

Relaxation Deferred Correction Methods and their Applications to Residual Distribution Schemes

R. Abgrall^{†‡}, E. Le Méle[†], P. Öffner[★], and D. Torlo[‡]

[†]Institute of Mathematics, University of Zurich, Switzerland

[★] Johannes Gutenberg-University Mainz, Germany

[‡] Team CARDAMOM, INRIA Bordeaux - Sud-Ouest, France

June 16, 2021

Abstract

In [2] is proposed a simplified DeC method, that, when combined with the residual distribution (RD) framework, allows to construct a high order, explicit FE scheme with continuous approximation avoiding the inversion of the mass matrix for hyperbolic problems. In this paper, we close some open gaps in the context of deferred correction (DeC) and their application within the RD framework. First, we demonstrate the connection between the DeC schemes and the RK methods. With this knowledge, DeC can be rewritten as a convex combination of explicit Euler steps, showing the connection to the strong stability preserving (SSP) framework. Then, we can apply the relaxation approach introduced in [22] and construct entropy conservative/dissipative DeC (RDeC) methods, using the entropy correction function proposed in [3].

Contents

1	Introduction	2
2	Numerical Methods and Theoretical Considerations	3
2.1	Strong Stability Preserving Methods	3
2.2	Relaxation Runge–Kutta Methods	4
2.3	Deferred Correction Methods	5
2.4	Semidiscrete Entropy Conservative Residual Distribution Methods	8
2.5	Residual Distribution for Hyperbolic Problems	10
3	Deferred Correction Methods – Connection to RK, SSP and Relaxation Technique	11
3.1	The Connection between DeC and RK with possible Application to SSP	11
3.2	Relaxation DeC	12
4	A Fully Entropy Conservative/Dissipative DeC-RD Approach	15
5	Numerical Simulations	17
5.1	Numerical test in the ODE case	17
5.1.1	Nonlinear Oscillator	17
5.1.2	Damped Nonlinear Oscillator	19
5.1.3	Nonlinear Pendulum	19
5.2	Numerical test in the PDE case	21

*Corresponding author: mail@philippoeffner.de

5.2.1	RDeC-FV for Burgers' Equation	21
5.2.2	Entropy Conservative/ Dissipative RDeC-RD Schemes - Linear Case	23
5.2.3	Entropy Conservative/ Dissipative RDeC-RD Schemes - Nonlinear Case	25
6	Conclusion	27
A	Another possible relaxation formulation	28

1 Introduction

Many problems in nature are described either by ordinary differential equations (ODEs) or partial differential equations (PDEs) and the numerical methods that approximate their solutions should preserve the physical quantities of the underlying problem. To keep the positivity for example Patankar approaches [32, 21, 28] or adaptive/limiting strategies [25, 29] can be found in the literature, while, recently, Ketcheson proposed relaxation Runge-Kutta (RRK) methods to guarantee conservation or stability with respect to any inner-product norm. In a series of papers, he and collaborators have further extended the relaxation approach to multistep schemes and applied it to different kind of problems, cf. [22, 35, 34, 38]. Special attention has been given on entropy conservation/dissipation in the context of hyperbolic equations where the relaxation approach is combined with a semidiscrete entropy conservative/dissipative scheme via a simple method of line.

In this work, we focus also on the relaxation approach, but we apply it, differently from before, on the deferred correction (DeC) method. DeC is based on the Picard-Lindelöf theorem and gives a simple algorithm to construct arbitrarily high order schemes. In the context of time-dependent PDEs, DeC has been recently combined with the residual distribution (RD) method [2, 4] to obtain a matrix-free finite element based high order method for hyperbolic PDE. In [3] a correction on the definition of the residuals allows to preserve the entropy in the spatial discretization.

Here, we combine the two ideas, entropy residual correction and relaxation of the time integration, to obtain a scheme which is fully discrete entropy conservative/dissipative. Moreover, we are able to preserve the matrix-free character of the DeC RD scheme avoiding the inversion of the mass matrix.

Therefore, the paper is organized as follows: in section 2, we introduce the numerical methods under consideration. First, classical strong stability preserving Runge-Kutta (SSPRK) schemes and the relaxation approach proposed by Ketcheson [22] are explained. After that we introduce the DeC method in his simplified version from [2] which will be the main topic in this work. Since we combine DeC later with the residual distribution (RD) methods, we shortly explain also the main ideas of RD, the construction of entropy conservative/dissipative RD schemes for steady state problems, and finally the application of DeC together with RD schemes to solve hyperbolic conservation laws. In section 3, we focus on DeC and explain the connection between general RK schemes, the SSP framework and the application of the relaxation technique. Since DeC can be interpreted as an RK scheme for ODE problems (or simple PDE problems with the method of lines (MOL) approach), the relaxation results done by Ketcheson et al. can be transferred directly to the DeC methods. Nevertheless, we need to prove analogous results also for the fully DeC framework as this should lead to a better understanding of the relaxation approach, but it is also necessary due to the fact that we want to apply the relaxation DeC (RDeC) with the RD where a MOL approach is avoided to keep the high-order accuracy property of the method. In section 4, we demonstrate how to combine RDeC with our semidiscrete entropy conservative/dissipative RD method resulting in fully discrete entropy conservative/dissipative RD schemes where we still avoid the inversion of the mass matrix. In section 5, we validate all our theoretical results through various numerical simulations both on ODEs and PDEs. Finally, in section 6 we summarize the work done and we give an outlook on possible future applications. In appendix A we provide another interpretation of RDeC-RD for completeness.

2 Numerical Methods and Theoretical Considerations

We are interested in the numerical solution of hyperbolic conservation laws

$$\begin{aligned} \frac{\partial U}{\partial t} + \operatorname{div} F(U) &= 0, \quad x \in \Omega \subset \mathbb{R}^d, \quad t \in [0, T] \\ U(x, 0) &= U_0(x), \quad x \in \Omega \end{aligned} \quad (1)$$

where Ω is the spatial domain. Like in every finite element based scheme, we divide Ω in subdomains (e.g. triangles) denoted by Ω_h . Further, we use K to denote the generic element and h is the mesh size. σ will be the index of any degree of freedom (DOF) in Ω_h (some nodal basis, etc.). The time interval $[0, T]$ is divided into N segments T_n and the time steps are given by $\Delta t^n = t^{n+1} - t^n$.

The following subsections will explain the temporal and spatial discretization methods which will be considered through the text in order to create a fully discrete high-order scheme for (1). In this work, we focus mainly on the time integration side and we close the gaps on DeC in combination with RD in several ways. Important is here the connection with Runge–Kutta framework and, therefore, we shortly repeat the strong stability preserving (SSP) and relaxation approach for the ODE case.

