Runge-Kutta Residual Distribution Schemes

Andrzej Warzynski, Matthew Hubbard, Mario Ricchiuto

Runge-Kutta Residual Distribution Schemes

Andrzej Warzyński, Matthew E. Hubbard, Mario Ricchiuto

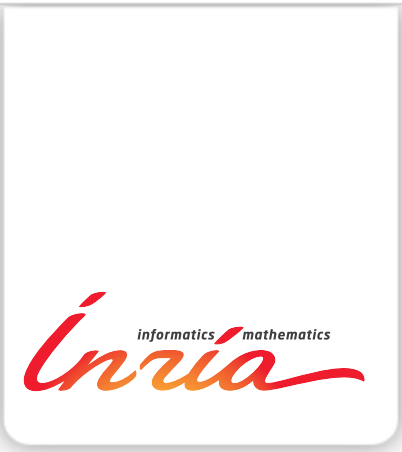
RESEARCH REPORT

N° 8370

September 2013

Project-Teams BACCHUS

ISSN 0249-6399 ISRN INRIA/RR--8370--FR+ENG



Runge-Kutta Residual Distribution Schemes

Andrzej Warzyński*, Matthew E. Hubbard†, Mario Ricchiuto‡

Project-Teams BACCHUS

Research Report n° 8370 — September 2013 — 34 pages

Abstract: We are concerned with the solution of time-dependent nonlinear hyperbolic partial differential equations. We investigate the combination of residual distribution methods with a consistent mass matrix (discretisation in space) and a Runge-Kutta-type time stepping (discretisation in time). The introduced nonlinear blending procedure allows us to retain the explicit character of the time stepping procedure. The resulting methods are second order accurate provided that both spatial and temporal approximations are. The proposed approach results in a global linear system that has to be solved at each time-step. An efficient way of solving this system is also proposed. To test and validate this new framework, we perform extensive numerical experiments on a wide variety of classical problems.

Key-words: Hyperbolic Conservation Laws, Time-dependent problems, Second order schemes, Residual Distribution, Runge-Kutta time-stepping

* School of Computing, University of Leeds, UK
† leeds
‡ Inria Bordeaux - Sud-Ouest

RESEARCH CENTRE
BORDEAUX – SUD-OUEST

351, Cours de la Libération
Bâtiment A 29
33405 Talence Cedex

Runge-Kutta Residual Distribution Schemes

Résumé : On considère l'approximation de solution de lois de conservation hyperboliques avec une combinaison des mèthodes de type Reisdual Distribution avec des schémas Runge-Kutta en temps. On propose une construction de schéma non-linéaire de type Blended qui ne nécéssite pas la résolution d'un systéme non-linéaire, donc permettan de retenir le caractére explicite de la mèthode. L'approche proposée est validée sur des nombreaux cas test.

Mots-clés : Lois de conservation Hyperboliques, problèmes instationnaires, schémas d'ordre deux, Residual Distribution, Runge-Kutta

Contents

1 Introduction

In this paper we study the numerical solution of hyperbolic partial differential equations. In the scalar case these equations take the following form:

$$\partial_t u + \nabla \cdot \mathbf{f}(u) = 0 \qquad \text{in } \Omega \times [0, T]. \tag{1}$$

Ω is the spatial domain and T is the given final time. Equation (1) is equipped with an initial solution:

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

and boundary conditions defined on $\partial\Omega$ or a properly defined subset. The above equation models the evolution of the conserved unknown $u(\mathbf{x}, t)$, the flux of which is denoted here by $\mathbf{f}$. Systems of equations are introduced later in the text. The framework we shall design our schemes in is that of Residual Distribution ($\mathcal{RD}$).

The residual distribution framework (see [19] for a thorough overview) is widely recognised as an efficient and accurate way of discretising steady state hyperbolic PDEs. It was originally introduced by Roe [35] as an alternative to the frequently-used finite volume and finite difference methods. Its ability to perform genuinely multidimensional upwinding [1, 17] enables the construction of approximations free of spurious oscillations, even in the vicinity of shocks in the solution. More important, it facilitates construction of methods which are capable of capturing the underlying physical processes incorporated within the mathematical model in a truly multidimensional manner. Extension to systems, although not straightforward, is well understood and covered in the literature. Usually it is performed with the aid of Roe's parameter vector in the case of the Euler equations [20,34] or, for instance, the CRD approach of Csík et al. [15] in more complex cases. In [1,24] it was demonstrated that when residual distribution methods do provide plausible solutions, these are usually more accurate than those obtained with the aid of more popular finite volume methods.

However, the residual distribution framework lacks the robustness of the finite volume approach, in particular, when it comes to discretising time-dependent equations. Various techniques of extending residual distribution methods to transient problems exist, but none is considered fully satisfactory. The existing approaches can be grouped into two categories: the space-time and consistent mass matrix frameworks.

The *Space-Time* residual distribution framework [4, 12, 13, 16, 21] is very faithful to the $\mathcal{RD}$ and multidimensional upwinding spirit. Although it allows construction of discretisations with *all* the desired properties, those methods are subject to a CFL-type restriction on the time-step. This restriction is particularly disappointing when taking into account that they are, by construction, implicit. In the *two layer* variant [14] one couples two space-time slabs at a time and solves the equations simultaneously in both. The resulting system to be solved at each step is larger, but 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 [12]. More recently, Hubbard and Ricchiuto [25] 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 properties are retained.

In this paper we focus on a different approach. The *Consistent Mass Matrix* formulation was inspired by the discovery of a close relation between the residual distribution and finite element frameworks [10]. This, quite naturally, led to the introduction of a mass matrix m_{ij} (see, in particular, [32] and [18]) and coupling in space of the time derivatives of the nodal values. It was the first successful attempt to construct second-order residual distribution schemes for time-dependent problems. This approach was implemented and investigated in a number of references, i.e. [4, 16, 18, 32] or [9]. 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 this approach is the fact that the resulting discretisations are implicit, i.e. the resulting linear system is not diagonal (even if explicit multi-step methods are utilised) and therefore expensive. In [33] Ricchiuto et al. modified the consistent mass matrix framework by introducing the so-called shifted time operator. By combining it with a multi-stage Runge-Kutta method, they derived a genuinely explicit scheme. The resulting methods are indeed explicit, but the formulation is somewhat complicated and leaves open the question of constructing a genuinely positive method. Recently, the authors of this paper proposed to draw together the discontinuous-in-space residual distribution scheme [2, 26] and the first order forward Euler time-stepping procedure. The resulting scheme [39] is explicit and positive, but so far only first order accurate.

In this contribution we focus on the consistent mass matrix formulation in its original form. To integrate the underlying PDE in time we use the second order TVD Runge-Kutta discretisation method due to Shu and Osher [37]. Although this time-stepping procedure is explicit, our methods are implicit in the sense that at each time step (and each RK stage) a system of linear equations (regardless whether the PDE is linear or non-linear) has to be solved. We show how to solve these linear systems efficiently and investigate the resulting discretisations in terms of accuracy and positivity. The order of accuracy we obtain, i.e. two, is similar as in [33]. However, by avoiding the shifted time

operator we construct a formulation that is more straightforward. The case in which the resulting system of equations is non-linear (i.e. genuinely non-linear schemes) will not be covered here. It is a subject of ongoing research.

This paper is structured as follows. In the following section we introduce the notation and basic assumptions. In Section 3 we introduce the Consistent Mass Matrix formulation and then move to the Runge-Kutta Residual Distribution ($\mathcal{RKCRD}$) framework in Section 4. Extension to systems of non-linear equations is discussed in Section 5 and extensive numerical results are presented in Section 6.

2 The Notation

Although our approach extends to $\mathbb{R}^3$, for clarity and brevity we assume that the spatial domain Ω is embedded in $\mathbb{R}^2$. We also assume that Ω is subdivided into non-overlapping triangular elements, denoted by E, belonging to $\mathcal{T}_h$, such that:

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

$\mathcal{D}_i$ will stand for the subset of triangles containing node $\mathbf{x}_i$. 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:

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

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:

$$\psi_i^E(\mathbf{x}_j) \;=\; \delta_{ij} \;\; \forall i,j \in \mathcal{T}_h, \qquad \sum_{j\in E} \psi_j^E = 1 \;\; \forall E \in \mathcal{T}_h. \tag{2}$$

As long as it does not introduce any ambiguity, the superscript ^{E} will be omitted. At each time level t^n the approximate solution u_h^n is assumed to be globally continuous and linear within each element $E \in \mathcal{T}_h$, and is given by:

$$u_h^n(\mathbf{x}) = \sum_i \psi_i(\mathbf{x})\, u_i^n, \tag{3}$$

where $u_i^n = u_h^n(\mathbf{x}_i)$ are the nodal values of the approximate solution at time t^n. Having decided to focus on piece-wise linear approximations, we aim to design a scheme that is second order accurate.

