in mind, though, that adding variation in time not only facilitates models capable of capturing more information but also introduces extra complexity into the process of solving the underlying PDE. Extending steady state methods to time-dependent problems is therefore not always as straightforward as one may wish. In particular, finding a construction that will enable retention of all the nice properties from the steady-state setting very often turns out to be a serious challenge.

The above discussion bears direct relevance here. The framework of residual distribution schemes for steady state problems, at least in the case of scalar equations, has reached a high level of sophistication and understanding. This was summarised in Chapter 2. Even though further research is still being carried out, the emphasis is now mainly laid on the development of $\mathcal{RD}$ methods for time-dependent problems.

Or, to be more precise, on efficiency, accuracy and robustness of such methods. Reviewing and contributing to this study is the main goal in this thesis. To this end, two new approaches to solving unsteady hyperbolic PDEs are introduced, one assuming continuous-in-space and the other assuming discontinuous-in-space data representation. In this chapter continuity of the underlying discretization is assumed and only the first approach will be discussed. Introduction of the second method is the subject of Chapter 5.

It should be pointed out that all the schemes developed within the continuous steady-state $\mathcal{RD}$ framework introduced in Chapter 2 are feasible for time-dependent problems. Indeed, it suffices to prescribe appropriate boundary and initial conditions to make them applicable to such problems. Those methods, however, reduce to first order the accuracy for time-dependent problems, no matter what the order of the special or temporal discretization is. To be more precise, the order of accuracy is at most one for linearity preserving schemes even if the time derivative is discretised using a second or higher order method. This is due to an inconsistency in the spatial discretization (see Section 1 in reference [73] for details) and for this reason alternative approaches need to be explored. Various competing solutions exists, each having its advantages and flaws. The main challenge, i.e. construction of a *second order, positive* and *efficient* scheme, remains open.

In the next section a brief overview of available $\mathcal{RD}$ methods for time dependent hyperbolic PDEs is given, namely the framework of residual distribution schemes with consistent mass matrix and the space-time framework. Only the former category will be considered here in more detail. In Sections 4.3.1 and 4.3.2 examples of two sub-frameworks falling into it are given: the implicit Runge-Kutta Residual Distribution and explicit Runge-Kutta Residual Distribution methods. The main difference between the two is that the resulting linear system is non-diagonal in the implicit case and diagonal in the explicit. Exhaustive numerical results are presented in Section 4.4.

4.2 The Framework

It is assumed that the temporal domain $[0, T]$ is discretized into a set of $N + 1$ discrete levels $\{t^n\}_{n=0,1\ldots,N}$ such that:

$$t^0 = 0, \quad t^N = T, \quad t^n < t^{n+1} \quad \text{and} \quad \Delta t^n = t^{n+1} - t^n.$$

At each time level t^n the approximate solution u_h^n is assumed to be globally continuous (this will be relaxed in the next chapter) and linear within each element $E \in \mathcal{T}_h$, and is given by (cf. Equation (2.6)):

$$u_h^n(\mathbf{x}) = \sum_i \psi_i(\mathbf{x})\, u_i^n, \tag{4.1}$$

where $u_i^n = u_h^n(\mathbf{x}_i)$ are the nodal values of the approximate solution at time t^n. As in the previous sections, ψ_i is the linear Lagrange basis function associated with $\mathbf{x}_i$. Whenever the time level is clear from the context the superscript ^{n} will be omitted.

4.2.1 The Consistent Mass Matrix Formulation

The first successful attempts to construct second-order residual distribution schemes for time-dependent problems were based on the observation, published in 1995 [20], of the close link between the residual distribution and finite element frameworks. This, quite naturally, led to the introduction of a mass matrix m_{ij} (see, in particular, [73] and [36]) and coupling in space of the time derivatives of the nodal values so that the semi-discrete counterpart of (2.2) became:

$$\sum_{E\in\mathcal{D}_i} \sum_{j\in E} m_{ij}^E \frac{du_j}{dt} + \sum_{E\in\mathcal{D}_i} \beta_i \phi^E = 0, \tag{4.2}$$

rather than (cf. Equation (2.8)):

$$u_i^{n+1} = u_i^n - \frac{\Delta t}{|S_i|} \sum_{E\in\mathcal{D}_i} \beta_i \phi^E \qquad \forall i. \tag{4.3}$$

Note that here u_i^n denotes the approximate solution at node $\mathbf{x}_i$ at time $t = t^n$. Expression (4.3) is a simplified version of Formulation (4.2) for which the time derivative was discretized with the aid of the Forward Euler formula, and m_{ij} was set to $|S_i|$ for $i = j$ and 0 for $i \neq j$ (i.e. the mass matrix was lumped). Although the above approach provides a framework for developing higher than first order methods, it leaves open the issue of construction of non-oscillatory schemes. Its close relation to finite elements has, however, enabled application of the same analytical tools and therefore deeper investigation.

In order to construct a second order scheme using formulation (4.2) one has to employ a linearity preservation distribution for cell residuals and discretise the time derivative with a second order scheme, i.e. second order Runge-Kutta method.

The definition of the mass matrix that leads in such a situation to a second order scheme (in both space and time) is not unique and four different approaches are known. Refer to [85] for a thorough overview and extensive numerical comparison. Here only one of them will be employed, the reason being twofold. First of all, as reported in [85] and [81], Formulation 2 (naming as in [85]) gives best (in terms of accuracy and stability) results. Secondly, in the discontinuous setting (see Chapter 5) all computations will be localised and instead of one global mass matrix there will be a separate local mass matrix in each cell. Only Formulation 2 guarantees that those matrices are non-singular. In order to be consistent, Formulation 2 will be employed throughout this thesis. For each cell $E \in \mathcal{T}_h$ the local mass matrix is defined as:

$$m_{ij}^E = \frac{|E|}{36}(3\delta_{ij} + 12\beta_i - 1) \tag{4.4}$$

with δ_{ij} Kronecker's delta. It can be expanded into a matrix form as:

$$\mathbf{M}^{\mathcal{RKDG}} = |E| \begin{bmatrix} \frac{\beta_1}{3} + \frac{1}{18} & \frac{\beta_1}{3} - \frac{1}{36} & \frac{\beta_1}{3} - \frac{1}{36} \\ \frac{\beta_2}{3} - \frac{1}{36} & \frac{\beta_2}{3} + \frac{1}{18} & \frac{\beta_2}{3} - \frac{1}{36} \\ \frac{\beta_3}{3} - \frac{1}{36} & \frac{\beta_3}{3} - \frac{1}{36} & \frac{\beta_3}{3} + \frac{1}{18} \end{bmatrix}.$$

The consistency of this mass matrix with the distribution strategy follows from the dependency of m_{ij} on β_i. This formulation was derived in [73] in which the authors based it on the analogy of the $\mathcal{RD}$ framework with stabilized Galerkin finite element schemes (discussed in Chapter 2). Those considerationsare recalled in Appendix C.

Formulation (4.2) was implemented and investigated in a number of references, i.e. [34, 36, 73] or [19]. In all of these references the authors used multi-step methods to integrate the underlying PDE in time. It is usually argued that the major disadvantage of these methods is the fact that they are implicit, i.e. the resulting linear system is not diagonal (even if explicit multi-step methods are utilised) and therefore expensive. It should come as no surprise that there have not been any attempts to combine this approach with multi-stage time stepping, i.e. Runge-Kutta methods, as such modification will not affect the implicit nature of the method. In [85] Ricchiuto et al. modified the above framework and combined it with multi-stage Runge-Kutta methods to obtain a genuinely explicit scheme. The resulting scheme is indeed explicit, but the formulation is somewhat complicated. It is presented in Section 4.3.2. Lack of any results testing Formulation (4.2) (without any modifications) combined with a multi-stage discretization in time is clearly a gap in the literature the filling of which is one of the main contributions of this thesis.

An example of such methods, Implicit Runge-Kutta Residual Distribution methods (referred hereafter to as implicit $\mathcal{RKRD}$ methods), are presented in Section 4.3.1. It should be pointed out that *implicit* refers here to the fact that the resulting linear system is not diagonal rather than to the fact that the time-stepping procedure is implicit. In this thesis all considered time-stepping methods are explicit! Numerical results presented in Section 4.4, surprisingly enough, show that the loss in efficiency due to solving a global mass matrix is not profound.

4.2.2 The Space-Time Framework

The space-time framework investigated in [29] (see also [38] and references therein) allows construction of second order and positive discretizations. Moreover, it is very faithful to the original spirit of $\mathcal{RD}$ methods which makes this approach a very appealing solution.

In order to proceed, extra notation is now introduced. First, note that in the space-time slab $\Omega \times [t^n, t^{n+1}]$, each element E in the mesh defines a prism in space-time, defined as (see Figure 4.1):

$$E_{t^n} := E \times [t^n, t^{n+1}].$$

By abuse of notation, E_{t^n} will be considered to belong to $\mathcal{D}_i$ if $E \in \mathcal{D}_i$. Denoting by

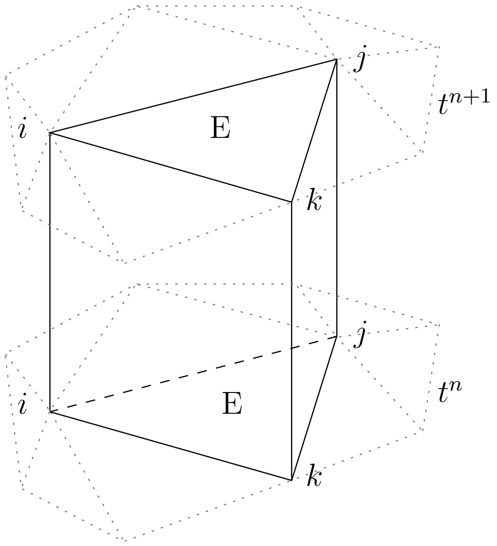


Figure 4.1: Space-time prism $E_{t^n} := E \times [t^n, t^{n+1}]$.

u_h^n and u_h^{n+1} the piecewise linear discrete approximations:

$$u_h^n = \sum_{i \in \mathcal{T}_h} \psi_i(\mathbf{x}) u_i^n \qquad u_h^{n+1} = \sum_{i \in \mathcal{T}_h} \psi_i(\mathbf{x}) u_i^{n+1},$$

the approximation of u in *space* and *time* on the space-time slab $\Omega \times [t^n, t^{n+1}]$ reads

$$u_h^{st}(\mathbf{x}, t) \;=\; \frac{t - t^n}{\Delta t} u^{n+1} \;+\; \frac{t^{n+1} - t}{\Delta t} u^n \qquad \text{for } t \in [t^n, t^{n+1}].$$

This definition is slightly different from the one used in the previous section and is introduced here only in order to demonstrate the space-time $\mathcal{RD}$ framework.

A space-time Residual Distribution scheme is defined as one that, given u_h^n, the discrete approximation in space of u at time t^n, and given a continuous discrete representation in space and time of the unknown u, denoted by u_h^{st}, computes the unknowns $\{u_i^{n+1}\}_{i \in \mathcal{T}_h}$ as follows:

1. $\forall_{E \in \mathcal{T}_h}$ compute the space-time residual

$$\Phi_{E_{t^n}} \;=\; \int_{t^n}^{t^{n+1}} \int_E \left(\frac{\partial u_h^{st}}{\partial t} \;+\; \nabla \cdot \mathbf{f}(u_h^{st}) \right) \, d\Omega \; dt \tag{4.5}$$

2. $\forall_{E \in \mathcal{T}_h}$ distribute fractions of $\Phi_{E_{t^n}}$ to each vertex of E_{t^n}. These fractions (signals) will be denoted by $\Phi_{i,n+1}^{E_{t^n}}$ where i is one of the vertices of E.

3. $\forall i \in \mathcal{T}_h$ assemble the elemental contributions from all $E \in \mathcal{D}_i$ and compute the nodal values of u_h^{n+1} by solving the algebraic system

$$\sum_{E \in \mathcal{D}_i} \Phi_{i,n+1}^{E_{t^n}} = 0 \qquad \forall i \in \mathcal{T}_h. \tag{4.6}$$

This framework allows construction of discretizations with *all* the desired properties, but, unfortunately, leads to schemes which are a subject to a CFL-type restriction on the time step. This is particularly disappointing when taking into account that these schemes are by construction implicit. The positivity condition for this approach is given by:

$$\Delta t = t^{n+1} - t^n \leq \min_{E \in \mathcal{T}_h} \min_{j \in E} \frac{2|E|}{3k_j^+}, \quad \forall n \;=\; 1, \ldots, N.$$

Derivation of this condition can be found in [82]. In the literature it is also referred to as the past-shield condition.

The CFL restriction in the space-time framework follows from the following reasoning. In each space-time prism E_{t^n} only the solution at the new time level, u_h^{n+1}, should be updated and hence receive signals. The one at the previous time level, u_h^n, is already known and its value is to remain unaltered. Imposing this constraint introduces a restriction on the time-step. Interestingly enough, in the *two layer* variant of the space-time framework [32] this restriction is no longer present. It works by coupling two space-time slabs at a time and solving the equations simultaneously in both. On one hand, the resulting system to be solved at each step is larger. On the other, the construction removes from one of the layers the restriction on the time-step. In theory this means that an arbitrarily large time-step can be used. For a full discussion see [29].

This framework is beyond the scope of this thesis and is presented here only for a brief comparison. No further details will be given.

4.3 Examples of Consistent Mass Matrix Frameworks

In this dissertation the main focus of interest are methods falling into the framework of consistent mass matrix schemes. Examples of two such $\mathcal{RD}$ discretizations for time-dependent hyperbolic PDEs are introduced below. The first one, the implicit $\mathcal{RKRD}$ framework, has not been investigated in the literature yet. The second one, the explicit $\mathcal{RKRD}$ framework, was originally introduced in [85] and can be viewed as an approximation to the former.

4.3.1 Implicit Runge-Kutta Residual Distribution Methods

The implicit Runge-Kutta Residual Distribution framework is derived by first integrating (2.1) in time using the second order TVD Runge-Kutta time-stepping, due to Osher and Shu [97]. It gives the following semi-discrete formulation:

$$\begin{cases} \dfrac{\delta u^1}{\Delta t} + \nabla \cdot \mathbf{f}(u^n) = 0, \\ \dfrac{\delta u^{n+1}}{\Delta t} + \dfrac{1}{2}\left(\nabla \cdot \mathbf{f}(u^n) + \nabla \cdot \mathbf{f}(u^1)\right) = 0. \end{cases} \tag{4.7}$$

Here, $\delta u^k = u^k - u^n$ is the increment during the current Runge-Kutta stage and u^1 is the intermediate Runge-Kutta estimate approximating u at time $t = t^{n+1}$. Using

(4.2) to integrate both stages in (4.7) in space leads to:

$$\begin{cases} \displaystyle\sum_{E\in\mathcal{D}_i}\sum_{j\in E} m_{ij}^E \frac{\delta u_j^1}{\Delta t} + \sum_{E\in\mathcal{D}_i} \beta_i \phi^E(u^n) = 0, \\ \displaystyle\sum_{E\in\mathcal{D}_i}\sum_{j\in E} m_{ij}^E \frac{\delta u_j^{n+1}}{\Delta t} + \sum_{E\in\mathcal{D}_i} \frac{1}{2}\beta_i \left(\phi^E(u^n) + \phi^E(u^1)\right) = 0. \end{cases} \tag{4.8}$$

The above defines two linear systems to be solved at every time-step. These systems can be written in a more general form as:

$$\begin{cases} \mathbf{M}\,\dfrac{\delta u^1}{\Delta t} + \boldsymbol{\phi}^1 = 0, \\ \mathbf{M}\,\dfrac{\delta u_j^{n+1}}{\Delta t} + \boldsymbol{\phi}^2 = 0, \end{cases}$$

where M is the global mass matrix, entries of which are defined by (4.4), and $\boldsymbol{\phi}^1$ and $\boldsymbol{\phi}^2$ are the vectors of signals each node has received. Note that the above can be further simplified as:

$$\begin{cases} u^1 = u^n - \Delta t\,\mathbf{M}^{-1}\,\boldsymbol{\phi}^1, \\ u^{n+1} = u^1 - \Delta t\,\mathbf{M}^{-1}\,\boldsymbol{\phi}^2. \end{cases} \tag{4.9}$$

This is the form that was employed to carry out numerical experiments in Section 4.4.

Naturally, in order to finalize the definition of a particular scheme, one still needs to decide which distribution strategy to implement. In this work four approaches were examined, namely the LDA, N, SU and BLEND distribution strategies (outlined in Chapter 2) leading to, respectively, the RKRD-LDA, RKRD-N, RKRD-SU and RKRD-BLEND schemes. Since the N scheme cannot be more than first order accurate, the mass matrix in this particular case can be set as:

$$m_{ij}^N = \delta_{ij}\frac{|E|}{3}, \tag{4.10}$$

in which, as previously, δ_{ij} is Kronecker's delta. The above is simply the lumped version of (4.4), which means that for the N scheme the resulting linear system is diagonal. This definition will be used throughout this thesis (for the N scheme only, though). For the BLEND scheme the mass matrix is defined as:

$$m_{ij}^{BLEND} = \theta m_{ij}^N + (1-\theta)m_{ij}^{LDA},$$

and, similarly, the spatial residuals:

$$\beta_i^{BLEND}\phi^E \;=\; \theta\phi_i^N \,+\, (1-\theta)\phi_i^{LDA},$$

for both the first and the second stage of the Runge-Kutta time-stepping. The definition of the blending parameter θ is outlined in the next section.

The scheme presented in this section will be referred to as either the implicit $\mathcal{RKRD}$ or simply $\mathcal{RKRD}$ scheme (with no direct reference to its implicit nature), as opposed to the explicit $\mathcal{RKRD}$ approach outlined in Section 4.3.2 (always with direct reference to its explicit nature).

4.3.2 Explicit Runge-Kutta Residual Distribution Methods

The method presented in Section 4.3.1 is implicit in the sense that at every time-step two linear systems have to be solved. In [85] Ricchiuto et al. derived an approximation to that approach, namely the framework of explicit Runge-Kutta Residual Distribution methods in which case the resulting linear systems are diagonal. It is based on the observation that for every cell $E \in \mathcal{T}_h$ and set of distribution coefficients β_i there exists a uniformly bounded and locally differentiable *bubble* function γ_i, such that $\sum_{i\in E}\gamma_i = 0$, and the following relation holds (cf. Equation (4.2)):

$$\begin{aligned}\sum_{E\in\mathcal{D}_i}\sum_{j\in E} m_{ij}^E \frac{du_j}{dt} + \sum_{E\in\mathcal{D}_i}\beta_i\phi^E =& \\ = \int_E \psi_i\left(\frac{\partial u_h}{\partial t} + \nabla\cdot\mathbf{f}(u_h)\right) d\Omega &+ \int_E \gamma_i\left(\frac{\partial u_h}{\partial t} + \nabla\cdot\mathbf{f}(u_h)\right) d\Omega.\end{aligned} \tag{4.11}$$

For a proof of this statement and examples of bubble functions satisfying the above refer to [85]. The Lagrange basis function ψ_i acts here as Galerkin test function. The above means that every residual distribution discretization that fits into Formulation (4.2) can be rewritten as a sum of a finite element-type term and a stabilizing bubble function contribution. It follows immediately that the first stage in System (4.8) can be rewritten as:

$$\begin{aligned}\sum_{E\in\mathcal{D}_i}\sum_{j\in E} m_{ij}^E \frac{\delta u_j^1}{\Delta t} + \sum_{E\in\mathcal{D}_i}\beta_i\phi^E(u^n) \;=& \\ = \int_E \psi_i\left(\frac{\delta u^1}{\Delta t} + \nabla\cdot\mathbf{f}(u^n)\right) d\Omega &+ \int_E \gamma_i\left(\frac{\delta u^1}{\Delta t} + \nabla\cdot\mathbf{f}(u^n)\right) d\Omega.\end{aligned} \tag{4.12}$$

Similarly, the second stage in (4.8) can be rewritten as:

$$\begin{aligned}
&\sum_{E\in\mathcal{D}_i}\sum_{j\in E} m_{ij}^E \frac{\delta u_j^{n+1}}{\Delta t} + \sum_{E\in\mathcal{D}_i} \frac{1}{2}\beta_i \left(\phi^E(u^n) + \phi^E(u^1)\right) = \\
&\quad = \frac{1}{2}\int_E \psi_i \left(\frac{\delta u^{n+1}}{\Delta t} + \nabla\cdot\mathbf{f}(u^1)\right) d\Omega + \frac{1}{2}\int_E \gamma_i \left(\frac{\delta u^{n+1}}{\Delta t} + \nabla\cdot\mathbf{f}(u^1)\right) d\Omega \\
&\quad + \frac{1}{2}\int_E \psi_i \left(\frac{\delta u^{n+1}}{\Delta t} + \nabla\cdot\mathbf{f}(u^n)\right) d\Omega + \frac{1}{2}\int_E \gamma_i \left(\frac{\delta u^{n+1}}{\Delta t} + \nabla\cdot\mathbf{f}(u^n)\right) d\Omega.
\end{aligned} \tag{4.13}$$

Note that using the above formulation would lead to a global non-diagonal mass matrix that would have to be solved at every stage of the Runge-Kutta time-stepping. This is despite the fact that explicit time-stepping routine is used. In order to construct a genuinely explicit method, i.e. such that the mass matrix is diagonal, Ricchiuto and Abgrall introduced the so-called *shifted time-operator*:

$$\overline{\delta u^k} = u^{k-1} - u^n \tag{4.14}$$

and substituted it into the right-hand-side of Equations (4.12)-(4.13), but only in the bubble contribution. In the case of the first stage (Equation (4.12)) it leads to:

$$\begin{aligned}
&\sum_{E\in\mathcal{D}_i}\sum_{j\in E} m_{ij}^E \frac{\delta u_j^1}{\Delta t} + \sum_{E\in\mathcal{D}_i} \beta_i\phi^E(u^n) \approx \\
&\quad \approx \int_E \psi_i \left(\frac{\delta u^1}{\Delta t} + \nabla\cdot\mathbf{f}(u^n)\right) d\Omega + \int_E \gamma_i \left(\frac{\overline{\delta u^1}}{\Delta t} + \nabla\cdot\mathbf{f}(u^n)\right) d\Omega,
\end{aligned} \tag{4.15}$$

The two formulations are no longer equal and hence the approximation sign ≈. A similar relation holds for the second stage, i.e. Equation (4.13). The next steps involve mainly algebraic manipulations and are rather technical so will not be presented here. The final form of the scheme is given by (referred to in [85] as the globally lumped formulation):

$$\begin{cases}
|S_i|\dfrac{u_i^1 - u_i^n}{\Delta t} + \displaystyle\sum_{E\in\mathcal{D}_i} \beta_i\phi^E(u^n) = 0, \\
|S_i|\dfrac{u_i^{n+1} - u_i^1}{\Delta t} + \displaystyle\sum_{E\in\mathcal{D}_i} \beta_i\Phi^{RK}(u^n, u^1) = 0.
\end{cases} \tag{4.16}$$

in which Φ^{RK} is the Runge-Kutta residual defined as:

$$\Phi^{RK}(u^n, u^1) = \sum_{j \in E} m_{ij}^E \frac{u_j^1 - u_j^n}{\Delta t} + \frac{1}{2}\beta_i \left(\phi^E(u^n) + \phi^E(u^1)\right).$$

Formulation (4.16), as opposed to (4.8), is explicit and no linear systems have to be solved. The authors prove in their paper that the above construction does not spoil the overall accuracy of the scheme. Their experimental investigation also proves that the resulting discretization is second order accurate. Numerical results in Section 4.4 do confirm that. Unfortunately, it remains unclear how to design a scheme within this framework that would be both second order accurate and positive. The authors suggest that the positivity is lost when the approximation (4.15) is introduced, which may indicate that it cannot be recovered. This remains an open question. It is worth pointing out that even though the above formulation is referred to as globally lumped, no lumping in its traditional meaning is performed. The diagonal matrix is obtained by simply applying an appropriate quadrature rule, i.e. guaranteeing accuracy of order two.

As in the case of the (implicit) $\mathcal{RKRD}$ framework, four different distribution strategies will be considered here, namely the LDA, SU, N and BLEND schemes. These will lead to, respectively, the explicit RKRD-LDA, explicit RKRD-SU, explicit RKRD-N and explicit RKRD-BLEND schemes. In every cell E the blending parameter for the BLEND scheme is defined as:

$$\theta^k(u_h) = \frac{\left|\overline{\Phi^{E(k)}}\right|}{\sum_{j \in E}\left|\overline{\Phi_j^{N(k)}}\right|}$$

for which $k = 1, 2$ denotes Runge-Kutta stage and $\overline{\Phi^{E(k)}}$ the total shifted residual:

$$\overline{\Phi^{E(k)}} = \int_E \left(\overline{\delta u^k} + e^k\right) d\Omega,$$

with e^1 and e^2 being the corresponding evolution operators:

$$e^1 = \nabla \cdot \mathbf{f}(u^n), \qquad e^2 = \frac{1}{2}\nabla \cdot \mathbf{f}(u^1) + \frac{1}{2}\nabla \cdot \mathbf{f}(u^n).$$

Finally, $\overline{\Phi_j^{N(k)}}$ is determined by signals sent by distributing the residuals with the aid of the N scheme and is defined as:

$$\overline{\Phi_j^{N(k)}} = \frac{|E|}{3}\frac{\overline{\delta u^k}}{\Delta t} + \beta_j^N \int_E e^k \, d\Omega.$$

An identical definition of the blending parameter was used for the implicit RKRD-BLEND scheme for which the above formulation guarantees that the resulting system of equations is *linear*. Indeed, had $\theta^k(u_h)$ depended on u_h^{n+1} (or, to be more precise, on δu^k rather than on $\overline{\delta u^k}$), this would not be the case and a system of *non-linear* equations would be constructed instead. Note also that due to the simplified definition of the mass matrix (4.10), the implicit RKRD-N scheme reduces to the explicit RRKD-N scheme.

It should be pointed out that in [85] the authors, apart from Scheme (4.11), presented one more formulation of the explicit $\mathcal{RKRD}$ framework: the so-called selectively lumped explicit $\mathcal{RKRD}$ scheme. The two differ only slightly, the latter being somewhat more complicated and slightly less stable (based on experimental observations). Here only the globally lumped formulation will be considered as this document is only meant to give an overview rather than a complete review of possible alternatives. Moreover, as already pointed out, between the two the globally lumped formulation is more straightforward and gives better results.

The authors in [85] do not raise the issue of stability. Instead, they report that '*A Fourier analysis on unstructured triangulations is under way to have a better estimate of the time step stability limit for the linear schemes.*' [85].

4.4 Numerical Results

In order to investigate properties of the frameworks introduced in this chapter, extensive numerical results are presented and discussed. A further study of the explicit $\mathcal{RKRD}$ framework, including comparison of different types of lumping and mass-matrices, can be found in reference [85]. To the author's best knowledge no other results than those presented here have been published on the implicit $\mathcal{RKRD}$ framework so far. The results presented here have two objectives: to verify the accuracy of the formulations discussed in this chapter and to test the non-oscillatory nature of the results obtained.

Three distinct test cases were implemented. Test Cases C and D are linear equations with *smooth* initial conditions which were used to measure convergence rates. Test Case E is a non-linear equation with a piece-wise constant initial condition, the solution to which exhibits shocks and rarefaction waves. It was employed to investigate positivity. In all experiments, the final time was set as:

- $T = 1$ for Test Cases D and F;
- $T = \frac{\pi}{2}$ for Test Case E.

Test Case D: The so-called *constant advection equation* given by

$$\partial_t u + \mathbf{a} \cdot \nabla u = 0 \qquad \text{on} \quad \Omega_t = \Omega \times [0, 1]$$

with $\Omega = [-1, 1] \times [-1, 1]$ and $\mathbf{a} = (1, 0)$. The exact solution to this problem (which was also used to specify the initial condition at $t = 0$) is given by

$$u(\mathbf{x}, t) = \begin{cases} z^5\,(70z^4 - 315z^3 + 540z^2 - 420z + 126) & \text{if} \qquad r < 0.4, \\ 0 & \text{otherwise} \end{cases}$$

in which $r = \sqrt{(x + 0.5 - t)^2 + y^2}$ and $z = -\frac{r-0.4}{0.4}$ and $\mathbf{x} = (x, y)$. Note that this function is $C^4(\Omega)$ regular. The boundary conditions were set to

$$u(\mathbf{x}, t) = 0 \qquad \text{on } \partial\Omega.$$

Note that for structured grids the advection velocity given above is aligned with the mesh.

Test Case E: The rotational advection equation, given by:

$$\partial_t u + \mathbf{a} \cdot \nabla u = 0 \qquad \text{on} \quad \Omega_t = \Omega \times [0, \frac{\pi}{2}]$$

with $\Omega = [-1, 1] \times [-1, 1]$ and $\mathbf{a} = (-y, x)$. The exact solution to this problem (which was also used to specify the initial condition at $t = 0$) is given by

$$u(\mathbf{x}, t) = \begin{cases} z^5(70z^4 - 315z^3 + 540z^2 - 420z + 126) & \text{if} \qquad r < 0.4, \\ 0 & \text{otherwise} \end{cases}$$

where $r = \sqrt{(x - x_c)^2 + (y - y_c)^2}$ and

$$z = -\frac{r - 0.4}{0.4}, \quad x_c = \frac{1}{2}\cos\left(t - \frac{\pi}{2}\right), \quad y_c = \frac{1}{2}\cos\left(t - \frac{\pi}{2}\right).$$

The boundary conditions were set to:

$$u(\mathbf{x}, t) = 0 \qquad \text{on } \partial\Omega.$$

Contrary to Test Case D, here the advection velocity is generally not aligned with the mesh. This test case is used to make sure that results obtained for Test Case D are not biased by the direction of the flow.

Test Case F: The inviscid Burgers' equation is given by:

$$\partial_t \, u \, + \, \nabla \cdot \mathbf{f}(u) \; = \; 0 \qquad \text{on} \quad \Omega_t = \Omega \times [0, 1]$$

with $\mathbf{f} = (\frac{u^2}{2}, \frac{u^2}{2})$. As for Test Cases D and E, the spatial domain is a square: $\Omega = [-1, 1] \times [-1, 1]$. The initial condition was set to be piece-wise constant:

$$u(\mathbf{x}, 0) = \begin{cases} 1 & \text{if} \qquad \mathbf{x} \in [-0.6, -0.1] \times [-0.5, 0] \\ 0 & \text{otherwise} \end{cases}$$

The boundary conditions were set to:

$$u(\mathbf{x}, t) = 0 \qquad \text{on } \partial\Omega.$$

The solution to this problem is discontinuous and exhibits rarefaction and shock waves. It is therefore a very challenging and interesting problem that was used to test for positivity. The exact solution to this problem is given in Appendix A.

In this chapter two types of triangulations were used, i.e. structured (regular and isotropic) and unstructured, examples of which are illustrated in Figure 4.2. Linear equations, as in the two previous chapters, were solved on structured grids. To demonstrate robustness of the methods discussed here, in particular to show that they can be used with both structured and unstructured discretizations of the domain, an unstructured mesh with 26054 elements (topology similar to that on the right of Figure 4.2) was used in the case of the non-linear Burgers' equation. The time step was calculated using (cf. Equation (68) in [85])

$$\Delta t = CFL \min_{i \in \mathcal{T}_h} \frac{|S_i|}{\sum_{E|_{i \in E}} \alpha^E}, \tag{4.17}$$

with CFL number set to 0.9 and the α^E coefficient defined as:

$$\alpha^E = \frac{1}{2} \max_{j \in E} \left\| \frac{\partial \mathbf{f}(u_j)}{\partial u} \right\| h_E, \tag{4.18}$$

with h_E being the reference length for element E. The linear system resulting from the implicit $\mathcal{RKRD}$ discretization was solved using PETSc [15] (see also the manual [16]) within which the ILU preconditioned GMRES solver was used. This is the default setting in PETSc which agrees with the type of solver that is usually suggested in the case of general non-symmetric systems of equations (see, for instance, Section 6.6.6 and Figure 6.8 in [43]). Since it gave good results, no other solver was implemented. To guarantee convergence, the relative tolerance in PETSc, i.e. the stopping criterion, was always set to 10^{-8}. The initial estimate was always set to zero.

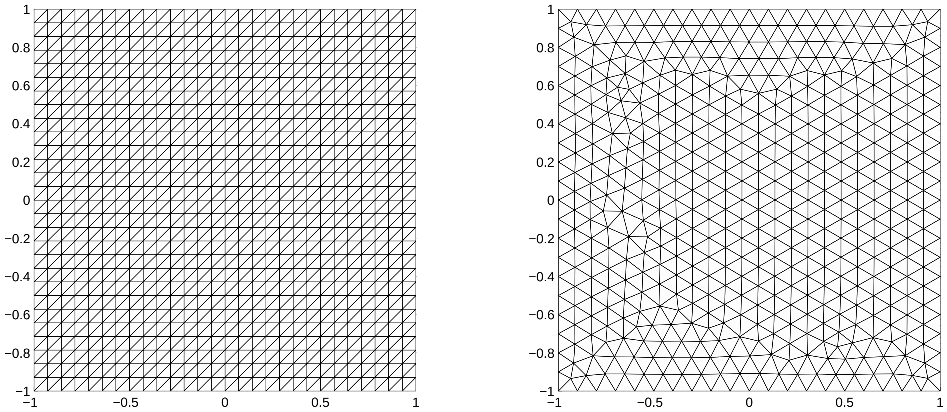


Figure 4.2: Representative structured (left) and unstructured (right) grids used for transient problems.

The grid convergence analysis confirmed that, within both the implicit and explicit $\mathcal{RKRD}$ frameworks, the N scheme is only first order accurate whereas the LDA, SU and BLEND schemes exhibit convergence of order two. These results, presented in Figures 4.3 and 4.4, indicate that with respect to accuracy both frameworks perform qualitatively the same. In the implicit $\mathcal{RKRD}$ framework the LDA and SU schemes gave best results, the SU scheme being noticeably more accurate than LDA. The BLEND scheme is slightly less accurate then both of them. This is most likely due to its nonlinear nature. Interestingly enough, moving to the explicit $\mathcal{RKRD}$ framework makes the LDA and SU schemes by an order of magnitude less accurate. Suddenly the LDA, SU and the BLEND scheme start to perform in a very similar manner, comparable to the implicit RKRD-BLEND scheme. In other words, the explicit schemes are less accurate than their implicit counterparts. These experiments were carried out on a set of regular triangular meshes (topology as on the left of Figure 4.2) with the coarsest mesh of a 14×14 regular grid refined 6 times by a factor 2 in each direction. The accuracy was monitored by the convergence of the L^2 norm of error (2.27) at the final time of the simulation with respect to the exact solution. The behaviour of the L^1 and L^∞ norms was qualitatively and quantitatively very similar. Switching to unstructured meshes also led to qualitatively identical results.