2.1 Strong Stability Preserving Methods

There are various approaches to solve numerically an ODE. A first ansatz is given by finite differences, where the derivative in time is replaced by differences of states in different timesteps. Backward (implicit) and forward (explicit) Euler are examples of this kind of strategy. Another approach would be to reformulate the ODE by integrating it in time. With different quadrature formulas and approximation techniques, one can obtain various RK methods (explicit and implicit ones). These are a standard tools for solving ODEs. We consider the following time-dependent initial value problem

$$y'(t) = f(t, y(t)) \quad y(t_0) = y_0 \quad (2)$$

where $y : \mathbb{R} \rightarrow \mathbb{R}^m$ and $f : \mathbb{R} \times \mathbb{R}^m \rightarrow \mathbb{R}^m$.

A Runge-Kutta method applied to (2) takes the form

$$\begin{aligned} u_i &= y^n + \Delta t \sum_{j=1}^s a_{ij} f(t_n + c_j \Delta t, u_j) \\ y(t_n + \Delta t) &\approx y^{n+1} = y^n + \Delta t \sum_{j=1}^s b_j f(t_n + c_j \Delta t, u_j) \end{aligned} \quad (3)$$

We assume that $c_j = \sum_i a_{ij}$ holds and we use for brevity $f_i = f(t_n + c_i \Delta t, u_i)$ for the i th stage derivate. This can also be written into a Butcher tableau

$$\begin{array}{c|c} c & A \\ \hline & b \end{array} \quad (4)$$

with $A \in \mathbb{R}^{s \times s}$ and $b, c \in \mathbb{R}^s$.

Instead of using (3), the Runge-Kutta scheme can also be expressed using a Shu-Osher formulation as described in [13, 24]. An explicit RK scheme has the Shu-Osher form [39]

$$\begin{aligned} u_0 &= y^n \\ u_i &= \sum_{k=0}^{i-1} (\alpha_{i,k} u_k + \Delta t \beta_{i,k} f(u_k)), \quad \alpha_{i,k} \geq 0, \quad i = 1, \dots, s, \\ y^{n+1} &= u_s, \end{aligned} \quad (5)$$

where we assume that f does only depend on t through y . Then, consistency requires $\sum_{k=0}^{i-1} \alpha_{i,k} = 1$ and it is also imposed that $\beta_{i,k} = 0$ whenever $\alpha_{i,k} = 0$.

Remark 2.1. The representation (5) is quite important for the proof of the **strong stability preserving** (SSP) property of RK methods. Here, SSPRK methods are a generalization of TVD-RK methods, where instead of the total variation of the solution y , every convex functional of the solution y is ensured to decrease provided that this holds for a step of the simple explicit Euler method with step size Δt_{FE} . If $\beta_{i,k} \geq 0$, all the intermedia stages $y^{(i)}$ are simply convex combinations of forward Euler steps, each with Δt replaced by $\frac{\beta_{i,k}}{\alpha_{i,k}} \Delta t$. There, any bound on a norm, semi-norm, or convex functional of the the solution which is satisfied by the explicit Euler method will be preserved by the RK method under the timestep restriction $\max_{i,k} \frac{\beta_{i,k}}{\alpha_{i,k}} \Delta t \leq \Delta t_{FE}$. For more details about (TVD/SSP) RK methods see the extensive literature [16, 17, 18, 23, 26, 36].

Example 2.2 (SSPRK(3,3)). The third order using three stages (SSPRK(3,3)) method given in [17] by Gottlieb and Shu is defined as

$$\begin{aligned} u_1 &= u_0 + \Delta t f(u_0), \\ u_2 &= \frac{3}{4}u_0 + \frac{1}{4}u_1 + \frac{1}{4}\Delta t f(u_1), \\ y^{n+1} = u_3 &= \frac{1}{3}u_0 + \frac{2}{3}u_2 + \frac{2}{3}\Delta t f(u_2). \end{aligned} \tag{6}$$

2.2 Relaxation Runge–Kutta Methods

To explain the relaxation approach, we first follow the spirit of Ketcheson [22] and explain the basic framework. For simplicity reasons, we concentrate again only on the initial value problem (2). We focus on problems which are *dissipative* (*conservative*) with respect to some inner-product norm, i.e.

$$\frac{d}{dt} \|y(t)\|^2 = 2 \langle y, f(t, y) \rangle \stackrel{(\text{=})}{\leq} 0 \tag{7}$$

where the equality sign is used for conservative problems. Here and throughout, $\langle \cdot, \cdot \rangle$ denotes an inner product and $\|\cdot\|$ the corresponding norm. For dissipative (conservative) problems, it is desirable that the numerical solution verifies (7) discretely, i.e.,

$$\|y^{n+1}\| \stackrel{(\text{=})}{\leq} \|y^n\|, \tag{8}$$

where n represents the n -th timestep. The dissipative (conservative) problems naturally arise in the context of the semi-discretization of hyperbolic problems if the space discretization is done in a correct way as we are doing it following [3]. A method is called *monotonicity preserving* if it guarantees (8) for all problems satisfying (7).

Remark 2.3. The term $\|y\|^2$ is called *energy* in the following and in [22] the author introduced the relaxation approach to control the increase of the energy in classical RK methods. The energy is nothing else than a special entropy function in the hyperbolic setting. In [38], the relaxation approach is extend to general convex quantities, i.e. general entropies. For simplicity reasons, we first focus on the energy but we will include the extension to general entropies later.

As it is shown in [15, 22, 31] the change of the energy between two steps is given by

$$\begin{aligned} \|y^{n+1}\|^2 - \|y\|^2 &= 2\Delta t \sum_{j=1}^s b_j \langle u_j, f_j \rangle + 2\Delta t \sum_{j=1}^s b_j \langle y^n - u_j, f_j \rangle + \Delta t^2 \sum_{i,j=1}^s b_i b_j \langle f_i, f_j \rangle \\ &\stackrel{(\text{3})}{=} 2\Delta t \sum_{j=1}^s b_j \langle u_j, f_j \rangle - 2\Delta t^2 \sum_{j,i=1}^s a_{ij} b_j \langle f_j, f_i \rangle + \Delta t^2 \sum_{i,j=1}^s b_i b_j \langle f_i, f_j \rangle \end{aligned}$$

and we have to control the increase of the energy. For conservative problems, the first sum is zero and for dissipative ones it is non positive if $b_j \geq 0$. However, the remaining terms can destroy these conditions. The

main idea of the relaxation approach is to change the update formula of RK methods by changing the local time step in such way that the remaining terms are canceling out. To obtain this, a relaxation coefficient γ_n is introduced so that the RK update reads like

$$y_{\gamma_n}^{n+1} = y^n + \gamma_n \Delta t \sum_{j=1}^s b_j f(t_n + c_j \Delta t, u_j), \quad (9)$$

and the energy difference becomes

$$\|y_{\gamma_n}^{n+1}\|^2 - \|y\|^2 = 2\gamma_n \Delta t \sum_{j=1}^s b_j \langle u_j, f_j \rangle - 2\gamma_n \Delta t^2 \sum_{j,i=1}^s a_{ij} b_j \langle f_j, f_i \rangle + \Delta t^2 \gamma_n^2 \sum_{i,j=1}^s b_i b_j \langle f_i, f_j \rangle.$$