Last, but not least, we introduce the cell residual $\phi^E(u)$:

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

Cell residuals are one of the key ingredients of all residual distribution approximations. This will become apparent in the following sections.

3 The Consistent Mass Matrix Formulation

Cell residuals introduced in the previous section are used to construct the consistent mass matrix semi-discrete approximation of (1):

$$\sum_{E\in\mathcal{D}_i}\sum_{j\in E} m_{ij}^E \frac{du_j}{dt} + \sum_{E\in\mathcal{D}_i} \beta_i \phi^E = 0. \tag{4}$$

The distribution coefficients β_i $(i = 1, 2, 3)$ define the so-called distribution strategy. Note that each set of these coefficients leads to a distinct approximation. Four examples of such distributions are given in Section 4. Here we only mention that for conservation, in every element $E \in \mathcal{T}_h$, the following condition has to be satisfied:

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

In the case of steady state problems for which piece-wise linear approximations are implemented, existence of $C \in \mathbb{R}$ such that:

$$\beta_i \leq C \qquad \forall E \in \mathcal{T}_h \quad \forall i \in E, \tag{5}$$

i.e. the boundedness of β_i, guarantees accuracy of order two (see [4] for details). The definition of the mass matrix, m_{ij}, that guarantees accuracy of order two in both space and time when transient problems are considered is not unique. Four different approaches are known, only one of which will be employed here. A thorough overview was given in [33] in which it was observed that their Formulation 2 (naming as in [33]) gives best (in terms of accuracy and stability) results. For each cell $E \in \mathcal{T}_h$ this mass matrix is defined as:

$$m_{ij}^E = \frac{|E|}{36}(3\delta_{ij} + 12\beta_i - 1) \tag{6}$$

with δ_{ij} Kronecker's delta. The consistency of this mass matrix with the distribution strategy follows from the dependency of m_{ij}^E on β_i. This formulation was originally derived by März [32]. Its construction is based on the analogy between the linearity preserving $\mathcal{RD}$, i.e. methods for which Condition (5) is satisfied, and the stabilized Galerkin finite element methods.

4 The Runge-Kutta Residual Distribution Framework

The Runge-Kutta Residual Distribution framework is derived by first integrating (1) in time using Runge-Kutta scheme. Here we choose the second order TVD Runge-Kutta time-stepping indicated by Shu and Osher [37], which leads to the following semi-discrete formulation:

$$\begin{cases} \dfrac{\delta u^1}{\Delta t} + \nabla\cdot\mathbf{f}(u^n) = 0, \\ \dfrac{\delta u^{n+1}}{\Delta t} + \dfrac{1}{2}\left(\nabla\cdot\mathbf{f}(u^n) + \nabla\cdot\mathbf{f}(u^1)\right) = 0, \end{cases} \tag{7}$$

in which $\delta u^k = u^k - u^n$ 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 the consistent mass matrix formulation (4) to integrate both stages in (7) in space leads to:

$$\begin{cases} \displaystyle\sum_{E\in\mathcal{D}_i}\sum_{j\in E} m_{ij}^E \frac{\delta u_j^1}{\Delta t} + \sum_{E\in\mathcal{D}_i} \beta_i \phi^E(u^n) = 0, \\ \displaystyle\sum_{E\in\mathcal{D}_i}\sum_{j\in E} m_{ij}^E \frac{\delta u_j^{n+1}}{\Delta t} + \sum_{E\in\mathcal{D}_i} \frac{1}{2}\beta_i \left(\phi^E(u^n) + \phi^E(u^1)\right) = 0. \end{cases} \tag{8}$$

Equation (8) defines two linear systems to be solved at each time-step. These systems can be written in a more compact form as:

$$\begin{cases} u^1 = u^n - \Delta t\, \mathbf{M}^{-1}\, \boldsymbol{\phi}^1, \\ u^{n+1} = u^1 - \Delta t\, \mathbf{M}^{-1}\, \boldsymbol{\phi}^2, \end{cases} \tag{9}$$

which is the form that was employed to carry out numerical experiments in Section 6. In Formulation (9) $\mathbf{M}$ is the global mass matrix, the entries of which are defined by Formula (6), and $\boldsymbol{\phi}^1$ and $\boldsymbol{\phi}^2$ are the vectors of signals each node has received, i.e.

$$\boldsymbol{\phi}_i^1 = \sum_{E\in\mathcal{D}_i} \beta_i \phi^E(u^n), \qquad \boldsymbol{\phi}_i^2 = \sum_{E\in\mathcal{D}_i} \frac{1}{2}\beta_i \left(\phi^E(u^n) + \phi^E(u^1)\right).$$

Formulation (9) (or, equivalently, (8)) defines the second order TVD Runge-Kutta Residual Distribution ($\mathcal{RKRD}$) framework. By choosing different time-stepping routines one obtains different frameworks. However, we believe that our choice is appropriate. The spatial accuracy of the residual distribution framework in the form considered here is at most two. Combining it with higher than second order time-stepping routine is unlikely to increase the accuracy. This was investigated in [33]. Among the second order Runge-Kutta methods the second order TVD method (7) is the most frequent choice, which brings us to Formulation (9). For brevity, hereafter this formulation will be referred to as the Runge-Kutta Residual Distribution ($\mathcal{RKRD}$) framework.

Let us now introduce four examples of algorithms falling into the framework of $\mathcal{RKRD}$ methods. Each such scheme is constructed by first choosing a distribution strategy for cell residuals. This distribution determines the β coefficients which are then substituted into (8) to complete the construction. There is a number of well understood and established distribution techniques for cell residuals within the $\mathcal{RD}$ framework. An overview can be found in [19]. Here we will focus on the

- N, LDA, SU and BLEND

schemes leading to, respectively, the

- RKRD-N, RKRD-LDA, RKRD-SU and RKRD-BLEND

methods.

The RKRD-N scheme

Among linear and positive schemes, the N scheme of Roe [36] allows the

largest time-step and has the smallest cross diffusion [38]. The N scheme is usually defined with the aid of the so-called flow sensors. For each vertex $i \in E$, the corresponding flow sensors are defined as:

$$k_i = -\frac{\mathbf{a}(u)\cdot\mathbf{n}_i}{2}|e_i|, \quad k_i^+ = \max(0,k_i), \quad k_i^- = \min(0,k_i), \tag{10}$$

in which $\mathbf{n}_i$ is the outward pointing unit normal vector to edge e_i (opposite the ith vertex). $|e_i|$ denotes the length of e_i and $\mathbf{a}(u) = \frac{\partial \mathbf{f}}{\partial u}$ is the flux Jacobian. We will also need the following quantities:

$$N = \left(\sum_{j\in E} k_j^+\right)^{-1}, \quad u_{in} = -\sum_{j\in E} N k_j^- u_j.$$

Finally, the distribution itself reads:

$$\beta_i^N \phi^E = k_i^+(u_i - u_{in}).$$

The N scheme is only first order accurate and hence the RKRD-N scheme cannot be expected to be more accurate than that. Recall that consistency of the mass matrix discussed in Section 3 is to guarantee that the accuracy remains two when a linearity preserving scheme is applied to time-dependent problems. However, in the case of the N scheme the linearity preservation condition (i.e. Equation (5)) is not satisfied and hence there is no need for such a consistency condition to be considered. Moreover, the equivalence between the stabilised Galerkin finite element and residual distribution methods was derived on the assumption that the distribution coefficients are bounded (linearity preservation). Again, the N scheme is not linearity preserving and hence Formulation (6) cannot be used. Instead, we replace the mass matrix in this case with the diagonal matrix:

$$m_{ij}^N = \delta_{ij}\frac{|E|}{3}.$$

Note that an identical formulation is obtained by row-lumping the high order mass matrix (6).

The RKRD-LDA scheme

The Low Diffusion A scheme of Roe [36], more often referred to as the LDA scheme, like the N scheme is linear, but not positive. However, this scheme has one nice property which the N scheme lacks, namely the linearity preservation. This means that in smooth regions, provided that the discrete representation of the data is linear, it is second order accurate (see [1] for details). The distribution coefficients for this scheme are given by:

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

The consistent mass matrix that guarantees second order of accuracy is defined by (6).

The RKRD-SU scheme

The SU distribution was inspired by the close link between the residual distribution and Galerkin finite element frameworks (in particular the Streamline Upwind Petrov Galerkin approach [8,28,29]). Its derivation can be found in [19]. The distribution coefficients are given by:

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

in which τ is a scaling parameter, taken here as

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

This scheme is linear and linearity preserving, but not positive. As in the case of the LDA scheme, the consistent mass matrix is defined by (6).