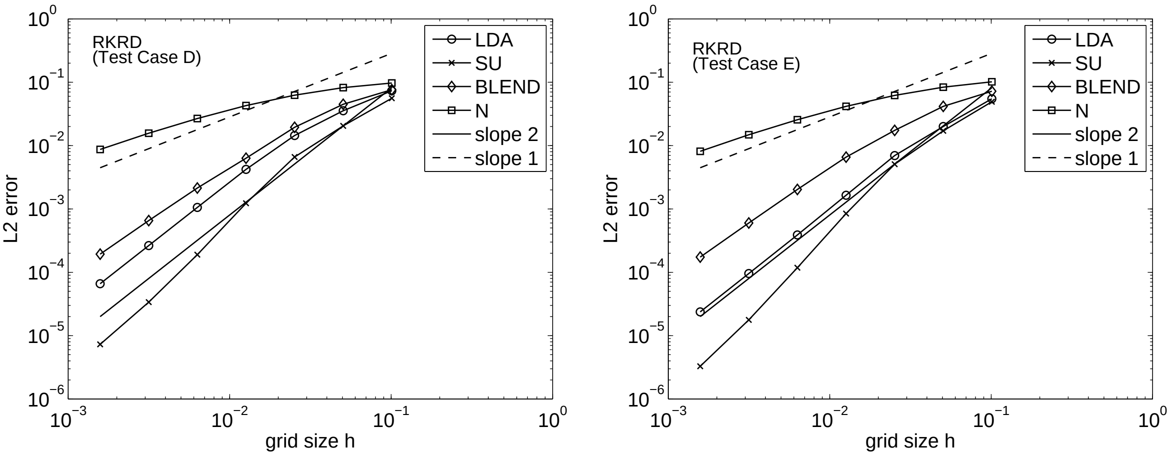


Figure 4.3: Grid convergence for the implicit $\mathcal{RKRD}$ framework for Test Cases D (left) and E (right).

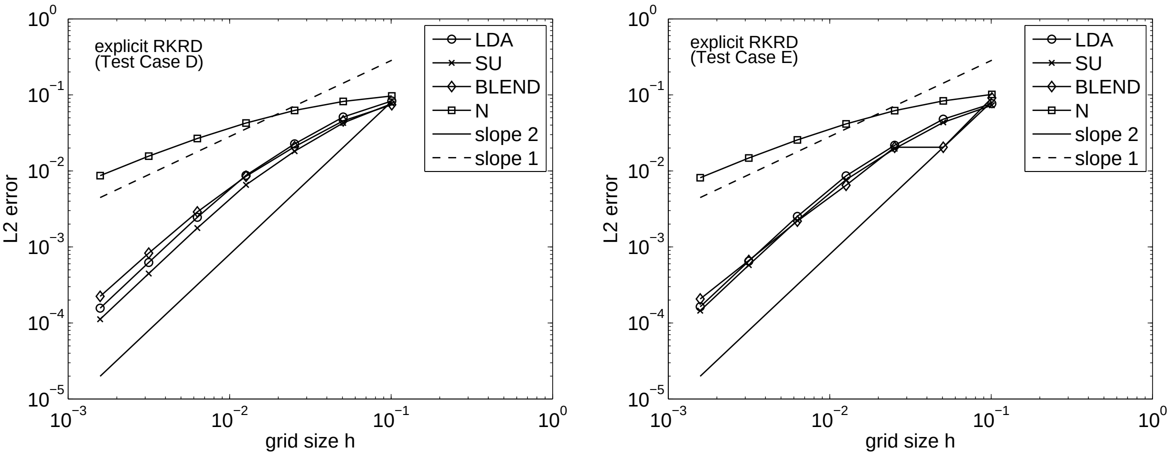
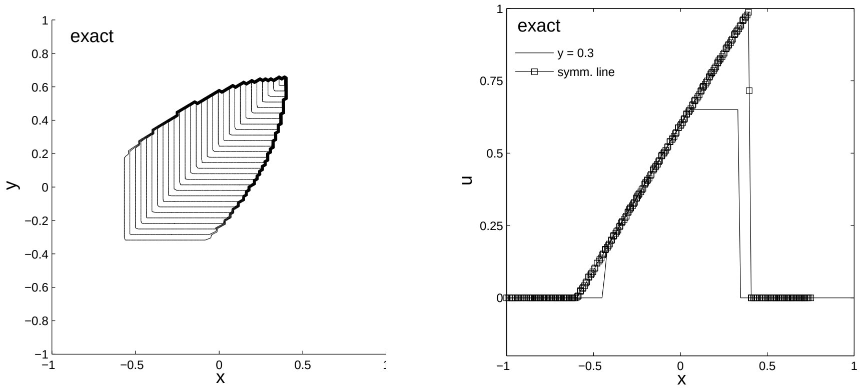


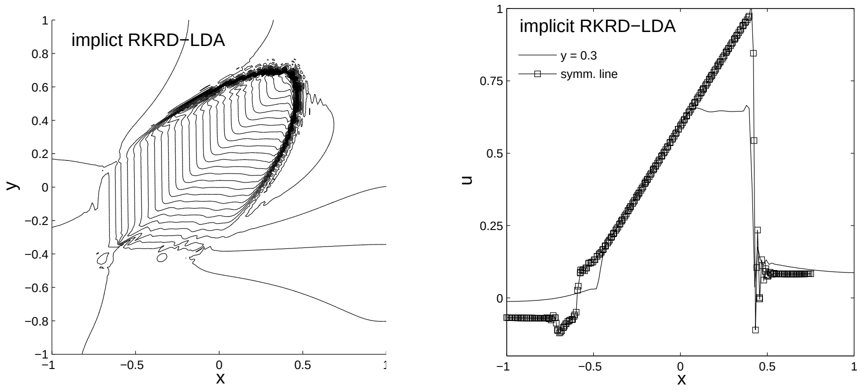
Figure 4.4: Grid convergence for the explicit $\mathcal{RKRD}$ framework for Test Cases D (left) and E (right).

In Figure 4.5 the contours, cross sections (along the symmetry line $y-x = 0.1$ and $y = 0.3$) are plotted and the maximum and minimum values of the exact solution to Burgers' equation (Test Case F) are given. Similar plots and quantities are given for the approximate solutions obtained with the aid of the implicit and explicit $\mathcal{RKRD}$ frameworks, see Figures 4.6-4.13. As expected, the N scheme gave a solution free of spurious oscillations (it is positive), though more diffusive than other schemes. The solution obtained with the aid of the LDA scheme exhibits oscillations near discontinuities (again, as expected). Compared to the explicit $\mathcal{RKRD}$ approach, these oscillations are much more pronounced when the implicit $\mathcal{RKRD}$ framework is used. To show that this was not due to the poor performance of the linear solver, two extra experiments were carried out. First, the CFL number was decreased to 0.1, all other parameters being the same as before. The result of this experiment is shown in Figure 4.14. Clearly the new solution is much smoother. Next, the RKRD-LDA scheme was tested with CFL set to, as previously, 0.9 and the relative tolerance in PETSc decreased to 10^{-16}. The final residual in this case was roughly (at each time-step and at each Runge-Kutta stage) equal to 10^{-18}. Results are shown in Figure 4.14. Clearly tuning PETSc did not help, which implies it is the scheme itself, not the linear solver, that is unstable. Other schemes behaved similarly regardless whether the $\mathcal{RKRD}$ discretization was explicit or implicit. The implicit and explicit RKRD-BLEND schemes performed much better than the implicit and explicit RKRD-LDA schemes, respectively. Blending helped smooth the solutions out and the resulting approximations have smaller under/over-shoots. Although less diffusive then the N scheme, the BLEND scheme is not 100% oscillation-free. To summarise, the BLEND scheme gives the best trade-off between being oscillations-free and second order accurate. In terms of accuracy the implicit framework is more accurate, but more oscillatory than its explicit counterpart. The PSI scheme discussed in Chapters 2 and 3 was not considered in this context as it would lead to a genuinely implicit scheme, i.e. the resulting system of equations would be non-linear. In this thesis the focus is laid on schemes that lead to linear systems of equations.



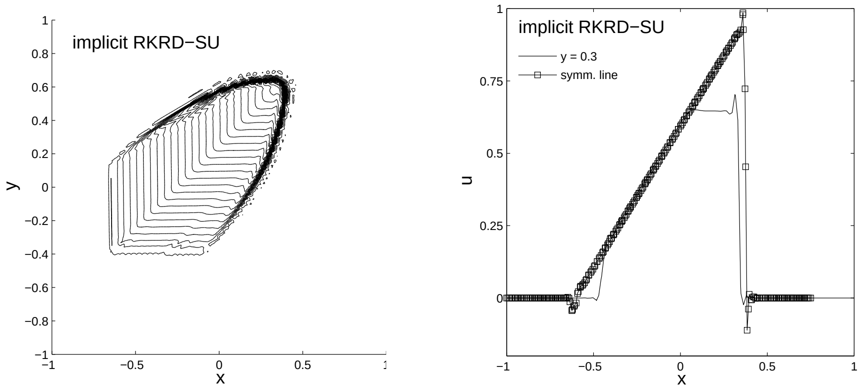
u_{min}	u_{max}
0.0	1.0

Figure 4.5: 2d Burgers' equation: the analytical solution. Left: contours at time $t = 1$. Middle: solution along line $y = 0.3$ and along the symmetry line. Right: minimum and maximum values of the solution.



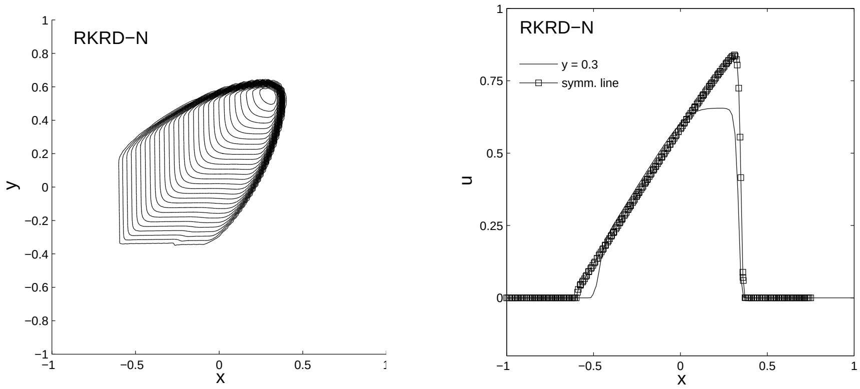
u_{min}	u_{max}
-0.181	1.041

Figure 4.6: 2d Burgers' equation: implicit RKRD-LDA scheme. Left: contours at time $t = 1$. Middle: solution along line $y = 0.3$ and along the symmetry line. Right: minimum and maximum values of the solution.



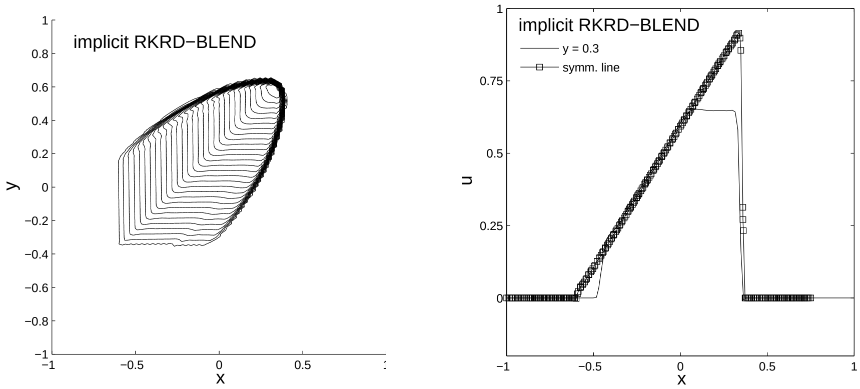
u_{min}	u_{max}
-0.167	1.058

Figure 4.7: 2d Burgers' equation: implicit RKRD-SU scheme. Left: contours at time $t = 1$. Middle: solution along line $y = 0.3$ and along the symmetry line. Right: minimum and maximum values of the solution.



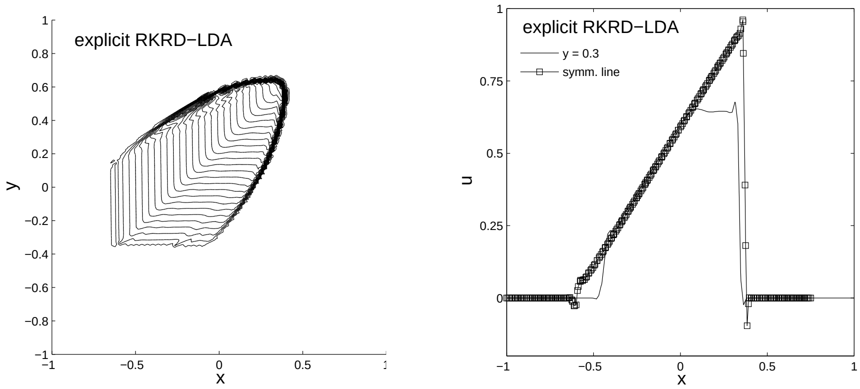
u_{min}	u_{max}
0.0	0.839

Figure 4.8: 2d Burgers' equation: RKRD-N scheme. Left: contours at time $t = 1$. Middle: solution along line $y = 0.3$ and along the symmetry line. Right: minimum and maximum values of the solution.



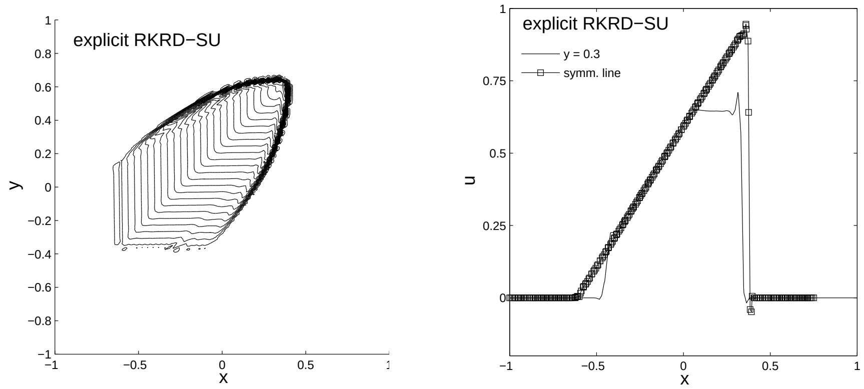
u_{min}	u_{max}
-0.017	0.915

Figure 4.9: 2d Burgers' equation: implicit RKRD-BLEND scheme. Left: contours at time $t = 1$. Middle: solution along line $y = 0.3$ and along the symmetry line. Right: minimum and maximum values of the solution.



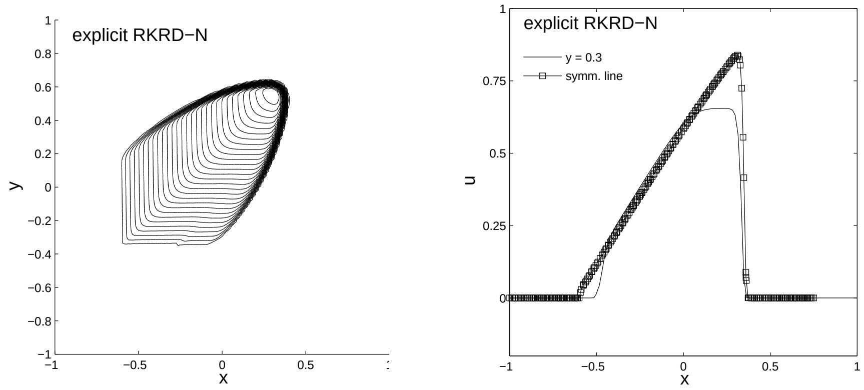
u_{min}	u_{max}
-0.141	0.961

Figure 4.10: 2d Burgers' equation: explicit RKRD-LDA scheme. Left: contours at time $t = 1$. Middle: solution along line $y = 0.3$ and along the symmetry line. Right: minimum and maximum values of the solution.



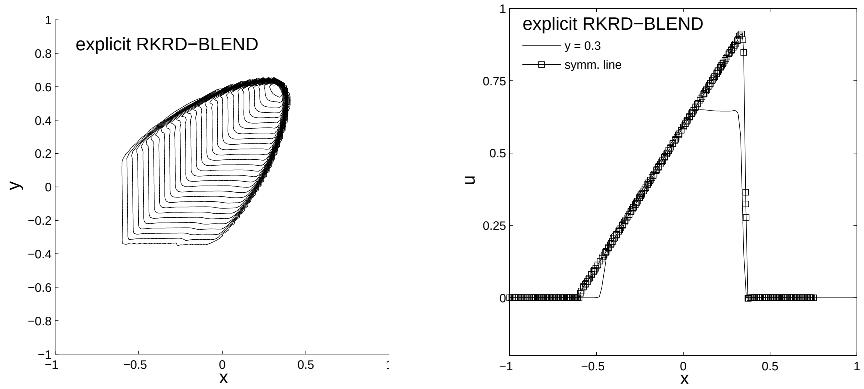
u_{min}	u_{max}
-0.140	1.017

Figure 4.11: 2d Burgers' equation: explicit RKRD-SU scheme. Left: contours at time $t = 1$. Middle: solution along line $y = 0.3$ and along the symmetry line. Right: minimum and maximum values of the solution.



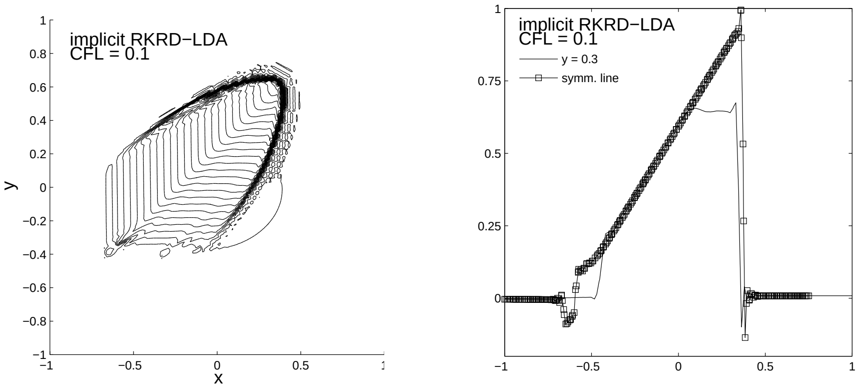
u_{min}	u_{max}
0.0	0.839

Figure 4.12: 2d Burgers' equation: explicit RKRD-N scheme. Left: contours at time $t = 1$. Middle: solution along line $y = 0.3$ and along the symmetry line. Right: minimum and maximum values of the solution.



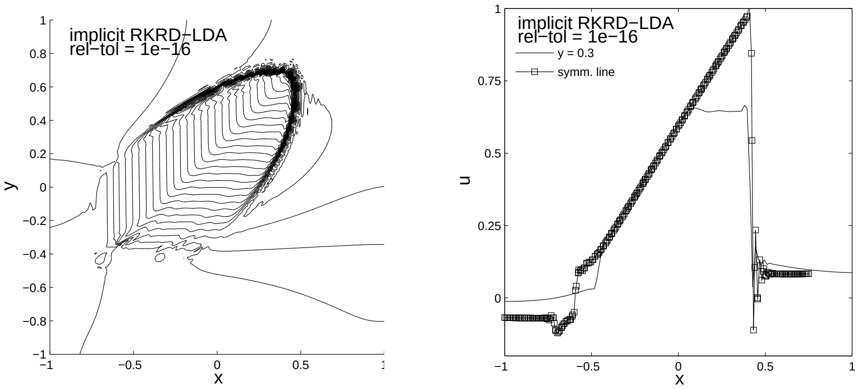
u_{min}	u_{max}
-0.011	0.914

Figure 4.13: 2d Burgers' equation: explicit RKRD-BLEND scheme. Left: contours at time $t = 1$. Middle: solution along line $y = 0.3$ and along the symmetry line. Right: minimum and maximum values of the solution.



u_{min}	u_{max}
-0.191	1.014

Figure 4.14: 2d Burgers' equation: implicit RKRD-LDA scheme with CFL set to 0.1. Left: contours at time $t = 1$. Middle: solution along line $y = 0.3$ and along the symmetry line. Right: minimum and maximum values of the solution.



u_{min}	u_{max}
-0.181	1.041

Figure 4.15: 2d Burgers' equation: implicit RKRD-LDA scheme with relative tolerance set to 10^{-16}. Left: contours at time $t = 1$. Middle: solution along line $y = 0.3$ and along the symmetry line. Right: minimum and maximum values of the solution.

Finally, one should comment on scaling and performance of the linear solver that was applied to solve linear systems resulting from the (implicit) $\mathcal{RKRD}$ discretization. As mentioned earlier, only GMRES preconditioned with ILU was used. To guarantee convergence, the linear solver was set to iterate until the relative tolerance r_{tol}:

$$r_{tol} = \frac{||r||_{l^2}}{||b||_{l^2}},$$

reached 10^{-8}. In the above r is the current residual and b is the right-hand-side vector (since the initial estimate was set to zero, b is also the initial residual). For all test cases and for all schemes the linear solver converged rather rapidly (on average, in less than 10 iterations) with the final residual equal to roughly 10^{-11}. Some sample results are given in Table 4.1. The extremely rapid convergence in the case of the N scheme should come as no surprise as the resulting linear system is diagonal. The behaviour of the iterative solver when the BLEND scheme is used may seem odd as the number of iterations needed for convergence for the first and the second stage of the Runge-Kutta time-stepping differs by around 100%. This is due to the fact that during the first stage the blending parameter picks the first order N scheme most of the time and the system of equations is very close to a diagonal matrix. The opposite situation is taking place during the second stage.

		1568	6272	25088	100352	401408	1605632
LDA	GMRES iter.	9.84/9.84	8.52/8.52	7.95/7.95	7.76/7.76	7.74/7.74	7.63/7.63
	$\|\|r_F\|\|_2$	7.8e-11	1.39e-10	1.9e-11	1.92e-11	4.8e-12	6.55e-12
BLEND	GMRES iter.	4.3/6.9	4.30/7.56	4.44/8.21	4.33/8.68	3.29/7.82	4.27/8.57
	$\|\|r_F\|\|_2$	3.96e-11	9.84e-11	2.09e-11	2.18e-11	1.03e-11	1.03e-11
N	GMRES iter.	2/2	2/2	2/2	2/2	2/2	2/2
	$\|\|r_F\|\|_2$	2.5e-11	3e-17	4e-17	6e-17	7e-17	6e-17
SU	GMRES iter.	9.03/9.03	7.76/7.78	6.41/6.41	6/6	5.88/5.88	5.87/5.87
	$\|\|r_F\|\|_2$	1.04e-10	2.4e-11	8.78e-11	8.61e-12	1.59e-12	5.13e-13

Table 4.1: Performance of the GMRES solver when applied to the linear systems resulting from the $\mathcal{RKRD}$ discretizations (Test Case E). The table shows the average number of iterations it took to reach the stopping criterion during the first/second stage of the Runge-Kutta time-stepping and the l_2 norm of the final residual (when GMRES converged at the final time-step) at the second stage of the RK time stepping (denoted by $||r_F||_2$). Results are given for the meshes used earlier in the grid convergence analysis (with 1568, 6272, 25088, 100352, 401408 and 1605632 elements, cf. top row of the table).

No comparison between execution times of the explicit and (implicit) $\mathcal{RKRD}$ frameworks is given. This is primarily due to the fact that in the latter case a very advanced and mature software library was used whereas the code for the explicit framework was not optimised and all procedures were written with relatively little emphasis on efficiency. Regardless the lack of actual results, it is worth mentioning that the observed execution times in both cases were comparable. This, on one hand, indicates that PETSc does indeed implement GMRES very efficiently. On the other it suggest that the (implicit) $\mathcal{RKRD}$ framework is *not* too expensive for applications and should be considered as an interesting alternative. Further notes on this matter are given in Chapter 6 in which systems of non-linear equations are considered.

4.5 Summary

In this chapter different techniques of approximating time-dependent hyperbolic PDEs using the $\mathcal{RD}$ framework were outlined. The discrete solution was assumed to be piecewise linear and continuous. The focus was laid on the framework of consistent mass matrix formulations combined with multi-stage second order TVD Runge-Kutta method for integration in time. In particular, two competing techniques were considered: implicit and explicit $\mathcal{RKRD}$ methods.

Regarding the accuracy and positivity the two approaches are very similar and there is no clear indication which of the considered frameworks is superior. The implicit $\mathcal{RKRD}$ methods are in general more accurate than their explicit counterparts, but slightly less stable when it comes to non-linear equations (in particular the implicit RKRD-LDA scheme). From a practical point of view, explicit methods are cheaper and significantly simplify the process of parallelization. The whole framework is a bit more complex than the implicit formulation. This becomes particularly apparent when studying the original paper [85] in which numerous variants of the explicit framework are discussed. Unfortunately no clear indication as which is the optimal choice is given. Moreover, the explicit framework is not as much more efficient than its implicit counterpart as expected (at least its serial implementation). Still, there is a space for potential improvements (i.e. better implementation, parallelization) which will speed the calculations up and which are not that obvious in the case of implicit discretisations. Construction of a second order *and* positive scheme still remains an open question. As far as implicit $\mathcal{RKRD}$ methods are concerned, developing a genuinely non-linear scheme is a possible solution. This will, however, lead to a set of non-linear (as opposed to linear in both the implicit and explicit $\mathcal{RKRD}$ cases) set of equations. In the case of the explicit $\mathcal{RKRD}$ framework one has to first investigate the impact of introducing the shifted time operator $\overline{\delta u^k}$ with regard to positivity. Another possibility is the limiting procedure of Hubbard and Mebrate [59] developed for steady-state high-order methods. However in [74] it gave only modest results when applied to time-dependent problems (the approximate solutions are not 100% oscillation-free).

One of the most interesting things observed in this chapter is the efficiency with which PETSc solves linear systems resulting from the (implicit) $\mathcal{RKRD}$ discretisations. A very natural extension of the presented results would involve carrying out a series of numerical experiments that would further compare the efficiency of various approaches to time-dependent hyperbolic PDEs. Another possible extension would a rigorous study of the effect of introducing the shifted time operator,$\overline{\delta u^k}$, on positivity.

Chapter 5

The Discontinuous $\mathcal{RKRD}$ Framework

5.1 Introduction

In the case of steady state residual distribution methods, relaxing the constraint on the continuity of the data led to very promising results. A new, more flexible framework was introduced and, as a consequence, a construction of scheme exhibiting all the desired properties was possible. Moreover, the resulting scheme was localised which facilitates $h-$ and $p-$ adaptation as well as parallelization. Extending those results to time-dependent problems is a natural step forward which is the main goal in this chapter.

To the author's best knowledge the only attempt to combine discontinuous-in-space data representation with $\mathcal{RD}$ schemes for time dependent problems was carried out by Warzyński et al. in [105]. However, the authors of that paper used only first order discretization in time and tested the resulting framework in terms of positivity. Their results are discussed in Section 5.5. The goal of this chapter is to further extend those results by designing a second order accurate discontinuous-in-space $\mathcal{RD}$ framework for time dependent problems. This, quite naturally, will be achieved by drawing together the framework of discontinuous $\mathcal{RD}$ schemes for steady state problems, outlined in Chapter 3, and (implicit) $\mathcal{RKRD}$ framework from

Chapter 4 for transient problems.

This chapter is organised as follows. First, the framework of discontinuous Runge-Kutta Residual Distribution schemes is introduced. Next, it is thoroughly analysed in terms of its relation to the discontinuous Galerkin framework. To a large extent this will be a continuation of the discussion already started in Section 3.3. In Section 5.4 the framework of unsteady discontinuous residual distribution methods is introduced. Finally, extensive numerical results summarising observations made in this chapter are given.

5.2 The Framework

The notation from Chapter 4 is kept unchanged. Only the approximate solution, $u_h^n(\mathbf{x})$, takes now a slightly more general form:

$$u_h^n(\mathbf{x})\big|_E \;=\; \sum_{i\in E} \psi_i(\mathbf{x})\, u_i^n, \qquad \forall \mathbf{x}\in E \quad \forall E \in \mathcal{T}_h. \tag{5.1}$$

This reflects the fact that it is no longer assumed to be globally continuous. In the remainder of this chapter, for clarity of presentation, the subscript $_h$ will be omitted. It is assumed that whenever a superscript is used (e.g. ^{n} or $^{n+1}$) then the approximate rather than the exact solution is considered, i.e. $u_h^n = u^n$. This is mainly to clarify the discussion.

The discontinuous Runge-Kutta Residual Distribution scheme is constructed by first integrating Equation (2.1) in time using the second order TVD Runge-Kutta procedure, as outlined in Chapter 4. The resulting Formulation (4.7) was originally discretised with the aid of continuous $\mathcal{RD}$ methods. Here, that semi-discrete equation will be discretized with the aid of discontinuous $\mathcal{RD}$ methods presented in Chapter 3. The resulting formulation reads:

$$\left\{\begin{aligned} &\sum_{j\in E} m_{ij}^E \frac{\delta u_j^1}{\Delta t} \;+\; \beta_i \phi^E(u^n) \;+\; \sum_{e\in E} \alpha_i\, \phi^e(u^n) \;=\; 0,\\ &\sum_{j\in E} m_{ij}^E \frac{\delta u_j^{n+1}}{\Delta t} \;+\; \frac{1}{2}\,\beta_i \left(\phi^E(u^n) \,+\, \phi^E(u^1)\right)\\ &\qquad\qquad\qquad +\; \frac{1}{2}\sum_{e\in E} \alpha_i \left(\phi^e(u^n) \,+\, \phi^e(u^1)\right) \;=\; 0. \end{aligned}\right. \tag{5.2}$$

As in the previous chapters, ϕ^E and ϕ^e denote cell and edge residuals, respectively, and the distribution coefficients β and α are used to split them between the vertices of E. The mass matrix, m_{ij}^E, was introduced in Section 4.2.1 and the time-increment operator $\delta u^k = u^k - u^n$ was introduced in Section 4.3.1. In this thesis, the distribution coefficients β_i and α_i are calculated using u^n during the first stage of the Runge-Kutta time stepping, and u^1 during the second stage. Particular methods developed within this framework will be referred to as: (discontinuous) RKRD-A-B method in which A and B stand for the distribution strategy for cells and edges, respectively.

Note that Scheme (5.2) is very similar to the continuous $\mathcal{RKRD}$ scheme (cf. Formulation (4.8)). One difference is the fact that now every degree of freedom i belongs to only one cell, i.e. $\mathcal{D}_i = E$, and therefore there is no extra summation over the elements belonging to $\mathcal{D}_i$. Another difference is the presence of edge residuals which impose communication between cells.

Linear system (5.2) is block-diagonal with 3 × 3 blocks corresponding to each cell. This effectively means that the scheme is explicit as one can easily solve 3 × 3 systems analytically. There is therefore little, if any, justification in trying to combine discontinuous-in-space data representation with the continuous explicit $\mathcal{RKRD}$ framework investigated in Chapter 4. Instead, the discontinuous $\mathcal{RKRD}$ framework should be considered as an alternative to both continuous implicit and explicit $\mathcal{RKRD}$ approaches. Being explicit, it is more promising than the first one as no additional work related to solving global linear systems is needed. Recall that in the continuous $\mathcal{RKRD}$ setting an external numerical library was used to solve the resulting linear system. The current framework can also be viewed as superior to the continuous explicit $\mathcal{RKRD}$ discretisations in the sense that in the discontinuous setting the explicit nature is achieved without introducing the shifted time-operator, $\overline{\delta u^k}$ (cf. Equation (4.14)).

5.3 Relation with the $\mathcal{RKDG}$ Framework

The Runge-Kutta Discontinuous Galerkin methods, due to Cockburn and Shu [23–27], are, ever increasingly, a very popular way of discretising time-dependent hyperbolic PDEs. A brief introduction to their steady-state counterpart and the relation between that framework and discontinuous $\mathcal{RD}$ methods was given in Section 3.3. This section aims at extending those observations. First, however, the $\mathcal{RKDG}$ framework is introduced.

5.3.1 The Runge-Kutta Discontinuous Galerkin Framework

As in the case of discontinuous $\mathcal{RKRD}$ schemes, one starts by discretising in time. Although different time-stepping techniques can be implemented, only the second order TVD Runge-Kutta time-stepping will be employed here. In other words, the first step is identical as in the previous section. Next comes the discretization in space, which is done following the methodology outlined in Section 3.3. Again, piecewise linear (and piecewise continuous) representation of the discrete solution 5.1 is assumed. The following fully discrete formulation is obtained:

$$\left\{\begin{aligned} \int_E \frac{\delta u^1}{\Delta t}\psi_i \, d\Omega + \int_E \nabla\cdot\mathbf{f}_h(u^n)\psi_i \, d\Omega - \sum_{e\in E}\int_e \left(\hat{\mathbf{f}}_{E,e} - \mathbf{f}_h\right)(u^n)\ \psi_i\cdot\mathbf{n}_{E,e}\, d\Gamma \ &= \ 0 \\ \int_E \frac{\delta u^{n+1}}{\Delta t}\psi_i \, d\Omega + \frac{1}{2}\int_E \nabla\cdot\mathbf{f}_h(u^n)\psi_i \, d\Omega - \sum_{e\in E}\int_e \left(\hat{\mathbf{f}}_{E,e} - \mathbf{f}_h\right)(u^n)\ \psi_i\cdot\mathbf{n}_{E,e}\, d\Gamma & \\ + \frac{1}{2}\int_E \nabla\cdot\mathbf{f}_h(u^1)\psi_i \, d\Omega - \sum_{e\in E}\int_e \left(\hat{\mathbf{f}}_{E,e} - \mathbf{f}_h\right)(u^1)\, \psi_i\cdot\mathbf{n}_{E,e}\, d\Gamma \ &= 0, \end{aligned}\right. \tag{5.3}$$

for every degree of freedom i. The notation was introduced in Section 3.3 and here only a brief overview is given. Each integrand in the above is multiplied by a test function, ψ_i, which in this case is the Lagrange linear basis function associated with vertex $i \in E$. Both the numerical flux $\hat{\mathbf{f}}_{E,e}$ and the unit outward pointing normal vector $\mathbf{n}_{E,e}$ change their value depending on whether they are considered from within E or E' (see Figure 3.2), and hence the subscript, $_{E,e}$. Recall that in the discontinuous $\mathcal{RKDG}$ framework, each degree of freedom i belongs to only one cell E and hence there is no summation over $\mathcal{D}_i$ in (5.3). As previously, e is used to denote edges.

Two schemes falling into the $\mathcal{RKDG}$ framework will be considered: the RKDG-DG-upwind is the $\mathcal{RKDG}$ approximation for which the upwind flux was used and RKDG-DG-LF is the $\mathcal{RKDG}$ approximation for which the Lax-Friedrichs flux was implemented.