The last two terms can be deleted by a proper choice of γ_n . We determine the roots of this equation in respect to γ_n and get

$$\gamma_n = \frac{2 \sum_{j,i=1}^s a_{ij} b_j \langle f_j, f_i \rangle}{\Delta t \sum_{i,j=1}^s b_i b_j \langle f_i, f_j \rangle}, \quad (10)$$

where the second root is $\gamma_n = 0$ and is not further considered. If the denominator of (10) vanishes, we have $y^{n+1} = y^n$ and we achieve our aims (i.e. conservation) by taking $\gamma_n = 1$. Thus, we define

$$\gamma_n := \begin{cases} \frac{2 \sum_{j,i=1}^s a_{ij} b_j \langle f_j, f_i \rangle}{\sum_{i,j=1}^s b_i b_j \langle f_i, f_j \rangle}, & \text{if } \|\sum_{i=1}^s b_i f_i\|^2 \neq 0, \\ 1, & \text{else.} \end{cases} \quad (11)$$

Since we update the solution time with $\gamma_n \Delta t$, it is important that γ_n is bigger than zero. In the Runge-Kutta setting Ketcheson formulate the following lemma [22, Lemma 1]:

Lemma 2.4. *Let $\sum b_i a_{i,j} > 0$, let f be sufficiently smooth, and let γ_n be defined by (11). Then $\gamma_n > 0$ for sufficiently small $\Delta t > 0$.*

This is naturally fulfilled for every RK method of order two or higher since from the order conditions of the RK, it is known that $\sum b_i a_{i,j} = \frac{1}{2}$.

Remark 2.5. *Further results about relaxation RK methods can be found in [22, 38, 35] including also extension to multistep methods and to general entropy relaxation (instead of energy) and so on. However, for our purpose this introduction is enough. In the above mentioned literature, one can find also a discussion about consistency and accuracy related to the fact that $\sum_j \gamma_n b_j \neq 1$ is included. However, to shorten this part, we stress out (proofs can be found in above literature) that $\gamma_n = 1 + \mathcal{O}(\Delta t^{p-1})$ hold where p denotes the order of the RK method.*

Remark 2.6. *The relaxation methods are very useful in the conservative tests, as they allow to preserve the exact entropy/energy level. In the dissipative case, they are reliable providing physically coherent and accurate simulations, though not reaching the exact entropy/energy level. Hence, we will focus more on entropy/energy conservative tests than dissipative ones, giving, nevertheless, a general description of the methods.*

2.3 Deferred Correction Methods

The idea of DeC schemes as introduced in [12] is based on the Picard-Lindelöf Theorem in the continuous setting and the classical proof makes use of Picard iterations to minimize the error and prove convergence. The foundation of DeC relies on mimicking these Picard iterations at the discrete level and also decrease the approximation error in several iterative steps. To describe DeC, we follow the approach presented in [2]. For the description, two operators are introduced: \mathcal{L}^1 and \mathcal{L}^2 . Here, the \mathcal{L}^1 operator represents a low-order easy-to-solve numerical scheme, e.g. the explicit Euler method, and \mathcal{L}^2 is a high-order operator that can

present difficulties in its practical solution, e.g. an implicit RK scheme. The DeC method can be written as a combination of these two operators.

Given a time interval $[t^n, t^{n+1}]$, we subdivide it into M subintervals $\{[t^{n,m-1}, t^{n,m}]\}_{m=1}^M$, where $t^{n,0} = t^n$ and $t^{n,M} = t^{n+1}$. There, we mimic for every subinterval $[t^0, t^M]$ the Picard–Lindelöf Theorem for both operators \mathcal{L}^1 and \mathcal{L}^2 . We drop the dependency on the timestep n for subtimesteps $t^{n,m}$ and substates $y^{n,m}$ as denoted in Figure 1. Then, the \mathcal{L}^2 operator is given by

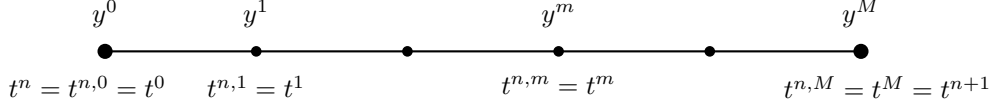


Figure 1: Time interval divided into subintervals

$$\mathcal{L}^2(y^0, \dots, y^M) := \begin{cases} y^M - y^0 - \int_{t^0}^{t^M} \mathcal{I}_M(f(y^0), \dots, f(y^M)) \\ \vdots \\ y^1 - y^0 - \int_{t^0}^{t^1} \mathcal{I}_M(f(y^0), \dots, f(y^M)) \end{cases} . \quad (12)$$

Here, the term \mathcal{I}_M denotes an interpolation polynomial of order M evaluated at the points $\{t^r\}_{r=0}^M$. In particular, we use Lagrange polynomials $\{\varphi_r\}_{r=0}^M$, which fulfills $\varphi_r(t^m) = \delta_{r,m}$ and satisfy the property $\sum_{r=0}^M \varphi_r(s) = 1$ for any $s \in [t^0, t^M]$. Using these properties, we can actually compute the integral of the interpolants thanks to a high order quadrature rule in the same points $\{t^m\}_{m=0}^M$, with weights

$$\theta_r^m := \frac{1}{\Delta t} \int_{t^0}^{t^m} \varphi_r(s) ds.$$

resulting in

$$\mathcal{L}^2(y^0, \dots, y^M) = \begin{cases} y^M - y^0 - \Delta t \sum_{r=0}^M \theta_r^M f(y^r) \\ \vdots \\ y^1 - y^0 - \Delta t \sum_{r=0}^M \theta_r^1 f(y^r) \end{cases} . \quad (13)$$

The \mathcal{L}^2 operator represents an $(M+1)$ th order numerical scheme if set equal to zero, i. e., $\mathcal{L}^2(y^0, \dots, y^M) = 0$. Unfortunately, the resulting scheme is implicit and, further, the terms f may be nonlinear. As a consequence, the only \mathcal{L}^2 formulation is not explicit and more efforts have to be made to solve it. For this purpose, we introduce a simplification of the \mathcal{L}^2 operator. Instead of using a quadrature formula at the points $\{t^m\}_{m=0}^M$ we evaluate the integral in equation (12) applying the left Riemann sum approximation. The resulting operator \mathcal{L}^1 is given by the forward Euler discretization for each state y^m in the time interval, i. e.,

$$\mathcal{L}^1(y^0, \dots, y^M) := \begin{cases} y^M - y^0 - \beta^M \Delta t f(y^0) \\ \vdots \\ y^1 - y^0 - \beta^1 \Delta t f(y^0) \end{cases} \quad (14)$$

with coefficients $\beta^m := \frac{t^m - t^0}{t^M - t^0}$.