The RKRD-BLEND scheme

Desire to construct methods which are simultaneously linearity preserving and positive brings the need to consider non-linear distributions. As a representative of this class of discretisations, the BLEND scheme will be now introduced. It is obtained by *blending* the two linear schemes presented so far, namely the positive N and the linearity preserving LDA schemes. The signals in this case are defined as:

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

in which $\theta(u_h)$ is a blending coefficient. In this work we propose a blending parameter allowing retention of the explicit nature of the time-marching scheme. In particular, we propose to combine our $\mathcal{RKRD}$ framework with the time-shifted residual evaluation proposed in [33]. In every cell E this blending coefficient is defined as:

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

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

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

$\overline{\delta u^k}$ is the so-called *shifted time-operator* introduced in [33]:

$$\overline{\delta u^k} = u^{k-1} - u^n \tag{11}$$

and e^1 and e^2 are 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, $\overline{\Xi_j^{N(k)}}$ is determined by signals sent by distributing the residuals with the aid of the N scheme and is defined as:

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

The mass matrix for the BLEND scheme is defined as:

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

Note that 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 have been the case and a system of *non-linear* equations would have been constructed instead. Therefore the RKRD-BLEND scheme is not genuinely non-linear. Our aim is to construct efficient schemes and hence the desire to avoid non-linear approximations.

5 Non-linear Systems of Equations

To demonstrate the robustness of our approach, we shall employ it to solve a system of non-linear hyperbolic PDEs, namely the Euler equations of fluid dynamics. The system can be written in a vector form as

$$\partial_t \mathbf{w} + \nabla \cdot \mathbf{F} = \mathbf{0} \tag{12}$$

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 ρ is the density and 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 $\gamma = 1.4$.

5.1 Conservative Linearisation

In the context of residual distribution methods, the Euler equations are most frequently solved under the assumption that the "parameter vector" of Roe [34]:

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

varies linearly within each mesh cell. In the above H stands for the total enthalpy $H = \frac{E_{total}+p}{\rho}$. The parameter vector $\mathbf{z}$ and its linearity facilitate construction of a conservative linearisation. It is a very desirable feature when dealing with hyperbolic equations as it guarantees that the position and strength of non-linear discontinuities in the solution are approximated correctly.

By analogy with the scalar case, the cell residual, Φ^E, lies at the basis of all $\mathcal{RD}$ approximations of (12):

$$\Phi^E = \int_E \nabla \cdot \mathbf{F}(\mathbf{w}_h)\, d\Omega = \oint_{\partial E} \mathbf{F}(\mathbf{w}_h) \cdot \mathbf{n}\, d\Gamma. \tag{13}$$

$\mathbf{n}$ is the outward pointing unit normal vector. In order to derive a discrete system approximating (12), one has to find an efficient and accurate way of calculating (13). Evaluating it in terms of the parameter vector gives:

$$\Phi^E = \int_E \left(\frac{\partial \mathbf{g}}{\partial \mathbf{z}} \mathbf{z}_x + \frac{\partial \mathbf{h}}{\partial \mathbf{z}} \mathbf{z}_y \right) d\Omega. \tag{14}$$

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 (14) as:

$$\Phi^E = \left(\int_E \frac{\partial \mathbf{g}}{\partial \mathbf{z}} d\Omega \right) \mathbf{z}_x + \left(\int_E \frac{\partial \mathbf{h}}{\partial \mathbf{z}} d\Omega \right) \mathbf{z}_y. \tag{15}$$

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:

$$\Phi^E = |E| \left(\frac{\partial \mathbf{g}(\bar{\mathbf{z}})}{\partial \mathbf{z}} \mathbf{z}_x + \frac{\partial \mathbf{h}(\bar{\mathbf{z}})}{\partial \mathbf{z}} \mathbf{z}_y \right) \tag{16}$$

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

$$\bar{\mathbf{z}} = \frac{\mathbf{z}_1 + \mathbf{z}_2 + \mathbf{z}_3}{3}, \qquad \text{with } \mathbf{z}_i = \mathbf{z}(\mathbf{x}_i) \text{ and } \mathbf{x}_i \in E. \tag{17}$$

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 (16), 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:

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

It now follows that (16) is equivalent to:

$$\Phi^E = |E|\left(\frac{\partial \mathbf{g}(\bar{\mathbf{z}})}{\partial \mathbf{w}}\widehat{\mathbf{w}_x} + \frac{\partial \mathbf{h}(\bar{\mathbf{z}})}{\partial \mathbf{w}}\widehat{\mathbf{w}_y}\right), \tag{18}$$

which is the formula that we used to get our results.

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.

A detailed description of how to evaluate the time derivative $\partial_t\mathbf{w}$ using the hypothesis of linear $\mathbf{z}$ can be found in [4]. Here we decided to use a mixed approach and to simplify the calculations by evaluating the time derivative using the hypothesis of linear $\mathbf{w}$ rather than $\mathbf{z}$.

5.2 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). To this end we implemented the so called *matrix distribution* approach devised in [3, 40, 41]. We note that contrary to the previous section in which the parameter vector of Roe was presented, 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 the 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 (10)):

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

with $\bar{\mathbf{w}}$ being the cell average of $\mathbf{w}$ (cf. Equation (17)) and $\mathbf{A}$ and $\mathbf{B}$ defined as Jacobian matrices of the fluxes:

$$\mathbf{A} = \frac{\partial \mathbf{g}}{\partial \mathbf{w}}, \qquad \mathbf{B} = \frac{\partial \mathbf{h}}{\partial \mathbf{w}}. \tag{19}$$

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 (1) are substituted into (19) then the resulting quantity will be equal to the scalar flow sensor, k_i, introduced in Section 4.

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 [23]. 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 (10). 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. We will now present particular matrix distribution schemes.

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, 5].

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 $\mathbf{N}$ was proven in [1, 5].

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 computed using the following formula:

$$\boldsymbol{\Theta}_{k,k} = \frac{\left|\boldsymbol{\phi}_k^E\right|}{\sum_{i\in E}\left|\boldsymbol{\phi}_{i,k}^N\right|}. \tag{20}$$

In expression (20), index k refers to the k^{th} 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 [15]. 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 (6) for systems is derived by applying the procedure outlined in [32] to systems. Since at every vertex $i \in E$ there are four degrees of freedom, 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.

6 Numerical Results

We used two types of triangulations, i.e. structured (regular and isotropic) and unstructured, examples of which are illustrated in Figure 1. Further details regarding the meshes are discussed when particular results are presented. The linear systems resulting from the $\mathcal{RKRD}$ discretization were solved using PETSc [6] (see also the manual [7]) within which the ILU preconditioned GMRES solver was used. Since it gave good results, no other solver was implemented. To guarantee convergence, the relative tolerance in PETSc, i.e. the stopping criterion, was set to 10^{-8} in the case of scalar equations and to 10^{-5} for the Euler equations. Reducing it, i.e. setting to values lower than 10^{-5}, did not show any noticeable improvements (qualitative nor quantitative). However, in the case of the scalar equations the extra overhead related to setting a lower tolerance did not have a significant effect on the efficiency and we decided to run our experiments using the reduced value. The initial estimate was always set to zero.

In all computations the time-step Δt was calculated using the following formula:

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

The α^E coefficient is defined as:

$$\alpha^E = \frac{1}{2}\max_{j\in E}\left\|\frac{\partial \mathbf{f}(u_j)}{\partial u}\right\| h_E, \tag{21}$$

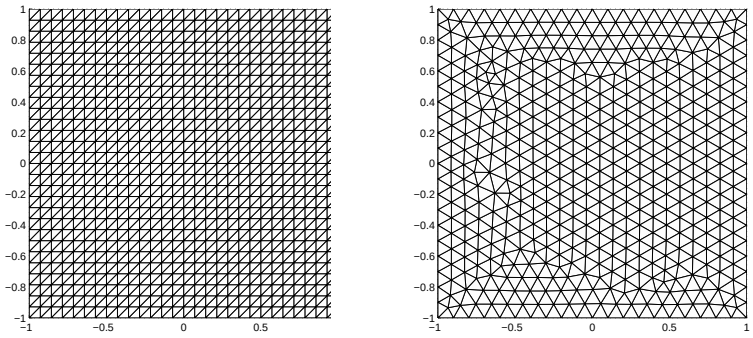


Figure 1: Representative structured (left) and unstructured (right) grids used for transient problems.

h_E stands for the reference length for element E in the scalar case, while for the Euler equations this coefficient was set to:

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

h_E stands for the reference length for element E. 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:

$$a_j = \sqrt{\frac{\gamma p_j}{\rho_j}}. \tag{22}$$

The Courant-Friedrichs-Lewy (CFL) number was set to 0.9 in the case of scalar equations and between 0.1 and 0.9 in the case of the Euler equations. Precise values are given when the corresponding results are presented.

6.1 Scalar Equations

Three distinct scalar test problems were implemented. Test Cases A and B are linear equations with *smooth* initial conditions which were used to measure convergence rates. Test Case C 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 A and C;
- $T = \frac{\pi}{2}$ for Test Case B.

Test Case A: The *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

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

in which $r = \sqrt{(x + 0.5 - t)^2 + y^2}$, $z = -\frac{r-0.4}{0.4}$ 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 B: 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

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

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 A, 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 A are not biased by the direction of the flow.