Formulation (5.3) can be rewritten as a discontinuous $\mathcal{RKRD}$ scheme. Indeed, the distribution coefficients are given in Equation (3.8) and the mass matrix can be calculated by using (4.4) and taking $\beta_i = \frac{1}{3}$ (equal to the actual distribution coefficients when the advection velocity is constant). This algorithm can therefore be viewed as a particular example of discontinuous $\mathcal{RKRD}$ discretization. This will become even more apparent after introducing the so-called *alternative* basis functions.

5.3.2 Alternative Basis Functions

Coming back to the relation between residual distribution and finite element type approximations (in this thesis the discontinuous Galerkin method is considered as such), it is interesting to note that the discussion carried out in Section 2.3 can be further extended and lead to somewhat surprising conclusions. To this end, observe that in every cell E one can consider a set of alternative basis functions $\psi_i^{\mathcal{RD}}$ defined as:

$$\psi_i^{\mathcal{RD}} = \psi_i + \alpha_i^E. \tag{5.4}$$

In the above expression α_i^E is a weighting coefficient yet to be specified. These functions were already introduced in Section 3.6.4 where the m1ED distribution for edge residuals was defined (α_i^E was then set to $\beta_i^{\mathcal{RD}} - \frac{1}{3}$). Here, finally, the reasoning that led to the introduction of $\psi_i^{\mathcal{RD}}$ is given. $\psi_i^{\mathcal{RD}}$ is assumed to satisfy the following relation:

$$\beta_i\,\phi^E = \int_E \nabla\cdot\mathbf{f}(u_h)\,\psi_i^{\mathcal{RD}}\,d\Omega,$$

where β_i is the $\mathcal{RD}$ distribution coefficient corresponding to node i in cell E. α_i^E can be calculated quite straightforwardly by taking the above and writing:

$$\beta_i\,\phi^E \;=\; \int_E \nabla\cdot\mathbf{f}(u_h)\,\psi_i\,d\Omega \;+\; \alpha_i^E\,\phi^E \quad\Longrightarrow\quad \alpha_i^E \;=\; \beta_i \;-\; \beta_i^{\mathcal{FE}}.$$

The finite element distribution coefficients, $\beta_i^{\mathcal{FE}}$, were introduced in Section 2.3. In the case of linear equations with constant advection velocity, Formula (C.1) reduces to (distribution coefficients for the finite element method are all equal to $\frac{1}{3}$ in this case):

$$\psi_i^{\mathcal{RD}} \;=\; \psi_i \;+\; \beta_i^{\mathcal{RD}} \;-\; \frac{1}{3}. \tag{5.5}$$

Note that expression (5.5) is identical to the formula that was used in the definition of the m1ED distribution strategy considered in Chapter 3. It will be used as the definition of $\psi_i^{\mathcal{RD}}$ regardless of the equation being solved.

The above reasoning is quite standard in the $\mathcal{RD}$ community. März et al. [73] used an identical technique to first derive $\psi_i^{\mathcal{RD}}$ and then to calculate the mass matrix (4.4). To be more precise, Formulation (4.4) results from the evaluation of the following integrals:

$$m_{ij}^E = \int_E \psi_i\,\psi_j^{\mathcal{RD}}\,d\Omega.$$

Similar approach was used in Chapter 3 to construct the m1ED splitting for edge residuals.

5.3.3 Equivalence of the discontinuous $\mathcal{RKRD}$ and $\mathcal{RKDG}$ approximations

The most interesting consequences of the derivation of $\psi_i^{\mathcal{RD}}$, at least from the point of view of this thesis, are related to the current scenario, i.e. discontinuous-in-space $\mathcal{RKRD}$ discretisations. To see this, choose any $\mathcal{RD}$ distribution coefficients β_i to distribute cell residuals and the corresponding m1ED splitting to distribute edge residuals (this technique was outlined in Chapter 3). This leads to the following discontinuous $\mathcal{RKRD}$ scheme:

$$\left\{\begin{aligned} \sum_{j\in E} m_{ij}^E \frac{\delta u_j^1}{\Delta t} + \beta_i \phi^E(u^n) - \sum_{e\in E}\int_e \Big(\hat{\mathbf{f}}_{E,e} - \mathbf{f}_h\Big)(u^n)\,\psi_i^{\mathcal{RD}}\cdot\mathbf{n}_{E,e}\,d\Gamma &= 0,\\ \sum_{j\in E} m_{ij}^E \frac{\delta u_j^{n+1}}{\Delta t} + \frac{1}{2}\left(\beta_i \phi^E(u^n) - \sum_{e\in E}\int_e \Big(\hat{\mathbf{f}}_{E,e} - \mathbf{f}_h\Big)(u^n)\,\psi_i^{\mathcal{RD}}\cdot\mathbf{n}_{E,e}\,d\Gamma\right) &\\ + \frac{1}{2}\left(\beta_i \phi^E(u^1) - \sum_{e\in E}\int_e \Big(\hat{\mathbf{f}}_{E,e} - \mathbf{f}_h\Big)\left(u^1\right)\,\psi_i^{\mathcal{RD}}\cdot\mathbf{n}_{E,e}\,d\Gamma\right) &= 0. \end{aligned}\right. \tag{5.6}$$

In what follows it will also be referred to as the RKRD-m1ED method. In light of the above observation, it can be rewritten as:

$$\left\{\begin{aligned} &\int_E \frac{\delta u^1}{\Delta t}\psi_i^{\mathcal{RD}}\,d\Omega + \\ &\qquad + \int_E \nabla\cdot\mathbf{f}_h(u^n)\psi_i^{\mathcal{RD}}\,d\Omega - \sum_{e\in E}\int_e \Big(\hat{\mathbf{f}}_{E,e} - \mathbf{f}_h\Big)(u^n)\;\psi_i^{\mathcal{RD}}\cdot\mathbf{n}_{E,e}\,d\Gamma\\ &\qquad = 0\\ &\int_E \frac{\delta u^{n+1}}{\Delta t}\psi_i^{\mathcal{RD}}\,d\Omega + \\ &\qquad + \frac{1}{2}\left(\int_E \nabla\cdot\mathbf{f}_h(u^n)\psi_i^{\mathcal{RD}}\,d\Omega - \sum_{e\in E}\int_e \Big(\hat{\mathbf{f}}_{E,e} - \mathbf{f}_h\Big)(u^n)\;\psi_i^{\mathcal{RD}}\cdot\mathbf{n}_{E,e}\,d\Gamma\right)\\ &\qquad + \frac{1}{2}\left(\int_E \nabla\cdot\mathbf{f}_h(u^1)\psi_i^{\mathcal{RD}}\,d\Omega - \sum_{e\in E}\int_e \Big(\hat{\mathbf{f}}_{E,e} - \mathbf{f}_h\Big)\left(u^1\right)\;\psi_i^{\mathcal{RD}}\cdot\mathbf{n}_{E,e}\,d\Gamma\right)\\ &\qquad = 0. \end{aligned}\right. \tag{5.7}$$

This formulation is very similar to the $\mathcal{RKDG}$ discretization (5.3), the only difference being the definition of test functions (basis functions are kept unchanged). A natural question arises: what is the relationship between the two formulations? It turns out that they are, in fact, identical, but some extra work is needed to justify this statement. It can be done in three steps. First, however, note that one can, without loss in generality, limit the discussion to one generic cell E. This simplification is possible due to the fact that all the schemes considered in this chapter are localised.

Step 1 In every cell E, Scheme (5.3) gives three separate equations (one for each vertex). Summing them up and using the fact that $\sum_{i\in E}\psi_i^E = 1$, one shows that the discrete solution, u_h, satisfies:

$$\begin{cases} \displaystyle\int_E \frac{\delta u^1}{\Delta t}\,d\Omega + \int_E \nabla\cdot\mathbf{f}_h(u^n)\,d\Omega - \sum_{e\in E}\int_e \left(\hat{\mathbf{f}}_{E,e}-\mathbf{f}_h\right)(u^n)\cdot\mathbf{n}_{E,e}\,d\Gamma = 0 \\ \displaystyle\int_E \frac{\delta u^{n+1}}{\Delta t}\,d\Omega + \frac{1}{2}\int_E \nabla\cdot\mathbf{f}_h(u^n)\,d\Omega - \sum_{e\in E}\int_e \left(\hat{\mathbf{f}}_{E,e}-\mathbf{f}_h\right)(u^n)\cdot\mathbf{n}_{E,e}\,d\Gamma \\ \displaystyle\qquad + \frac{1}{2}\int_E \nabla\cdot\mathbf{f}_h(u^1)\,d\Omega - \sum_{e\in E}\int_e \left(\hat{\mathbf{f}}_{E,e}-\mathbf{f}_h\right)(u^1)\cdot\mathbf{n}_{E,e}\,d\Gamma = 0. \end{cases} \tag{5.8}$$

Applying this procedure to Scheme (5.6) (also in this case the test functions sum up to 1, i.e. $\sum_{i\in E}\psi_i^{\mathcal{RD}} = 1$) shows that also the solution obtained with the aid of Scheme (5.6) satisfies Formulation (5.8).

Step 2 The approximate solution $u_h^{\mathcal{DG}}$ obtained with the aid of the $\mathcal{RKDG}$ scheme, i.e. Formulation (5.3), satisfies Formulation (5.6). This follows from the fact that every equation in (5.6) can be written as a linear combination of the corresponding equation in (5.3) and the corresponding equation in (5.8) multiplied by $\left(\beta_i^{\mathcal{RD}}-\frac{1}{3}\right)$ (cf. Equation (5.5)). Both are satisfied by $u_h^{\mathcal{DG}}$. Recall that β_i are the distribution coefficients, which are constant in every cell.

Step 3 Finally, solutions to $\mathcal{RKRD}$ approximation (5.6) and $\mathcal{RKDG}$ approximation (5.3) are unique. This follows from the non-singularity of the corresponding local mass matrices, given by (cf. Formulation (4.4)):

$$\mathbf{M}^{\mathcal{RKDG}} = |E|\begin{bmatrix} \frac{1}{6} & \frac{1}{12} & \frac{1}{12} \\ \frac{1}{12} & \frac{1}{6} & \frac{1}{12} \\ \frac{1}{12} & \frac{1}{12} & \frac{1}{6} \end{bmatrix}, \quad \mathbf{M}^{\mathcal{RKRD}} = \frac{|E|}{36}\begin{bmatrix} 2+12\beta_1 & 12\beta_1-1 & 12\beta_1-1 \\ 12\beta_2-1 & 2+12\beta_2 & 12\beta_2-1 \\ 12\beta_3-1 & 12\beta_3-1 & 2+12\beta_3. \end{bmatrix},$$

The determinants of these matrices are equal to:

$$\det \mathbf{M}^{\mathcal{RKDG}} \;=\; 3|E|, \qquad \det \mathbf{M}^{\mathcal{RKRD}} \;=\; 3|E|(\,\beta_1 \,+\, \beta_2 \,+\, \beta_3\,).$$

Since the solution to (5.3) and to (5.6) are unique, and the solution to (5.3) also solves (5.6), then the two formulations are indeed identical. This assertion also holds in the steady-state case. The above result was verified numerically, i.e. formulations (5.3) and (5.6) gave identical results. It is very important as among all investigated discontinuous $\mathcal{RKRD}$ schemes, only the discontinuous RKRD-m1ED/$\mathcal{RKDG}$ scheme is genuinely second order accurate (this is investigated with detail in Section 5.5). This also explains why the m1ED distribution leads to discretisations producing spurious oscillations. Indeed, the above discussion shows that choosing the m1ED edge distribution means that in fact the discontinuous Galerkin method without limiting is being used. In [23] it was shown that this method is second order accurate and this is confirmed experimentally in Section 5.5. According to the famous Godunov Theorem [49] such method will not be positive. Again, this is confirmed experimentally in Section 5.5.

5.3.4 Equivalence of the mED and DG-upwind Distribution Strategies

Yet, one more quite striking observation can be made with regard to similarities between the discontinuous $\mathcal{RKRD}$ and $\mathcal{RKDG}$ frameworks. Recall that piece-wise linear representation of the approximate solution is assumed throughout this thesis. This means that the resulting schemes can be at most second order accurate (super-convergence is not taken into account here) and such accuracy rate should be regarded as optimal. Constructing an algorithm that does exhibit such accuracy rate is one of the main challenges here. This goal cannot be achieved without, first, identifying and then satisfying conditions that will guarantee the desired result. One of the more natural conditions for accuracy is a constraint on the quadrature rules used to evaluate the integrals appearing in the numerical scheme. These cannot always be computed exactly and the procedure used to evaluate their approximations should be accurate enough not to spoil the overall order of accuracy. As explained in Section 2.5, this is not an issue in the case of $\mathcal{RD}$ discretisations (as far as the test cases considered in this thesis are concerned), but the situation becomes a bit more complicated with discontinuous Galerkin methods for which integrands are usually a product of two functions. For instance, for a generic cell E the signal for vertex

$i \in E$ resulting from the $\mathcal{DG}$ discretization is given by:

$$\phi_i^{\mathcal{DG}} = \int_E \nabla \cdot \mathbf{f}_h \, \psi_i \, d\Omega.$$

This corresponds to the following signal resulting from the $\mathcal{RD}$ scheme:

$$\phi_i^{\mathcal{RD}} = \beta_i \int_E \nabla \cdot \mathbf{f}_h \, d\Omega.$$

It is clear enough that evaluating the first integral is somewhat more involved. According to Cockburn and Shu (see Theorem 2.10 in reference [23]), in the case of piece-wise linear approximations, if the quadrature rule over the edges is exact for polynomials of degree 3, and the quadrature rule over the elements is exact for polynomials of degree 2 then the resulting scheme is second order accurate. In this thesis Gaussian quadrature rules were implemented (2-point for the edges and 3-point for cells) to guarantee accuracy. Consider, however, the following scenario. Let the advection velocity be constant (also recall that all polynomial approximations in this thesis are piecewise linear, see also Section 3.5 to recall the type of fluxes considered in the context of discontinous schemes) and set the numerical flux in (3.20) to be the upwind flux. Furthermore, use the trapezium rule, which is not exact for third order polynomials, to evaluate edge integrals in (5.3). Noting that ψ_i is equal to 1 at one end of the considered edge e and 0 on the other, and that for the constant advection equation and the upwind flux the following holds:

$$\left(\hat{\mathbf{f}}_{E,e} - \mathbf{f}_h\right)(u_i) \cdot \mathbf{n}_{E,e} \;=\; [\mathbf{a} \cdot \mathbf{n}_{E_R,e}]^+ \, |e| (u_{i,L} - u_{i,R}),$$

one can, by direct calculations, show that within edge $e \in \mathcal{T}_h$ the resulting signals will be given by (notation as in Figure 3.1):

$$\begin{aligned}
\phi_1^{DG-upwind-TR} &= \frac{1}{2} [\mathbf{a} \cdot \mathbf{n}_{E_R,e}]^+ \, |e| (u_1 - u_2) = \alpha_1 \, \phi^e, \\
\phi_2^{DG-upwind-TR} &= \frac{1}{2} [\mathbf{a} \cdot \mathbf{n}_{E_R,e}]^- \, |e| (u_1 - u_2) = \alpha_2 \, \phi^e, \\
\phi_3^{DG-upwind-TR} &= \frac{1}{2} [\mathbf{a} \cdot \mathbf{n}_{E_R,e}]^- \, |e| (u_4 - u_3) = \alpha_3 \, \phi^e, \\
\phi_4^{DG-upwind-TR} &= \frac{1}{2} [\mathbf{a} \cdot \mathbf{n}_{E_R,e}]^+ \, |e| (u_4 - u_3) = \alpha_4 \, \phi^e.
\end{aligned} \tag{5.9}$$

This distribution strategy (i.e. DG-upwind evaluated using the trapezium rule) will be referred to as the DG-upwind-TR scheme. For constant advection equation it is identical to the mED distribution presented in Section 3.6 (see Equation (3.17)) as the averaged advection velocities used in the definition of the mED splitting are equal and satisfy:

$$\tilde{\mathbf{a}}_{12} \;=\; \tilde{\mathbf{a}}_{43} \;=\; \mathbf{a}.$$

Note that this is the case regardless of whether the mathematical problem being considered is steady or transient. In the general case when the advection velocity is not constant, the DG-upwind-TR scheme is given by:

$$\begin{aligned}
\phi_1^{DG-upwind-TR} &= \frac{1}{2}\,[\mathbf{a}_1 \cdot \mathbf{n}_{E_R,e}]^+\,|e|(u_1 - u_2) = \alpha_1\,\phi^e,\\
\phi_2^{DG-upwind-TR} &= \frac{1}{2}\,[\mathbf{a}_2 \cdot \mathbf{n}_{E_R,e}]^-\,|e|(u_1 - u_2) = \alpha_2\,\phi^e,\\
\phi_3^{DG-upwind-TR} &= \frac{1}{2}\,[\mathbf{a}_3 \cdot \mathbf{n}_{E_R,e}]^-\,|e|(u_4 - u_3) = \alpha_3\,\phi^e,\\
\phi_4^{DG-upwind-TR} &= \frac{1}{2}\,[\mathbf{a}_4 \cdot \mathbf{n}_{E_R,e}]^+\,|e|(u_4 - u_3) = \alpha_4\,\phi^e.
\end{aligned} \tag{5.10}$$

The advection velocity $\mathbf{a}_k$ $(k = 1, 2, 3, 4)$ is simply $\mathbf{a}$ evaluated at $\mathbf{x}_k \in e$ (see Figure 3.1 for notation).

As already mentioned, the trapezium rule is *not* exact for polynomials of order 3, but according Cockburn and Shu the third order accurate quadrature rule is only a sufficient condition for accuracy, not a necessary one. Interestingly enough, numerical results in Section 5.5 show that it is usually possible to get a second order scheme when the trapezium rule is used. Not in all situations, though! Similar behaviour is observed when the mED distribution strategy is used, i.e. the resulting discretization is second order accurate, but only in the particular situations when the flow is not aligned with the mesh. For a fuller discussion see Section 5.5. Note that the above remains in agreement with the results of Cockburn and Shu [23], i.e. as long as the quadrature rules are accurate enough then the accuracy of order two is guaranteed.

5.4 The Discontinuous Unsteady Residual Distribution Framework

It is worth mentioning that as in the case of continuous $\mathcal{RD}$ methods for time-dependent problems, one is free to use a simplified procedure to integrate in time. One natural choice would be the forward Euler time-stepping procedure used in steady-state computations (cf. Schemes (2.8) and (3.3)). The resulting approximation reads:

$$u_i^{n+1} = u_i^n - \frac{3\Delta t}{|E|}\left(\beta_i\phi^E + \sum_{e\in E}\alpha_i\phi^e\right) \qquad \forall i. \tag{5.11}$$

Obviously such an approach leads to, at most, first order methods. It is, though, suitable for construction of positive schemes. As such it was examined in [105]. Those results are presented and summarized in Section 5.5. Hereafter this approach will be referred to as the discontinuous unsteady residual distribution scheme. Particular methods developed within this framework will be referred to as: discontinuous unsteady A-B method in which A and B stand for the distribution strategy for cells and edges, respectively.

5.5 Numerical Results

This section is devoted to a thorough experimental examination of the numerical frameworks introduced in this chapter. There are two main goals here. First is to assess the framework of discontinuous unsteady schemes by showing that indeed they are first order accurate and, more importantly, that they facilitate construction of a positive scheme. The second goal is to examine the discontinuous $\mathcal{RKRD}$ framework by showing that the resulting discretizations are second order accurate.

From the point of view of possible applications, the scenario considered here is identical to the one investigated in Chapter 4 (similar mathematical models), and hence an identical set of test cases (and corresponding grids) was used. The time step was calculated using Formulae (4.17)–(4.18) with the CFL number set to 0.3, or 0.1 if a particular test case was unstable for $CFL > 0.1$. These values, as in the case of all experiments carried out in this thesis, were chosen empirically. All errors presented here were measured using the L^2 norm, results in the L^1 and L^∞ norms being qualitatively similar.

Only linear distribution strategies were considered here. In the case of discontinuous unsteady $\mathcal{RD}$ methods a non-linear splitting would only complicate the scheme not being able to offer any benefits (the scheme will remain at most first order regardless the distribution strategy). In such a case the distribution strategy for cells was kept fixed (with one exception when the LDA scheme was used) and set to the N scheme. As observed in Chapter 2, this is the least diffusive linear positive scheme. As far as discontinuous $\mathcal{RKRD}$ schemes are concerned, it has yet to be understood how to incorporate non-linear splittings into this framework so that the resulting approximation is both positive and second order accurate. The blending procedure outlined in Section 4.3.2 led to, at most, a first order scheme exhibiting small oscillations. These results are not presented here. Since out of all linear splittings for cell residuals presented in this thesis only the LDA and SU schemes are linearity preserving, these are the ones that were used to perform experiments for this section. Additionally, the discontinuous Galerkin method was implemented so that discontinuous $\mathcal{RKRD}$ methods can be compared with the $\mathcal{RKDG}$ framework.

Results of the grid convergence analysis for the unsteady discontinuous $\mathcal{RD}$ framework are presented in Figure 5.1. The cell residuals were distributed with the aid of the N scheme and for edge residuals four different distribution strategies were implemented: the mED, the LF, the DG-upwind and DG-LF schemes. These splittings were introduced in Chapter 3. Since the way cell residuals were distributed was kept fixed, Figure 5.1 shows how switching from one splitting methodology for edges to another affects accuracy. All the schemes except for the LF distribution exhibit first order accuracy even for coarser meshes. The order of accuracy of the LF scheme, estimated with the aid of errors for the two finest meshes, was 0.68. In all cases the CFL number was set to 0.3.

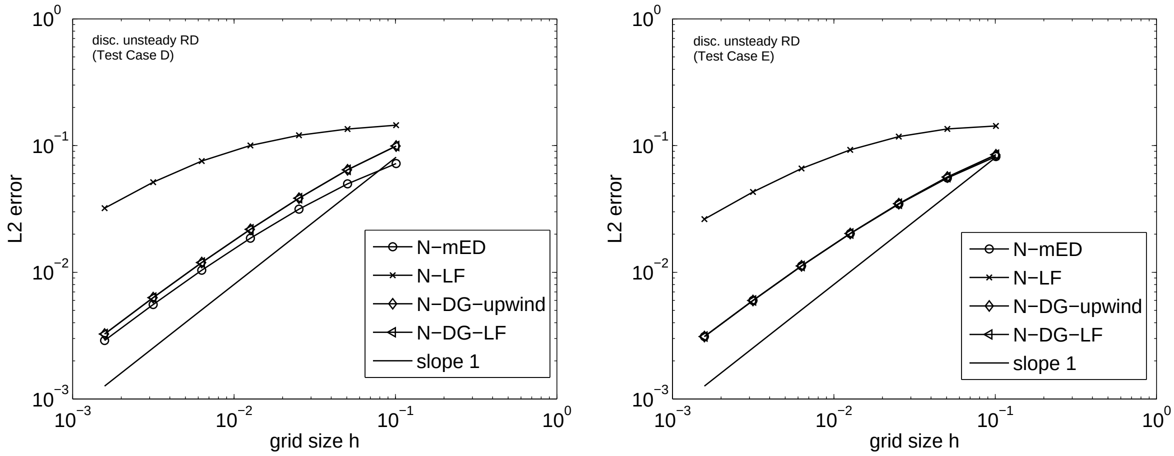


Figure 5.1: Grid convergence for the discontinuous unsteady $\mathcal{RD}$ framework for Test Cases D (left) and E (right). The cell residuals were distributed with the aid of the N scheme and the mED, LF, DG-upwind and DG-LF schemes were utilised to split the edge residuals. All schemes apart from the LF distribution gave similar results and hence some of the plots overlap each other.

The grid convergence analysis for the discontinuous $\mathcal{RKRD}$ framework gave somehow less expected results. These are presented on Figure 5.2. The cell residuals were distributed with the aid of three different methods: the LDA, SU and DG. To distribute the edge residuals the

- mED, m1ED-upwind, m1ED-LF (in the case of the LDA and SU schemes),

- DG-upwind, DG-LF, DG-upwind-TR (in the case of DG cell distribution)

strategies were used. As switching from the upwind to the LF flux caused only minor quantitative alterations to the solution, only results for the former are shown. This is to make the presentation clearer. The results reveal that when the Test Case E (circular advection) was used all schemes, as expected, exhibit second order accuracy. This, however, is no longer the case when Test Case D (constant advection) is used. The RKRD-LDA-m1ED, RKRD-SU-m1ED and $\mathcal{RKDG}$ schemes are indeed second order accurate (and give identical results, which, according to observations made in Section 5.3, was expected). On the other hand, the RKRD-LDA-mED, RKRD-SU-mED and RKDG-DG-upwind-TR are only first order accurate. Recall from Section 4.4 that the direction of the flow is aligned with the mesh when Test Case D is considered. In order to develop a better understanding of this phenomenon, a series of further experiments were carried out. Figure 5.3 shows the results of grid convergence analysis conducted for the RKDG-DG-upwind-TR (left) and RKRD-LDA-mED (right) schemes on a set of test cases generated by modifying Test Case D, i.e. by altering the advection velocity between $\mathbf{a} = (1, 0.1)$ (not aligned with the mesh) and $\mathbf{a} = (1, 0.0005)$ (almost aligned with the mesh). Note that the mesh edges are aligned with 3 distinct vectors:

$$\mathbf{v_1} = (1,0), \qquad \mathbf{v_2} = (1,1) \quad \text{and} \quad \mathbf{v_3} = (0,1).$$

The results show that the two implemented schemes exhibit qualitatively identical behaviour, i.e. the order of accuracy is closer to one for advection velocities close to $\mathbf{a} = (1, 0)$ (aligned with the mesh) and becomes gradually two when one moves away from this velocity. This may at first strike as unexpected behaviour, but one should bear in mind that the theory for discontinuous Galerkin methods covers only scenarios in which the quadrature rules are accurate enough. Using the trapezium rule to integrate the edge residuals, i.e. selecting the DG-upwind-TR distribution, means that some integrals in the discrete formulation are under-integrated. Therefore, counter-intuitive results are in this case possible. Furthermore, the fact that the mED distribution is so similar to the DG-upwind-TR distribution (see Section 5.3) suggest that similar behaviour in the case of the RKRD-LDA-mED scheme should not surprise. Apart from the situations in which the mED or DG-upwind-TR schemes were used, to carry out the above experiments the CFL number was set to 0.3. The mED or DG-upwind-TR splittings were prone to instabilities and the CFL number was decreased to 0.1 to obtain the results.

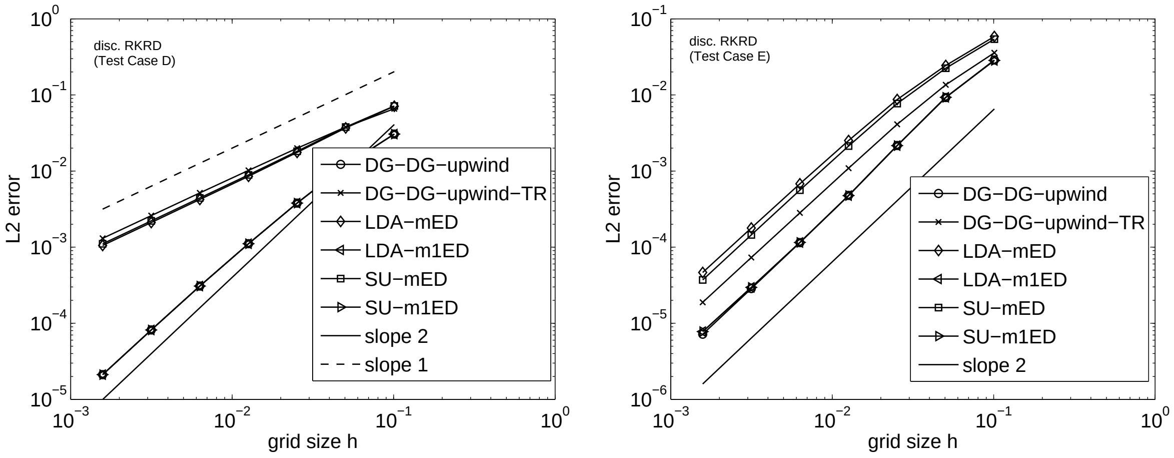


Figure 5.2: Grid convergence for the discontinuous $\mathcal{RKRD}$ framework for Test Cases D (left) and E (right). The DG, LDA and SU schemes were used to distribute cell residuals. These were combined with different splittings for the edges. The DG-upwind and m1ED splittings (combined with the DG and LDA/SU schemes, respectively) were used to guarantee convergence of order two. The DG-upwind-TR and mED splittings (again, for the DG and LDA/SU schemes, respectively) only give second order convergence when the advection velocity is not aligned with the mesh (Test Case E).

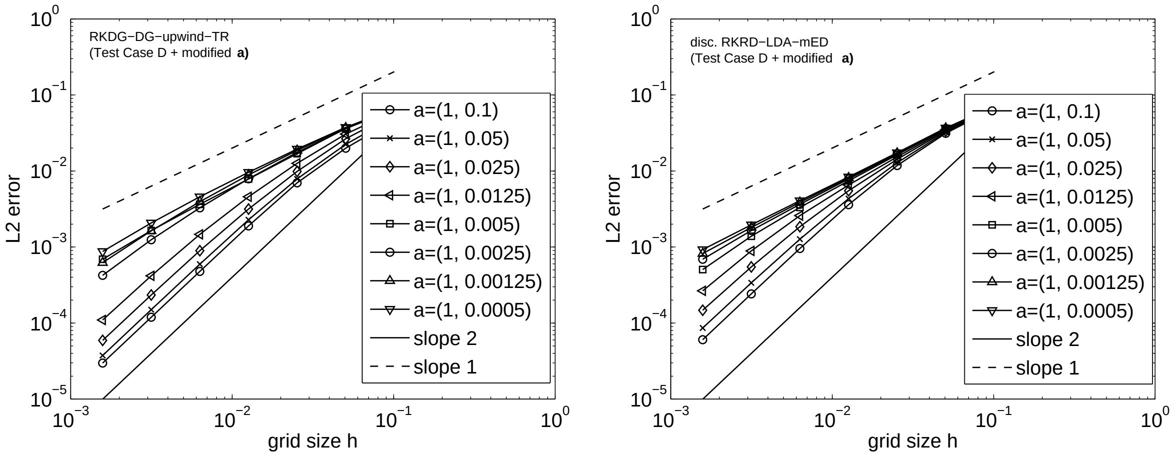
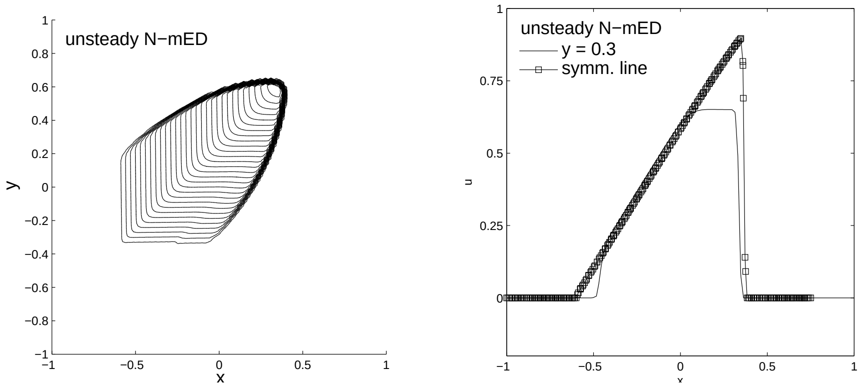


Figure 5.3: Grid convergence for the discontinuous $\mathcal{RKRD}$ framework for Test Case D with modified advection velocity $\mathbf{a}$. The distribution strategy was set to be the DG scheme for cell residuals with the DG-upwind-TR for edge residuals (left) and the LDA scheme combined with the mED splitting (right). In both cases the scheme is first order accurate for $\mathbf{a} = (1.0, 0.0005)$ and becomes gradually second order accurate as $\mathbf{a}$ diverges away from $\mathbf{v_1} = (1.0, 0.0)$.

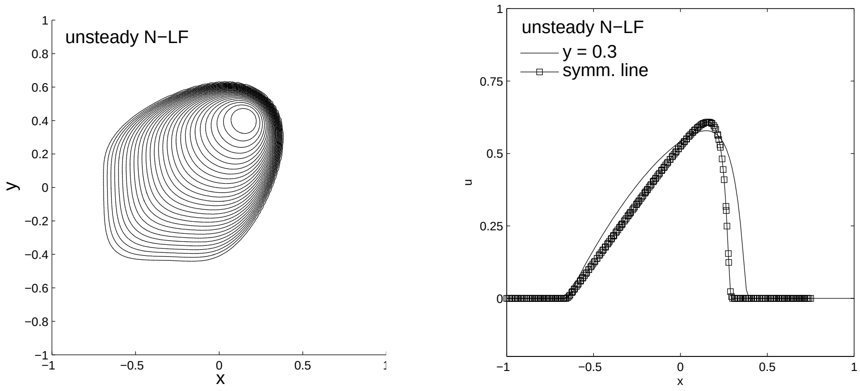
Finally, results for Test Case F, i.e. non-linear Burgers' equation, are shown. On Figures 5.4-5.7 one finds the contour lines and cross-section of approximate solutions obtained with the aid of the discontinuous unsteady N scheme combined with the mED, LF, DG-LF and DG-upwind splittings for edge residuals. As in Chapter 3, the mED distribution turns out to give best results, i.e. the solution is positive (no spurious oscillations) and is only mildly diffusive. The solution given by the LF scheme is also positive, but much more diffusive than the one obtained with the mED scheme. Neither the DG-LF or the DG-upwind scheme gave a positive solution. This shows that in order to construct a discontinuous positive scheme not only do the cell residuals have to be positive, but also the edge residuals have to be distributed with the aid of a positive splitting. To get an idea of what happens if the cell distribution is not positive and the edge distribution is, the discontinuous unsteady LDA-mED scheme was also implemented. Results presented on Figure 5.8 clearly show that the resulting discretization is not positive, though the solution is physically plausible.

Within the discontinuous $\mathcal{RKRD}$ framework only schemes for which the first order N scheme was used gave a plausible answer (shown on Figure 5.9). To the author's best knowledge, discontinuous Galerkin approximations considered in the literature have always been implemented with a limiting procedure. All available results indeed indicate that this approximation should give a plausible solution for Burgers' equation *as long as* a limiting procedure is incorporated. Here, however, such procedure was not included and the solution *exploded* before it reached time $t = 1.0$. Again, since no limiting procedure was used such behaviour should not surprise. As outlined in Section 5.3, other second order schemes considered in this chapter are very similar to discontinuous Galerkin methods. Such being the case, it comes as no surprise that these discretisations also failed to produce plausible results.