To simplify the notation and to describe DeC, we introduce the matrix of states for the variable y at all subtimesteps.

$$\mathbf{y} := (y^0, \dots, y^M) \in \mathbb{R}^{M \times I}, \text{ such that} \quad (15)$$

$$\mathcal{L}^1(\mathbf{y}) := \mathcal{L}^1(y^0, \dots, y^M) \text{ and } \mathcal{L}^2(\mathbf{y}) := \mathcal{L}^2(y^0, \dots, y^M). \quad (16)$$

So far, the DeC algorithm uses a combination of the \mathcal{L}^1 and \mathcal{L}^2 operators to provide an iterative procedure. The aim is to recursively approximate \mathbf{y}^* , the numerical solution of the $\mathcal{L}^2 = 0$ scheme, similarly to the Picard iterations in the continuous setting. The successive states of the iteration process will be denoted by the superscript (k) , where k is the iteration index, e.g. $\mathbf{y}^{(k)} \in \mathbb{R}^{M \times I}$. The total number of iterations (also called correction steps in the following) is denoted by K . To describe the procedure, we have to refer to both the m -th subimestep and the k -th iteration of the DeC algorithm. We will indicate the variable by $y^{m,(k)} \in \mathbb{R}^I$. Finally, the DeC method can be written as

DeC Algorithm

$$\begin{aligned} y^{0,(k)} &:= y(t^n), \quad k = 0, \dots, K, \\ y^{m,(0)} &:= y(t^n), \quad m = 1, \dots, M \\ \mathcal{L}^1(\mathbf{y}^{(k)}) &= \mathcal{L}^1(\mathbf{y}^{(k-1)}) - \mathcal{L}^2(\mathbf{y}^{(k-1)}) \text{ with } k = 1, \dots, K, \end{aligned} \tag{17}$$

where K is the number of iterations that we want to compute. Using the procedure (17), we need, in particular, as many iterations as the desired order of accuracy p , i.e., $K = p = M + 1$.

Example 2.7. *Second and third order DeC*

- Actually, we have only the two endpoints¹ $t^0 = 0$ and $t^1 = 1$ for a second order scheme. We calculate the first approximation at $t^1 = 1$ using the explicit Euler method. Afterwards, the iteration step (17) is used which results finally in the following update:

$$\begin{aligned} y^1 &= y^0 + \Delta t f(t^0, y^0), \\ y^{1,(1)} &= y^0 + \frac{1}{2} \Delta t (f(t^0, y^0) + f(t^1, y^1)). \end{aligned}$$

which is equivalent to the SSPRK(2,2) of two stages and second order given in its Butcher tableau:

$$\begin{array}{c|cc} 0 & & \\ 1 & 1 & \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array}.$$

- Next, we want to consider the third order DeC scheme. We use equispaced nodes which intersect with Gauss-Lobatto nodes in this case. The values are $t^0 = 0$, $t^1 = \frac{1}{2}$ and $t^2 = 1$. The algorithm leads to the following

$$\begin{aligned} y^{1,(1)} &= y^0 + \frac{1}{2} \Delta t f(t^0, y^0), \\ y^{2,(1)} &= y^0 + \Delta t f(t^0, y^0), \end{aligned}$$

First Correction :

$$\begin{aligned} y^{1,(2)} &= y^0 + \Delta t \left(\frac{5}{24} f(t^0, y^0) + \frac{1}{3} f(t^1, y^{1,(1)}) - \frac{1}{24} f(t^2, y^{2,(1)}) \right), \\ y^{2,(2)} &= y^0 + \Delta t \left(\frac{1}{6} f(t^0, y^0) + \frac{2}{3} f(t^1, y^{1,(1)}) + \frac{1}{6} f(t^2, y^{2,(1)}) \right), \end{aligned}$$

Second Correction :

$$y^{n+1} = y_2^{2,(3)} = y_0 + \Delta t \left(\frac{1}{6} f(t^0, y^0) + \frac{2}{3} f(t^1, y^{1,(2)}) + \frac{1}{6} f(t^2, y^{2,(2)}) \right).$$

¹For simplicity, we neglect the dependence on n here.

Here, we have ignored the update for $y^{1,(3)}$ since it does not effect y^{n+1} . This is different for the classical DeC as described in [12] and investigated in [27].

However, as before, we can interpret DeC(3) in form of a RK methods. The Butcher tableau reads

$$\begin{array}{c|cccccc} 0 & 0 & & & & & \\ \frac{1}{2} & \frac{1}{2} & 0 & & & & \\ 1 & 1 & 0 & 0 & & & \\ \frac{1}{2} & \frac{5}{24} & \frac{1}{3} & -\frac{1}{24} & 0 & 0 & \\ 1 & \frac{1}{6} & \frac{4}{6} & \frac{1}{6} & 0 & 0 & \\ \hline 1 & \frac{1}{6} & 0 & 0 & \frac{4}{6} & \frac{1}{6} & \end{array} .$$

2.4 Semidiscrete Entropy Conservative Residual Distribution Methods

The focus of this work is on the DeC framework in combination with RD. Since a method of lines approach, i.e., splitting between space and time discretization, will destroy the high order of accuracy of RD methods (cf. [1]), we start to explain RD for a steady state problem

$$\operatorname{div} F(U) = 0. \quad (18)$$

Nowadays RD schemes are embed in the finite element (FE) setting. Therefore, V_h is defined as the space of globally continuous, piecewise polynomial functions of degree p :

$$\mathcal{V}^h := \{U \in C^0(\Omega_h), U|_K \in \mathbb{P}^p, \forall K \in \Omega_h\}. \quad (19)$$

\mathbb{P}^p denotes the space of polynomials of order p . With this definition an approximation of a solution of (18) can be written as a finite linear combination of a suitable choice of base functions of V_h , which are denoted by φ_σ :

$$U(x, t_n) \approx U^{h,n} = \sum_{\sigma \in \Omega_h} U_\sigma^n \varphi_\sigma(x), \quad x \in \Omega. \quad (20)$$

In general one is free regarding the choice of basis functions as long as $\operatorname{span}\{\varphi_{\sigma|K}\} = \mathbb{P}^p$. However, it has been shown in [4] that there are additional restrictions on the basis functions when combining RD with DeC. Here, the condition $\int_K \varphi_\sigma dx > 0$ must be also fulfilled [4]. Therefore, Bernstein polynomials or cubature elements (Lagrangian polynomials defined on quadrature points [11]) will be used in our case (for the space discretization). Once the setup is done, the approach works in three steps:

1. Define $\Phi^K(U) := \int_K \nabla \mathbf{F}(U) dx$ which is called the total residual of an element K ;
2. Define Φ_σ^K as the contribution of a DOF σ to the total residual of the element K . Thus, the following equation holds: $\Phi^K(U^h) = \sum_{\sigma \in K} \Phi_\sigma^K$, $\forall K \in \Omega_h$. The whole RD strategy is determined by the way the total residual of an element is distributed among its DOFs σ . Important is that for any element K and any $U^h \in \mathcal{V}^h$ the conservation relation holds:

$$\sum_{\sigma \in K} \Phi_\sigma^K(U^h) = \oint_{\partial K} \sum_j f^{\text{num},j} \left(U_{|K}^h, U_{|K-}^h \right) \cdot \nu_j, \quad (21)$$

where the symbol \oint denotes the integral done via quadrature rules;

3. Finally, all surrounding residual contributions at one DOF σ are collected and summed up. We get $\sum_{K|\sigma \in K} \Phi_\sigma^K = 0$, $\forall \sigma \in \Omega_h$ will finally enable the calculation of the coefficients U_σ^n in (20).