Test Case C: 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 A and B, the spatial domain is a square: $\Omega = [-1,1] \times [-1,1]$. The initial condition was set to be piece-wise constant:

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

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 [30, 31] and was used to test for positivity.

Linear equations (Test Case A and B) 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 discretisations of the domain, an unstructured mesh with 26054 elements (topology similar to that on the right of Figure 1) was used in the case of the non-linear Burgers' equation.

The grid convergence analysis confirmed that the N scheme is only first order accurate whereas the LDA, SU and BLEND schemes exhibit convergence of order two. These results are presented in Figure 2. The LDA and SU schemes gave best results, the SU scheme being noticeably more accurate than the LDA scheme. The BLEND scheme is slightly less accurate then both of them. This is most likely due to its nonlinear nature. These experiments were carried out on a set of regular triangular meshes (topology as on the left of Figure 1) 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 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 similar results.

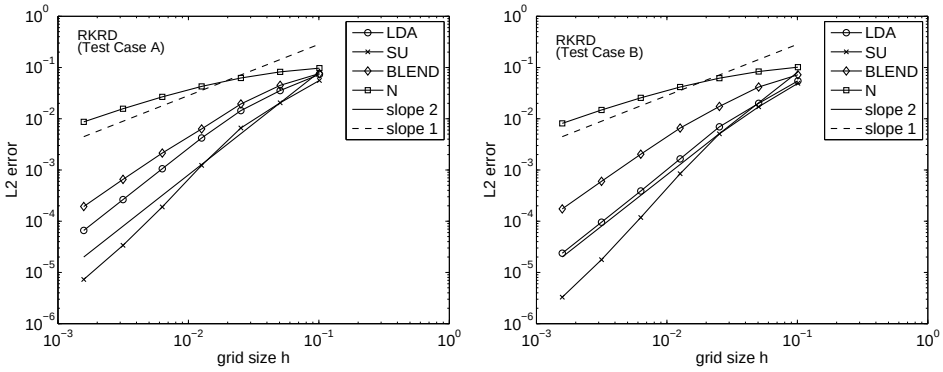
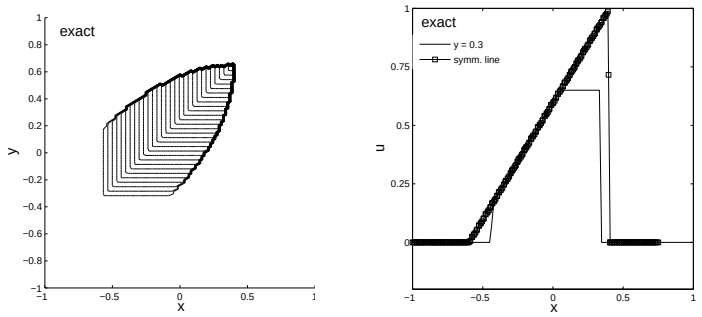


Figure 2: Grid convergence for the implicit $\mathcal{RKRD}$ framework for Test Cases A (left) and B (right).

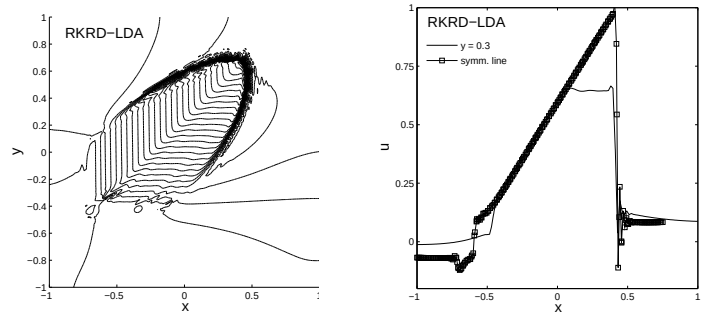
Concerning Test Case C, i.e. the Burgers' equation, in Figure 3 we plotted the contours and cross sections (along the symmetry line $y-x = 0.1$ and $y = 0.3$) of the exact solution. Next to these plots the reader will find the maximum and minimum values of the profile. Similar plots and quantities are given for the approximate solutions obtained with the aid of the $\mathcal{RKRD}$ framework, see Figures 4-7. 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). To show that these were 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 8. 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 9. Clearly tuning PETSc did not help, which implies it is the scheme itself, not the linear solver, that is unstable. The RKRD-BLEND schemes performed much better than the RKRD-LDA scheme. 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.



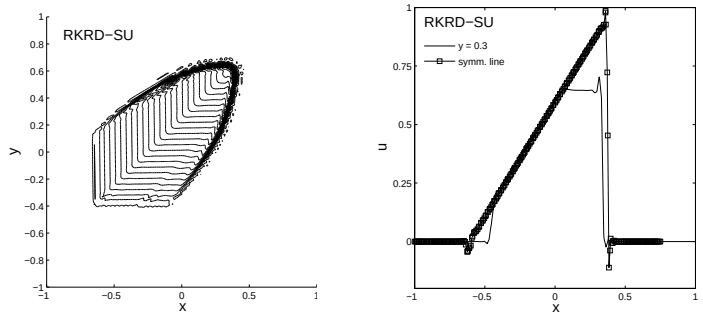
u_{min}	u_{max}
0.0	1.0

Figure 3: 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.



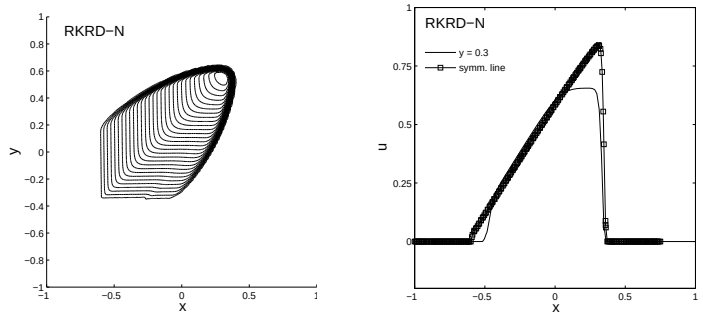
u_{min}	u_{max}
-0.181	1.041

Figure 4: 2d Burgers' equation: 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.



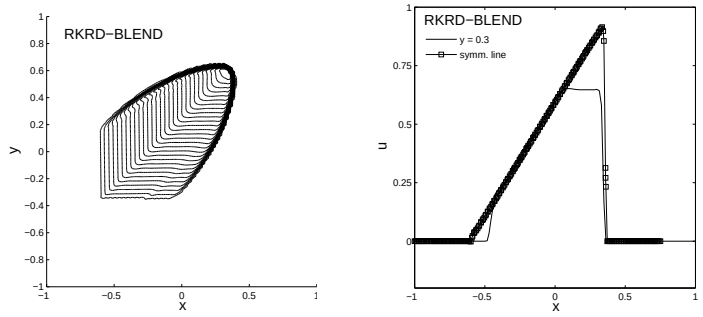
u_{min}	u_{max}
-0.167	1.058

Figure 5: 2d Burgers' equation: 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 6: 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.

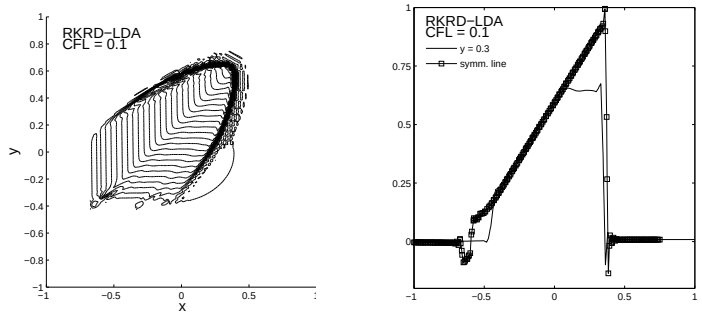


u_{min}	u_{max}
-0.017	0.915

Figure 7: 2d Burgers' equation: 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.

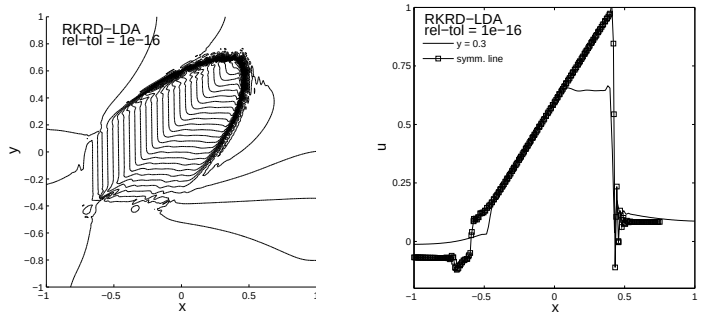
Finally, we want to comment on scaling and performance of the linear solver that was applied to solve linear systems resulting from the $\mathcal{RKRD}$ discretization. As mentioned earlier, only GMRES preconditioned with ILU was used. To guarantee convergence, the linear solver was set to iterate until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

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



u_{min}	u_{max}
-0.191	1.014

Figure 8: 2d Burgers' equation: 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 9: 2d Burgers' equation: 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.