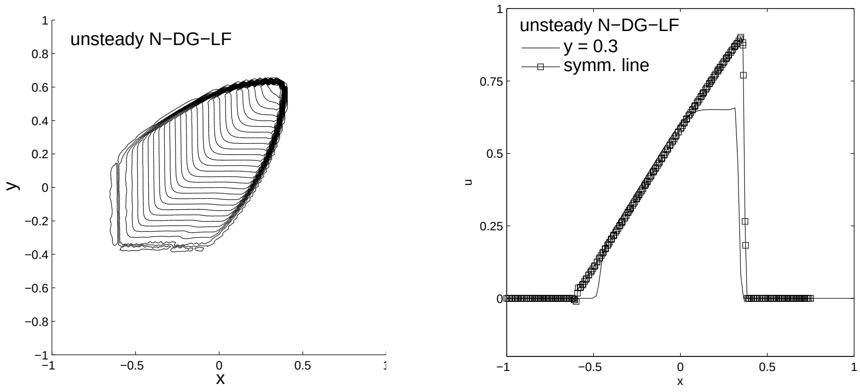
u_{min}	u_{max}
0.0	0.9

Figure 5.4: 2d Burgers' equation: unsteady N-mED scheme with CFL set to 0.3. Left: contours at time $t = 1$. Middle: solution along line $y = 0.3$ and along the symmetry line. Right: minimum and maximum values of the solution.



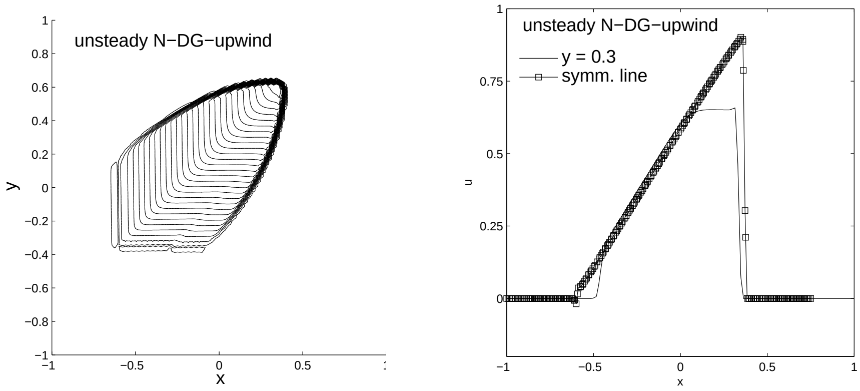
u_{min}	u_{max}
0.0	0.609

Figure 5.5: 2d Burgers' equation: unsteady N-LF scheme with CFL set to 0.3. Left: contours at time $t = 1$. Middle: solution along line $y = 0.3$ and along the symmetry line. Right: minimum and maximum values of the solution.



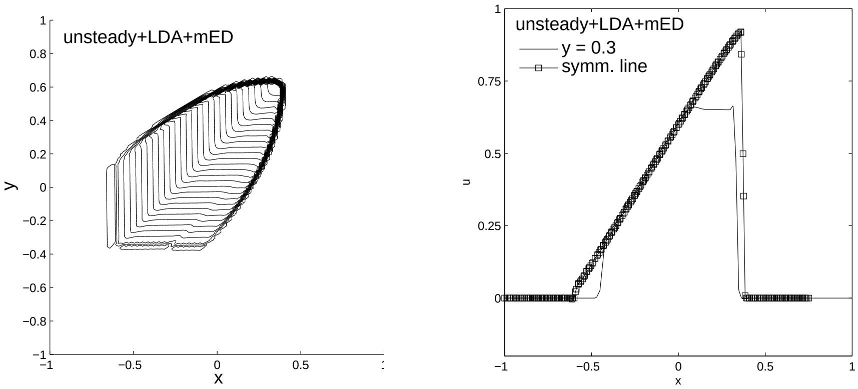
u_{min}	u_{max}
-0.09	0.925

Figure 5.6: 2d Burgers' equation: unsteady N-DG-LF scheme with CFL set to 0.3. Left: contours at time $t = 1$. Middle: solution along line $y = 0.3$ and along the symmetry line. Right: minimum and maximum values of the solution.



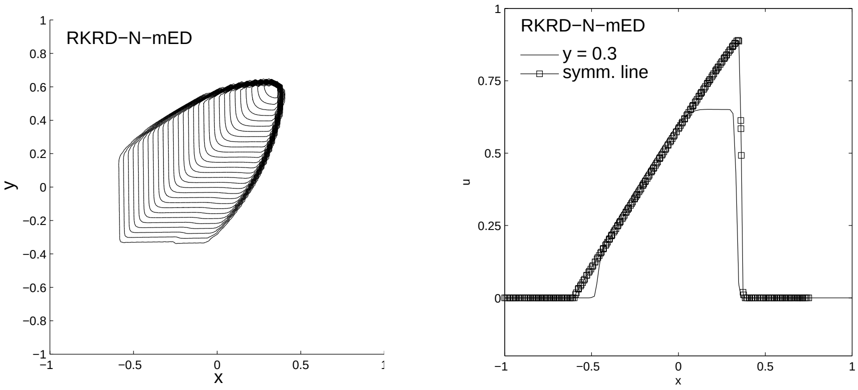
u_{min}	u_{max}
-0.073	0.926

Figure 5.7: 2d Burgers' equation: unsteady N-DG-upwind scheme with CFL set to 0.3. Left: contours at time $t = 1$. Middle: solution along line $y = 0.3$ and along the symmetry line. Right: minimum and maximum values of the solution.



u_{min}	u_{max}
-0.076	0.962

Figure 5.8: 2d Burgers' equation: unsteady LDA-mED scheme with CFL set to 0.3. Left: contours at time $t = 1$. Middle: solution along line $y = 0.3$ and along the symmetry line. Right: minimum and maximum values of the solution.



u_{min}	u_{max}
0.0	0.894

Figure 5.9: 2d Burgers' equation: discontinuous RKRD-N-mED scheme with CFL set to 0.3. Left: contours at time $t = 1$. Middle: solution along line $y = 0.3$ and along the symmetry line. Right: minimum and maximum values of the solution.

5.6 Summary

In this chapter two new frameworks for solving time-dependent hyperbolic PDEs were presented. Both assume that the underlying representation of the approximate solution is piece-wise linear, but not globally continuous. In this respect the methods presented here are very similar to discontinuous Galerkin approximations.

The discontinuous unsteady $\mathcal{RD}$ framework enables construction of at most first order accurate schemes. However, it leads to discretisations which are much simpler (and cheaper) than their counterparts developed within the discontinuous $\mathcal{RKRD}$ framework. Moreover, as presented in the previous section, the discontinuous unsteady N-mED scheme is less diffusive than the discontinuous RKRD-N-mED method. It is, therefore, a quite interesting alternative and can be considered in future as a building block of higher than first order positive schemes.

The discontinuous $\mathcal{RKRD}$ framework facilitates construction of linear second order accurate schemes (by, for instance, taking the LDA and m1ED-DG-upwind distributions) and linear positive schemes (by, for instance, taking the N and mED distributions). The resulting discretisations are only first order accurate when the flow is aligned with the mesh and the mED scheme is used to distribute edge residuals (regardless of the way cell residuals are treated). On the other hand, in more interesting and realistic scenarios when the flow is *not* aligned with the mesh the order of accuracy is indeed two. Construction of a scheme that would be both positive and linearity preserving remains an open question. Also, extension to non-linear equations has yet to be investigated. The discontinuous $\mathcal{RKRD}$ schemes (unless the N scheme is used) *explode* almost immediately after the simulation begins. This was observed regardless of the choice of the distribution strategy for edge based residuals. The fact that this framework in general fails to give plausible solutions when applied to nonlinear equations (again, unless the N scheme is used) is disappointing. Especially, when compared to continuous $\mathcal{RKRD}$ methods discussed in Chapter 4. On the other hand, investigating the discontinuous $\mathcal{RKRD}$ framework revealed further similarities between the discontinuous residual distribution and discontinuous Galerkin approaches. This implies that the two should be considered as one framework rather than competing ways of discretising PDEs. Note that there are no indications (mathematical or experimental) that the discontinuous Galerkin method *without* limiting should give plausible solution to non-linear equations. Such being the case, the discontinuous $\mathcal{RKRD}$ framework cannot be expected to work well for such problems.

To briefly summarize, the main contributions of this chapter are the developments of:

- discontinuous unsteady $\mathcal{RD}$ methods for linear and non-linear equations (first order and positive provided appropriate distributions are implemented);
- discontinuous $\mathcal{RKRD}$ methods for linear equations (linearity preserving);
- the discontinuous RKRD-N method for non-linear equations (positive and first order);
- better understanding of the discontinuous $\mathcal{RD}$ framework, in particular its close links with discontinuous Galerkin methods.

Even though the work on discontinuous $\mathcal{RKRD}$ methods is not complete, the discussed close relation between the discontinuous Galerkin framework and the discontinuous residual distribution framework give some indication what the potential next steps could be. One possibility is to incorporate a $\mathcal{DG}-$type limiting procedure into the discontinuous $\mathcal{RKRD}$ framework. Another interesting extension to the framework would be a genuinely second order accurate (for flows both aligned and not aligned with the mesh) *and* positive distribution strategy for edge residuals.

Chapter 6

The Euler Equations

6.1 Introduction

Thus far, only scalar equations have been considered. It is, however, the desire to tackle more realistic problems captured by systems of non-linear equations that drives the development of new numerical schemes. This chapter is devoted to numerical investigation of residual distribution schemes, introduced earlier in this thesis, when applied to the Euler equations – the system of non-linear hyperbolic partial differential equations for which the $\mathcal{RD}$ framework was originally incepted.

The compressible Euler equations modelling dynamics of inviscid fluids, one of the most important and sound mathematical models in fluid dynamics, have been thoroughly studied, both mathematically and numerically, in a number of monographs, i.e. [46, 47, 65, 67, 72] or [48], to name just a few. Here, only a brief discussion of the equations is given the focus being laid on solving them numerically. In particular with the aid of residual distribution methods. The system can be written in a vector form as

$$\partial_t \mathbf{w} + \nabla \cdot \mathbf{F} = \mathbf{0} \tag{6.1}$$

in which $\mathbf{w}$ is the vector of conserved variables and $\mathbf{F} = (\mathbf{g}, \mathbf{h})$ are the conservative fluxes. In the two-dimensional setting, i.e. in $\mathbb{R}^2$, these are given by:

$$\mathbf{w} = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ E_{total} \end{pmatrix}, \qquad \mathbf{g} = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ u(p + E_{total}) \end{pmatrix} \qquad \mathbf{h} = \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ v(p + E_{total}) \end{pmatrix}.$$

In the above u and v are the x and y components of the velocity, respectively. The total energy E_{total} is related to the other quantities by a state equation which, for a perfect gas, takes the form:

$$E_{total} = \frac{p}{\gamma - 1} + \frac{1}{2}\rho\left(u^2 + v^2\right).$$

Here γ is the ratio of specific heats (the *Poisson adiabatic constant*) and p is the pressure. Only the case of air will be considered, that is $\gamma = 1.4$.

6.2 The Parameter Vector of Roe

Depending on the set of independent variables used, the Euler equations (6.1) can take different forms. Although mathematically equivalent, the way the resulting set of equations is solved numerically differs.

Conserved variables, introduced in the previous section, are the most natural choice from the point of view of mechanics. Another commonly considered variant is primitive variables $\mathbf{v} = (\rho, u, v, p)^T$, which, at least at the first glance, may seem to be easier to work with as the momentum (which depends on two primitive variables) is substituted with the velocity vector. However, when it comes to numerical computations they do not offer anything extra when compared to the conservative variables. As a matter of fact, it is the so called "parameter vector" of Roe [88] that adds the most in terms of numerical integration of the Euler equations. This, yet another set of variables, enables conservative linearisation (discussed in the next section), which facilitates construction of conservative discretisations. It is a very desirable feature, especially when solving systems of nonlinear hyperbolic PDEs solutions to which exhibit discontinuities. Conservation guarantees that those shocks are captured consistently. For this reason the parameter vector of Roe is the most frequently used set of variables in the residual distribution framework. This thesis remains faithful to this trend.

The "parameter vector" of Roe, denoted here by $\mathbf{z}$, is defined by

$$\mathbf{z} = \begin{pmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{pmatrix} = \sqrt{\rho}\begin{pmatrix} 1 \\ u \\ v \\ H \end{pmatrix},$$

where $H = \frac{E_{total}+p}{\rho}$ is the total enthalpy. Its key property is the quadratic dependence of the conservative variables $\mathbf{w}$ on it:

$$\mathbf{w}(\mathbf{z}) = \left(z_1^2, z_1 z_2, z_1 z_3, \frac{z_1 z_4}{\gamma} + \frac{\gamma - 1}{2\gamma}(z_2^2 + z_3^2) \right)^T .$$

The same holds for the fluxes:

$$\begin{aligned}
\mathbf{g}(\mathbf{z}) &= \left(z_1 z_2, z_2^2 + \frac{\gamma - 1}{\gamma} \left[z_1 z_4 - \frac{1}{2}(z_2^2 + z_3^2) \right], z_2 z_3, z_2 z_4 \right), \\
\mathbf{h}(\mathbf{z}) &= \left(z_1 z_3, z_2 z_3, z_3^2 + \frac{\gamma - 1}{\gamma} \left[z_1 z_4 - \frac{1}{2}(z_2^2 + z_3^2) \right], z_3 z_4 \right).
\end{aligned}$$

It follows immediately that the corresponding Jacobians are linear in terms of $\mathbf{z}$:

$$\frac{\partial \mathbf{w}}{\partial \mathbf{z}} = \sqrt{\rho} \begin{pmatrix} 2 & 0 & 0 & 0 \\ u & 1 & 0 & 0 \\ v & 0 & 1 & 0 \\ \frac{1}{\gamma} H & \frac{\gamma-1}{\gamma} u & \frac{\gamma-1}{\gamma} v & \frac{1}{\gamma} \end{pmatrix}$$

in the case of conservative variables, and:

$$\frac{\partial \mathbf{g}}{\partial \mathbf{z}} = \sqrt{\rho} \begin{pmatrix} u & 1 & 0 & 0 \\ \frac{\gamma-1}{\gamma} H & \frac{\gamma+1}{\gamma} u & -\frac{\gamma-1}{\gamma} v & \frac{\gamma-1}{\gamma} \\ 0 & v & u & 0 \\ 0 & H & 0 & u \end{pmatrix}$$

$$\frac{\partial \mathbf{h}}{\partial \mathbf{z}} = \sqrt{\rho} \begin{pmatrix} v & 0 & 1 & 0 \\ 0 & v & u & 0 \\ \frac{\gamma-1}{\gamma} H & -\frac{\gamma-1}{\gamma} u & \frac{\gamma+1}{\gamma} v & \frac{\gamma-1}{\gamma} \\ 0 & 0 & H & u \end{pmatrix}$$

in the case of the fluxes. These rather technical properties enable conservative linearisation, which is one of the key ingredients of the considerable majority of residual distribution methods when applied to the Euler equations.

6.3 Conservative Linearisation

The application of multidimensional upwinding techniques to nonlinear systems of equations such as (6.1) requires the construction of an appropriate discrete form. To ensure that the scheme captures discontinuities accurately, such a discrete formulation should be conservative. The procedure outlined below shows how this can be achieved in practice.

By analogy with the scalar case, the cell residual, Φ^E, lies at the basis of all $\mathcal{RD}$ approximations of (6.1). As previously, it is defined by substituting the numerical solution into the system and integrating it over each cell E :

$$\Phi^E = \int_E \nabla \cdot \mathbf{F}(\mathbf{w}_h)\, d\Omega = \oint_{\partial E} \mathbf{F}(\mathbf{w}_h) \cdot \mathbf{n}\, d\Gamma. \tag{6.2}$$

The subscript $_h$ is suppressed in the remainder of this chapter, though all the terms are understood as approximations of their continuous counterparts.

In order to derive a discrete system approximating (6.1), one has to find an efficient and accurate way of calculating (6.2). Evaluating it in terms of the parameter vector gives:

$$\Phi^E = \int_E \left(\frac{\partial \mathbf{g}}{\partial \mathbf{z}} \mathbf{z}_x + \frac{\partial \mathbf{h}}{\partial \mathbf{z}} \mathbf{z}_y \right) d\Omega. \tag{6.3}$$

Assuming that $\mathbf{z}$ is piece-wise linear (and hence both $\mathbf{z}_x$ and $\mathbf{z}_y$ are piece-wise constant), one can further expand (6.3) as:

$$\Phi^E = \left(\int_E \frac{\partial \mathbf{g}}{\partial \mathbf{z}} d\Omega \right) \mathbf{z}_x + \left(\int_E \frac{\partial \mathbf{h}}{\partial \mathbf{z}} d\Omega \right) \mathbf{z}_y. \tag{6.4}$$

From quadratic dependence of the numerical flux on $\mathbf{z}$ (and hence the linear dependence of the flux Jacobian on it), Φ_E can be evaluated exactly using a one point quadrature rule:

$$\Phi^E = |E| \left(\frac{\partial \mathbf{g}(\bar{\mathbf{z}})}{\partial \mathbf{z}} \mathbf{z}_x + \frac{\partial \mathbf{h}(\bar{\mathbf{z}})}{\partial \mathbf{z}} \mathbf{z}_y \right) \tag{6.5}$$

in which $\bar{\mathbf{z}}$ is taken as the average of the values of $\mathbf{z}$ at the vertices of the corresponding triangle E:

$$\bar{\mathbf{z}} = \frac{\mathbf{z}_1 + \mathbf{z}_2 + \mathbf{z}_3}{3}, \qquad \text{with } \mathbf{z}_i = \mathbf{z}(\mathbf{x}_i) \text{ and } \mathbf{x}_i \in E. \tag{6.6}$$

Within each cell E, the gradient of $\mathbf{z}$ is constant. Denoting by $\mathbf{n}_i$ the unit outward pointing normal to edge $e_i \in E$ (opposite the i^{th} vertex), it can be calculated using:

$$\nabla \mathbf{z} = -\frac{1}{2|E|} |e_i| \sum_{i=1}^{3} \mathbf{z}_i\, \mathbf{n}_i.$$

Equation (6.5), gives a very simple formula for evaluating cell residuals, but expressed in terms of Roe's parameter vector. A similar formula in terms of the conservative variables would be more practical and natural to work with. This can be achieved by first noting that:

$$\mathbf{z}_x = \frac{\partial \mathbf{z}}{\partial \mathbf{w}} \mathbf{w}_x, \qquad \mathbf{z}_y = \frac{\partial \mathbf{z}}{\partial \mathbf{w}} \mathbf{w}_y.$$

and then showing that the averaged gradient of $\mathbf{w}$:

$$\widehat{\mathbf{w}_x} = \frac{1}{|E|} \int_E \mathbf{w}_x \, d\Omega, \qquad \widehat{\mathbf{w}_y} = \frac{1}{|E|} \int_E \mathbf{w}_y \, d\Omega$$

can be evaluated as:

$$\begin{aligned}
\widehat{\mathbf{w}_x} &= \frac{1}{|E|} \int_E \frac{\partial \mathbf{w}}{\partial \mathbf{z}} \mathbf{z}_x \, d\Omega = \frac{1}{|E|} \int_E \frac{\partial \mathbf{w}}{\partial \mathbf{z}} \, d\Omega \, \mathbf{z}_x = \frac{\partial \mathbf{w}(\bar{\mathbf{z}})}{\partial \mathbf{z}} \mathbf{z}_x, \\
\widehat{\mathbf{w}_y} &= \frac{1}{|E|} \int_E \frac{\partial \mathbf{w}}{\partial \mathbf{z}} \mathbf{z}_y \, d\Omega = \frac{1}{|E|} \int_E \frac{\partial \mathbf{w}}{\partial \mathbf{z}} \, d\Omega \, \mathbf{z}_y = \frac{\partial \mathbf{w}(\bar{\mathbf{z}})}{\partial \mathbf{z}} \mathbf{z}_y.
\end{aligned}$$

It now follows that (6.5) is equivalent to:

$$\Phi^E = |E| \left(\frac{\partial \mathbf{g}(\bar{\mathbf{z}})}{\partial \mathbf{w}} \widehat{\mathbf{w}_x} + \frac{\partial \mathbf{h}(\bar{\mathbf{z}})}{\partial \mathbf{w}} \widehat{\mathbf{w}_y} \right), \tag{6.7}$$

which is the formula that is used in practice.

The linearisation process described above shows how to evaluate the cell residuals Φ^E exactly. This means the procedure outlined here is conservative as:

$$\sum_{E \in \Omega} \Phi^E = \sum_{E \in \Omega} \oint_{\partial E} \mathbf{F}_h \cdot \mathbf{n} \, d\Gamma = \oint_{\Omega} \mathbf{F}_h \cdot \mathbf{n} \, d\Gamma.$$

In other words, the discrete flux balance (summed up over the whole domain) reduces to boundary contributions, even though it is evaluated numerically. It is worth pointing out again that conservation is important as it guarantees that the discontinuities are captured accurately. Consult [38] for further details on this matter.

6.4 Matrix Distribution Schemes

Conservative linearisation discussed in the previous section is simply a tool that is implemented to calculate cell residuals when the underlying system of PDEs being solved is the Euler equations. The next step is to distribute those residuals among the vertices of the given cell *and* degrees of freedom located at each of those vertices (four unknowns per vertex in the case of two-dimensional Euler equations). Originally, this was done with the aid of wave decomposition models, investigated in [35,39,40] and further developed in [75,78,94]. The idea behind this strategy is to decouple the system into distinct transport equations, referred to as waves, travelling with different speeds and in different directions. When full diagonalisation is possible (i.e. for steady two-dimensional Euler equations in the supersonic case), methods developed for scalar equations can be directly applied. Otherwise, other approaches have to be implemented. Nevertheless, wave decomposition significantly simplifies the process of solving the underlying system. This approach, although developed to mimic the underlying physical phenomena as accurately as possible and hence very promising, is not as robust as one would wish. In particular, it does not generalise to time–dependent and three–dimensional problems. Instead, the so called *matrix distribution* approach has been devised [7, 102, 103]. Although not as physically sound as the wave decomposition, the matrix distribution framework proved to be a very robust approach and has become the most popular way of extending residual distribution methods to systems of non-linear hyperbolic equations. In particular, definitions presented here are independent of the underlying system of PDEs being discretized. The only condition is that the underlying system is hyperbolic.

Matrix distribution schemes are constructed by heuristically generalising their scalar counterparts to systems of equations. Only matrix LDA, N, and BLEND schemes will be considered here, all of which are defined with the aid of matrix flow parameters. For every cell $E \in \mathcal{T}_h$, these are defined as (cf. Equation (2.18) in Section 2.6):

$$\mathbf{K}_j \;=\; -\frac{1}{2}\left(\mathbf{A}(\bar{\mathbf{w}}), \mathbf{B}(\bar{\mathbf{w}})\right)\, \mathbf{n}_j\, |e_j|,$$

with $\bar{\mathbf{w}}$ being the cell average of $\mathbf{w}$ (cf. Equation (6.6)) and $\mathbf{A}$ and $\mathbf{B}$ defined as Jacobian matrices of the fluxes:

$$\mathbf{A} = \frac{\partial \mathbf{g}}{\partial \mathbf{w}}, \qquad \mathbf{B} = \frac{\partial \mathbf{h}}{\partial \mathbf{w}}. \tag{6.8}$$

Vector $\mathbf{n}_j$ is the unit normal to edge e_j (opposite the j^{th} vertex) pointing outward from cell E. $|e_j|$ denotes the length of e_j. Note that this definition is consistent with the definition of scalar flow sensors. Indeed, if $\mathbf{f}$ and u from Equation (2.2) are substituted into (6.8) then the resulting quantity will be equal to the scalar flow sensor, k_i, introduced in Section 2.6.

Since the system is hyperbolic, the matrix flow sensor admits real eigenvalues and a complete set of right and left eigenvectors. In other words, it can be diagonalised:

$$\mathbf{K}_j = \mathbf{R}_j \mathbf{\Lambda}_j \mathbf{R}_j^{-1},$$

with $\mathbf{R}_j$ being composed of the right eigenvectors of $\mathbf{K}_j$ and $\mathbf{\Lambda}_j$ containing the corresponding eigenvalues on its diagonal and zero elsewhere. These matrices can be found in, for example, Section 4.3.2 of the monograph by Godlewski and Raviart [48]. The authors also give a very detailed presentation of the conservative linearisation for the two-dimensional Euler equations.

Let now $\lambda_1, \lambda_2, \lambda_3$ and λ_4 denote the non-zero entries of $\mathbf{\Lambda}_j$ (eigenvalues of $\mathbf{K}_j$). The following matrices based on $\mathbf{\Lambda}_j$:

$$\mathbf{\Lambda}_j^+ \ = \ \mathrm{diag}\{\max(0, \lambda_k)\}_{k=1}^4, \qquad \mathbf{\Lambda}_j^- \ = \ \mathrm{diag}\{\min(0, \lambda_k)\}_{k=1}^4,$$

and

$$|\mathbf{\Lambda}_j| \ = \ \mathrm{diag}\,|\lambda_k|_{k=1}^4 \ = \ \mathbf{\Lambda}_j^+ \ - \ \mathbf{\Lambda}_j^-,$$

can now be used to define:

$$\mathbf{K}_j^+ \ = \ \mathbf{R}_j \mathbf{\Lambda}_j^+ \mathbf{R}_j^{-1}, \qquad \mathbf{K}_j^- \ = \ \mathbf{R}_j \mathbf{\Lambda}_j^- \mathbf{R}_j^{-1}, \qquad |\mathbf{K}_j| \ = \ \mathbf{R}_j \, |\mathbf{\Lambda}_j| \, \mathbf{R}_j^{-1}.$$

The above definitions are, again, consistent with the corresponding ones in the scalar case, cf. Equation (2.18). It is worth recalling that for all scalar residual distribution methods/frameworks considered here, the flow sensors are evaluated using only the *previous* (already calculated) solution. This guarantees that the resulting systems of equations are linear. Matrix flow sensors are consistent with their scalar counterparts and hence a similar property holds in the case considered here. Particular matrix distribution schemes can now be presented.

The LDA scheme The split residuals for the matrix LDA scheme are defined as:

$$\boldsymbol{\phi}_i^{LDA} = \boldsymbol{B}_i^{LDA} \boldsymbol{\phi}^E, \qquad \boldsymbol{B}_i^{LDA} = \mathbf{K}_i^+ \mathbf{N}, \qquad \mathbf{N} = \left(\sum_{j \in E} \mathbf{K}_j^+ \right)^{-1},$$

The existence of matrix product $\mathbf{K}_i^+ \mathbf{N}$ was proven in [1, 11].

The N scheme The matrix N scheme is defined by:

$$\boldsymbol{\phi}_i^N \;=\; \mathbf{K}_i^+(\mathbf{w}_i \,-\, \mathbf{w}_{in}), \qquad \mathbf{w}_{in} \;=\; -\mathbf{N}\sum_{j\in E}\mathbf{K}_j^-\mathbf{w}_j,$$

The existence of matrix **N** was proven in [1, 11].

The BLEND scheme The matrix BLEND scheme is given by:

$$\boldsymbol{\phi}_i^{BLEND} \;=\; \boldsymbol{\Theta}\boldsymbol{\phi}_i^N \,+\, (\boldsymbol{I} \,-\, \boldsymbol{\Theta})\boldsymbol{\phi}_i^{LDA},$$

with $\boldsymbol{I}$ the identity matrix. The entries of the non-linear blending matrix $\boldsymbol{\Theta}$ were computing using the following formula:

$$\boldsymbol{\Theta}_{k,k} = \frac{\left|\boldsymbol{\phi}_k^E\right|}{\sum_{i\in E}\left|\boldsymbol{\phi}_{i,k}^N\right| \,+\, \epsilon}, \quad \epsilon = 10^{-15}. \tag{6.9}$$

In expression (6.9), index k refers to the kth equation of the system; i.e., $\boldsymbol{\phi}_k^E$ and $\boldsymbol{\phi}_{i,k}^N$ are the k^{th} components of vectors $\boldsymbol{\phi}^E$ and $\boldsymbol{\phi}_i^N$, respectively [33]. Note that $\boldsymbol{\Theta}$ is a diagonal matrix. Depending on the problem being solved (smooth or exhibiting shocks), one is free to either give preference to the LDA scheme for smooth problems (set all the diagonal values to minimum), or to the N scheme for non-smooth problems (set all the diagonal values to maximum).

The mass matrix (4.4) for systems is derived by applying the procedure outlined in [73] to systems. Since at every vertex $i \in E$ there are four degrees of freedoms, the mass matrix coefficient m_{ij}^E becomes a 4×4 matrix $\mathbf{M}_{ij}^E$ defined as:

$$\mathbf{M}_{ij}^E \;=\; \frac{|E|}{36}(3\delta_{ij}\boldsymbol{I} \,+\, 12\boldsymbol{B}_i^E \,-\, \boldsymbol{I}),$$

in which $\boldsymbol{B}_i^E$ is the corresponding distribution matrix and $\boldsymbol{I}$ is the identity matrix.

Recall that the PSI scheme has not been implemented in the $\mathcal{RKRD}$ framework because it would to lead to a genuinely non-linear discretisation. For this reason it will not be considered in this chapter.

6.5 The Time Step

Local time-stepping was employed in the steady state case. The time step was calculated using a straightforward extension of formula used in the scalar case (cf.

Expression (2.22)):

$$\Delta t_i = \mathrm{CFL}\,\frac{|S_i|}{\sum_{E\in\mathcal{D}_i}\sigma(\mathbf{K}_i^+)} \qquad \forall i \in \mathcal{T}_h,$$

in which $\sigma(\mathbf{K}_i^+)$ denotes the spectral radius of $\mathbf{K}_i^+$, i.e. its maximal eigenvalue. A Courant-Friedrichs-Lewy (CFL) number of 0.9 was used in the interior of the domain for all of the test cases. Boundaries were more sensitive to instabilities and $CFL = 0.25$ was used for nodes located at the boundaries.

For time-dependent problems a formula similar to the one used in Chapter 4 was implemented (cf. Expression (4.17)):

$$\Delta t_i = \mathrm{CFL}\,\frac{|S_i|}{\sum_{E\in\mathcal{D}_i}\alpha^E} \qquad \forall i \in \mathcal{T}_h,$$

with the definition of the α^E coefficient modified to reflect the fact that now systems rather than scalar equations are considered (cf. Formula (68) in [85]):

$$\alpha^E = \frac{1}{2}\max_{j\in E}\left(||\mathbf{u}_j|| + a_j\right).$$

The velocity vector $\mathbf{u}_j = (u_j, v_j)$ is evaluated at vertex $j \in E$ and the speed of sound a_j is given by:

$$a_j = \sqrt{\frac{\gamma p_j}{\rho_j}}. \tag{6.10}$$

In the transient case the CFL number was set to values between 0.9 and 0.05. Precise values are given when the corresponding results are presented.

6.6 The Boundary Conditions

The discussion on boundary conditions is carried out here rather than in one of the earlier chapters on scalar problems as in the latter case straightforward strong imposition of the boundary conditions (see Section 6.6.1) gave good results. Here one additional alternative is reviewed.

The imposition of boundary conditions is, fundamentally, a physical problem, but it must correspond to the mathematical character of the solved equations. Relatively few results are available regarding their mathematical properties, let alone their numerical implementation. A thorough, though not very up to date, review of the available results can be found in [48], Chapter V. When it comes to $\mathcal{RD}$

discretizations the most popular approaches to applying boundary conditions are those developed within the $\mathcal{FV}$ framework. It is no wonder as the former can be recast in the formalism of the latter and vice versa. Very little, however, has been investigated to see how different techniques affect residual distributions schemes. A somehow more systematic approach was presented in the recently published monograph of Ricchiuto [83]. Also in the PhD theses of Guzik [50] and Pailléré [77] one can find a relatively extended discussion on how the boundary conditions can be imposed.

All the available approaches can be grouped into two categories. In order to impose the boundary conditions *strongly* one basically prescribes appropriate unknowns with the desired values. In the *weak* approach the solution at the boundary is considered to be unknown and treated as in the interior of the domain. For so called *ghost* cells located outside the domain additional boundary residuals are defined and corresponding signals are distributed. This is basically an $\mathcal{RD}$ philosophy applied to the boundary. A similar approach is very often used for discontinuous Galerkin approximations [23].

6.6.1 Strong Boundary Conditions

In this approach the far field state vector, $\mathbf{w}_\infty$, is substituted for the numerical solutions, $\mathbf{w}_h$. One should bear in mind, that $\mathbf{w}_\infty$:

$$\mathbf{w}_\infty = \begin{pmatrix} \rho_\infty \\ (\rho u)_\infty \\ (\rho v)_\infty \\ E_\infty \end{pmatrix}$$

has to be specified in such a way that both the underlying mathematical and mechanical problems are well posed. The underlying challenge is to specify the number of unknowns to be prescribed/extrapolated at each boundary edge e. This is usually done by looking at the signs of eigenvalues of

$$\mathbf{C_n} = \mathbf{f} n_x + \mathbf{g} n_y,$$

in which $\mathbf{n} = (n_x, n_y)$ is a unit outward pointing normal vector to e. The eigenvalues of $\mathbf{C_n}$ are given by

$$\lambda_1 = \lambda_2 = \mathbf{u} \cdot \mathbf{n}, \qquad \lambda_3 = \mathbf{u} \cdot \mathbf{n} + a, \qquad \lambda_4 = \mathbf{u} \cdot \mathbf{n} - a$$

where a is the speed of sound (6.10). The rule is that only information coming from outside the computational domain (i.e. with a negative speed) may be imposed at a physical boundary. The remaining information is naturally provided/extrapolated by the upwind scheme used in the interior of the domain. In other words, the number of unknowns to be prescribed is equal to the number of negative eigenvalues of $\mathbf{C}_n$. It can be shown that in the two-dimensional case (see, for instance, [47] for details) there are:

- 4 negative eigenvalues at supersonic inlet;
- 0 negative eigenvalues at supersonic outlet
- 3 negative eigenvalues at subsonic inlet;
- 1 negative eigenvalue at subsonic outlet.

At a solid wall, the slip condition $\mathbf{u} \cdot \mathbf{n} = 0$ implies that only one eigenvalue is positive and hence only one quantity should be prescribed. Usually it is the tangency of the flow itself that is imposed.

A more detailed discussion on the matter of well-posedness of the boundary conditions can be found in [47] and [48]. In both references the authors not only specify *how many* but also stipulate *which* quantities should be prescribed. This, however, is done using different heuristics and it is not clear which of the approaches is most reliable and robust. In this thesis the methodology proposed by Feistauer et al. [47] was implemented:

- prescribe ρ, u, v and p at supersonic inlet;
- extrapolate ρ, u, v and p at supersonic outlet;
- prescribe ρ, u, v and extrapolate p at subsonic inlet;
- prescribe p and extrapolate ρ, u, v at subsonic outlet.

6.6.2 Weak Boundary Conditions

Weak imposition of the boundary conditions was inspired by the well known ‘ghost cell’ technique used in cell-centered finite volumes. It is probably the most robust and faithful to the original $\mathcal{RD}$ concept way of prescribing boundary conditions.