When needed, we can also include the boundary elements Γ in our consideration and then the update scheme reads like

$$\sum_{K|\sigma \in K} \Phi_\sigma^K + \sum_{\Gamma|\sigma \in \Gamma} \Phi_\sigma^\Gamma = 0, \quad \forall \sigma \in \Omega_h \quad (22)$$

The choice of Φ_σ^K determines the scheme.

Example 2.8. A pure Galerkin scheme can be written as $\Phi_\sigma^K(U^h) = \int_K \varphi_\sigma \nabla \cdot \mathbf{F}(U^h) dx$ with additional jump stabilization and using Gauss' theorem it becomes:

$$\Phi_\sigma^K(U^h) = \oint_{\partial K} \varphi_\sigma \mathbf{F}(U^h) \cdot \mathbf{n} d\gamma - \oint_K \nabla \varphi_\sigma \cdot \mathbf{F}(U^h) dx + \sum_{e \in K} \theta h_e^2 \oint_e [\nabla U^h] \cdot [\nabla \varphi_\sigma] d\lambda$$

with jump $[\nabla U^h] = \nabla U|_K^h - \nabla U|_{K^+}^h$. More RD formulations for different schemes including discontinuous Galerkin, flux reconstruction and nonlinear limiters can be found in [5, 7].

Entropy Correction Term

Since we want to construct fully entropy conservative/dissipative RD schemes, we follow the approach presented in [3, 8] and add an entropy correction function to our steady-state space residual. Let η be an entropy, g the corresponding entropy flux and $\partial_u \eta(u) = v$ is the entropy variable [20]. The entropy equality using the RD framework reads

$$\sum_{\sigma \in K} \langle V_\sigma, \tilde{\Phi}_\sigma^K \rangle = \oint_{\partial K} g^{\text{num}}(V|_K^h, V|_{K^-}^h) \mathbf{n} d\gamma, \quad (23)$$

where V_σ is the approximation at the DOF σ . Since (23) is not fulfilled for general Φ_σ^K , the entropy correction terms r_σ^K is added to the residuals Φ_σ^K to guarantee (23). In addition, we have to select these correction terms such that they do not violate the conservation relation. We introduce the following definition of the entropy-corrected residuals

$$\tilde{\Phi}_\sigma^K = \Phi_\sigma^K + r_\sigma^K \quad (24)$$

with the goal of fulfilling the discrete entropy condition (23). In [3], the following correction terms are presented

$$r_\sigma^K := \alpha(V_\sigma - \bar{V}), \quad \text{with } \bar{V} = \frac{1}{\#K} \sum_{\sigma \in K} V_\sigma, \quad (25)$$

$$\alpha = \frac{\mathcal{E}}{\sum_{\sigma \in K} (V_\sigma - \bar{V})^2}, \quad \mathcal{E} := \oint_{\partial K} g^{\text{num}}(V|_K^h, V|_{K^-}^h) \mathbf{n} - \sum_{\sigma \in K} \langle V_\sigma, \Phi_\sigma^K \rangle. \quad (26)$$

In addition, extensions and re-interpretations of the terms can be found in [8] where the following theorem is also proven:

Theorem 2.9. The correction term (25) with (26) satisfies

$$\sum_{\sigma \in K} r_\sigma^K = 0, \quad \sum_{\sigma \in K} \langle V_\sigma, r_\sigma^K \rangle = \mathcal{E}. \quad (27)$$

By adding (25) to the residual Φ_σ^K , the resulting scheme using $\tilde{\Phi}_\sigma^K$ is locally conservative in u and entropy conservative.

However, entropy conservation is most of the time not enough in the context of nonlinear hyperbolic conservation laws. Especially, in the presence of discontinuities (i.e. shocks), the scheme should not just fulfill the equality (23) but rather an inequality

$$\sum_{\sigma \in K} \langle V_\sigma, \hat{\Phi}_\sigma^K \rangle \geq \oint_{\partial K} g^{\text{num}}(V|_K^h, V|_{K^-}^h) \mathbf{n} d\gamma. \quad (28)$$

To obtain a semidiscrete entropy dissipative scheme, we apply the previous construction and write the new residual as

$$\hat{\Phi}_\sigma^K = \Phi_\sigma^K + r_\sigma^K + \Psi_\sigma^K, \quad (29)$$

where r_σ^k are defined by (25). The Ψ_σ^K satisfy

$$\sum_{\sigma \in K} \Psi_\sigma^K = 0 \text{ and } \sum_{\sigma \in K} \langle V_\sigma, \Psi_\sigma^K \rangle \geq 0. \quad (30)$$

Two expressions for Ψ_σ^K that can be used to enforce the inequality, not violating the conservation requirement, are **streamline or jump diffusion**. In this work, we apply only the jump diffusion, defined by

$$\Psi_\sigma^K := \nu h_K^2 \oint_{\partial K} [\nabla_{\mathbf{x}} \varphi_\sigma] \cdot [\nabla_{\mathbf{x}} V^h] d\lambda \quad (31)$$

which ensure $\sum_{\sigma \in K} \langle V_\sigma, \Psi_\sigma^K \rangle = \nu h_K^2 \sum_{e \in K} \oint_{\partial K} [\nabla_{\mathbf{x}} V^h]^2 d\lambda \geq 0$ for any $\nu > 0$. We apply this correction term in the nonlinear case to guarantee the inequality in presence of shocks.

Remark 2.10. *The presented entropy corrections for residual distribution schemes must be chosen entropy conservative, as in (27), or entropy dissipative, as in (30). This choice must be done a priori knowing the behavior of the problem.*

2.5 Residual Distribution for Hyperbolic Problems

After describing the general construction of RD and DeC schemes, they can be coupled to a explicit space-time FE scheme for the initial value problem (1). We like to point out that this approach has several similarities and connections to the modern ADER approach [19].