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 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.

6.2 The Euler Equations

Also in the case of the Euler equations 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);

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

Table 1: Performance of the GMRES solver when applied to the linear systems resulting from the $\mathcal{RKRD}$ discretisations (Test Case B). 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 Runge-Kutta time-stepping (denoted by $||r_F||_2$). Results are given for the meshes used earlier in the grid convergence analysis (with 6272, 25088, 100352, 401408 and 1605632 elements, cf. top row of the table).

- Advection of a Vortex (the analytic solution is C^2 regular).

Shocks appearing in the Double Mach Reflection and Mach 3 test cases were too strong for the LDA scheme to cope with. This being the case, only the RKRD-BLEND scheme was 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.

Double Mach Reflection

This problem was originally introduced by Woodward et al. in [42]. 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 10. 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 11 for the geometry and initial values of the solution. The CFL number was set to 0.9.

For this test case it is customary to plot contours of the density field. These are presented in Figures 12-17. 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 [11, 33] and [42] 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. In the case of the

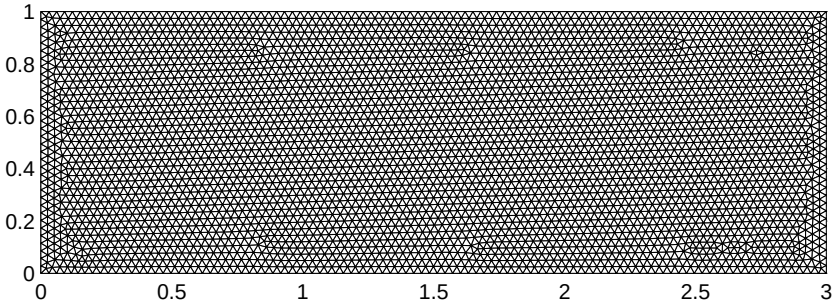


Figure 10: The coarsest grid for the Double Mach Reflection test case, 7865 cells.

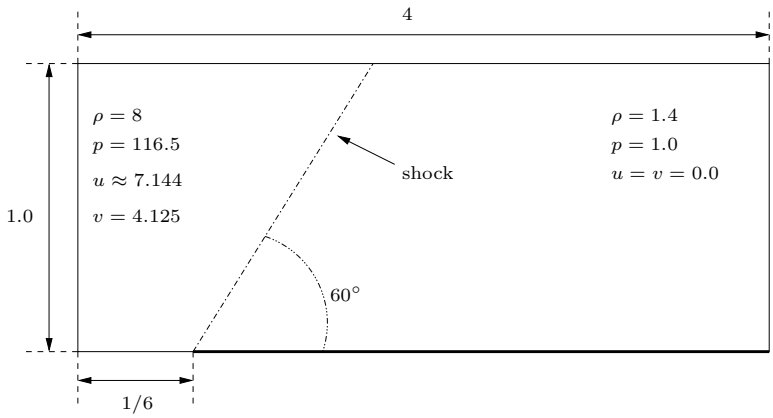


Figure 11: The geometry and initial condition for the Double Mach Reflection test case.

RKRD-BLEND scheme, values on the diagonal of the blending matrix $\mathbf{\Theta}$ (cf. Equation (20)) 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 17 is comparable with those obtained in [11] and [42] on meshes with similar resolution.

Mach 3 Flow Over a Step

This test was originally introduced in the paper by Emery [22] and more recently reviewed by Woodward et al. in [42]. 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 18 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 $(3, 0)$. Gas with this density, pressure, and

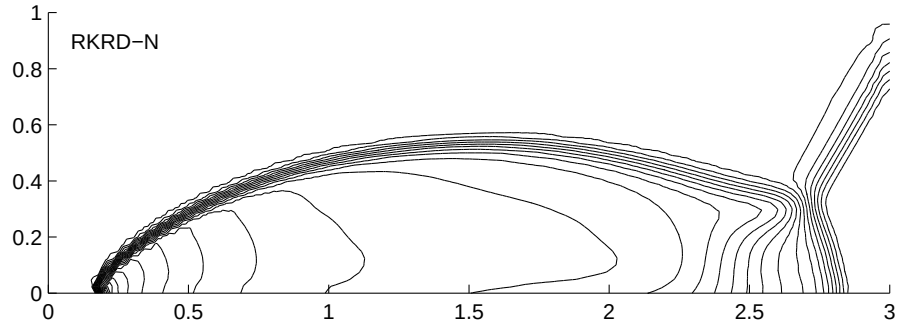


Figure 12: Double Mach reflection: density contours for the RKRD-N scheme. 7865 cells (the reference mesh size is $h = 1/30$)

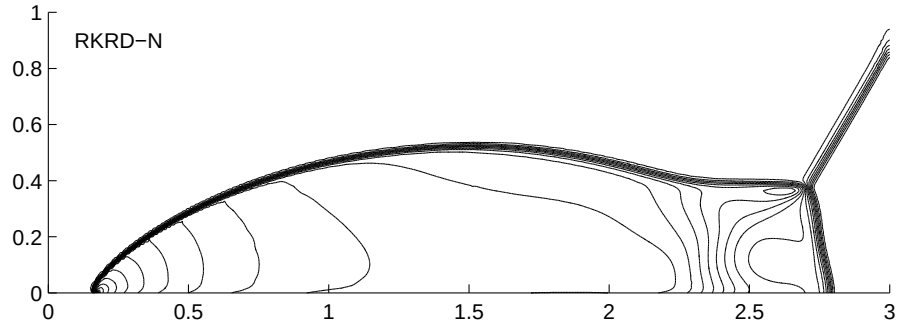


Figure 13: Double Mach reflection: density contours for the RKRD-N scheme. 55927 cells (the reference mesh size is $h = 1/80$)

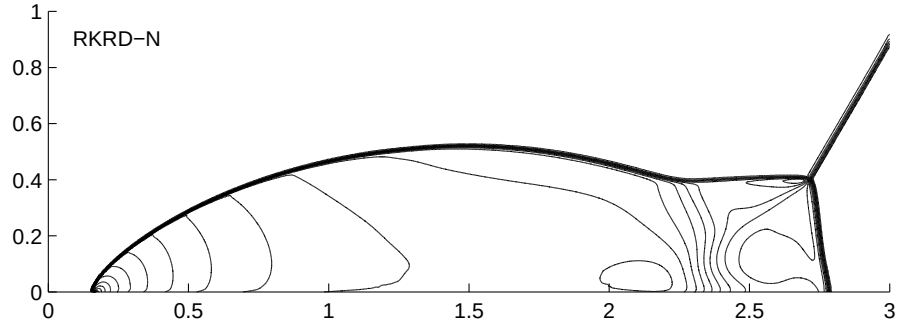


Figure 14: Double Mach reflection: density contours for the RKRD-N scheme. 278141 cells (the reference mesh size is $h = 1/190$)

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 [42], in order to minimize numerical errors generated at

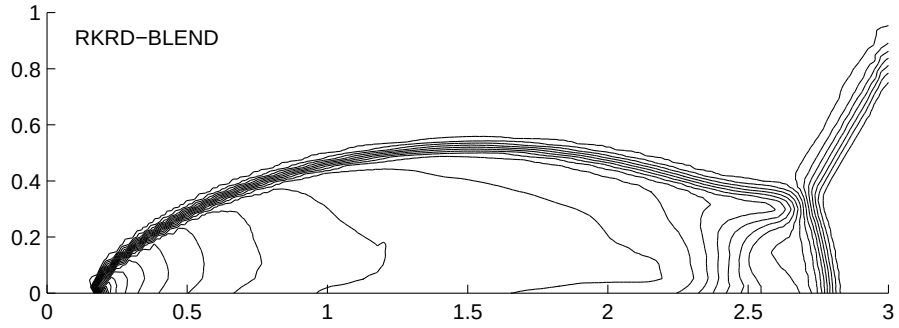


Figure 15: Double Mach reflection: density contours for the RKRD-BLEND scheme. 7865 cells (the reference mesh size is $h = 1/30$)

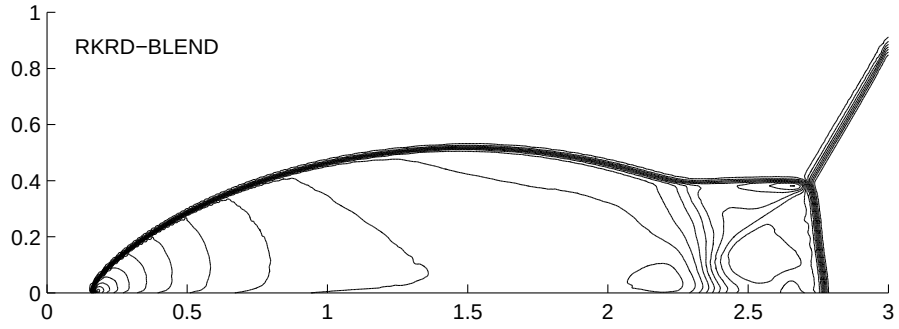


Figure 16: Double Mach reflection: density contours for the RKRD-BLEND scheme. 55927 cells (the reference mesh size is $h = 1/80$)