In this approach, boundary nodes are treated in the similar manner as the values from interior of the domain. In other words, a similar *update* procedure is being applied to them. The only difference is the way the signals for boundary nodes are defined, i.e. for each boundary node $i \in \mathcal{T}_h$ and edge $e \in \mathcal{T}_h$, such that $i \in e$, an additional contribution from the boundary residuals, $\Phi^{e,bd}$, is added. This means that at the boundary Γ, the steady state scheme becomes (cf. the scalar scheme (2.7))):

$$\sum_{E \in \mathcal{D}_i} \Phi_i^E + \sum_{e|i \in e} \Phi_i^{e,bd} = 0.$$

Edge residuals are defined in such a way that the signals distributed from each boundary edge e sum up to:

$$\sum_{i \in e} \Phi_i^{e,bd} = \Phi^{e,bd} = \int_e \Big(\hat{\mathbf{f}}_{E,e}(\mathbf{w}_h, \mathbf{w}_\infty, \mathbf{n}) - \mathbf{F}(\mathbf{w}_h) \cdot \mathbf{n} \Big)\, d\,\Gamma, \tag{6.11}$$

in which $\hat{\mathbf{f}}_{E,e}$ is a numerical flux and $\mathbf{w}_h$ is a vector of the local states. The far field state, $\mathbf{w}_\infty$, represents here the flow in a fictitious cell adjacent to the boundary (*ghost cell*), defined in Section 6.6.1. The unit normal $\mathbf{n}$ is assumed to be outward-pointing.

Following Abgrall [1], the modified Steger & Warming numerical flux will be used:

$$\hat{\mathbf{f}}(\mathbf{w}_h, \mathbf{w}_\infty, \mathbf{n}) = \mathbf{C}_\mathbf{n}^+(\mathbf{w_h})\mathbf{w_h} + \mathbf{C}_\mathbf{n}^-(\mathbf{w_h})\mathbf{w}_\infty.$$

By analogy with the original reference, particular signals are calculated with the aid of the linear Lagrange basis functions, ψ_i :

$$\Phi_i^{e,bd} = \int_e \Big(\hat{\mathbf{f}}(\mathbf{w}_h, \mathbf{w}_\infty, \mathbf{n}) - \mathbf{F}(\mathbf{w}_h) \cdot \mathbf{n} \Big)\, \psi_i \, d\,\Gamma. \tag{6.12}$$

In Equation (6.12), ψ_i and $\hat{\mathbf{f}}$ are used to split $\Phi^{e,bd}$ into signals and distribute them among the vertices of e, i.e. define the distribution strategy. To the author’s best knowledge, with regard to the boundary conditions no other fluxes have been suggested in the literature.

This approach, originally introduced by Abgrall [1], was also applied in [8] and [4]. Ricchiuto in his recent monograph [83] suggested a similar technique, though discussed it only for scalar equations. In other references, i.e. [3, 6, 51, 86] and [87], the authors also refer to weak boundary conditions, but omit some or most of the details related to the implementation. The above methodology is obviously very similar to the way interior and boundary edges are treated in the discontinuous Galerkin discretizations. In this respect, the approach outlined here can be viewed as a hybrid RD-DG method. One might be tempted to experiment with other numerical fluxes and different distribution strategies. This, however, is a separate research strand and will not be considered here.

Applying weak boundary conditions in the case of steady state computations revealed that boundaries are prone to instabilities and therefore the CFL number was decreased for the corresponding nodes. Such an approach is feasible in the steady state case in which local time stepping can be used. It is no longer practical in the time-dependent setting as adjusting the time step at the boundary means that it has to be adjusted uniformly throughout the domain. For this reason weak boundary conditions were used only for the steady-state Euler equations.

6.7 Numerical Results

The goal of this section is twofold. First, to briefly report on the numerical performance of steady state residual distribution methods when applied to the steady Euler equations. By no means is this an attempt to conduct a thorough study - this was done by a number of authors in the past. See for instance [50, 77, 82, 99] and [56]. The second and the key aim of this section is to present a thorough and extensive numerical comparison of the explicit and implicit $\mathcal{RKRD}$ frameworks with respect to their performance when implemented to solve the time-dependent Euler equations. Only continuous-in-space schemes are considered. This is primarily because the results for non-linear equations presented in Chapter 5 suggest that the discontinuous-in-space residual distribution framework is not fully developed.

In the steady state case the boundary conditions were prescribed weakly. To avoid stability related issues, strong boundary conditions were used in the time-dependent setting. Both structured and unstructured meshes were used. Further details are given when particular examples are discussed.

6.7.1 Steady State Euler Equations

Four different test problems were studied in the steady state case: two modeling supersonic, one modeling transonic and one modeling subsonic fluid flow. The CFL number was set to 0.9 for interior nodes and to 0.25 for boundaries, i.e. local time-stepping was employed. For some experiments, setting the CFL number at the boundary to values higher than 0.25 (i.e. 0.9) led to instabilities. For consistency, all simulations were carried out with the same time-step restriction (i.e. CFL number). Only weak boundary conditions were used in this section. All simulations were run on unstructured meshes.

In what follows, contour plots of the local Mach number of solutions obtained with the aid of the schemes described in this chapter are given. Since the Mach number depends on all four physical quantities present in the equations (pressure, density and the velocity field), it tends to be very sensitive and hence such plots are a very good way of evaluating the results. In all four cases the N, LDA and BLEND schemes gave plausible and satisfactory results. Shocks were captured accurately and the obtained contour plots are similar to corresponding ones that can be found in the literature [65, 76, 110]. The N scheme gave the least oscillatory, but the most diffusive solutions. The solutions produced by the LDA scheme are less diffusive, but much more oscillatory. Finally, the BLEND scheme coped with the system better than the LDA scheme in terms of oscillations and better than N with respect to diffusion of the final solution. It is, however, more diffusive than the LDA scheme and marginally more oscillatory then the matrix N scheme.

Oblique Shock Reflection

The problem is of an oblique shock reflection [110] in the domain defined by $(x, y) \in [0, 4] \times [0, 1]$. The data for this case are chosen such that the solution consists of three states separated by shocks. The boundary conditions were set so that the incident shock angle was 29° and the free stream Mach number M_∞ was set to 2.9.

Results are shown on Figures 6.2-6.4. All schemes succeeded in capturing the shocks. The solution obtained with the LDA scheme exhibits small oscillations near the shocks and under-shoots in the central region of the domain (these are not profound, though). These under-shoots were also present in the case of the BLEND scheme, but to a considerably smaller extent. The N scheme solved this test problem without producing non-physical oscillations or over/under-shoots. Both the LDA and the BLEND scheme captured the shock sharply, whereas the N scheme smeared it across the surrounding cells (the solution is, as in the scalar case, diffused).

Initial conditions: $\rho = 1.4, u = 2.9, v = 0.0, p = 1.0.$

Boundary conditions:

–left boundary: supersonic inflow

$(\rho = 1.0, u = 2.9, v = 0.0, p = 0.714286)$;

–right boundary: supersonic outflow;

–upper boundary: supersonic inflow

$(\rho = 1.7, u = 2.61934, v = -0.50632, p = 1.52819)$;

–lower boundary: solid wall.

Grid: Topology as in Figure 6.1, 5422 nodes

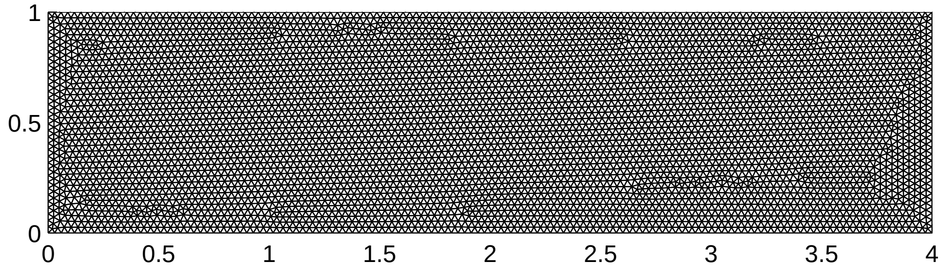


Figure 6.1: The grid used for the oblique shock reflection test case.

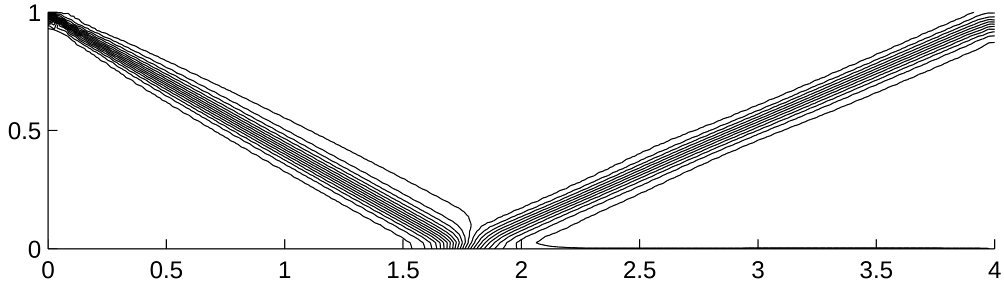


Figure 6.2: Local Mach number contours for the oblique shock reflection test case with the N scheme.

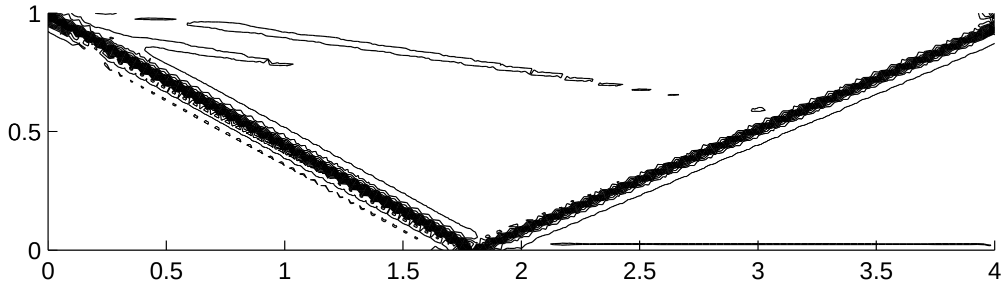


Figure 6.3: Local Mach number contours for the oblique shock reflection test case with the LDA scheme.

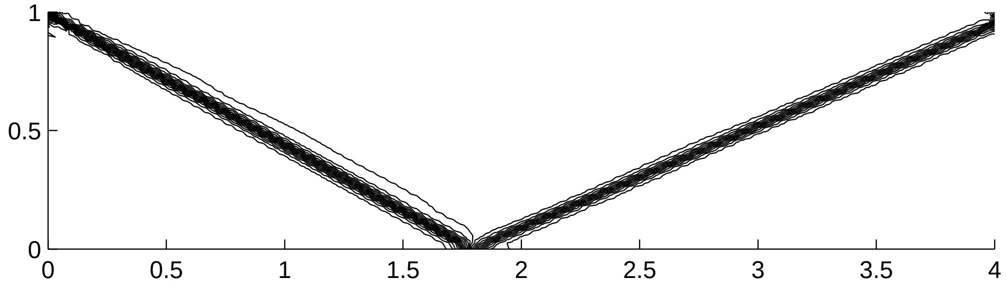


Figure 6.4: Local Mach number contours for the oblique shock reflection test case with the BLEND scheme.

Ni's Constricted Channel Flows - the subsonic case [76]

This test problem consists of flow over a ramp that is part of a circle in the domain defined by $(x, y) \in [0, 3] \times [0, 1]$. The circular arc is given a height of 0.1. The data for this case are chosen such that the free stream Mach number is 0.5. The resulting flow should be subsonic throughout the whole domain, shock-free and symmetric about the centre of the construction.

Results for this test cases are presented on Figures 6.6-6.8. For each scheme tested here, the solution is slightly smeared out towards the lower right-hand-side corner of the domain, but almost perfectly symmetric elsewhere. There are no major differences between the three solutions, but as usual the N scheme is more diffusive than the LDA and BLEND schemes.

Initial conditions: $\rho = 1.4, u = 0.5, v = 0.0, p = 1.0$.

Boundary conditions:

–left boundary: subsonic inflow $(\rho = 1.4, u = 0.5, v = 0.0)$;

–right boundary: subsonic outflow $(p = 1.0)$;

–upper boundary: solid wall;

–lower boundary: solid wall.

Grid: Topology as in Figure 6.5, 6660 nodes

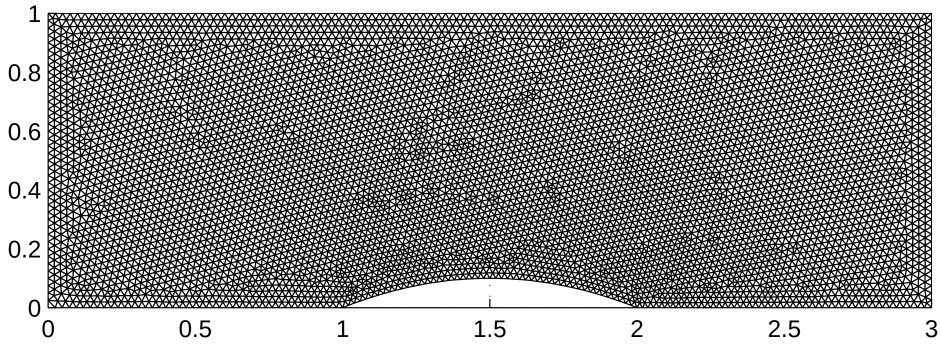


Figure 6.5: The grid for the 10% circular arc bump test case.

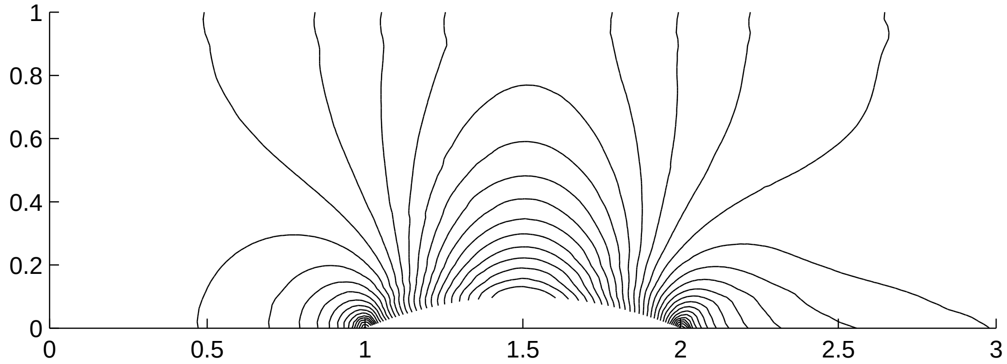


Figure 6.6: Local Mach number contours for the 10% circular arc bump test case with the N scheme.

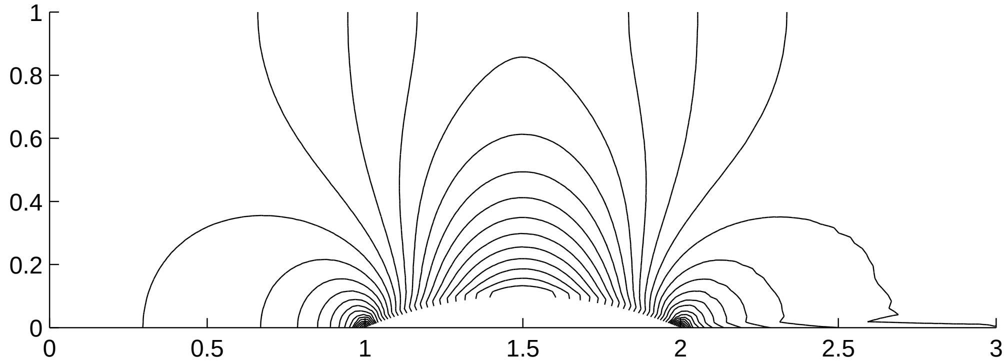


Figure 6.7: Local Mach number contours for the 10% circular arc bump test case with the LDA scheme.

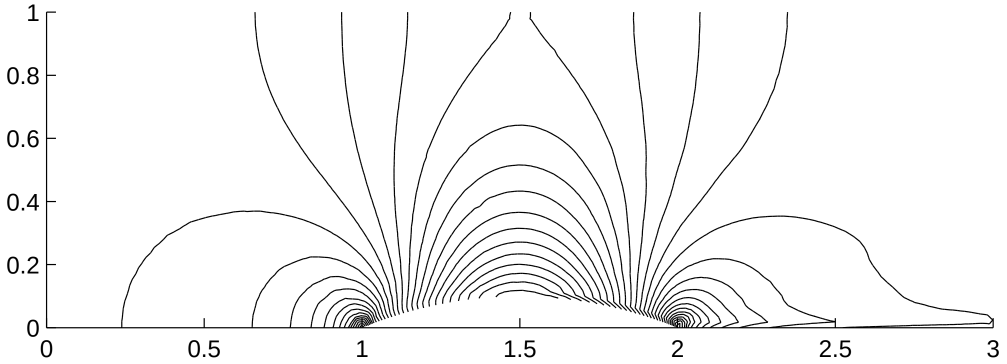


Figure 6.8: Local Mach number contours for the 10% circular arc bump test case with the BLEND scheme.

Ni's Constricted Channel Flows - the transonic case [76]

The geometrical setting in this case is identical as in the previous section. The initial and boundary conditions are set so that the Mach number at the inflow is equal to 0.675. The resulting flow contains a single shock on the lower surface of the domain.

Figures 6.9-6.11 show the local Mach number contours of the steady state solutions obtained for this problem using the N, LDA and the BLEND scheme, respectively. In all three cases the shock was captured sharply. The solution obtained with the LDA scheme exhibits small overshoots close to the shock, which vanished in the case of the N scheme and almost vanished for the BLEND scheme. The N scheme, as previously, exhibits large amounts of numerical diffusion. The resolution with which the LDA and BLEND schemes approximated the solution is noticeably higher.

Initial conditions: $\rho = 1.4, u = 0.675, v = 0.0, p = 1.0$.

Boundary conditions:

–left boundary: subsonic inflow ($\rho = 1.4, u = 0.675, v = 0.0$);

–right boundary: subsonic outflow ($p = 1.0$);

–upper boundary: solid wall;

–lower boundary: solid wall.

Grid: Topology as in Figure 6.5, 6660 nodes

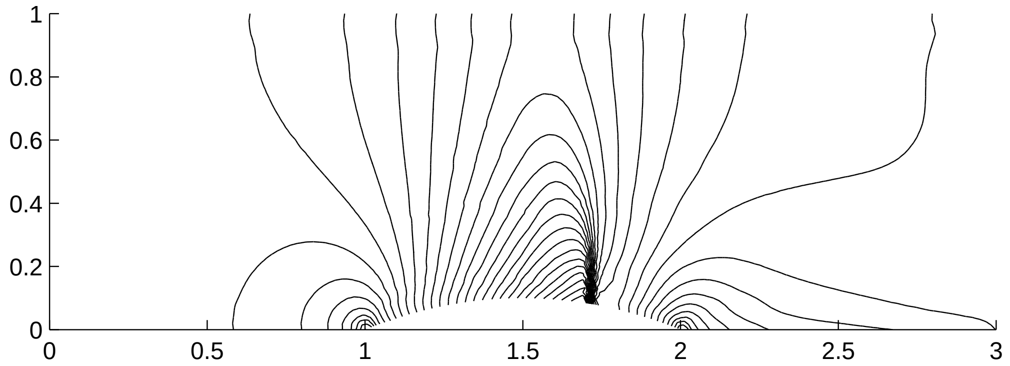


Figure 6.9: Local Mach number contours for the 10% circular arc bump test case with the N scheme.

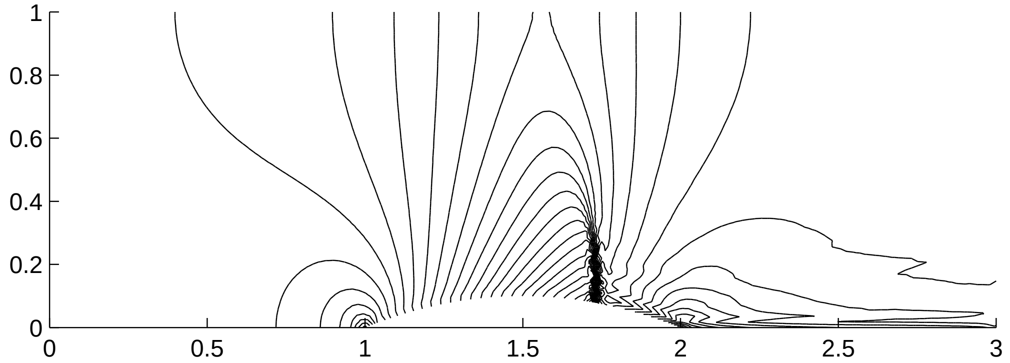


Figure 6.10: Local Mach number contours for the 10% circular arc bump test case with the LDA scheme.

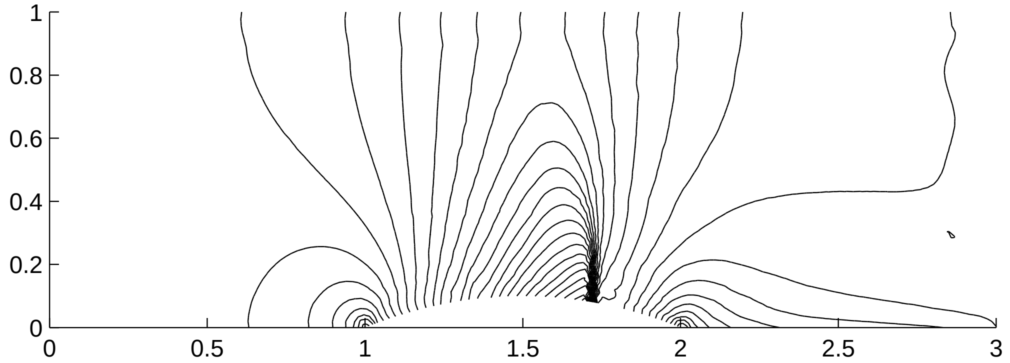


Figure 6.11: Local Mach number contours for the 10% circular arc bump test case with the BLEND scheme.

Ni's Constricted Channel Flows - the supersonic case [76]

The geometrical setting for this test case is similar as in the previous section except that the circular arc is given a height of 0.04 instead of 0.1. The data for this case are chosen such that the free stream Mach number is 1.4. The resulting flow should be almost completely supersonic with strong shocks at both front and the rear of the bump which are reflected off the walls of the channel further downstream.

Also in this case all schemes captured the shocks sharply, see Figures 6.13 -6.15. No under/over-shoots were observed in the case of the N scheme. The solution obtained with the BLEND scheme does exhibit some over- and under-shoots, though these are not profound. In the case of the LDA scheme there are clearly visible oscillations along the shocks. As expected, the LDA scheme clearly performs poorly when the solution to the underlying problem exhibits shocks.

Initial conditions: $\rho = 1.4, u = 1.4, v = 0.0, p = 1.0$.
Boundary conditions:

–left boundary: supersonic inflow ($\rho = 1.4, u = 1.4, v = 0.0, p = 1.0$);

–right boundary: supersonic outflow;

–upper boundary: solid wall;

–lower boundary: solid wall.

Grid: Topology as in Figure 6.12, 6456 nodes

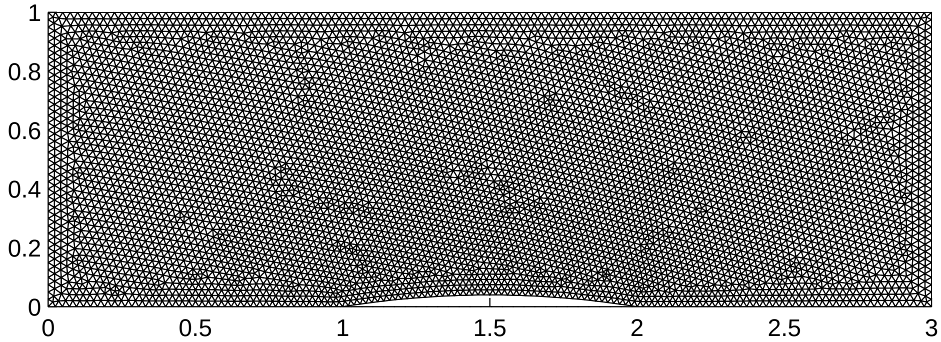


Figure 6.12: The grid for the 4% circular arc bump test case.

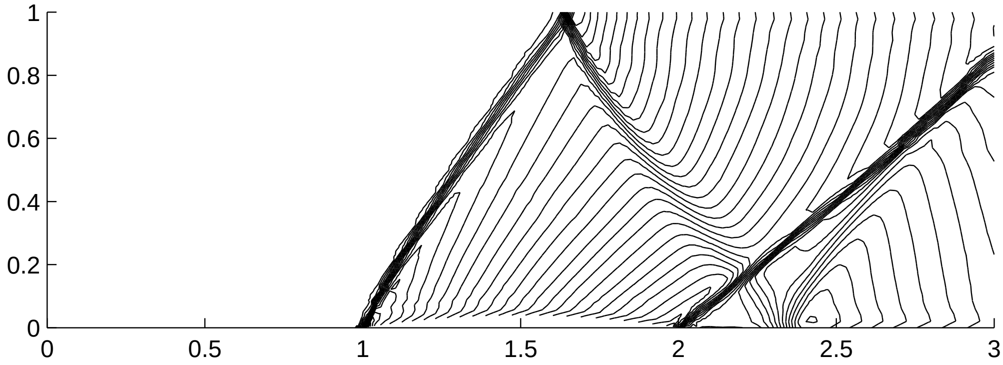


Figure 6.13: Local Mach number contours for the 4% circular arc bump test case with the N scheme.

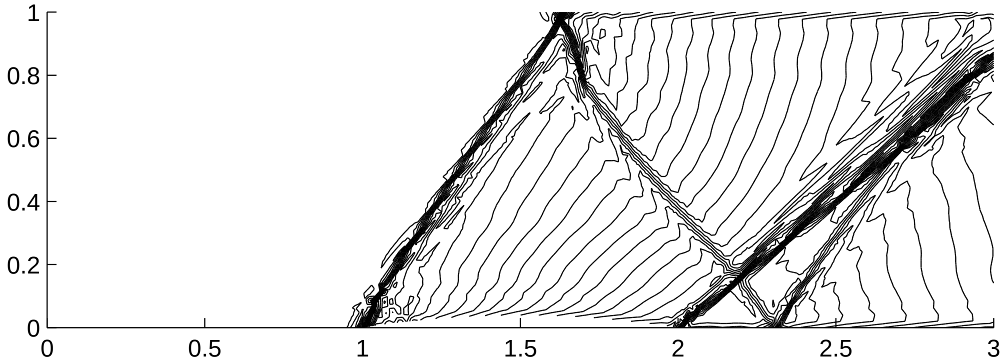


Figure 6.14: Local Mach number contours for the 4% circular arc bump test case with the LDA scheme.

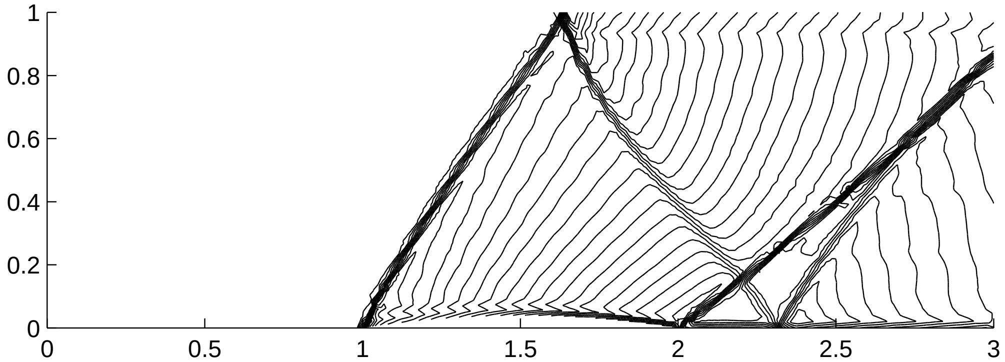


Figure 6.15: Local Mach number contours for the 4% circular arc bump test case with the BLEND scheme.

6.7.2 Evolutionary Euler Equations

In the time-dependent setting three distinct test problems were implemented:

- Double Mach Reflection (the solution exhibits strong shocks);
- Mach 3 Flow Over a Step (the solution exhibits strong shocks);
- Advection of a Vortex (the analytic solution is C^2 regular).

The focus in this section is laid on comparing the implicit $\mathcal{RKRD}$ and explicit $\mathcal{RKRD}$ frameworks, both introduced in Chapter 4. Both structured and unstructured meshes were used and the CFL number was varied between 0.1 and 0.9. In the case of the Advection of a Vortex test case, the CFL number was decreased in order to gain a clearer indication of the order of accuracy of the underlying numerical approximations. For the Mach 3 Flow Over a Step test case the CFL number was decreased only in the case of the implicit $\mathcal{RKRD}$ framework, for which higher CFL numbers caused instabilities which were too strong for the algorithm to finish the simulation. The detailed configuration for each test case is given in the corresponding paragraph. As noted in [81], shocks appearing in the Double Mach Reflection and Mach 3 test cases are too strong for the LDA scheme to cope with. Such being the case, only the implicit RKRD-BLEND and explicit RKRD-BLEND schemes were considered in these cases. For a comparison with a first order scheme, the Double Mach Reflection test case was additionally solved with the aid of the RKRD-N scheme (for which there is no distinction between the implicit and explicit frameworks, see Section 4.3.2). As discussed in Section 4.4, the PETSc [15, 16] numerical library was used to solve linear systems resulting from the implicit $\mathcal{RKRD}$ discretisations. The configuration of the linear solver remained unchanged from the one used in Chapter 4, except that the relative tolerance, in order to speed the calculations up, was set to 10^{-5} rather than 10^{-8}. In most cases, reducing it, i.e. setting it to values lower than 10^{-5}, did not show any noticeable improvements (qualitative nor quantitative).

Double Mach Reflection

This problem was originally introduced by Woodward et al. in [109]. It constitutes a very severe test for the robustness of schemes designed to compute discontinuous flows. The flow consists of the interaction of a planar right-moving Mach 10 shock with a 30° ramp. In the frame of reference used, the x axis is aligned with the ramp. The computational domain is the rectangle $[0, 4] \times [0, 1]$, with the ramp starting at

$x = \frac{1}{6}$ and stretching till the right-hand-side corner of the domain ($x = 4, y = 0$). The simulations were run until time $T = 0.2$ on three unstructured meshes with topology similar to that in Figure 6.16. The coarsest mesh had 7865 cells, then it was refined to give a mesh with 55927 cells and finally the experiment was run on a mesh with 278141 elements. At the initial state, the shock forms a 60° angle with the x axis. See Figure 6.17 for the geometry and initial values of the solution. The CFL number was set to 0.9.

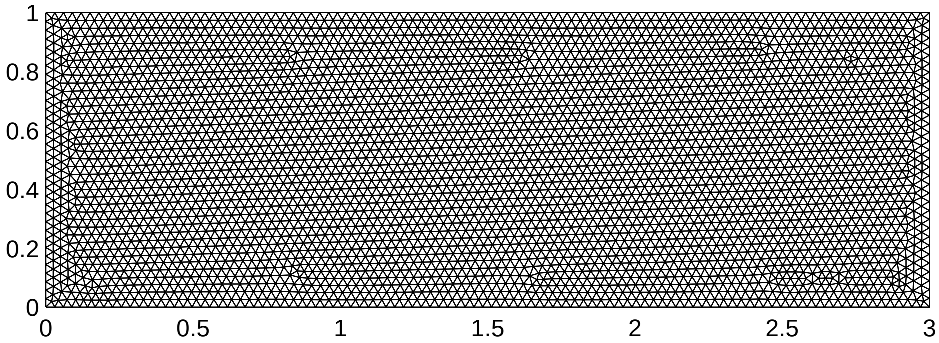


Figure 6.16: The coarsest grid for the Double Mach Reflection test case, 7865 cells.

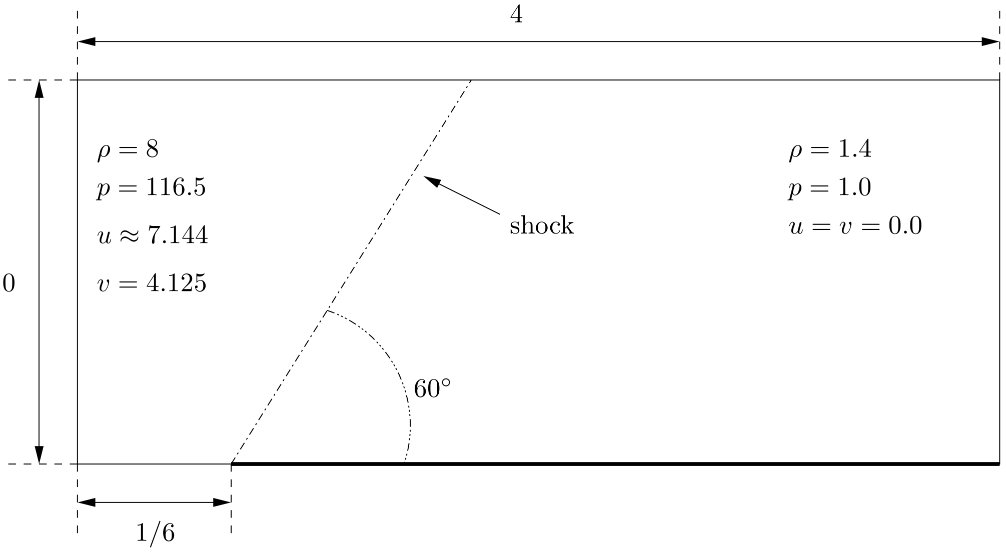


Figure 6.17: The geometry and initial condition for the Double Mach Reflection test case.

For this test case it is customary to plot contours of the density field. These are presented in Figures 6.18-6.26. Only the region between $x = 0$ and $x = 3$ is displayed, although the grid continues to $x = 4$. The air ahead of the shock remains undisturbed and the shorter domain makes the presentation clearer. All the considered schemes successfully captured the interaction between the shock and the ramp (see [27,85] and [109] for reference results). As expected, refining the mesh increased the resolution and the accuracy with which that interaction was resolved. In all cases, the BLEND scheme gave a solution exhibiting higher resolution and thus capturing the shocks more accurately than the N scheme. The coarsest mesh was insufficient to capture the contact emanating from the triple point and refining it led to a significant improvement. The explicit RKRD-BLEND and implicit RKRD-BLEND schemes gave comparable results, however the one obtained with the aid of the explicit RKRD-BLEND scheme is of noticeably higher resolution. This is probably due to the fact that in the case of the implicit RKRD-BLEND scheme values on the diagonal of the blending matrix $\mathbf{\Theta}$ (cf. Equation (6.9)) were set to the maximum value (i.e. the preference was given to the first order N scheme). Otherwise, instabilities would stop the algorithm from completing the simulation. The result in Figure 6.20 is comparable with those obtained in [27] and [109] on meshes with similar resolution.