The classic DeC algorithm does not change what actually is changing are the \mathcal{L}^1 and \mathcal{L}^2 operators. Here, we have to take the steady space residuals $\Phi_{\sigma,x}^K$ into account. Following [2, 4], we have

$$\begin{aligned} \mathcal{L}_\sigma^1(U^{(k)}) &= \begin{pmatrix} |C_\sigma|(U_\sigma^{n,M,(k)} - U_\sigma^{n,0}) + \Delta t \beta_M \sum_{K|\sigma \in K} \Phi_{\sigma,x}^K(U^{n,0}) \\ \vdots \\ |C_\sigma|(U_\sigma^{n,1,(k)} - U_\sigma^{n,0}) + \Delta t \beta_1 \sum_{K|\sigma \in K} \Phi_{\sigma,x}^K(U^{n,0}) \end{pmatrix}, \\ \mathcal{L}_\sigma^2(U^{(k)}) &= \begin{pmatrix} \sum_{K|\sigma \in K} \left(\oint_K \varphi_\sigma (U^{n,M,(k)} - U^{n,0}) + \Delta t \sum_{r=0}^M \theta_r^M \Phi_{\sigma,x}^K(U^{n,r,(k)}) \right) \\ \vdots \\ \sum_{K|\sigma \in K} \left(\oint_K \varphi_\sigma (U^{n,1,(k)} - U^{n,0}) + \Delta t \sum_{r=0}^M \theta_r^1 \Phi_{\sigma,x}^K(U^{n,r,(k)}) \right) \end{pmatrix}. \end{aligned}$$

where β_i, θ_r^i are the quadrature weights for the time integration and $|C_\sigma| := \oint_K \varphi_\sigma$. The term $|C_\sigma|$ can be seen as some introduced mass lumping effect and should be positive definite. The l -th line simply becomes

$$U_\sigma^{n,l,(k)} = U_\sigma^{n,l,(k-1)} - |C_\sigma|^{-1} \sum_{K|\sigma \in K} \left(\oint_K \varphi_\sigma (U^{n,l,(k-1)} - U^{n,0}) + \Delta t \sum_{r=0}^M \theta_r^l \Phi_{\sigma,x}^K(U^{n,r,(k-1)}) \right), \quad (32)$$

that we will also use in its vector formulation, denoting with $\mathbf{M}_{ij} = \oint_\Omega \varphi_i \varphi_j dx$ the mass matrix, with $\mathbf{D}_{ii} = |C_i| = \oint_\Omega \varphi_i dx$ the lumped diagonal matrix and with \mathbf{I} the identity matrix, i.e.,

$$\mathbf{U}^{n,l,(k)} = \mathbf{U}^{n,l,(k-1)} - \mathbf{D}^{-1} \left(\mathbf{M}(\mathbf{U}^{n,l,(k-1)} - \mathbf{U}^{n,0}) + \Delta t \sum_{r=0}^M \theta_r^l \Phi(\mathbf{U}^{n,r,(k-1)}) \right) \quad (33)$$

$$= \mathbf{U}^{n,0} + (\mathbf{I} - \mathbf{D}^{-1} \mathbf{M})(\mathbf{U}^{n,l,(k-1)} - \mathbf{U}^{n,0}) - \Delta t \sum_{r=0}^M \theta_r^l \mathbf{D}^{-1} \Phi(\mathbf{U}^{n,r,(k-1)}). \quad (34)$$

Remark 2.11. *With this approach, we can avoid a mass matrix inversion which is typically in a FE approach. However, we have to deal with further terms in the semidiscretization. This has to be taken into account when the energy/entropy production is discussed. Nevertheless, it should be marked that we have now two different possibilities to apply the entropy correction terms (27) or (30) in (34). First, we can use it only in the pure space residual $\Phi_x(\mathbf{U}^{n,r,(k-1)})$ resulting in a semidiscrete entropy conservative or dissipative RD scheme. Second, we apply it to the hole*

$$(\mathbf{I} - \mathbf{D}^{-1}\mathbf{M})(\mathbf{U}^{n,l,(k-1)} - \mathbf{U}^{n,0}) - \Delta t \sum_{r=0}^M \theta_r^l \mathbf{D}^{-1} \Phi(\mathbf{U}^{n,r,(k-1)})$$

to obtain a fully discrete conservative scheme. In [8], first simulations were already been done considering this direction. The authors realized only small differences but a rather complex implementation strategy. Therefore, combining the correction approach with the relaxation framework should simplify the implementation. Because of this reasons, we apply the correction term (30) only in the semidiscrete setting together with the proposed relaxation approach.

In the future, a numerical comparison between two different approaches will be given extending also the application to different FE based schemes. However, this is not the point of this current work.

3 Deferred Correction Methods – Connection to RK, SSP and Relaxation Technique

In this section, we reinterpret DeC as a classical RK method and mention also the connection to SSP methods. Therefore, we focus again only on the simple ODE case (2). Further, we apply the relaxation technique to this interpretation demonstrating the desired properties. In the following sections we will compare the RDeC to the classical relaxed SSPRK methods numerically both for ODEs and PDEs.

3.1 The Connection between DeC and RK with possible Application to SSP

As we have seen already in example 2.7, the first two DeC approaches can be directly interpreted as RK schemes where the quadrature weights θ and β defined the coefficients of the Butcher tableau. However, this is not new at all. Already in [10] this embedding has been pointed out for the classical DeC approach resulting in a block structure matrix for A with repeated coefficients. Here, we adapt this to the simplified version from [2]. As it has been presented in [30, 40], DeC has the advantage that one does not specify the coefficients for every order of accuracy as usually necessary in classical RK methods. This is done automatically. On the other side, by rewriting a DeC method as a RK scheme, it requires a number of stages equal to $K \times M = d \times (d - 1)$, which is bigger than classical RK stages. Nevertheless, in our case we can even delete additional stages since the intermedia values of the last correction step are not needed anymore due to the simplification from [2], cf. example 2.7. Indeed, the \mathcal{L}^1 operator is always starting from y^0 , which is different to the DeC method described in [12, 27]. Therefore, we get $d \times (d - 1) - (d - 2) = (d - 1)^2 + 1$ total stages.