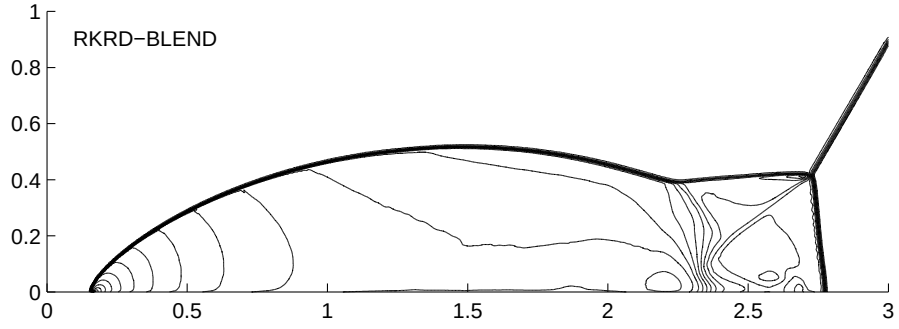


Figure 17: Double Mach reflection: density contours for the RKRD-BLEND scheme. 278141 cells (the reference mesh size is $h = 1/190$)

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 $\forall x \in E\ 0.6 \leq x \leq 0.6125$, 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 19. The CFL number was set to 0.5 and the values on the diagonal of the blending matrix $\mathbf{\Theta}$ (cf. Equation (20)) 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.

Density contours at times $t = 0.5, t = 1.5$ and $t = 4.0$ obtained with the aid of the RKRD-BLEND scheme are plotted on Figures 20-22. 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, [25, 33] and [12]).

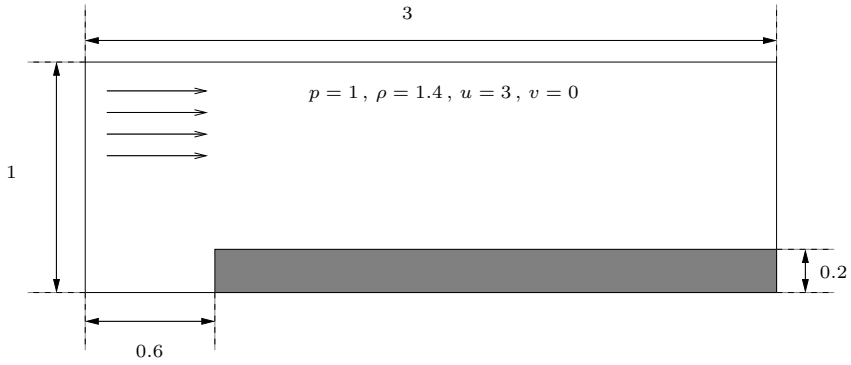


Figure 18: Geometry and the initial condition for the Mach 3 test case.

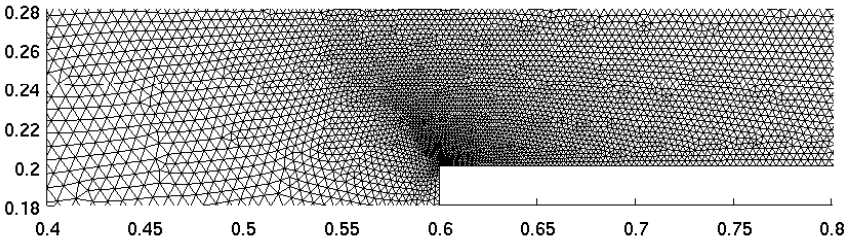


Figure 19: The zoom of the grid used for the Mach 3 Flow Over a Step test case near the singularity point.

Advection of a Vortex

The following problem was originally introduced in [21]. 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 an inflow boundary 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

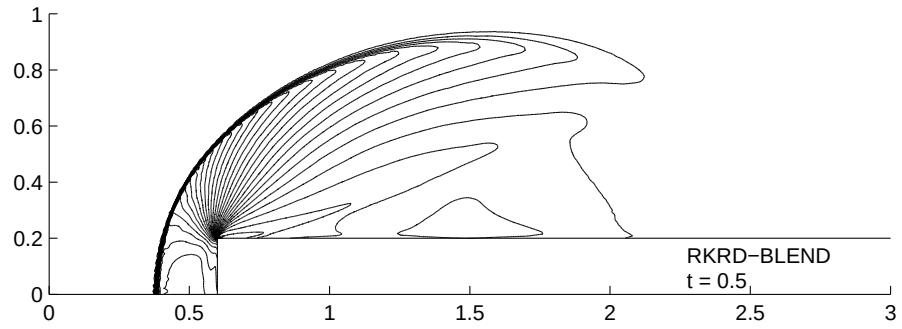


Figure 20: Mach 3 Flow Over a Step: RKRD-BLEND scheme, density contours at time $t = 0.5$, $CFL = 0.5$

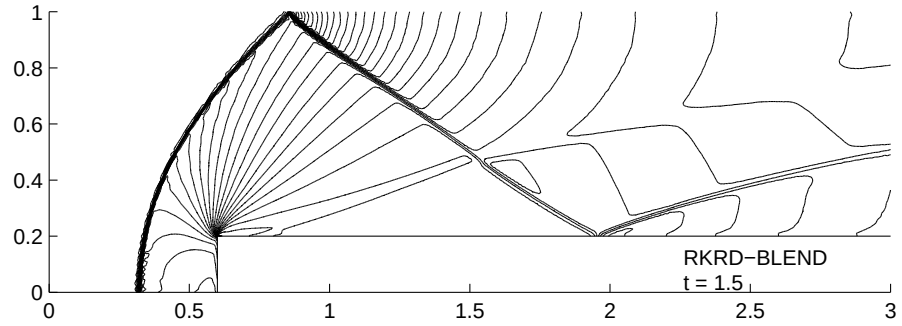


Figure 21: Mach 3 Flow Over a Step: RKRD-BLEND scheme, density contours at time $t = 1.5$, $CFL = 0.5$

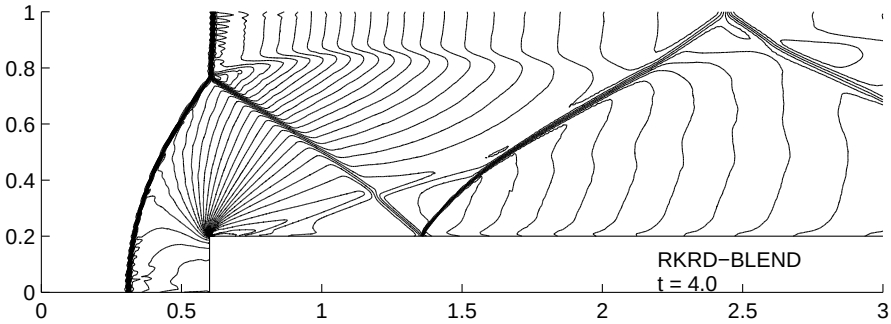


Figure 22: Mach 3 Flow Over a Step: RKRD-BLEND scheme, density contours at time $t = 4.0$, $CFL = 0.5$

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:

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

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. $p = p_m + p_p$:

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

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 27 with 161×81 nodes. The computations were performed with $CFL = 0.8$. In Table 2, 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 24-26. 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 solution obtained with the RKRD-BLEND scheme resembles the exact solution, Figure 23, the most. It should be noted, though, that in this section the RKRD-BLEND scheme was set in such a way that the preference was given to the LDA scheme (cf. Section 5.2). In [21] 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 2 show similar behaviour, but contour plots presented here (in particular those obtained with the RKRD-LDA and RKRD-BLEND schemes) are much more faithful to the exact solution than those presented in the literature [21].

Scheme	N	BLEND	LDA	exact
p_{min}	98.77133	93.5180	92.90018	93.213
p_{max}	100.1191	100.0004	100.0803	100

Table 2: The minimum and maximum value of the pressure obtained with the aid of the LDA, N and BLEND schemes using the $\mathcal{RKRD}$ framework.