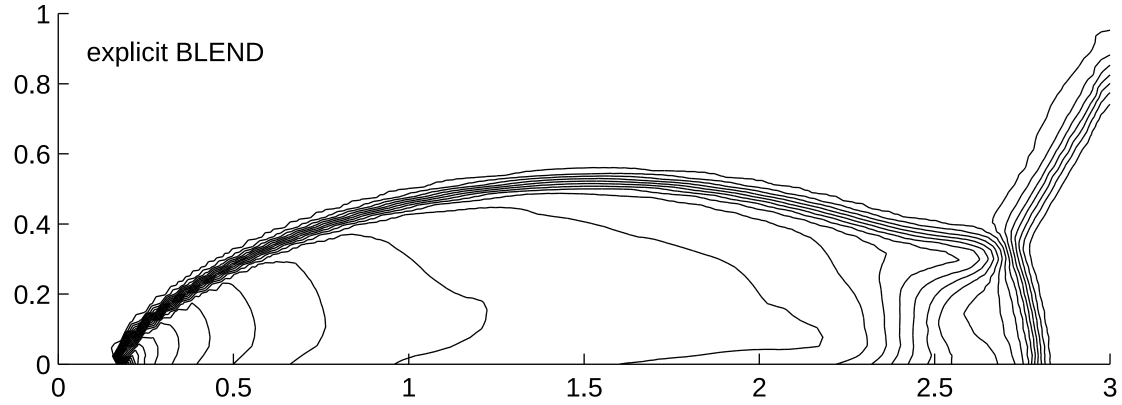


Figure 6.18: Double Mach reflection: density contours for the explicit RKRD-BLEND scheme. 7865 cells

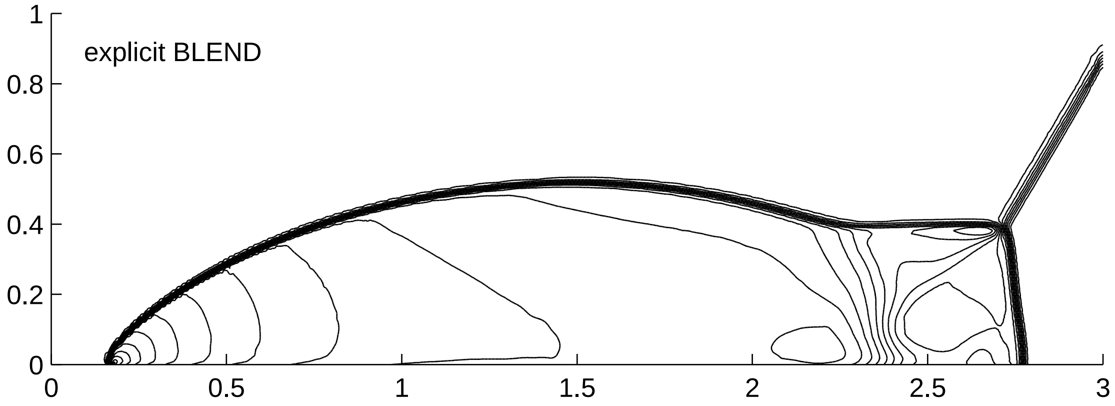


Figure 6.19: Double Mach reflection: density contours for the explicit RKRD-BLEND scheme. 55927 cells

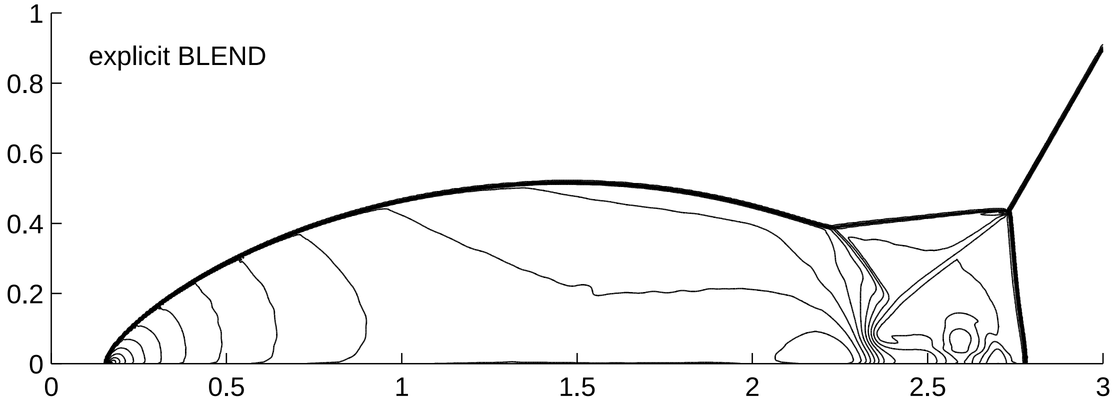


Figure 6.20: Double Mach reflection: density contours for the explicit RKRD-BLEND scheme. 278141 cells

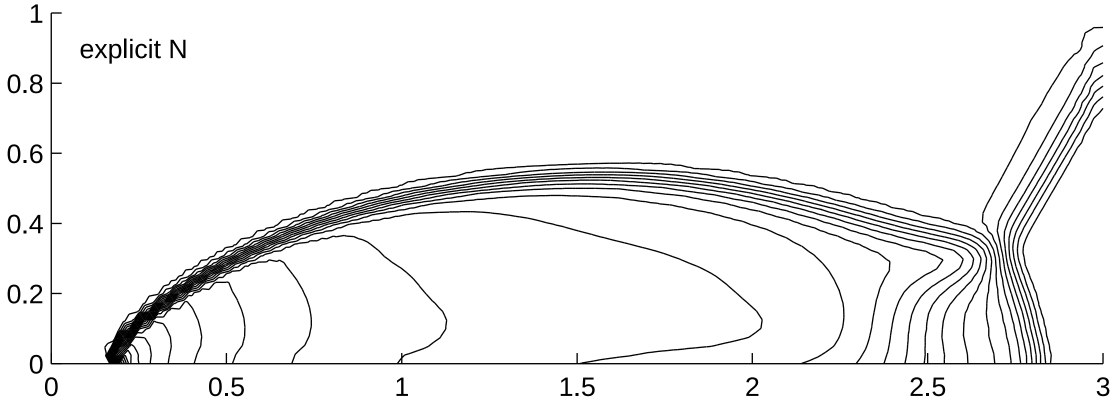


Figure 6.21: Double Mach reflection: density contours for the explicit RKRD-N scheme. 7865 cells

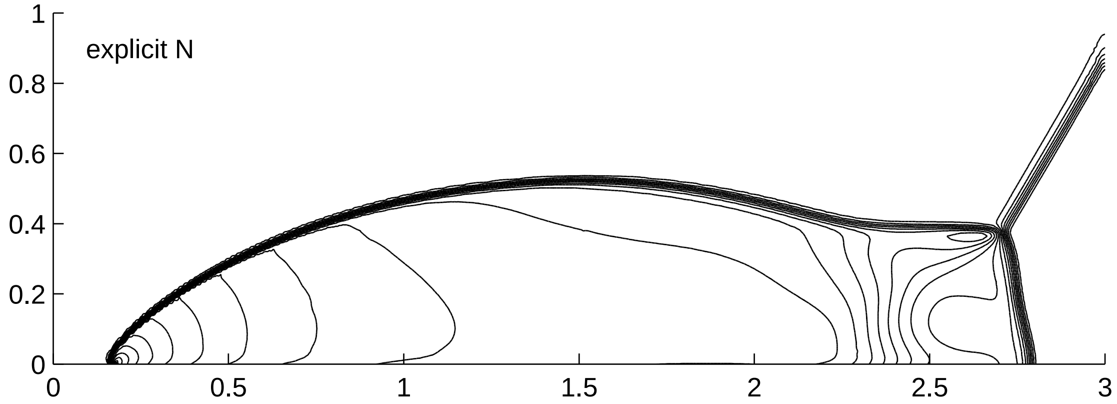


Figure 6.22: Double Mach reflection: density contours for the explicit RKRD-N scheme. 55927 cells

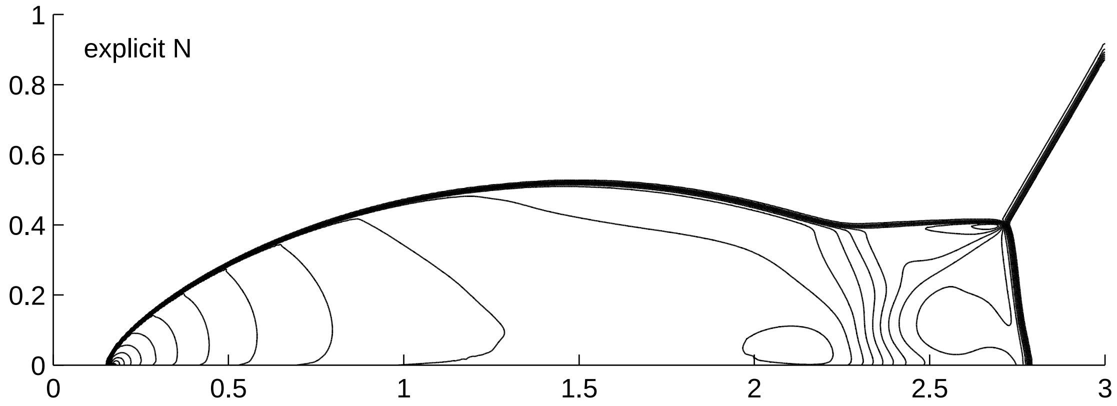


Figure 6.23: Double Mach reflection: density contours for the explicit RKRD-N scheme. 278141 cells

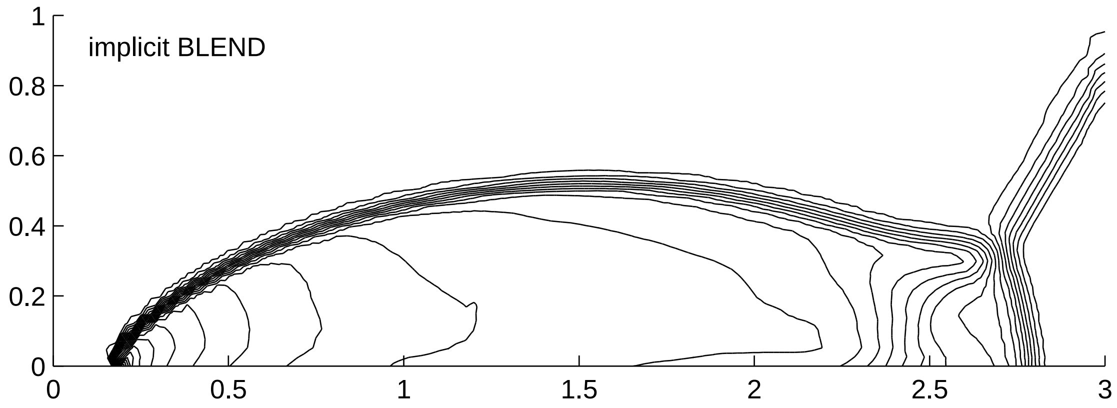


Figure 6.24: Double Mach reflection: density contours for the implicit RKRD-BLEND scheme. 7865 cells

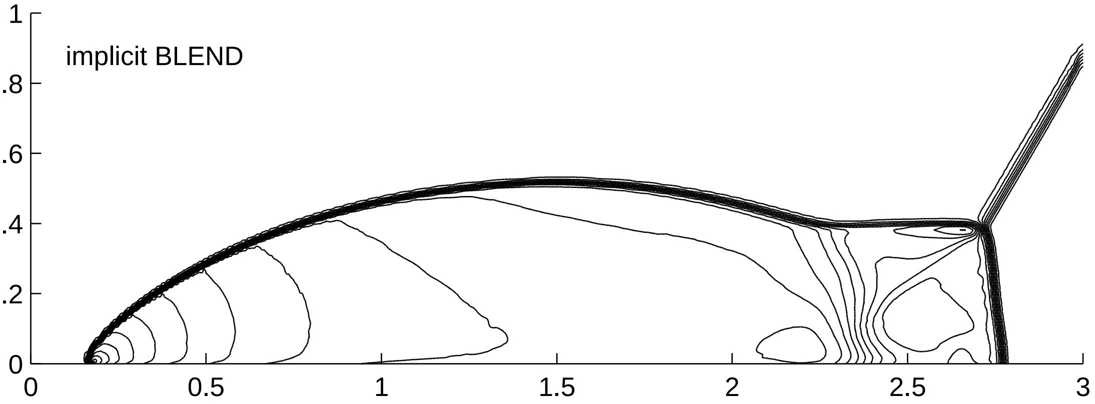


Figure 6.25: Double Mach reflection: density contours for the implicit RKRD-BLEND scheme. 55927 cells

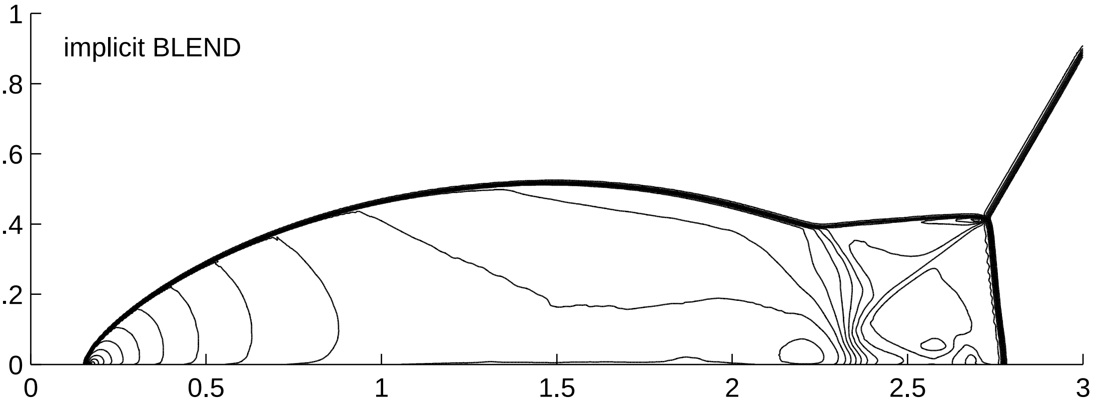


Figure 6.26: Double Mach reflection: density contours for the implicit RKRD-BLEND scheme. 278141 cells

Mach 3 Flow Over a Step

This test was originally introduced in the paper by Emery [45] and more recently reviewed by Woodward et al. in [109]. The problem begins with uniform Mach 3 flow in a wind tunnel containing a step. The wind tunnel is 1 length unit wide and 3 length units long. The step is 0.2 length units high and is located 0.6 length units from the left-hand end of the tunnel (see Figure 6.27 for the geometry and the initial condition). The inflow and outflow conditions are prescribed at the left and right boundaries ($y = 0.0$ and $y = 3.0$), respectively. The exit boundary condition has no effect on the flow, because the exit velocity is always supersonic. Initially the wind tunnel is filled with a gas, which everywhere has density 1.4, pressure 1.0, and velocity 3. Gas with this density, pressure, and velocity is continually fed in from the left-hand boundary. Along the walls of the tunnel reflecting boundary conditions are applied. The corner of the step is the centre of a rarefaction fan and hence is a singular point of the flow. Following Woodward and Colella [109], in order to minimize numerical errors generated at this singularity, additional boundary conditions near the corner of the step were prescribed. For every boundary cell E located behind the step and such that $0.6 \leq x \leq 0.6125 \; \forall x \in E$, all the variables were reset to their initial value. This condition is based on the assumption of a nearly steady flow in the region near the corner. The simulations were carried out on an unstructured mesh with 71080 nodes with the reference length set to approximately $\frac{1}{80}$ at the beginning and the end of the domain and $\frac{1}{1000}$ at the corner of the step. The zoom of the mesh near the singularity point is illustrated in Figure 6.28. The CFL number was to 0.8 for the explicit framework and 0.5 for the implicit.

Density contours at times $t = 0.5, t = 1.5$ and $t = 4.0$ obtained with the aid of explicit RKRD-BLEND and implicit RKRD-BLEND schemes are plotted on Figures 6.29-6.34. All the figures show a sharp resolution of the shocks and are comparable to results that one can find in the literature obtained on meshes with similar resolution (see, for instance, [60, 85] and [29]). The implicit RKRD-BLEND scheme captured the mach stem more accurately, see Figures 6.31 and 6.34. Otherwise both schemes behaved similarly with one exception. At time roughly equal $t \approx 2.35$ the implicit RKRD-BLEND (having set the CFL number to 0.8) gave a solution with negative density, which almost immediately led to instabilities and the simulation stopped. In order to obtain the solution at time $t = 4.0$, the CFL number was decreased to 0.5. In both the implicit and explicit case the values on the diagonal of the blending matrix $\boldsymbol{\Theta}$ (cf. Equation (6.9)) were set to maximum (i.e. the preference was given to the first order N scheme). Otherwise, instabilities close to the corner of the step would prevent the algorithm from completing the simulation.

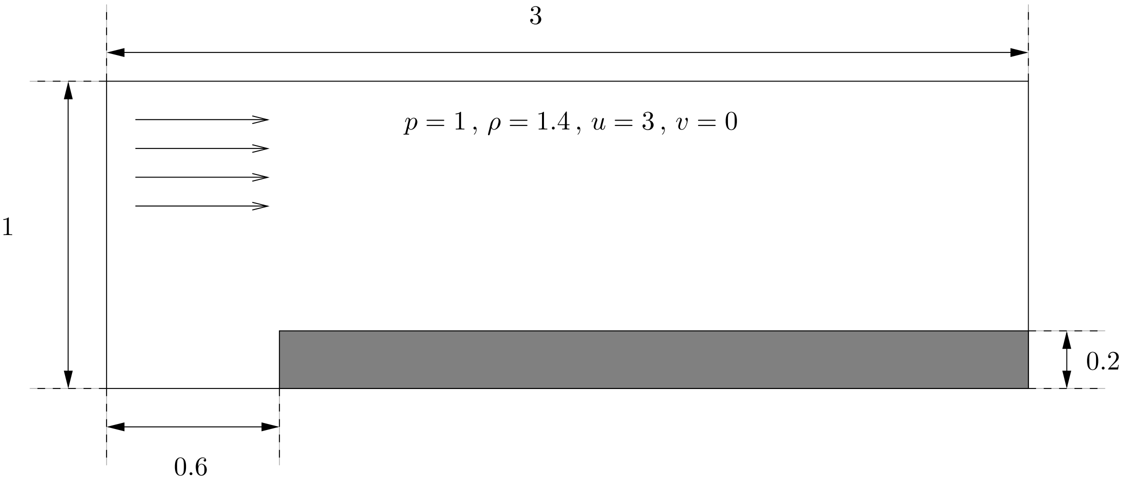


Figure 6.27: Geometry and the initial condition for the Mach 3 test case.

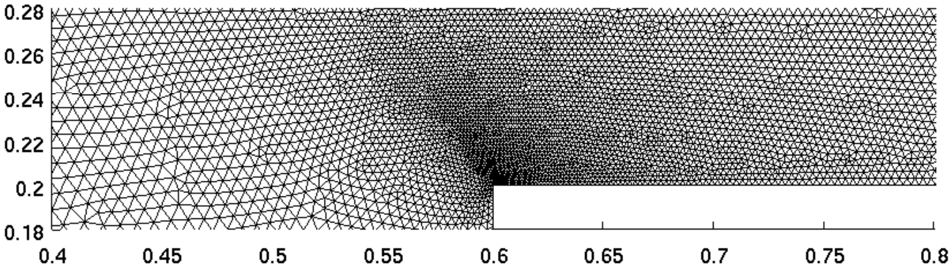


Figure 6.28: The zoom of the grid used for the Mach 3 Flow Over a Step test case near the singularity point.

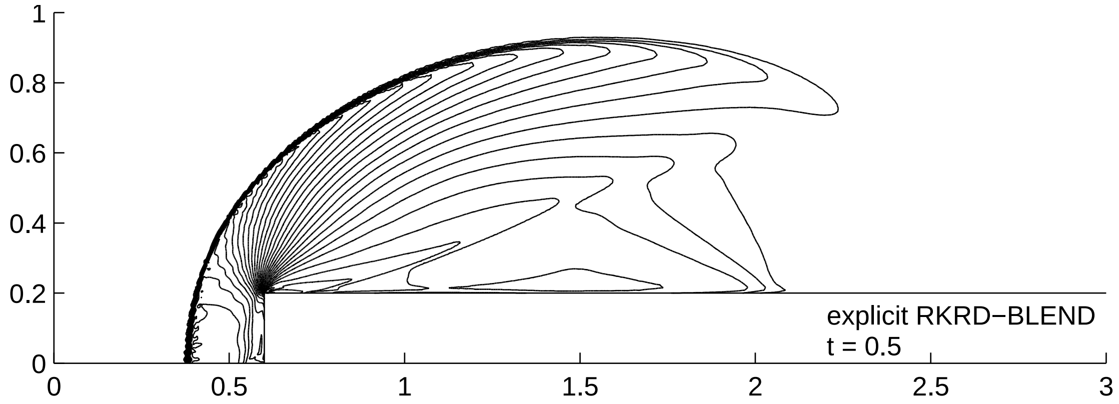


Figure 6.29: Mach 3 Flow Over a Step: Explicit RKRD-BLEND scheme, density contours at time $t = 0.5$, $CFL = 0.8$

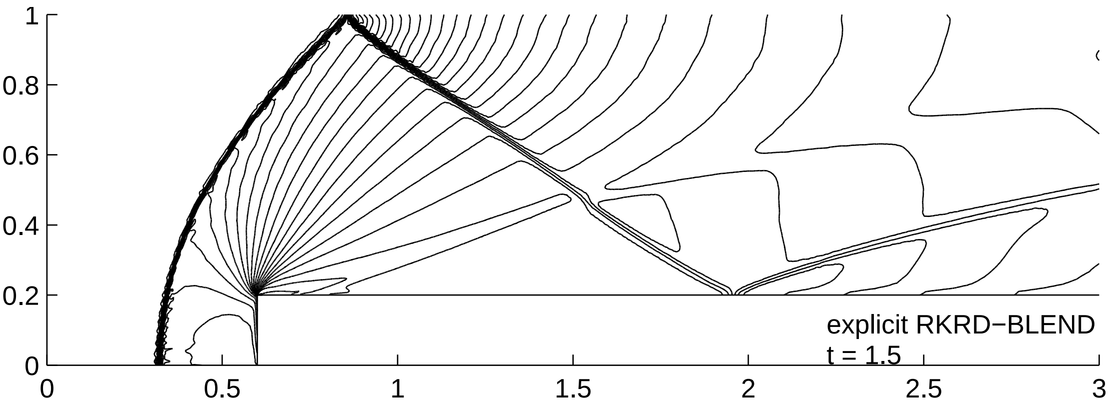


Figure 6.30: Mach 3 Flow Over a Step: Explicit RKRD-BLEND scheme, density contours at time $t = 1.5$, $CFL = 0.8$

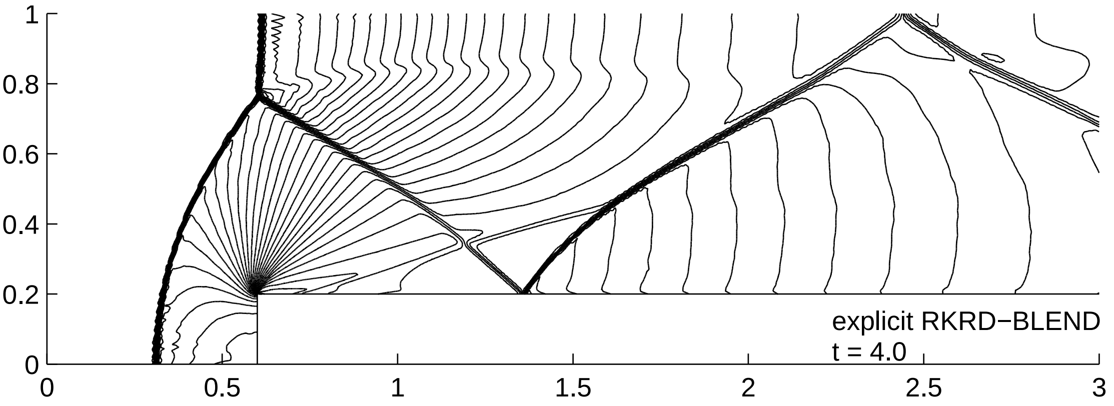


Figure 6.31: Mach 3 Flow Over a Step: Explicit RKRD-BLEND scheme, density contours at time $t = 4.0$, $CFL = 0.8$

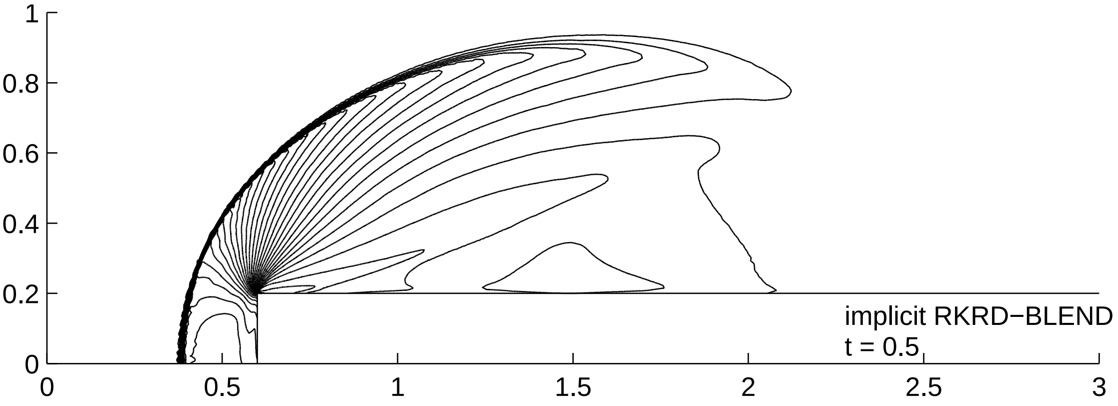


Figure 6.32: Mach 3 Flow Over a Step: Implicit RKRD-BLEND scheme, density contours at time $t = 0.5$, $CFL = 0.5$

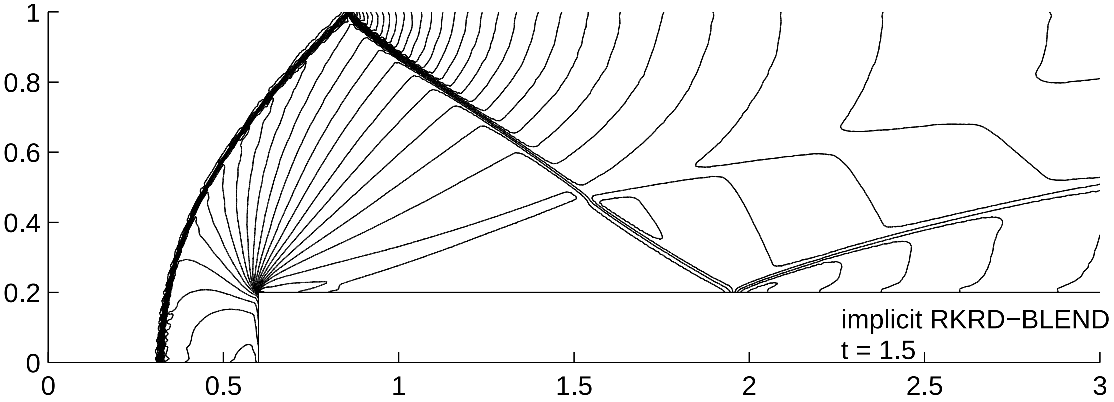


Figure 6.33: Mach 3 Flow Over a Step: Implicit RKRD-BLEND scheme, density contours at time $t = 1.5$, $CFL = 0.5$

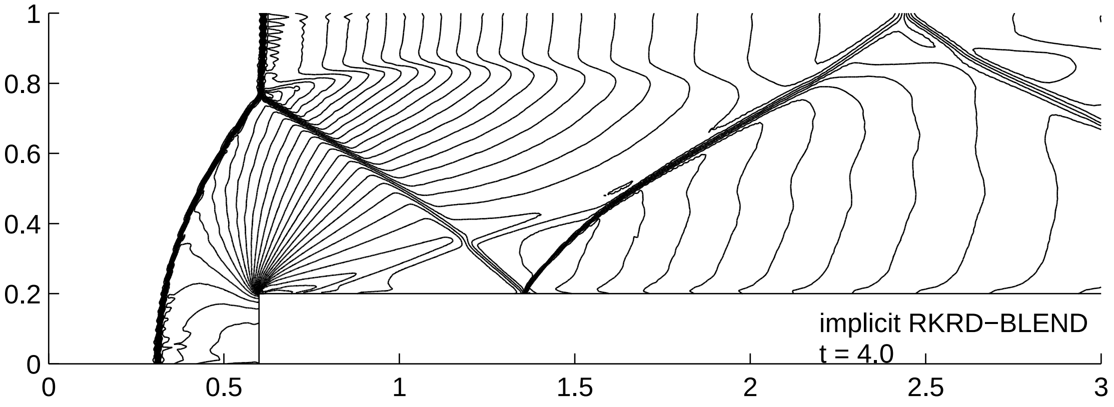


Figure 6.34: Mach 3 Flow Over a Step: Implicit RKRD-BLEND scheme, density contours at time $t = 4.0$, $CFL = 0.5$

Advection of a Vortex

The following problem was originally introduced in [44]. Its main appeal is the fact that the exact solution to this test case is known. The problem was solved on a rectangular domain $[0, 2] \times [0, 1]$ with inflow wall on its left side ($x = 0.0$), outflow at the right end of the domain ($x = 2$) and solid wall boundary conditions at the bottom and the top. The density for this test was constant and set to $\rho = 1.4$ throughout the domain. The centre of the vortex, (x_c, y_c), was initially set to $(0.5, 0.5)$ and was then advected during the simulation with the mean stream velocity $\mathbf{v}_m = (6, 0)$. The flow velocity was given by the mean $\mathbf{v}_m$ and the circumferential perturbation, i.e. $\mathbf{v} = \mathbf{v}_m + \mathbf{v}_p$, with:

$$\mathbf{v}_p = \begin{cases} 15\,(\cos(4\pi r) + 1)\,(-y, x) & \text{for} \qquad r < 0.25, \\ (0, 0) & \text{otherwise,} \end{cases}$$

with $r = \sqrt{(x - x_c)^2 + (y - y_c)^2}$. The pressure, similarly to the velocity vector, was given by its mean value $p_m = 100$ plus perturbation, i.e. $p = p_m + p_p$:

$$p_p = \begin{cases} \Delta p + C & \text{for} \qquad r < 0.25, \\ 0 & \text{otherwise,} \end{cases}$$

with $\Delta p + C$ defined so that the solution is C^2 regular:

$$\Delta p = \frac{15^2 \rho}{(4\pi)^2} \left(2\cos(4\pi r) + 8\pi r \sin(4\pi r) + \frac{\cos(8\pi r)}{8} + \pi r \sin(8\pi r) + 12\pi^2 r^2 \right).$$

The regularity is guaranteed by choosing C such that $p|_{r=0.25} = p_m = 100$. With the above setup the maximal Mach number in the domain is $M = 0.8$. The simulation was run until time $T = \frac{1}{6}$.

The first set of experiments was carried out on a structured mesh with topology as in Figure 6.41 with 161 × 81 nodes. The computations were performed with $CFL = 0.8$. In Table 6.1 values of the maximum and the minimum values of the pressure obtained are given. Isolines of the pressure inside and in the close vicinity of the vortex are shown in Figures 6.36-6.40. The N scheme gave the most smeared out and the least accurate result. The minimum value of the solution in this case is much higher than the exact one. The solutions produced with the explicit RKRD-BLEND scheme was much better in this respect, however noticeably worse than those obtained with the (implicit and explicit) LDA and implicit RKRD-BLEND schemes. The difference is not significant, but the solution obtained with the implicit RKRD-BLEND scheme resembles the exact solution, Figure 6.35, the most. No clear superiority of either the explicit or implicit frameworks was noticed, though the implicit RKRD-BLEND scheme gave somewhat smoother solution than its explicit counterpart. It should be noted, though, that in this section the implicit RKRD-BLEND scheme was set in such a way that the preference was given to the LDA scheme (cf. Section 6.4). In [44] similar experiments for this test problem were carried out (i.e. investigation of contour plots and the maximum/minimum values of the numerical solutions). Values presented in Table 6.1 show similar behaviour, but contour plots presented here (in particular those obtained with the aid explicit and implicit RKRD-LDA schemes and implicit RKRD-BLEND) are much more faithful to the exact solution than those presented in the literature.

Scheme	N	ex BLEND	im BLEND	ex LDA	im LDA	exact
p_{min}	98.77133	94.11941	93.5180	93.06300	92.90018	93.213
p_{max}	100.1191	100.1159	100.0004	100.0766	100.0803	100

Table 6.1: The minimum and maximum value of the pressure obtained with the aid of the LDA, N and BLEND schemes using the explicit (ex) and implicit (im) $\mathcal{RKRD}$ frameworks.