Moreover, one can notice that every subimestep is independent of another, so one can compute sequentially the corrections and in parallel the subimesteps, obtaining a computational cost of just $K = d$ corrections. Let us consider our version of DeC and rewrite it in a Butcher tableau

$$\begin{array}{c|c} c & A \\ \hline & b \end{array}.$$

Here, A is block diagonal after the first column. The column includes the explicit Euler steps from the y^0 to y^i , i.e., the β^i coefficients with $\beta^0 = 0$. Next, the θ_r^m are written in a matrix form where the superscript m describes the rows and r the columns. We have $\underline{\theta} = (\theta_r^m)$ and we split the matrix using $\underline{\theta} = (\underline{\theta}_0 | \tilde{\underline{\theta}})$, where $\underline{\theta}_0 = (\theta_0^1, \theta_0^2, \dots, \theta_0^M)^T$ represents the first column vector and $\tilde{\underline{\theta}}$ the remaining matrix. The first column of

the Butcher tableau then includes $\underline{\theta}_0$ repeatedly to the number of corrections steps minus one before the final step is reached. The remaining matrix is written on the right side of it but in each correction step it is shifted to the right depending on the used number of subinterval. The open positions are filled with zeros and this is done until the last correction. Here, the correction does not affect the intermediate values and has to be evaluated only at the last one. This means that it defines b in the Butcher tableau and A is not affected anymore. We obtain $b_1 = \theta_0^M$ and all of the following b_i values are zero with respect to the blocks of A . In connection, we include θ_r^M with $r \in 1, \dots, M$, denoting the vector $\underline{\theta}^{M,T} = (\theta_1^M, \dots, \theta_M^M)$. The Butcher tableau for an arbitrary DeC approach is given by:

$$\begin{array}{c|cccccc}
 0 & 0 & & & & \\
 \underline{\beta} & \underline{\beta} & & & & \\
 \underline{\beta} & \underline{\theta}_0 & \underline{\tilde{\theta}} & & & \\
 \vdots & \underline{\theta}_0 & \underline{0} & \underline{\tilde{\theta}} & & \\
 \vdots & \underline{\theta}_0 & \underline{0} & \underline{0} & \underline{\tilde{\theta}} & \\
 \vdots & \vdots & \vdots & \vdots & \ddots & \ddots \\
 \underline{\beta} & \underline{\theta}_0 & \underline{0} & \dots & \dots & \underline{0} & \underline{\tilde{\theta}} \\
 \hline
 & \underline{\theta}_0^M & \underline{0}^T & \dots & \dots & \underline{0}^T & \underline{\theta}_r^{M,T}
 \end{array} \quad (35)$$

A comparison to the presented example 2.7 demonstrated that the interpretations coincide. Please note that this description is from [30] where the different y^0 values have been set directly equal to the starting point. In [40], these values have been considered inside the RK methods yielding to a slight different Butcher tableau which is equivalent to the presented one (35).

From the Butcher representation, we can easily obtain the Shu-Osher formulation (5) using for example the provided Matlab scripts². However, due to the special structure of the Butcher tableau, the coefficients $[\alpha, \beta]$ from (5) can be easily read directly from (35). α, β are two matrices with one column more than the number of stages. They belong to $\mathbb{R}^{(s+1) \times s}$ with $s = (d-1)^2 + 1$. α contains zeros everywhere except in the first column starting with the second row where we have always ones in the entries and β is given by $\beta = \begin{bmatrix} A \\ b \end{bmatrix}$

with A, b from (35). From the Shu-Osher formulation the usual SSP analysis can be done as presented in [27] for the classical DeC in combination with an DG scheme. Please be aware that in case of negative coefficients in the formula, one has to change the wind directions in the spirit of [27]. This is also possible using the RD framework. It is currently work in progress and not topic of this paper. However, this part should clarify the connection between DeC and classical RK schemes. Due to the close connection between the modern ADER approach and DeC [19], this will also open the theoretical investigations of ADER. Next, we focus on the relaxation approach.

3.2 Relaxation DeC

Due to the RK interpretation, all the theoretical fundings of [22, 38, 35] will transfer directly to the DeC approach if one uses DeC to solve ODEs or as a time-integration scheme in simple splitting through the methods of lines. Therefore, for such problems, we do not demonstrate new results in this part, but a rather different technique to prove it using the DeC framework. This should lead to a better understanding for the relaxation approach in the following. Nevertheless, it is still important that we can adapt the relaxation to our simplified DeC framework, due to the fact that we want to apply it later together with the RD methods. It is important for the future analysis in Section 4. Here, we need actually our proven new results to guarantee that the RDeC-RD methods remain high order accurate. Again, a classical splitting technique through the method of lines has to be avoided for RD methods not to violate the high order property.

To adapt the relaxation approach, we have to change the \mathcal{L}^2 operator. For the step y^M and the last correction, we multiply Δt by γ_n from (10). This is a simple projection, cf. [35]. Again, γ_n can be calculated using the

²http://www.cfm.brown.edu/people/sg/SSPpage/sspsite/matlab_scripts.html

expression (35) with (10) in the ODE and simple PDE case, but for completeness reasons, we demonstrate that the change of the \mathcal{L}^2 operator does not destroy the accuracy properties of the RDeC scheme.

Accuracy Property

We denote in the following the relaxation \mathcal{L}^2 operator with $\mathcal{L}_{\gamma_n}^2$. Before, we start we like to point out that one may think that y_{γ}^{n+1} approximates $y(t^n + \Delta t)$ or $y(t^n + \gamma_n \Delta t)$. As it is formulated in [22], the first approach corresponds to the *incremental direction technique* (IDT) and it was proven that the scheme yield to a $d - 1$ order method if the corresponding RK method is off order d . This is also the case using the DeC approach. In this paper, we focus on the second interpretation (i.e. $y_{\gamma}^{n+1} \approx y(t^n + \gamma_n \Delta t)$) which gives a d order method.

We will mimic the original proof from [2]. Here, the term y^* is applied and denotes the solution of the classical \mathcal{L}^2 operator i.e. $\mathcal{L}^2(y^*) = 0$. Using the fact that $\gamma_n = 1 + \mathcal{O}(\Delta t^{d-1})$ stated in [22, Lemma 3], we obtain

$$\mathcal{L}_{\gamma_n}^2(y^*) = \mathcal{L}^2(y^*) + \mathcal{O}(\Delta t^d) = \mathcal{O}(\Delta t^d), \quad (36)$$

where we get another order from the time integration.

We prove that

$$\begin{aligned} \|y^{(M)} - y^*\| &\leq C_0 \|\mathcal{L}^1(y^{(M)}) - \mathcal{L}^1(y^*)\| = C_0 \|\mathcal{L}^1(y^{(M-1)}) - \mathcal{L}_{\gamma_n}^2(y^{(M-1)}) - \mathcal{L}^1(y^*) + \mathcal{L}_{\gamma_n}^2(y^*)\| \\ &\leq C^M \Delta t^M \|y_0 - y^*\|, \end{aligned} \quad (37)$$

since in every iteration step, we obtain one order of accuracy more. To prove that the inequalities in (37) are valid, we have to ensure the following conditions:

1. the coercivity of the operator \mathcal{L}^1
2. the high order accuracy of the operators $\mathcal{L}_{\gamma_n}^2$.
3. the Lipschitz inequality for operator $\mathcal{L}^1 - \mathcal{L}_{\gamma_n}^2$

Since the \mathcal{L}^1 operator is not changed, the coercivity follows directly from the analysis in [2, 32]. The classical \mathcal{L}^2 operators represent an implicit high-order scheme. Using the relaxation approach does not change this property due the analysis from [22], cf. equation (36). The operator $\mathcal{L}_{\gamma_n}^2$ represents also a high order accurate time integration scheme. Therefore, we have only to consider the Lipschitz continuity of $\mathcal{L}^1 - \mathcal{L}_{\gamma_n}^2$.