The grid convergence analysis was performed to investigate the order of accuracy of the RKRD-LDA and RKRD-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:

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

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. This 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

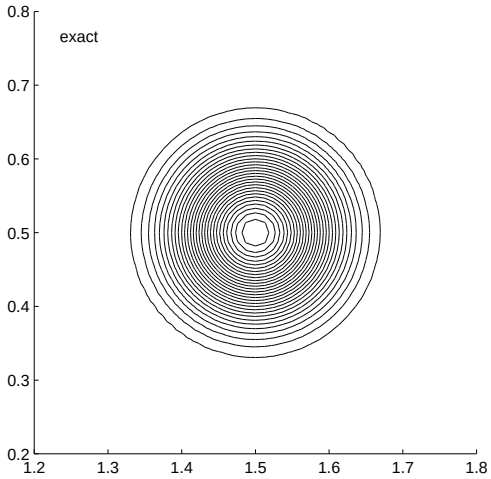


Figure 23: Travelling Vortex: pressure contours for the exact solution, 25600 cells

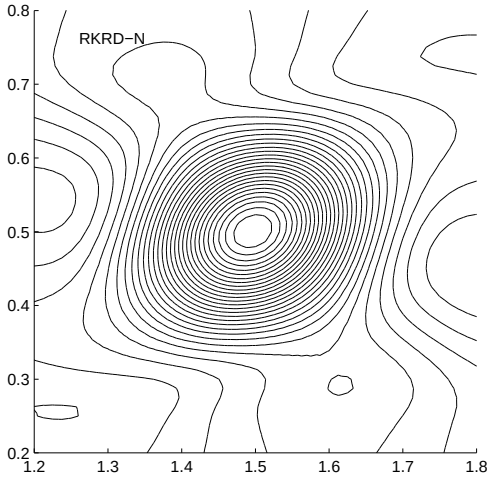


Figure 24: Travelling vortex: pressure contours for the RKRD-N scheme, 25600 cells

of structured and unstructured meshes (topology as in Figure 27), 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 in this case was reduced to 0.1. Recall that it was set to 0.8 to produce the contour plots, i.e. Figures 24-26. 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 from $(0.5, 0.5)$ to $(0.98, 0.5)$ instead of $(1.5, 0.5)$). The results on structured and unstructured meshes are illustrated in Figures 28 and 29, respectively. The second order of accuracy is reached quite rapidly, but only in the L^2 and L^1 norms. The

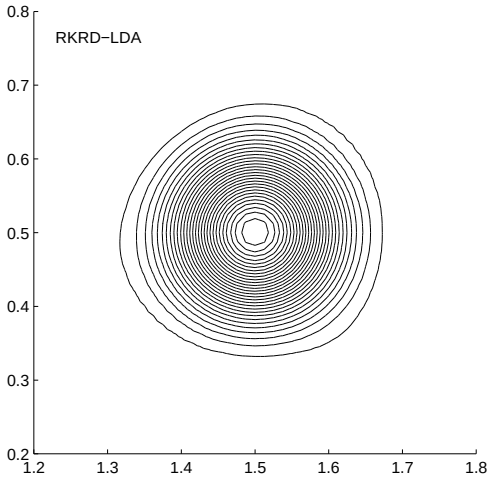


Figure 25: Travelling vortex: pressure contours for the RKRD-LDA scheme, 25600 cells

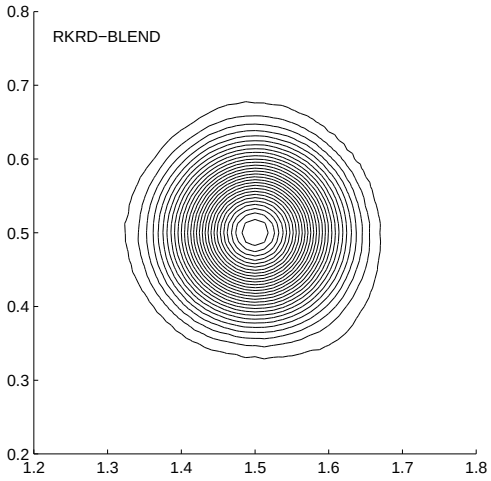


Figure 26: Travelling vortex: pressure contours for the RKRD-BLEND scheme, 25600 cells

RKRD-LDA scheme exhibited a small drop down in the order of accuracy when moving to the finest meshes.

7 Conclusions

In this paper we introduced a new class of numerical approximations to time-dependent hyperbolic PDEs, namely the framework of Runge-Kutta residual distribution methods. The proposed framework facilitates construction of second order accurate schemes and this was confirmed experimentally. The non-linear RKRD-BLEND scheme, although not completely oscillation-free, gave very encouraging results in terms of monotonicity. In particular it coped well

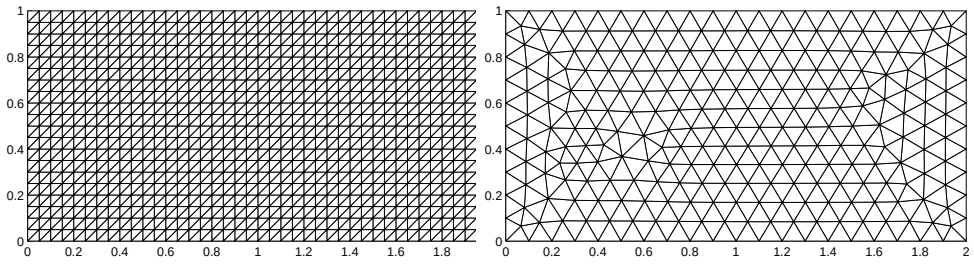


Figure 27: The coarsest structured (left) and unstructured (right) grid used in the grid convergence analysis for the Advection of a Vortex test case.

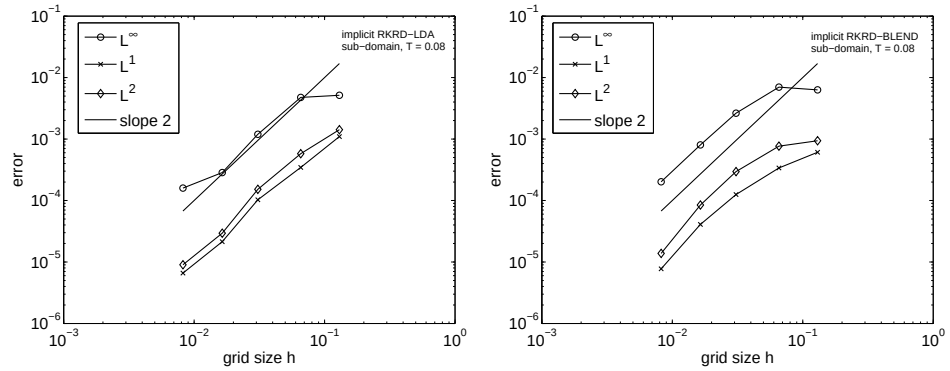


Figure 28: Grid convergence for the RKRD-LDA (left, $CFL = 0.1$) and -BLEND (right, $CFL = 0.1$) schemes for the Advection of a Vortex test case. Errors calculated within a sub-domain surrounding the vortex. Simulation run until $T = 0.08$. Unstructured meshes.

with all the severe test cases based on the system of non-linear Euler equations.

The future work will include incorporating discontinuous-in-space data representation following the methodology of [26, 27] and Abgrall [2]. This will aid the construction of a localised approximation for which there will be no need to solve a global linear system. Another possibility that will be considered and which is expected to lead to a construction of a second order accurate and *positive* scheme is a genuinely non-linear RKRD-BLEND scheme. Recall that in this paper the blending parameter was designed in such a way that the resulting discrete system of equations was linear. By substituting δu^k instead of $\overline{\delta u^k}$ into the definition one possibly will get a positive scheme. Finally, higher order time-stepping combined with higher order spatial representation could be used to design a higher than second order accurate scheme.

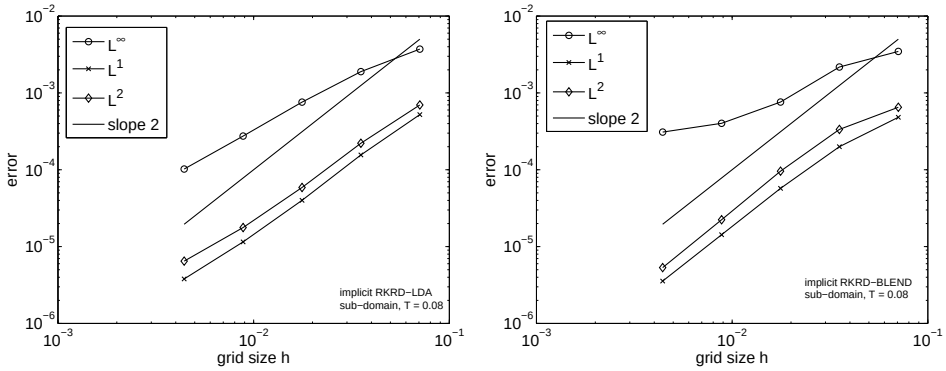


Figure 29: Grid convergence for the RKRD-LDA (left, $CFL = 0.1$) and -BLEND (right, $CFL = 0.1$) schemes for the travelling vortex test case. Errors calculated within a sub-domain surrounding the vortex. Simulation run until $T = 0.08$. Structured meshes.