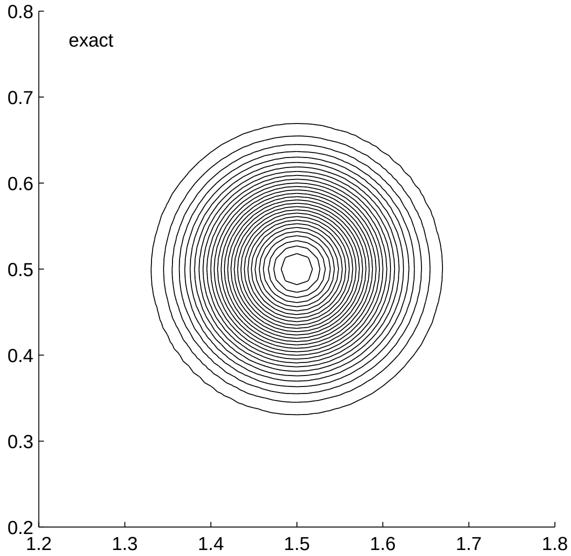


Figure 6.35: Travelling Vortex: pressure contours for the exact solution, 25600 cells

The grid convergence analysis was performed to investigate the order of accuracy of the implicit/explicit RKRD-LDA and -BLEND schemes. Errors were measured by means of the usual L^∞ norm and the L^2 and L^1 norms of the relative pressure error:

$$\epsilon_p = \frac{p^{exact} - p^{approx}}{p_m},$$

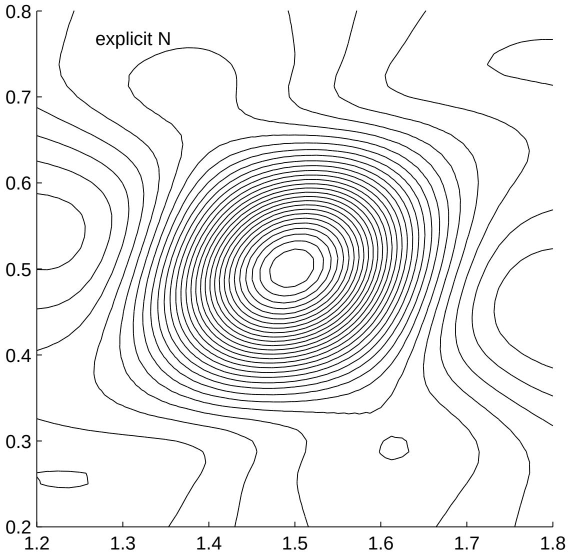


Figure 6.36: Travelling vortex: pressure contours for the explicit RKRD-N scheme, 25600 cells

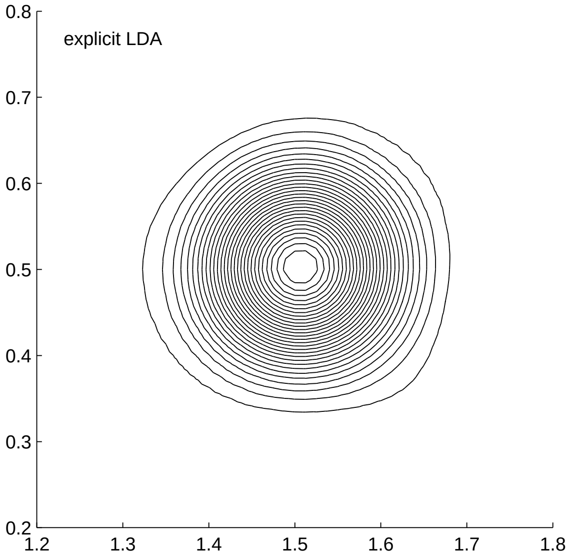


Figure 6.37: Travelling vortex: pressure contours for the explicit RKRD-LDA scheme, 25600 cells

in which p^{exact} and p^{approx} are the values of the analytical and numerical (approximate) pressure, respectively. Instead of calculating the error in the whole domain, only nodes inside and in the close vicinity of the vortex, i.e. nodes for which:

$$r \;=\; \sqrt{(x - x_c)^2 \,+\, (y - y_c)^2} \;\leq\; 0.35$$

were considered. Such approach guaranteed that there was no interference between boundary and interior nodes and that the imposition of boundary conditions did not affect the results. The experiments were performed on a set of structured and

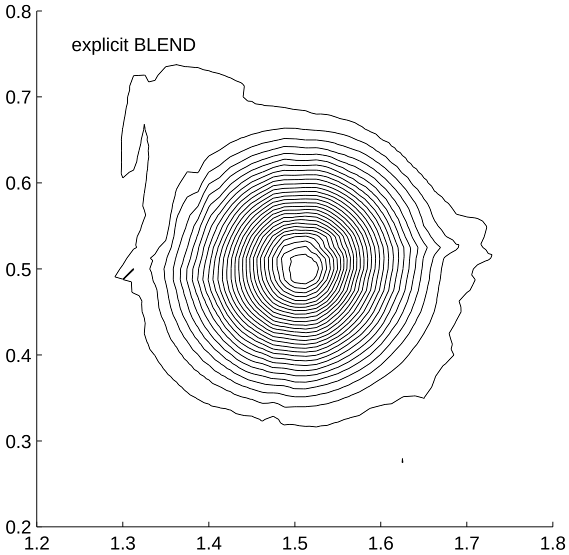


Figure 6.38: Travelling vortex: pressure contours for the explicit RKRD-BLEND scheme, 25600 cells

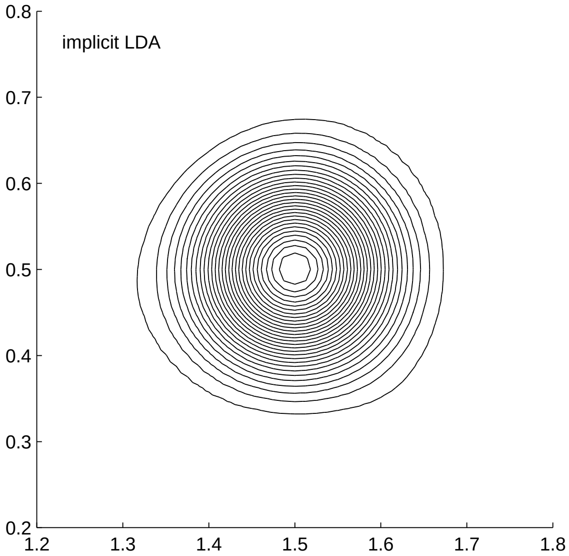


Figure 6.39: Travelling vortex: pressure contours for the implicit RKRD-LDA scheme, 25600 cells

unstructured meshes (topology as in Figure 6.41), for which the reference length was varied from approximately $\frac{1}{10}$ to $\frac{1}{160}$ in the case of unstructured meshes and from $\frac{1}{20}$ to $\frac{1}{320}$ in the case of structured grids. The CFL number had to be reduced to 0.4 and 0.1 for the explicit and implicit $\mathcal{RKRD}$ frameworks, respectively. Recall that it was set to 0.8 to produce the contour plots, i.e. Figures 6.36-6.40. Such a modification was necessary in order to demonstrate the accuracy for the coarsest meshes and to obtain results exhibiting second order convergence. The simulations were run until time $T = 0.08$ rather than $T = \frac{1}{6}$ (i.e. making the vortex travel

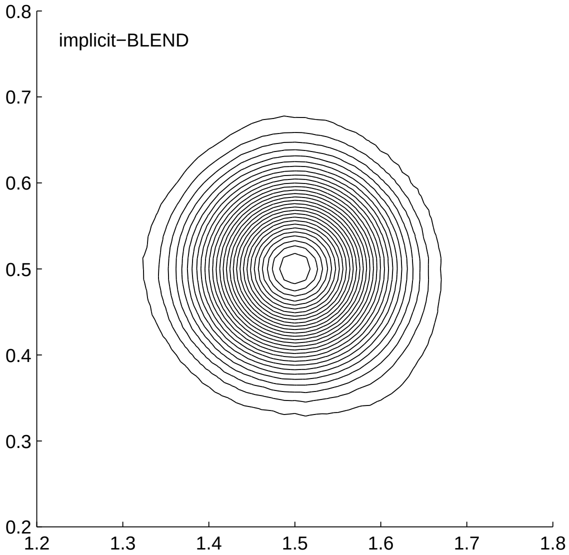


Figure 6.40: Travelling vortex: pressure contours for the implicit RKRD-BLEND scheme, 25600 cells

from $(0.5, 0.5)$ to $(0.98, 0.5)$ instead of $(1.5, 0.5)$). The results for the explicit RKRD framework on unstructured and structured meshes are presented in Figures 6.42 and 6.43, respectively. Apart from the L^∞ errors, the second order of accuracy is reached almost immediately. In [85] the authors presented errors only in the L^2 norm (comparable to those obtained here) claiming that the behaviour of their schemes (i.e. the explicit $\mathcal{RKRD}$ framework) is quantitatively and qualitatively very similar in all three norms considered: L^2, L^1 and L^∞. However, the configuration they used was somewhat different, i.e. periodic boundary conditions and a shorter domain were used. The results for the implicit framework on structured and unstructured meshes are illustrated in Figures 6.44 and 6.45, respectively. Also in this case the second order of accuracy is reached quite rapidly, but only in the L^2 and L^1 norms. Both, the implicit and explicit RKRD-LDA scheme exhibited a small drop down in the order of accuracy when moving to the finest meshes. For almost all the experiments, the implicit framework was more accurate then its explicit counterpart.

To investigate the overhead related to solving linear systems (and using PETSc) in the case of the implicit framework, selected execution times for the implicit and explicit frameworks are presented in Table 6.2. The *clock()* function from the C Programming Language Standard Library was used. The overhead that the implicit framework introduced is strictly related to solving two linear systems at every time step: one at each stage of the Runge-Kutta time stepping. This includes allocating the memory for the linear system, M (cf. Formulation (4.9)), assembling it and

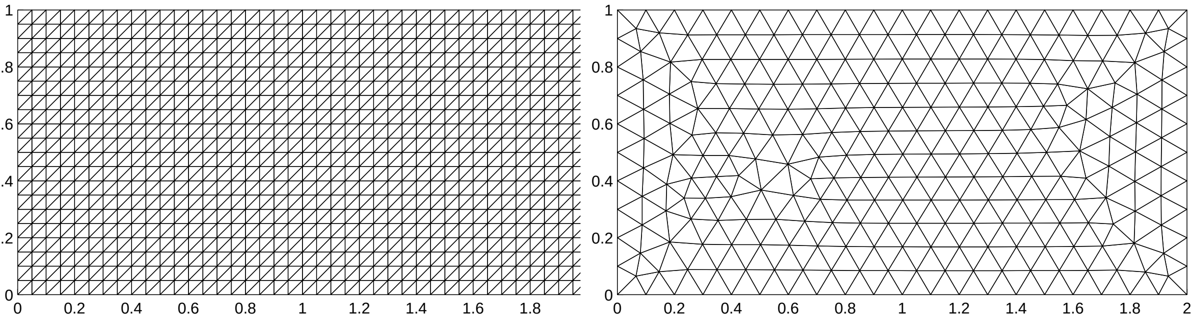


Figure 6.41: The finest structured (left) and unstructured (right) grid used in the grid convergence analysis for the Advection of A Vortex test case.

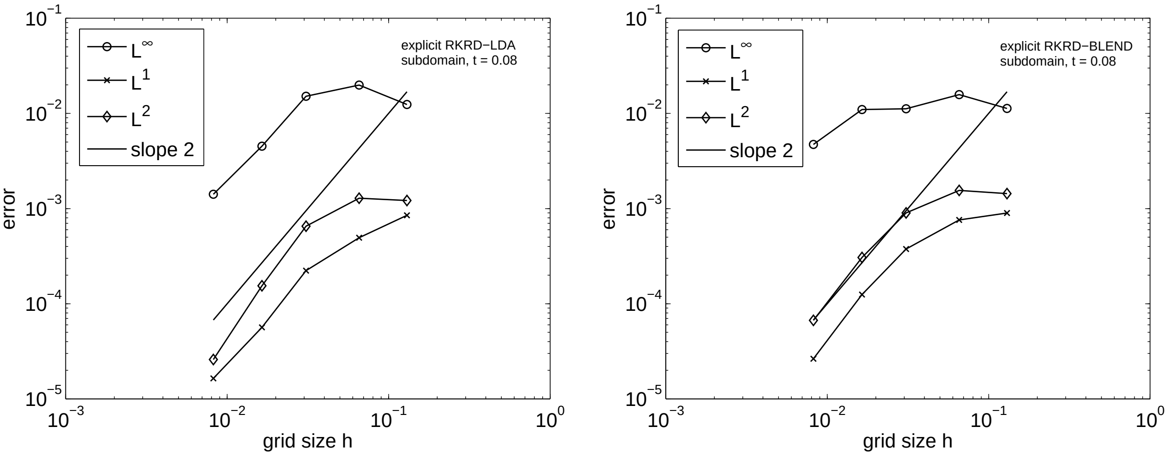


Figure 6.42: Grid convergence for the explicit RKRD-LDA (left, $CFL = 0.4$) and -BLEND (right, $CFL = 0.4$) schemes for the travelling vortex test case. Errors calculated within a sub-domain surrounding the vortex. Simulation run until $T = 0.08$. Unstructured meshes.

finally inverting to get the solution at the next time step. All of these tasks were performed in one update procedure, all other parts of the code being shared between the implicit and explicit frameworks. In the case of the explicit $\mathcal{RKRD}$ framework, the linear system M (cf. Formulation (4.9)) is diagonal and one can immediately proceed to calculating the solution at the new time step, i.e. $\mathbf{w}^{n+1}$, based on the solution at the current time, i.e. $\mathbf{w}^n$. Table 4.9 contains a set of two times (evaluated on five consecutively refined meshes) for two of the tested schemes. The implicit and explicit RKRD-LDA schemes were chosen as representatives for the implicit and explicit $\mathcal{RKRD}$ frameworks, respectively. The first value (Time 1) represent the amount of time (in seconds) it took for one time step (two Runge-Kutta stages), i.e. the whole process of calculating $\mathbf{w}^{n+1}$ based on $\mathbf{w}^n$. The second value (Time 2) represents the time it took for one update procedure (within one Runge-Kutta

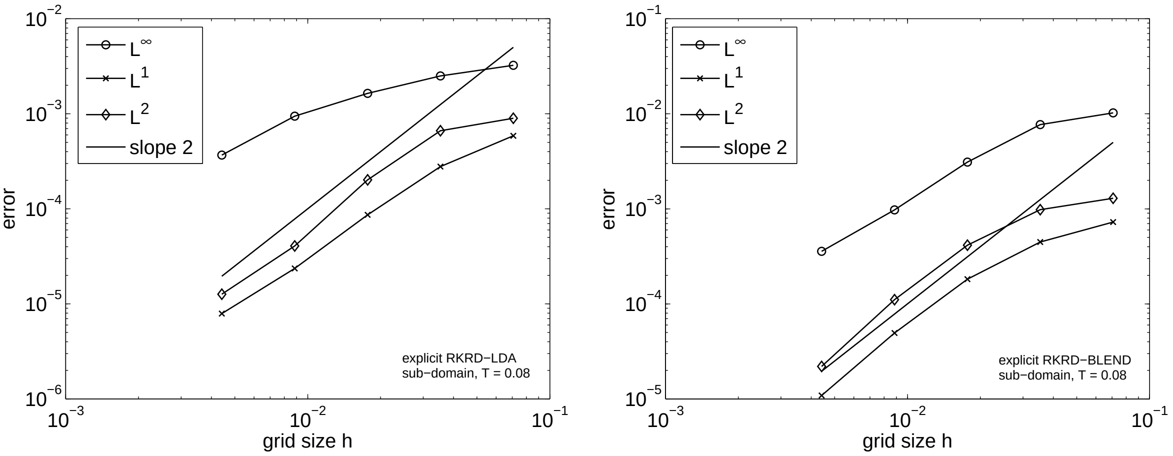


Figure 6.43: Grid convergence for the explicit RKRD-LDA (left, $CFL = 0.4$) and -BLEND (right, $CFL = 0.4$) schemes for the travelling vortex test case. Errors calculated within a sub-domain surrounding the vortex. Simulation run until $T = 0.08$. Structured meshes.

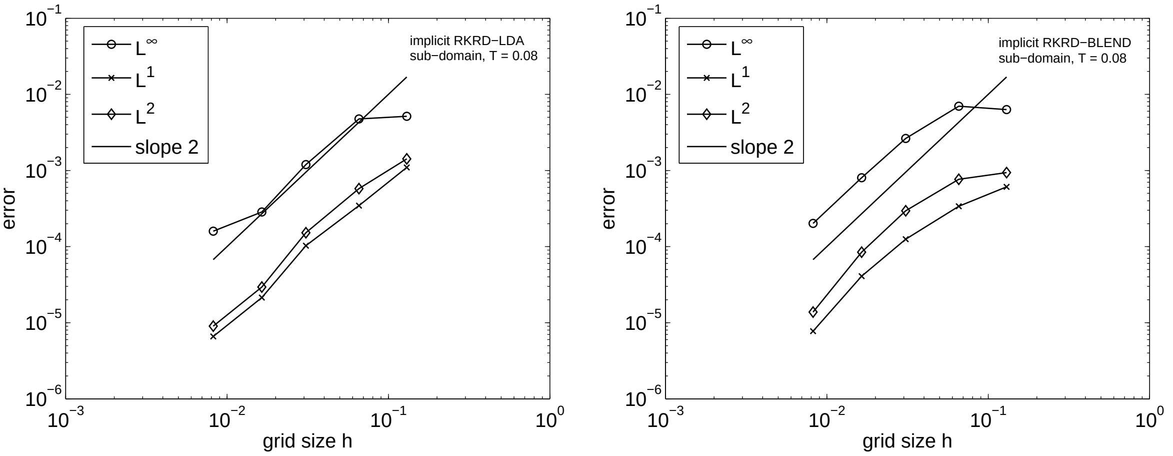


Figure 6.44: Grid convergence for the implicit RKRD-LDA (left, $CFL = 0.1$) and -BLEND (right, $CFL = 0.1$) schemes for the travelling vortex test case. Errors calculated within a sub-domain surrounding the vortex. Simulation run until $T = 0.08$. Unstructured meshes.

stage), that is creating and inverting the mass matrix, M, and then using it and $\mathbf{w}^n$ to calculate $\mathbf{w}^{n+1}$. In the case of the explicit $\mathcal{RKRD}$ framework the mass matrix is diagonal and hence there is no need for an expensive procedure assembling and inverting it. The implicit RKRD-LDA takes on average four times longer to obtain the desired solution. For both schemes the execution time increases by a factor of four when the mesh is refined. One should bear in mind that the above approach gives an estimation rather than the actual execution time of the investigated procedure, and that the result has a rather low resolution (microseconds).

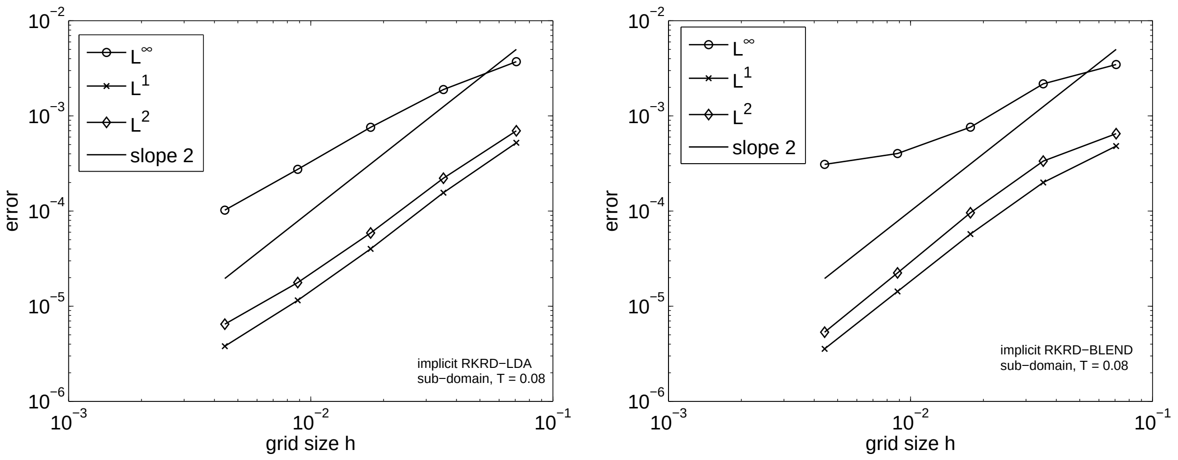


Figure 6.45: Grid convergence for the implicit RKRD-LDA (left, $CFL = 0.1$) and -BLEND (right, $CFL = 0.1$) schemes for the travelling vortex test case. Errors calculated within a sub-domain surrounding the vortex. Simulation run until $T = 0.08$. Structured meshes.

This is mainly due to the implementation of the clock() function. At the same time, the above is not an attempt to perform a thorough profiling or comparison of the implicit and explicit frameworks. Results from Table 6.2 are presented here to draw a general picture and to serve as guidance when considering these schemes in future. Experiments were performed on a Desktop PC equipped with an HT Intel Xeon core and twelve gigabytes of operating memory. All presented execution times are averages calculated during the corresponding simulation (the Advection of a Vortex test case, simulation run until time $T = 0.08$). The above study shows that the implicit RKRD-LDA, even though in most cases it is more accurate than its explicit counterpart, is relatively slow compared to the explicit RKRD-LDA scheme. In the tested scenarios the gain in accuracy does not outweigh the lost in efficiency. However, a more thorough and extensive study ought to be carried out before such comparison can be considered complete.

6.8 Summary

In this chapter an appropriate conservative discrete form for the two-dimensional Euler equations was presented. The outlined technique lays at the basis of every modern residual distribution method when applied to the Euler equations. Originally, residual distribution methods were applied to the Euler equations with the aid of wave decompositions. However, only the more popular and robust approach of matrix distribution methods was considered.

#Cells		474	1856	7374	29656	118522
im LDA	GMRES iter.	9.90/9.90	9.85/9.85	9.98/9.98	7.16/7.16	10.41/10.41
	L^2 Error	1.4219e-03	5.7804e-04	1.5207e-04	2.9426e-05	9.0165e-06
	Time 1	0.120e-2	4.709e-2	1.946e-1	7.359e-1	3.305
	Time 2	4.662e-3	1.818e-2	7.592e-2	2.822e-1	1.309
ex LDA	L^2 Error	1.2164e-03	1.2843e-03	6.5866e-04	1.5420e-04	2.5926e-05
	Time 1	2.628e-03	1.087e-02	4.3921e-02	1.760e-01	7.072e-01
	Time 2	0.0	1.7e-5	1.720e-04	5.08e-04	1.885e-03

Table 6.2: Performance of the implicit (im) and explicit (ex) RKRD-LDA methods when applied to the Advection of a Vortex test case. The table shows (1) the average number of iterations it took to reach the stopping criterion during the first/second stage of the Runge-Kutta time-stepping (the implicit scheme only), (2) L^2 errors and (3) the amount of time (in seconds) for: one time step (both stages, Time 1) and the update procedure (setting and solving the linear system, Time 2). Results are given for the unstructured meshes used earlier in the grid convergence analysis (with 474, 1856, 7374, 29656 and 118522 cells, cf. top row of the table and Figures 6.42 and 6.44).

Extensive numerical results comparing the explicit and implicit $\mathcal{RKRD}$ frameworks were presented. The former proved to be noticeably more stable in the sense that in a number of scenarios it allowed larger CFL number (the Advection of a Vortex and Mach 3 Flow Over a Step test problems) and more relaxed definition of the blending matrix $\mathbf{\Theta}$ (the Advection of a Vortex and the Double Mach Reflection test case). On the other hand, the implicit framework offers a more accurate discretization. This, obviously, comes at a price - the implicit framework is on average four times more expensive in terms of computational cost, or, to be more precise, execution time.

To sum up, it was presented that the implicit $\mathcal{RKRD}$ framework is a robust way of discretising systems of non-linear hyperbolic PDEs. In most cases it leads to more accurate approximations than its explicit counterpart. However, this gain in accuracy only in rare cases outweighs the computational overhead related to solving the underlying linear system. In the case of the explicit $\mathcal{RKRD}$ framework this linear system is diagonal and therefore very cheap to solve.

Chapter 7

Conclusions

In this work, the framework of multidimensional upwind residual distribution methods for solving hyperbolic conservation laws has been studied. The emphasis has been laid on methods for time-dependent problems, in particular those incorporating Runge-Kutta integration in time (e.g. the first order forward Euler methods and second order TVD Runge-Kutta method of Osher et al. [97]). In the steady state setting, residual distribution methods have already reached a high level of sophistication and proved to be a very successful alternative to other widely used frameworks, e.g. finite volume and discontinuous Galerkin methods. Extension to time-dependent problems, although possible and already achieved, brings additional challenges which yet need to be fully resolved. The main challenge is to design a scheme with all the desired properties, and which achieves this at a relatively low computational cost. Constructing such a method has been the main underlying goal of this thesis.

The main focus of interest in this thesis has been the derivation and investigation of *second* order accurate schemes. The underlying discrete representation of the numerical solution has been assumed to be piece-wise linear, in case of which such an accuracy requirement is a natural research goal. Whilst considering numerical methods for hyperbolic PDEs, it is a very frequent requirement that the resulting numerical approximation to the analytic solution of the underlying PDE is free of spurious and non-physical oscillations. Constructing a scheme that is both positive and second order accurate is even more challenging and has been intensively discussed throughout this work. In order to guarantee that the resulting discretization is efficient, only explicit time-stepping methods have been considered. Although such choice does not guarantee that overall the resulting scheme is explicit, two discretisations which are indeed fully explicit were considered, i.e. the unsteady $\mathcal{RD}$ and explicit $\mathcal{RKRD}$ [85] frameworks, discussed in Chapter 4. The simple combination of the second order TVD Runge-Kutta method [97] with the residual distribution framework, i.e. the implicit $\mathcal{RKRD}$ framework (see Chapter 4), is an example of an approach where explicit time-stepping does not guarantee that the overall discretisation is fully explicit. Fortunately, the resulting linear system is sparse and a robust and efficient way of solving it was presented. This was done with the aid of the PETSc numerical library. Moreover, the implicit scheme was, in majority of the tested situations, more accurate than its explicit counterpart, which somehow compensates for the extra overhead due to solving a linear system.

One of the most recent strands of research within the residual distribution community is discontinuous methods [3, 14, 57, 58, 60, 61, 105] (see also Chapter 3). Both, discontinuous-in-space and discontinuous-in-time approximations are being intensively studied and developed. In this thesis, the focus was laid on the former approach, the latter being derivative of frameworks based on implicit time-stepping methods (the so-called space-time residual distribution methods [10,29,32,34,38,44]) and which are not considered here. The first successful attempt to apply the discontinuous-in-space residual distribution framework to time-dependent problems [105] focused on first order approximations (i.e. the forward Euler method was used to integrate in time). Constructing a genuinely second order method for time-dependent problems turned out to be much more challenging than originally anticipated. During this process a few shortcomings of the discontinuous-in-space framework have been discovered (see Chapter 5). Extensive numerical experiments showed that when the underlying fluid flow is aligned (or almost aligned) with the mesh, then the order of accuracy drops down to one. The solution to this problem, a new distribution strategy for edges, removed this anomaly. However, this new distribution leads to a residual distribution method which is very similar to the discontinuous Galerkin approximation [23] and consequently makes the whole discontinuous-in-space $\mathcal{RD}$ approach less appealing. It is worth pointing out, though, that it is very unlikely that in practical applications for which grids are more often than not unstructured, the fluid flow will indeed be aligned with the mesh. This means that even the distribution strategies for which the order of accuracy for mesh-aligned flows drops down to one remain interesting. These distributions, in particular the mED method, are more faithful to the $\mathcal{RD}$ spirit and bear far less resemblance to solutions known from the $\mathcal{DG}$ approach. There is yet another major flaw of the discontinuous-in-space $\mathcal{RD}$ framework that was discovered while investigating second order schemes for time-dependent problems. This flaw is the fact that the discontinuous $\mathcal{RKRD}$ framework, at least in its current state, proved to be incapable of solving non-linear equations. This discussion was carried out in Chapter 5. In the future, a "slope limiting" procedure (see the approach employed in the discontinuous Galerkin framework [27]) will have to be introduced to overcome this poor behaviour when dealing with non-linear equations. Interestingly enough, the first order discontinuous-in-space approximations (i.e. the discontinuous unsteady $\mathcal{RD}$ methods) coped with non-linear equations with no extra effort [105]. Although the research presented here has not led to a fully developed second order accurate discontinuous-in-space residual distribution framework, the discussion and analysis carried out alongside gives a new and very thorough insight into this approach.

The above overview briefly summarises the contents of Chapters 2-5. Chapter 6 deals with the application of methods discussed in this thesis to solve the Euler equations of fluid dynamics. As such, it is not meant to introduce any new concepts. Instead, it focuses on introducing the existing methodology for extending residual distribution methods to the Euler equations [39, 102] and applying it to the algorithms investigated in this thesis. The chapter as whole should be regarded as a very extensive collection of numerical results that demonstrate robustness of discretisations techniques studied in this thesis.

7.1 Contributions

To summarise, major contributions of this thesis are listed below.

- **Discontinuous Residual Distribution Framework:** The framework of discontinuous residual distribution methods has been extended by designing one new distribution strategy for edge-based residuals, i.e. the dcmED method. A thorough review and numerical comparison of all the available distribution strategies for edge-based residuals demonstrated that, as far as steady state problems are concerned, the mED distribution of Hubbard [57] performs the best, i.e. the resulting scheme is second order accurate and positive. The dcmED distribution leads to a second order accurate scheme, but the solution cannot be guaranteed to be free of spurious oscillations. The LF distribution of Abgrall [3] gives only first order approximations and is too diffusive to be of any practical interest. This study was presented in Chapter 3.

- **Discontinuous Unsteady Residual Distribution Framework [105]:** This framework is the first successful attempt to construct discontinuous-in-space residual distribution methods for unsteady problems. Due to the low order time-stepping method used in this approach (first order forward Euler method), at most first order accurate schemes can be designed within this framework. Indeed, presented numerical results confirm that schemes developed within this framework are first order accurate. On the other hand, imposing positivity is very straightforward and this was validated experimentally for both linear and non-linear problems. This framework was introduced in Chapter 5.

- **Implicit Runge-Kutta Residual Distribution Framework:** Recently, Ricchiuto et al. [85] introduced the explicit $\mathcal{RKRD}$ framework. Their approach guarantees that the underlying system of equations describing the relation between the solution at two consequent time levels is linear and diagonal. Within the proposed approach, second order schemes were constructed. With respect to positivity, the presented results are promising, though not 100% oscillation-free. The authors did not give any indication how this could be improved. Instead, they focused on designing a relatively efficient and accurate scheme. The implicit $\mathcal{RKRD}$ approach presented here is an alternative to their approach. Although in this case the relation between the solution at two consecutive time steps is described by a non-diagonal matrix, extensive numerical results presented here show that the related overhead is not as profound as originally expected. More important, the implicit framework is in many cases more accurate than its explicit counterpart and provides a clear indication where to look for improvements - introducing a non-linear mass matrix (by modifying the blending procedure) will possibly improve the results presented here. In particular in terms of positivity. This should be studied in more detail in the future. Obviously, the expense of obtaining the solution will increase. Nevertheless, without a thorough investigation it is impossible to judge whether this extra cost will or will not introduced unbalanced overhead. This study was carried out in Chapters 4 and 6.

- **The Discontinuous Runge-Kutta Residual Distribution Framework:** Combining together ideas from the discontinuous-in-space and Runge-Kutta residual distribution frameworks turned out to be more challenging then originally expected. In terms of the constructed numerical methods, one can find that the results presented here are a bit disappointing. Although the proposed schemes are second order accurate, they *blow up* when applied to non-linear equations. On the other hand, the research carried out whilst working on those methods led to a number of interesting analytical results on the discontinuous-in-space framework, and ultimately to a thorough comparison between the discontinuous $\mathcal{RD}$ and discontinuous Galerkin methods. For instance, it was shown that the mED distribution strategy can be viewed as a first order approximation to DG-upwind distribution. Also, it was shown that any discontinuous $\mathcal{RKRD}$ for dcmED strategy was used to distribute edge residuals is equivalent to the discontinuous Galerkin approximation. See Chapter 5 for more details.

7.2 Future Work

Various research avenues have been opened up by the results presented in this thesis:

- The discontinuous implicit $\mathcal{RKRD}$ framework is not complete. Extending it to non-linear equations is currently the key challenge. Due to its close relation to the $\mathcal{DG}$ framework, it very likely can be achieved with the aid of limiting techniques used in the latter approach.

- In the implicit $\mathcal{RKRD}$ framework, the blending procedure was deliberately designed in such a way that the resulting system of equations was linear. Although the resulting scheme behaved very well with respect to the presence of spurious physical oscillations, there is still room for improvement. One possibility would be to modify the blending procedure so that the resulting system of equations is non-linear and see how it affects the positivity. Such an approach could be devised by heuristically extending a similar procedure for steady state problems.

- In order to develop better understanding of the discontinuous-in-space residual distribution framework, the truncation error analysis following the methodology of Abgrall [1] can be carried out. The methodology devised originally for continuous methods, could be extended to the discontinuous-in-space setting by incorporating the edge-based signals into the original analysis. This would potentially result in introducing more genuinely second order discontinuous-in-space $\mathcal{RKRD}$ methods.

- A thorough investigation and comparison of different techniques for prescribing boundary conditions for the Euler equations will contribute to the rigour of numerical methods for systems of non-linear hyperbolic PDEs.
- Explicit Runge-Kutta residual distribution methods have successfully been applied to the equations of Shallow Water Flows [84]. Similar extension in terms of the implicit $\mathcal{RKRD}$ framework can also be considered.

Additionaly, one could consider extending the presented framework to higher than second order discretisations by following the methodology of either Abgrall and Roe [13], Careani and Fuchs [19] or Mebrate [74]. For extension to 3-dimensional problems refer to the approach of Abgrall and Marpeau [9].

Appendices

Appendix A

Analytical Solution to Burgers' Equation

For the purpose of experimental investigation of Chapters 4 and 5, as one of the test cases, nonlinear inviscid Burgers' equation was implemented (i.e. Test Case F):

$$\partial_t \, u \, + \, \nabla \cdot \mathbf{f}(u) \;=\; 0 \qquad \text{on} \quad \Omega_t = \Omega \times [0,1], \tag{A.1}$$

with $\mathbf{f} = (\frac{u^2}{2}, \frac{u^2}{2})$ and $\Omega \subset \mathbb{R}^2$. Although plots of the exact solution were given (see Figure 4.5), the analytical formula describing it was omitted. In general, finding such a formula is not an easy task and the result very often is too complicated to be considered practical. In some special cases, though, it is possible to give a clear and simple answer. One particular example of such a situation was considered in Chapters 4 and 5, where piece-wise constant initial conditions were prescribed. To see how the solution to that problem is constructed, its one-dimensional counterpart will be first derived.

A.1 The 1D Riemann Problem

To start with, consider the one-dimensional equivalent of (A.1):

$$\partial_t \, u \, + \, \partial_x \, f(u) \;=\; 0 \qquad \text{on} \quad [a,b] \times [0,1],$$

with $f(u) = \frac{u^2}{2}$, $a, b \in \mathbb{R}$ $(a < 0 < b)$ and the initial condition set to:

$$u(x,0) = \begin{cases} u_l & \text{if} \quad x < 0, \\ u_r & \text{if} \quad x > 0, \end{cases}$$

where u_l and u_r are two constant states. The above problem (piece-wise constant initial data having a single discontinuity) is known as the Riemann problem. The solution takes one of two forms depending on the relation between u_l and u_r.

Case I $\quad u_l > u_r$

The unique weak solution in this case is given by:

$$u(x,t) = \begin{cases} u_l & \text{if} \quad x < st, \\ u_r & \text{if} \quad x > st, \end{cases}$$

with the shock speed s defined as:

$$s = \frac{f(u_l) \; - \; f(u_r)}{u_l \; - \; u_r}.$$

Case II $\quad u_l < u_r$

This time the weak solution is not unique. Only the so-called rarefaction wave is stable to perturbations and physically relevant. It is given by:

$$u(x,t) = \begin{cases} u_l & \text{if} \quad x < f'(u_l)t, \\ v(x/t) & \text{if} \quad f'(u_l)t \leq x \leq f'(u_r)t \\ u_r & \text{if} \quad x > f'(u_r)t, \end{cases}$$

where $v(\zeta)$ is the solution to $f'(v(\zeta)) = \zeta$. For $f(u) = \frac{u^2}{2}$ the derivative of $f(u)$ is equal to u and $v(\zeta) = \zeta$. As discussed in [68], the above formula is also true in the more general case where f is an arbitrary convex function.

Riemann problems were discussed in many classical text-books on hyperbolic PDEs, for instance in [47, 48, 69] and [101]. A very clear derivation of the above solution can be found in the monograph by LeVeque [68]. Although the above formulation specifies the solution in only two particular situations, it can be easily extended to more general case when there are more discontinuities in the data. To this end it suffices to divide the domain into sub-domains containing only one discontinuity each, and to solve the equation separately on every one of them.

A.2 The 2D Riemann Problem

In order to solve (A.1), rotate the coordinate system to get a family of adjacent one-dimensional problems. The new coordinates ξ and η are given by

$$\xi \;=\; \frac{\sqrt{2}}{2}x + \frac{\sqrt{2}}{2}y, \qquad \eta \;=\; \frac{\sqrt{2}}{2}x - \frac{\sqrt{2}}{2}y,$$

so that

$$\nabla \cdot \mathbf{f}(u) = u\,(u_x + u_y) \;=\; u\,(u_\xi \xi_x + u_\eta \eta_x + u_\xi \xi_y + u_\eta \eta_y) \;=\; \sqrt{2}\,\partial_\xi\, f(u).$$

In other words, the two-dimensional problem is equivalent to its one-dimensional counterpart in the new coordinate system:

$$\partial_t\, u + \nabla \cdot \mathbf{f}(u) \;=\; 0 \qquad \Leftrightarrow \qquad \partial_t\, u + \sqrt{2}\,\partial_\xi\, f(u) \;=\; 0.$$

Note that rotating the coordinate system does not affect the initial data. In Chapters 4 and 5 only piece-wise constant initial data was considered and hence the one-dimensional equivalent of (A.1) falls into the class of problems considered in the previous section. Therefore, construction of the analytical solution for Test Case F is accomplished.

Appendix B

Notation

To avoid ambiguity or confusion in the interpretation of the text, a brief description of the notation employed in this thesis is outlined.

Frameworks

Frameworks for steady state problems:

$\mathcal{RD}$	-	residual distribution
$\mathcal{FE}$	-	finite elements
$\mathcal{FV}$	-	finite volumes
$\mathcal{DG}$	-	discontinuous Galerkin
discontinuous $\mathcal{RD}$	-	discontinuous residual distribution

Frameworks for time-dependend problems

$\mathcal{RKRD}$	-	Runge-Kutta residual distribution (2nd order TVD Runge-Kutta method + $\mathcal{RD}$)
$\mathcal{RKDG}$	-	Runge-Kutta discontinuous Galerkin (2nd order TVD Runge-Kutta method + $\mathcal{DG}$)

unsteady $\mathcal{RD}$ - unsteady residual distribution
(1st order forward Euler + $\mathcal{RD}$)

disc. $\mathcal{RKRD}$ - discontinuous Runge-Kutta residual distribution
(2nd order TVD Runge-Kutta method + disc. $\mathcal{RD}$)

disc. unsteady $\mathcal{RD}$ - discontinuous unsteady residual distribution
(1st order forward Euler + discontinuous $\mathcal{RD}$)

Note that for the majority of the frameworks discussed here there is a number of particular schemes that fall into it. These are denoted with standard rather than curly font (e.g. LDA, N, RKRD-LDA). Note also that for the finite element framework, $\mathcal{FE}$, only one scheme is considered and by abuse of notation that scheme is denoted by $\mathcal{FE}$. The FE scheme is a particular type of a $\mathcal{RD}$ scheme constructed by looking at the similarities between the $\mathcal{FE}$ and $\mathcal{RD}$ frameworks. The $\mathcal{FE}$ and FE approximations are regarded as two distinct discretisations. In the case of the discontinuous Galerkin framework, $\mathcal{DG}$, two schemes are considered: DG-DG-upwind (the discontinuous Galerkin scheme with the upwind flux) and DG-DG-LF (the discontinuous Galerkin scheme with the Lax-Friedrichs flux).

Scalars, vectors and matrices

Three distinct forms of notation are used to represent different types of quantities in the text

- Scalar quantities are denoted with lower case letter, standard font (e.g. u, v, p)
- Vectors (regardless the dimensions) are denoted using lower case bold font (e.g. $\mathbf{n}, \mathbf{a}, \boldsymbol{\phi}$)
- Matrices (regardless the dimensions) are denoted using upper case bold font (e.g. $\mathbf{M}$)

In several instances upper case standard font letters are used to denote scalar and vector quantities. Such approach was motivated by the desire to remain consistent with the notation used in the literature. To avoid confusion, every quantity is clearly described in the text.

Mesh related quantities

Throughout the thesis the following notation is used:

E	-	mesh cell
e	-	mesh edge
h	-	mesh parameter
$\mathbf{n}$	-	unit outward pointing normal vector
$\mathcal{T}_h$	-	the triangulation

Conservation law variables

For the scalar equations:

u	-	solution variable
$\mathbf{f}$	-	the flux
$\mathbf{a}$	-	flux Jacobian (advection velocity)

For the Euler equations:

ρ	-	density
p	-	pressure
u, v	-	$x-$ and $y-$ velocities
E_{total}	-	total Energy
a	-	speed of sound
H	-	total enthalpy
$\mathbf{g}, \mathbf{h}$	-	fluxes
$\mathbf{w}$	-	vector of conservative variable
$\mathbf{z}$	-	parameter vector variable

Appendix C

Derivation of The Consistent Mass Matrix F2

In Chapters 4 and 5, the consistent mass matrix (4.4) was used. Recently referred to in the literature as Formulation 2 (or F2), see [85], it is given by the following formula:

$$m_{ij}^E = \frac{|E|}{36}(3\delta_{ij} + 12\beta_i - 1).$$

It was originally proposed in [73] and that derivation will now be recalled here.

Let $\psi_i^{\mathcal{RD}}$ be defined as:

$$\psi_i^{\mathcal{RD}} \;=\; \psi_i + \alpha_i^E, \tag{C.1}$$

with ψ_i being the standard linear Lagrange basis function associated with node i in cell E and α_i^E some weighting coefficient yet to be specified. This weighting coefficient is meant to guarantee that the following relation is true:

$$\int_E \mathbf{a}\cdot\nabla u_h\,\psi_i^{\mathcal{RD}}\,d\Omega \;=\; \beta_i\phi^E,$$

in which β_i is a distribution coefficient resulting from a residual distribution discretisation, see Chapter 2 for details. Now, since $\int_E \mathbf{a}\cdot\nabla u_h\,d\Omega \;=\; \phi^E$, the offset parameter α_i must satisfy the following relation:

$$\alpha_i \;=\; \beta_i - \frac{1}{\phi^E}\int_E \mathbf{a}\cdot\nabla u_h\,\psi_i\,d\Omega,$$

or, using notation from Section 2.3,

$$\alpha_i = \beta_i - \beta_i^{\mathcal{FE}}.$$

Now, approximating $\beta_i^{\mathcal{FE}}$ with $\beta_i^{FE} = \frac{1}{3}$ one obtains:

$$\psi_i^{\mathcal{RD}} = \psi_i + \beta_i - \beta_i^{\mathcal{FE}}.$$

This definition is used in the literature and in this thesis regardless of the equation being discretised. To the author's best knowledge, the effect of approximating $\beta_i^{\mathcal{FE}}$ with β_i^{FE} on the accuracy has not yet been investigated. However, the available numerical results show that the order of accuracy does not deteriorate.

The derivation of the mass matrix (4.4) is now obvious and reads:

$$m_{ij}^E = \int_E \psi_i^{\mathcal{RD}} \psi_j \, d\Omega = |E| \begin{bmatrix} \frac{\beta_1}{3} + \frac{1}{18} & \frac{\beta_1}{3} - \frac{1}{36} & \frac{\beta_1}{3} - \frac{1}{36} \\ \frac{\beta_2}{3} - \frac{1}{36} & \frac{\beta_2}{3} + \frac{1}{18} & \frac{\beta_2}{3} - \frac{1}{36} \\ \frac{\beta_3}{3} - \frac{1}{36} & \frac{\beta_3}{3} - \frac{1}{36} & \frac{\beta_3}{3} + \frac{1}{18} \end{bmatrix}.$$

Note that for the above reasoning to make sense, the distribution coefficientsβ_i have to be bounded, i.e. the underlying residual distribution scheme has to be linearity preserving.

Appendix D

Derivation of the Limit on the Time-Step for the PSI-mED Scheme

Equation (3.18) gives the limit on a time step that guarantees that a steady discontinuous $\mathcal{RD}$ scheme for which the PSI (or N) scheme was used to distribute cell residuals and mED scheme to distribute to edge-based residuals, is positive. This condition was originally presented in [57] but its derivation has never been published. The following reasoning is meant to fill this gap.

Following [82], consider schemes that can be recast in the following abstract form:

$$u_i^{n+1} \;=\; u_i^n - \frac{3\Delta t}{|E|}\left(\sum_{j\in E} c_{ij}(u_i^n - u_j^n) + \sum_{j\in e_{|e\in E}} c_{ij}(u_i^n - u_j^n)\right) \tag{D.1}$$

According to [82] (see also references therein), if

$$c_{ij} \geq 0 \qquad \forall j \in E \quad \text{and} \quad \forall j \in e_{|e\in E},$$

and the time-step Δt satisfies the following condition:

$$\Delta t \leq \frac{|E|}{3(\sum_{j\in E} c_{ij} + \sum_{j\in e|_{e\in E}} c_{ij})} \quad \forall_{i\in\mathcal{T}_h}, \tag{D.2}$$

then Scheme (D.1) is positive.

Assume now that the N scheme is used to distribute cell residuals. It follows (see Section 5.4.2.1 in [82])) that in every cell E the signal that is sent to node $i \in \mathcal{T}_h$ is equal to:

$$\phi_i^N = - \sum_{j \in E, j \neq i} k_i^+ N k_j^- (u_i - u_j),$$

with

$$N = - \left(\sum_{j \in E} k_j^- \right)^{-1}.$$

From the properties of the flow sensors one gets that:

$$- \sum_{j \in E, j \neq i} k_i^+ N k_j^- = k_i^+,$$

which is the sum of the c_{ij} coefficients corresponding to the cell-based residuals. It is clear that in the case of the mED distribution (3.17), the corresponding edge-based c_{ij} coefficients are equal to $(k_i^e)^+ = \frac{1}{2} \left(\mathbf{a}_i \cdot \mathbf{n}_{E,e} \right)^+ |e|$, in which e is one of the two edges that node i belongs to. This shows that Condition (D.2) is equivalent to:

$$\Delta t \ \le \ \frac{1}{3} \frac{|E|}{\left(k_i^E\right)^+ + \left(k_i^{e_1}\right)^+ + \left(k_i^{e_2}\right)^+} \qquad \forall E \in \mathcal{T}_h \quad \forall i \in E,$$

The final step is to show that similar reasoning is true when the PSI scheme (derived by limiting the N scheme) is used. This was shown in [82], Section 5.5.2.

Appendix E

Compact Presentation of the Discontinuous $\mathcal{RD}$ Framework

A Discontinuous Residual Distribution scheme is defined as a scheme that, given a discontinuous initial solution u_h :

$$u_h(\mathbf{x})|_E \;=\; \sum_{i\in E} \psi_i^E(\mathbf{x})u_i^E \qquad \forall \mathbf{x} \in E \quad \forall E \in \mathcal{T}_h,$$

computes the next approximation of the solution, i.e. evolves in time the nodal values of u_h, by implementing the following procedure:

1. $\forall E \in \mathcal{T}_h$ compute the cell residual:

   $$\phi^E \;=\; \int_E \nabla \cdot \mathbf{f}(u)\, d\Omega,$$

   and $\forall e \in E$ calculate the edge residuals:

   $$\phi^e(u_h) \;=\; -\int_e [\mathbf{f}(u_h) \cdot \mathbf{n}]\; d\Gamma,$$

2. $\forall E \in \mathcal{T}_h$ and $\forall e \in E$ distribute fractions of ϕ^E and ϕ^e between the nodes of E. Denoting by ϕ_i^E the signal sent to node i from cell E (i.e. the fraction of

the cell residual ϕ^E assigned to node $i \in E$), by construction one must have

$$\sum_{j \in E} \phi_i^E \;=\; \phi^E.$$

Equivalently, denoting by β_i the distribution coefficient corresponding to node i :

$$\beta_i^E \;=\; \frac{\phi_i^E}{\phi^E},$$

one must have by construction

$$\sum_{j \in E} \beta_j \;=\; 1.$$

Similar observation holds for ϕ_i^e, but one has to remember that each edge e belongs to two adjacent cells: E and E' (cf. Figure 3.1). Denoting by ϕ_i^e the signal sent to node i from edge $e \in E$ (i.e. the fraction of the edge residual ϕ^e assigned to node $i \in E$), by construction one must have

$$\sum_{j \in E \cup E'} \phi_i^e \;=\; \phi^e.$$

Equivalently, denoting by α_i the distribution coefficient corresponding to node i :

$$\alpha_i^e \;=\; \frac{\phi_i^e}{\phi^e},$$

one must have by construction

$$\sum_{j \in E \cup E'} \alpha_j \;=\; 1.$$

3. $\forall i \in \mathcal{T}_h$ assemble the contributions from E and all $e \in E$ ($i \in E$) and evolve u_i in time according to (see Equation (3.4))

$$u_i^{n+1} \;=\; u_i^n \;-\; \frac{3\Delta t}{|E|}\left(\beta_i \phi^E \;+\; \sum_{e \in E} \alpha_i \phi^e\right) \qquad \forall i.$$

Bibliography

[1] R. Abgrall. Toward the ultimate conservative scheme: following the quest. *J. Comput. Phys.*, 167(2):277–315, 2001.

[2] R. Abgrall. Essentially non-oscillatory residual distribution schemes for hyperbolic problems. *J. Comput. Phys.*, 214(2):773–808, 2006.

[3] R. Abgrall. A residual distribution method using discontinuous elements for the computation of possibly non smooth flows. *Adv. in Appl. Math. Mech.*, 2(1):32–44, 2010.

[4] R. Abgrall. A review of residual distribution schemes for hyperbolic and parabolic problems: the July 2010 state of the art. *Commun. Comput. Phys.*, 11(4):1043–1080, 2012.

[5] R. Abgrall and T. Barth. Residual distribution schemes for conservation laws via adaptive quadrature. *SIAM J. Sci. Comput.*, 24(3):732–769, 2002.

[6] R. Abgrall, G. Baurin, P. Jacq, and M. Ricchiuto. Some examples of high order simulations parallel of inviscid flows on unstructured and hybrid meshes by residual distribution schemes. *Comput & Fluids*, 61:6 – 13, 2012.

[7] R. Abgrall, H. Deconinck, and K. Sermeus. Status of multidimensional upwind residual distribution schemes and applications in aeronautics. In *AIAA Paper 2000-2328, Fluids 2000/Denver*, June 2000.

[8] R. Abgrall, A. Larat, and M. Ricchiuto. Construction of very high order residual distribution schemes for steady inviscid flow problems on hybrid unstructured meshes. *J. Comput. Phys.*, 230(11):4103–4136, 2011.

[9] R. Abgrall and F. Marpeau. Residual distribution schemes on quadrilateral meshes. *Journal of Scientific Computing*, 30(1):131–175, 2007.

[10] R. Abgrall and M. Mezine. Construction of second order accurate monotone and stable residual distribution schemes for unsteady flow problems. *J. Comput. Phys.*, 188(1):16–55, 2003.

[11] R. Abgrall and M. Mezine. Residual distribution schemes for steady problems. In *Computational Fluid Dynamics, VKI LS 2003-05*. von Karman Institute for Fluid Dynamics, 2003.

[12] R. Abgrall and M. Mezine. Construction of second-order accurate monotone and stable residual distribution schemes for steady problems. *J. Comput. Phys.*, 195(2):474–507, 2004.

[13] R. Abgrall and P. L. Roe. High order fluctuation schemes on triangular meshes. *J. Sci. Comput.*, 19(1-3):3–36, 2003. Special issue in honor of the sixtieth birthday of Stanley Osher.

[14] R. Abgrall and C.-W. Shu. Development of residual distribution schemes for the discontinuous Galerkin method: the scalar case with linear elements. *Commun. Comput. Phys.*, 5(2-4):376–390, 2009.

[15] S. Balay, J. Brown, K. Buschelman, W. D. Gropp, D. Kaushik, M. G. Knepley, L. Curfman McInnes, B. F. Smith, and H. Zhang. PETSc Web page, 2012. http://www.mcs.anl.gov/petsc.

[16] S. Balay, J. Brown, V. Buschelman, K. Eijkhout, W. D. Gropp, D. Kaushik, M. G. Knepley, L. C. McInnes, B. F. Smith, and H. Zhang. PETSc users manual. Technical Report ANL-95/11 - Revision 3.3, Argonne National Laboratory, 2012.

[17] D. Braess. *Finite elements*. Cambridge University Press, Cambridge, third edition, 2007.

[18] A. N. Brooks and T. J. R. Hughes. Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations. *Comput. Methods Appl. Mech. Engrg.*, 32(1-3):199–259, 1982.

[19] D. Caraeni and L. Fuchs. Compact third-order multidimensional upwind scheme for Navier-Stokes simulations. *Theor. Comput. Fluid Dyn.*, 15(6):373–401, 2002.

[20] J.-C. Carette, H. Deconinck, H. Paillère, and P. L. Roe. Multidimensional upwinding: its relation to finite elements. *Internat. J. Numer. Methods Fluids*, 20(8-9):935–955, 1995.

[21] C. J. Chapman. *High speed flow*. Cambridge Texts in Applied Mathematics. Cambridge University Press, Cambridge, 2000.

[22] B. Cockburn. Discontinuous Galerkin methods for convection-dominated problems. In *High-order methods for computational physics*, volume 9 of *Lect. Notes Comput. Sci. Eng.*, pages 69–224. Springer, Berlin, 1999.

[23] B. Cockburn, S. Hou, and C.-W. Shu. The Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws. IV. The multidimensional case. *Math. Comp.*, 54(190):545–581, 1990.

[24] B. Cockburn, S. Y. Lin, and C.-W. Shu. TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws. III. One-dimensional systems. *J. Comput. Phys.*, 84(1):90–113, 1989.

[25] B. Cockburn and C.-W. Shu. TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws. II. General framework. *Math. Comp.*, 52(186):411–435, 1989.

[26] B. Cockburn and C.-W. Shu. The Runge-Kutta local projection P^1-discontinuous-Galerkin finite element method for scalar conservation laws. *RAIRO Modél. Math. Anal. Numér.*, 25(3):337–361, 1991.

[27] B. Cockburn and C.-W. Shu. The Runge-Kutta discontinuous Galerkin method for conservation laws. V. Multidimensional systems. *J. Comput. Phys.*, 141(2):199–224, 1998.

[28] R. Courant and K. O. Friedrichs. *Supersonic flow and shock waves*. Springer-Verlag, New York, 1976. Reprinting of the 1948 original, Applied Mathematical Sciences, Vol. 21.

[29] Á. Csík and H. Deconinck. Space-time residual distribution schemes for hyperbolic conservation laws on unstructured linear finite elements. *Internat. J. Numer. Methods Fluids*, 40(3-4):573–581, 2002.

[30] Á. Csík, H. Deconinck, and S. Poedts. Monotone residual distribution schemes for the ideal magnetohydrodynamics equations on unstructured grids. *AIAA Journal*, 39(8):1532–1541, 2001.

[31] Á. Csík, H. Deconinck, and M. Ricchiuto. Residual distribution for general time-dependent conservation laws. *J. Comput. Phys.*, 209(1):249–289, 2005.

[32] Á. Csík, H. Deconinck, M. Ricchiuto, and S. Poedts. Space-time residual distribution schemes for hyperbolic conservation laws. In *15th AIAA Computational Fluid Dynamics Conference, Anaheim, CA, USA*, June 2001.

[33] Á. Csík, M. Ricchiuto, and H. Deconinck. A conservative formulation of the multidimensional upwind residual distribution schemes for general nonlinear conservation laws. *J. Comput. Phys.*, 179(1):286–312, 2002.

[34] P. De Palma, G. Pascazio, G. Rossiello, and M. Napolitano. A second-order-accurate monotone implicit fluctuation splitting scheme for unsteady problems. *J. Comput. Phys.*, 208(1):1–33, 2005.

[35] H. Deconinck. Upwind methods and multidimensional splittings for the Euler equations. In *Computational Fluid Dynamics, VKI LS 1991-01*. von Karman Institute for Fluid Dynamics, 1991.

[36] H. Deconinck and A. Ferrante. Solution of the unsteady Euler equations using residual distribution and flux corrected transport. Technical Report 97-08, von Karman Institute for Fluid Dynamics, 1997.

[37] H. Deconinck and M. Ricchiuto, editors. *34th CFD - higher order discretization methods, November 14-18, 2005*. VKI LS 2006-01. The von Karman Institute for Fluid Dynamics, 2006.

[38] H. Deconinck and M. Ricchiuto. Residual distribution schemes: foundations and analysis. In *Encyclopedia of Computational Mechanics*, volume 3. John Wiley and Sons, Ltd., 2007.

[39] H. Deconinck, P. L. Roe, and R. Struijs. A multidimensional generalization of Roe's flux difference splitter for the Euler equations. *Comput. & Fluids*, 22(2-3):215–222, 1993.

[40] H. Deconinck, R. Struijs, G. Bourgois, H. Paillére, and P. L. Roe. Multidimensional upwind methods for unstructured grids. In *Special Course on Unstructured Grid Methods for Advection Dominated Flows (AGARD Report 787)*. von Karman Institute for Fluid Dynamics, 1992.

[41] H. Deconinck, R. Struijs, G. Bourgois, and P. L. Roe. High resolution shock capturing cell vertex advection schemes on unstructured grids. In *Computational Fluid Dynamics, VKI LS 1994-05*. von Karman Institute for Fluid Dynamics, 1994.

[42] G. Degrez and van der Weide E. Upwind residual distribution schemes for chemical nonequilibrium flows. In *14th AIAA Computational Fluid Dynamics Conference,Norfolk*, 1999.

[43] J. W. Demmel. *Applied numerical linear algebra*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1997.

[44] J. Dobeš and H. Deconinck. Second order blended multidimensional upwind residual distribution scheme for steady and unsteady computations. *J. Comput. Appl. Math.*, 215(2):378–389, 2008.

[45] A. F. Emery. An evaluation of several differencing methods for inviscid fluid flow problems. *J. Comput. Phys.*, 2:306–331, 1968.

[46] M. Feistauer. *Mathematical methods in fluid dynamics*, volume 67 of *Pitman Monographs and Surveys in Pure and Applied Mathematics*. Longman Scientific & Technical, Harlow, 1993.

[47] M. Feistauer, J. Felcman, and I. Straškraba. *Mathematical and computational methods for compressible flow*. Numerical Mathematics and Scientific Computation. The Clarendon Press Oxford University Press, Oxford, 2003.

[48] E. Godlewski and P. A. Raviart. *Numerical approximation of hyperbolic systems of conservation laws*, volume 118 of *Applied Mathematical Sciences*. Springer-Verlag, New York, 1996.

[49] S. K. Godunov. A difference method for numerical calculation of discontinuous solutions of the equations of hydrodynamics. *Mat. Sb. (N.S.)*, 47 (89):271–306, 1959.

[50] S. M. J. Guzik. *Accurate Residual-Distribution Schemes for Accelerated Parallel Architectures*. PhD thesis, University of Toronto, 2010.

[51] S. M. J. Guzik and C. P. T. Groth. Comparison of solution accuracy of multidimensional residual distribution and Godunov-type finite-volume methods. *Int. J. Comput. Fluid Dyn.*, 22(1-2):61–83, 2008.

[52] Ami Harten and Stanley Osher. Uniformly high-order accurate nonoscillatory schemes. I. *SIAM J. Numer. Anal.*, 24(2):279–309, 1987.

[53] R. Hartmann and P. Houston. Adaptive discontinuous Galerkin finite element methods for nonlinear hyperbolic conservation laws. *SIAM J. Sci. Comput.*, 24(3):979–1004 (electronic), 2002.

[54] J. S. Hesthaven and T. Warburton. *Nodal discontinuous Galerkin methods*, volume 54 of *Texts in Applied Mathematics*. Springer, New York, 2008.

[55] P. Houston, C. Schwab, and E. Süli. Stabilized *hp*-finite element methods for first-order hyperbolic problems. *SIAM J. Numer. Anal.*, 37(5):1618–1643 (electronic), 2000.

[56] M. E. Hubbard. *Multidimensional Upwinding and Grid Adaptation for Conservation Laws*. PhD thesis, The University of Reading, 1996.

[57] M. E. Hubbard. Discontinuous fluctuation distribution. *J. Comput. Phys.*, 227(24):10125–10147, 2008.

[58] M. E. Hubbard. A framework for discontinuous fluctuation distribution. *Internat. J. Numer. Methods Fluids*, 56(8):1305–1311, 2008.

[59] M. E. Hubbard and N. Z. Mebrate. Very high order, non-oscillatory fluctuation distribution schemes. In *Computational Fluid Dynamics 2006*, pages 65–70. Springer–Verlag, 2009.

[60] M.E. Hubbard and M. Ricchiuto. Discontinuous fluctuation distribution: A route to unconditional positivity and high order accuracy. In *ICFD 2010 International Conference on Fluid Dynamics, Reading (UK)*, April 2010.

[61] M.E. Hubbard, M. Ricchiuto, and D. Sármány. Unconditionally stable space-time discontinuous residual distribution for shallow-water flows. Submitted.

[62] T. J. R. Hughes and M. Mallet. A new finite element formulation for CFD III: the generalized streamline operator for multidimensional advective-diffusive systems. *Comput. Methods Appl. Mech. Engrg.*, 58:305–328, 1986.

[63] Guang-Shan Jiang and Chi-Wang Shu. Efficient implementation of weighted ENO schemes. *J. Comput. Phys.*, 126(1):202–228, 1996.

[64] C. Johnson. *Numerical solution of partial differential equations by the finite element method*. Dover Publications Inc., Mineola, NY, 2009. Reprint of the 1987 edition.

[65] Dietmar Kröner. *Numerical schemes for conservation laws*. Wiley-Teubner Series Advances in Numerical Mathematics. John Wiley & Sons Ltd., 1997.

[66] A. Lani, M. Panesi, and H. Deconinck. Conservative residual distribution method for viscous double cone flows in thermochemical nonequilibrium. *Commun. Comput. Phys.*, 13(2):479–501, 2013.

[67] P. D. Lax. *Hyperbolic systems of conservation laws and the mathematical theory of shock waves*. Society for Industrial and Applied Mathematics, Philadelphia, Pa., 1973. Conference Board of the Mathematical Sciences Regional Conference Series in Applied Mathematics, No. 11.

[68] R. J. LeVeque. *Numerical methods for conservation laws*. Lectures in Mathematics ETH Zürich. Birkhäuser Verlag, Basel, second edition, 1992.

[69] R. J. LeVeque. *Finite volume methods for hyperbolic problems*. Cambridge Texts in Applied Mathematics. Cambridge University Press, Cambridge, 2002.

[70] B. Q. Li. *Discontinuous finite elements in fluid dynamics and heat transfer*. Computational Fluid and Solid Mechanics. Springer-Verlag London Ltd., London, 2006.

[71] Xu-Dong Liu, Stanley Osher, and Tony Chan. Weighted essentially non-oscillatory schemes. *J. Comput. Phys.*, 115(1):200–212, 1994.

[72] A. Majda. *Compressible fluid flow and systems of conservation laws in several space variables*, volume 53 of *Applied Mathematical Sciences*. Springer-Verlag, New York, 1984.

[73] J. März and G. Degrez. Improving time accuracy of residual distribution schemes. Technical Report 96-17, von Karman Institute for Fluid Dynamics, 1996.

[74] N. Z. Mebrate. *High Order Fluctuation Splitting Schemes for Hyperbolic Conservation Laws*. PhD thesis, University of Leeds, 2007.

[75] L.M. Mesaros and P. L. Roe. Multidimensional fluctuation splitting schemes based on decomposition methods. In *12th AIAA Computational Fluid Dynamics Conference*, 1995.

[76] R.H. Ni. A multiple-grid scheme for solving the Euler equations. *AIAA Journal*, 20(11):1565–1571, 1982.

[77] H. Paillére. *Multidimensional Upwind Residual Distribution Schemes for the Euler and Navier-Stokes equations on Unstructured Grids*. PhD thesis, VKI, 1995.

[78] H. Paillére, E. van der Weide, and H. Deconinck. Multidimensional upwind methods for inviscid and viscous compressible flows. In *Computational Fluid Dynamics, VKI LS 1995-02*. von Karman Institute for Fluid Dynamics, 1995.

[79] N Petrovskaya. Personal communication.

[80] N. B. Petrovskaya. On oscillations in discontinuous Galerkin discretization schemes for steady state problems. *SIAM J. Sci. Comput.*, 27(4):1329–1346, 2006.

[81] M Ricchiuto. Personal communication.

[82] M. Ricchiuto. *Construction and analysis of compact residual discretizations for conservation laws on unstructured meshes, PhD Thesis*. PhD thesis, VKI, 2005.

[83] M. Ricchiuto. Contributions to the development of residual discretizations for hyperbolic conservation laws with application to shallow water flows. HDR Thesis, 2011.

[84] M. Ricchiuto. Explicit residual discretizations for shallow water flows. *AIP Conference Proceedings*, 1389(1):919–922, 2011.

[85] M. Ricchiuto and Abgrall. Explicit Runge–Kutta residual distribution schemes for time dependent problems: second order case. *J. Comput. Phys.*, 229(16):5653–5691, 2010.

[86] M. Ricchiuto, R. Abgrall, and H. Deconinck. Application of conservative residual distribution schemes to the solution of the shallow water equations on unstructured meshes. *J. Comput. Phys.*, 222(1):287–331, 2007.

[87] M. Ricchiuto and A. Bollermann. Accuracy of stabilized residual distribution for shallow water flows including dry beds. In *Hyperbolic problems: theory, numerics and applications*, volume 67 of *Proc. Sympos. Appl. Math.*, pages 889–898. Amer. Math. Soc., Providence, RI, 2009.

[88] P. L. Roe. Approximate Riemann solvers, parameter vectors, and difference schemes. *J. Comput. Phys.*, 43(2):357–372, 1981.

[89] P. L. Roe. Fluctuations and signals - a framework for numerical evolution problems. In *Numerical Methods for Fluid Dynamics*, pages 219–257. Academic Press, 1982.

[90] P. L. Roe. Upwind schemes using various formulations of the Euler equations. In *Numerical methods for the Euler equations of fluid dynamics (Rocquencourt, 1983)*, pages 14–31. SIAM, Philadelphia, PA, 1985.

[91] P. L. Roe. A basis for upwind differencing of the two-dimensional unsteady Euler equations. In *Numerical methods for fluid dynamics, II (Reading, 1985)*, volume 7 of *Inst. Math. Appl. Conf. Ser. New Ser.*, pages 55–80. Oxford Univ. Press, New York, 1986.

[92] P. L. Roe. Linear advection schemes on triangular meshes. Technical Report Technical Report CoA 8720, Cranfield Institute of Technology, 1987.

[93] P. L. Roe and D. Sidilkover. Optimum positive linear schemes for advection in two and three dimensions. *SIAM J. Numer. Anal.*, 29(6):1542–1568, 1992.

[94] P.L Roe and L.M. Mesaros. Solving steady mixed conservation laws by elliptic/hyperbolic splitting. In *Proceedings of 15th ICNMFD Conference*, 1996.

[95] D. Sármány and M.E. Hubbard. Upwind residual distribution for shallow-water ocean modelling. *Ocean Modelling*, 64(0):1 – 11, 2013.

[96] K. Sermeus and H. Deconinck. An entropy fix for multi-dimensional upwind residual distribution schemes. *Comput. & Fluids*, 34(4-5):617–640, 2005.

[97] C.-W. Shu and S. Osher. Efficient implementation of essentially nonoscillatory shock-capturing schemes. *J. Comput. Phys.*, 77(2):439–471, 1988.

[98] G. D. Smith. *Numerical solution of partial differential equationp: Finite difference methods.* Oxford Applied Mathematics and Computing Science Series. The Clarendon Press Oxford University Press, New York, third edition, 1985.

[99] R. Struijs. *A Multi-dimensional Upwind Discretization Methods for the Euler Equations on Unstructured Grids.* PhD thesis, University of Delft, 1994.

[100] R. Struijs, H. Deconinck, and P. L. Roe. Fluctuation splitting schemes for the 2D Euler equations. In *Computational Fluid Dynamics, VKI LS 1991-01.* von Karman Institute for Fluid Dynamics, 1991.

[101] E. F. Toro. *Riemann solvers and numerical methods for fluid dynamics.* Springer-Verlag, Berlin, second edition, 1999.

[102] E. van der Weide and H. Deconinck. Positive matrix distribution schemes for hyperbolic systems, with application to the Euler equations. In JA Desideri, C Hirsch, P LeTallec, M Pandolfi, and J Periaux, editors, *Computational Fluid Dynamics '96*, pages 747–753, 1996.

[103] E. van der Weide, H. Deconinck, E. Issman, and G. Degrez. A parallel, implicit, multi-dimensional upwind, residual distribution method for the Navier-Stokes equations on unstructured grids. *Computational Mechanics*, 23(2):199–208, 1999.

[104] B. van Leer. Towards the ultimate conservative difference scheme. V. A second-order sequel to Godunov's method. *J. Comput. Phys.*, 32(1):101 – 136, 1979.

[105] A. Warzyński, M. E. Hubbard, and M. Ricchiuto. Discontinuous residual distribution schemes for time-dependent problems. In Li, J and Yang, HT and Machorro, E, editor, *Recent Advances In Scientific Computing And Applications*, volume 586 of *Contemporary Mathematics*, pages 375–382, 2013.

[106] G. B. Whitham. *Linear and nonlinear waves.* Pure and Applied Mathematics (New York). John Wiley & Sons Inc., New York, 1999. Reprint of the 1974 original, A Wiley-Interscience Publication.

[107] A. V. Wolkov, Ch. Hirsch, and N. B. Petrovskaya. Application of a higher order discontinuous Galerkin method in computational aerodynamics. *Math. Model. Nat. Phenom.*, 6(3):237–263, 2011.

[108] W. A. Wood and W. L. Kleb. Diffusion characteristics of finite volume and fluctuation splitting schemes. *J. Comput. Phys.*, 153(2):353–377, 1999.

[109] P. Woodward and P. Colella. The numerical simulation of two-dimensional fluid flow with strong shocks. *J. Comput. Phys.*, 54(1):115–173, 1984.

[110] H. C. Yee, R. F. Warming, and A. Harten. Implicit total variation diminishing (TVD) schemes for steady-state calculations. *J. Comput. Phys.*, 57(3):327–360, 1985.