References

[1] Abgrall, R.: Toward the ultimate conservative scheme: following the quest. J. Comput. Phys. **167**(2), 277–315 (2001)

[2] Abgrall, R.: 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)

[3] Abgrall, R., Deconinck, H., Sermeus, K.: Status of multidimensional upwind residual distribution schemes and applications in aeronautics. In: AIAA Paper 2000-2328, Fluids 2000/Denver (2000)

[4] Abgrall, R., Mezine, M.: Construction of second order accurate monotone and stable residual distribution schemes for unsteady flow problems. J. Comput. Phys. **188**(1), 16–55 (2003)

[5] Abgrall, R., Mezine, M.: Residual distribution schemes for steady problems. In: Computational Fluid Dynamics, VKI LS 2003-05. von Karman Institute for Fluid Dynamics (2003)

[6] Balay, S., Brown, J., Buschelman, K., Gropp, W.D., Kaushik, D., Knepley, M.G., Curfman McInnes, L., Smith, B.F., Zhang, H.: PETSc Web page (2012). Http://www.mcs.anl.gov/petsc

[7] Balay, S., Brown, J., Buschelman K. Eijkhout, V., Gropp, W.D., Kaushik, D., Knepley, M.G., McInnes, L.C., Smith, B.F., Zhang, H.: PETSc users manual. Tech. Rep. ANL-95/11 - Revision 3.3, Argonne National Laboratory (2012)

[8] Brooks, A.N., Hughes, T.J.R.: 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)

[9] Caraeni, D., Fuchs, L.: Compact third-order multidimensional upwind scheme for Navier-Stokes simulations. Theor. Comput. Fluid Dyn. **15**(6), 373–401 (2002)

[10] Carette, J.C., Deconinck, H., Paillère, H., Roe, P.L.: Multidimensional upwinding: its relation to finite elements. Internat. J. Numer. Methods Fluids **20**(8-9), 935–955 (1995)

[11] Cockburn, B., Shu, C.W.: The Runge-Kutta discontinuous Galerkin method for conservation laws. V. Multidimensional systems. J. Comput. Phys. **141**(2), 199–224 (1998)

[12] Csík, Á., Deconinck, H.: 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)

[13] Csík, Á., Deconinck, H., Ricchiuto, M.: Residual distribution for general time-dependent conservation laws. J. Comput. Phys. **209**(1), 249–289 (2005)

[14] Csík, Á., Deconinck, H., Ricchiuto, M., Poedts, S.: Space-time residual distribution schemes for hyperbolic conservation laws. In: 15th AIAA Computational Fluid Dynamics Conference, Anaheim, CA, USA (2001)

[15] Csík, Á., Ricchiuto, M., Deconinck, H.: A conservative formulation of the multidimensional upwind residual distribution schemes for general nonlinear conservation laws. J. Comput. Phys. **179**(1), 286–312 (2002)

[16] De Palma, P., Pascazio, G., Rossiello, G., Napolitano, M.: A second-order-accurate monotone implicit fluctuation splitting scheme for unsteady problems. J. Comput. Phys. **208**(1), 1–33 (2005)

[17] Deconinck, H.: Upwind methods and multidimensional splittings for the Euler equations. In: Computational Fluid Dynamics, VKI LS 1991-01. von Karman Institute for Fluid Dynamics (1991)

[18] Deconinck, H., Ferrante, A.: Solution of the unsteady Euler equations using residual distribution and flux corrected transport. Tech. Rep. 97-08, von Karman Institute for Fluid Dynamics (1997)

[19] Deconinck, H., Ricchiuto, M.: Residual distribution schemes: foundations and analysis. In: Encyclopedia of Computational Mechanics, vol. 3. John Wiley and Sons, Ltd. (2007)

[20] Deconinck, H., Roe, P.L., Struijs, R.: A multidimensional generalization of Roe's flux difference splitter for the Euler equations. Comput. & Fluids **22**(2-3), 215–222 (1993)

[21] Dobeš, J., Deconinck, H.: Second order blended multidimensional upwind residual distribution scheme for steady and unsteady computations. J. Comput. Appl. Math. **215**(2), 378–389 (2008)

[22] Emery, A.F.: An evaluation of several differencing methods for inviscid fluid flow problems. J. Comput. Phys. **2**, 306–331 (1968)

[23] Godlewski, E., Raviart, P.A.: Numerical approximation of hyperbolic systems of conservation laws, *Applied Mathematical Sciences*, vol. 118. Springer-Verlag, New York (1996)

[24] Guzik, S.M.J., Groth, C.P.T.: 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)

[25] Hubbard, M., Ricchiuto, M.: Discontinuous fluctuation distribution: A route to unconditional positivity and high order accuracy. In: ICFD 2010 International Conference on Fluid Dynamics, Reading (UK) (2010)

[26] Hubbard, M.E.: Discontinuous fluctuation distribution. J. Comput. Phys. **227**(24), 10,125–10,147 (2008)

[27] Hubbard, M.E.: A framework for discontinuous fluctuation distribution. Internat. J. Numer. Methods Fluids **56**(8), 1305–1311 (2008)

[28] Hughes, T.J.R., Mallet, M.: 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)

[29] Johnson, C.: Numerical solution of partial differential equations by the finite element method. Dover Publications Inc., Mineola, NY (2009). Reprint of the 1987 edition

[30] LeVeque, R.J.: Numerical methods for conservation laws, second edn. Lectures in Mathematics ETH Zürich. Birkhäuser Verlag, Basel (1992)

[31] LeVeque, R.J.: Finite volume methods for hyperbolic problems. Cambridge Texts in Applied Mathematics. Cambridge University Press, Cambridge (2002)

[32] März, J., Degrez, G.: Improving time accuracy of residual distribution schemes. Tech. Rep. 96-17, von Karman Institute for Fluid Dynamics (1996)

[33] Ricchiuto, M., Abgrall: Explicit Runge–Kutta residual distribution schemes for time dependent problems: second order case. J. Comput. Phys. **229**(16), 5653–5691 (2010)

[34] Roe, P.L.: Approximate Riemann solvers, parameter vectors, and difference schemes. J. Comput. Phys. **43**(2), 357–372 (1981)

[35] Roe, P.L.: Fluctuations and signals - a framework for numerical evolution problems. In: Numerical Methods for Fluid Dynamics, pp. 219–257. Academic Press (1982)

[36] Roe, P.L.: Linear advection schemes on triangular meshes. Tech. Rep. Technical Report CoA 8720, Cranfield Institute of Technology (1987)

[37] Shu, C.W., Osher, S.: Efficient implementation of essentially nonoscillatory shock-capturing schemes. J. Comput. Phys. **77**(2), 439–471 (1988)

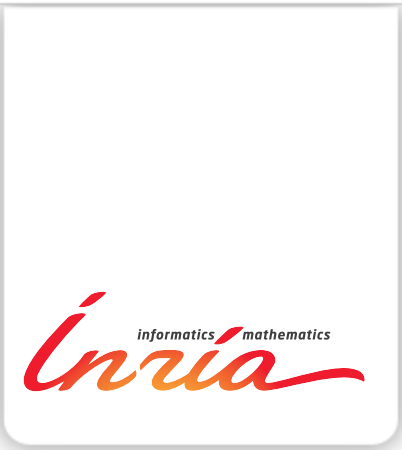
[38] Struijs, R., Deconinck, H., Roe, P.L.: Fluctuation splitting schemes for the 2D Euler equations. In: Computational Fluid Dynamics, VKI LS 1991-01. von Karman Institute for Fluid Dynamics (1991)

[39] Warzyński, A., Hubbard, M.E., Ricchiuto, M.: Discontinuous residual distribution schemes for time-dependent problems. In: Li, J and Yang, HT and Machorro, E (ed.) Recent Advances In Scientific Computing And Applications, *Contemporary Mathematics*, vol. 586, pp. 375–382 (2013)

[40] van der Weide, E., Deconinck, H.: Positive matrix distribution schemes for hyperbolic systems, with application to the Euler equations. In: J. Desideri, C. Hirsch, P. LeTallec, M. Pandolfi, J. Periaux (eds.) Computational Fluid Dynamics '96, pp. 747–753 (1996)

[41] van der Weide, E., Deconinck, H., Issman, E., Degrez, G.: A parallel, implicit, multi-dimensional upwind, residual distribution method for the Navier-Stokes equations on unstructured grids. Computational Mechanics **23**(2), 199–208 (1999)

[42] Woodward, P., Colella, P.: The numerical simulation of two-dimensional fluid flow with strong shocks. J. Comput. Phys. **54**(1), 115–173 (1984)



RESEARCH CENTRE
BORDEAUX – SUD-OUEST

351, Cours de la Libération
Bâtiment A 29
33405 Talence Cedex

Publisher
Inria
Domaine de Voluceau - Rocquencourt
BP 105 - 78153 Le Chesnay Cedex
inria.fr

ISSN 0249-6399