Theorem 3.1 (Lipschitz continuity of $\mathcal{L}^1 - \mathcal{L}_{\gamma_n}^2$). *We consider a DeC method of order d and assume further that*

$$\gamma_n = 1 + \mathcal{O}(\Delta t^{d-1}) \quad (38)$$

is fulfilled. Let $y^{(k)}$, $y^{(k-1)}$, y^ be given as above described. Then, the operator $\mathcal{L}^1 - \mathcal{L}_{\gamma_n}^2$ is Lipschitz continuous with constant $\Delta t C_L$, i.e.*

$$\|\mathcal{L}^1(y^{(k)}) - \mathcal{L}_{\gamma_n}^2(y^{(k)}) - \mathcal{L}^1(y^*) + \mathcal{L}_{\gamma_n}^2(y^*)\| \leq C_L \Delta t \|y^{(m)} - y^*\| \quad (39)$$

Proof. The operator \mathcal{L}_{γ}^2 intersect in every correction step with the classical \mathcal{L}^2 operator except in the last correction. Therefore, the Lipschitz continuity follows directly for every correction until the final one from the investigation done in [2]. We have to show the Lipschitz continuity for the last correction step. Through the assumption (38) and substitute $\mathcal{L}_{\gamma_n}^2$ with the original \mathcal{L}^2 operator we add an error of order $\mathcal{O}(\Delta t^d)$ to it. It is

$$\|\mathcal{L}^1(y^{(K-1)}) - \mathcal{L}_{\gamma_n}^2(y^{(K-1)}) - \mathcal{L}^1(y^*) + \mathcal{L}_{\gamma_n}^2(y^*)\| \leq \|\mathcal{L}^1(y^{(K-1)}) - \mathcal{L}^2(y^{(K-1)}) - \mathcal{L}^1(y^*) + \mathcal{L}^2(y^*)\| + \mathcal{O}(\Delta t^d)$$

We can concentrate on the single equation using the notation $\mathcal{L}^{2,l}$. The difference is given by

$$|\mathcal{L}^{2,l}(y^{(K-1)}) - \mathcal{L}^{1,l}(y^{(K-1)})| = \left| \Delta t \left(\sum_{r=0}^M \theta_r^l f(y^{r,(K-1)}) - \sum_{r=0}^M \beta_r^l f(y^{r,(K-1)}) \right) \right| = \left| \Delta t \sum_{r=0}^M (\theta_r^l - \beta_r^l) f(y^{r,(K-1)}) \right| \quad (40)$$

Here, we have used the coefficients β_r^l , which are an extension of the previously defined β^l coefficients and they are defined as

$$\beta_r^l := \begin{cases} \beta^l & \text{for } r = 0, \\ 0 & \text{for } r = 1, \dots, M. \end{cases}$$

Now, we can compute the differences of the two terms

$$\begin{aligned} |\mathcal{L}_l^1(y^{(K-1)}) - \mathcal{L}_l^2(y^{(K-1)}) - \mathcal{L}_l^1(y^*) + \mathcal{L}_l^2(y^*)| &\leq \left| \Delta t \sum_{r=0}^M (\theta_r^l - \beta_r^l) \left(f(y^{r,(K-1)}) - f(y^{r,*}) \right) \right| \\ &\leq \Delta t C_L \|y^{(K-1)} - y^*\| \end{aligned}$$

where we used with an abuse of notation in the last step the Lipschitz continuity of f . \square

With the different interpretations of the RDeC approach, we can formulate the following theorem:

Theorem 3.2 (Convergence of RDeC). *Let f sufficiently smooth and we consider a DeC approach of order $d > 1$. Let $\mathcal{L}^1(\cdot)$ and $\mathcal{L}_{\gamma_n}^2(\cdot)$ are the defined operators respectively. The RDeC procedure with the relaxation term (10) gives a numerical approximation with order of accuracy equal to d .*

Proof. With the coercivity of the \mathcal{L}^1 operator, the first inequality in (37) is verified. The definition of the RDeC scheme gives us the equality and the Lipschitz continuity Theorem 3.1 proves the last inequality (37). Moreover, we know that y^* is a d -th accurate approximation of the y^{ex} exact solution depending how it is interpreted. So, overall, we have

$$\|y^{(K)} - y^{ex}\| \leq \|y^* - y^{ex}\| + (C\Delta t)^d \|y^* - y^{(0)}\| \leq C^* \Delta t^d + C_L \Delta t^d \quad (41)$$

which proves the result. \square

Finally, we have to say that in (3.2), we have assumed that $\gamma_n = 1 + \mathcal{O}(\Delta t^{d-1})$ is fulfilled. However, this is not needed since as presented in [22, Lemma 3]:

Lemma 3.3. *Let a_{ij}, b_j denotes the coefficients of a RK method of order d , let f be a sufficiently smooth function, and let γ_n be defined by (11). Then $\gamma_n = 1 + \mathcal{O}(\Delta t^{d-1})$.*

Remark 3.4. *An alternative proof of this results is shown in [38]. Since we can interpret DeC as a RK scheme, we know that this condition is always fulfilled.*

Extension to General Entropies

We have presented the relaxation approach up to now only for the energy in the ODE case where we have assumed an energy conservative/dissipative system. Those systems naturally develop from hyperbolic conservation/balance laws using energy (entropy) conservative/dissipative space discretizations, cf. [6, 9, 14, 37] and references therein. One is not only interested in the energy behavior but on the behavior of general entropies, especially in the nonlinear case, e.g. Euler equations in gas dynamics. As presented in [38], the relaxation approach can easily adapted to general entropy functions. Again, we denote with η the strictly convex entropy and with g is associated entropy flux for an hyperbolic conservation law (1). From the theory, we know that the entropy satisfies the additional hyperbolic equation $\partial_t \eta + \text{div } g = 0$, and we have the following relation to be fulfilled: $g = \langle \eta', f \rangle$. The space discretisation is done by an semidiscrete entropy conservative/dissipative scheme resulting in our ODE system and now the time-integration comes into play. We know that explicit time-integration methods produces always entropy and the change of the entropy is given by $\partial_t \eta = g_{se} = \langle \eta'_{se}, f_{se} \rangle$ where we use the index to clarify that the right side is a discretization. Therefore, the following nonlinear equation has to be solved

$$\eta(y^{n+1}) - \eta(y^n) = \Delta t \gamma_n \sum_{r=0}^M \theta_r^M \left\langle \eta'_{se}(y^{r,(K)}), f_{r,se} \right\rangle,$$

where θ_r^M denote the coefficients of the Butcher tableau corresponding to DeC and $y^{r,(K)}$ are the intermedia steps. Instead of searching the roots as for the energy, the following equation has to be solved

$$r(\gamma_n) = \eta \left(y_0 + \Delta t \gamma_n \sum_{r=0}^M \theta_r^M \left\langle \eta'_{se}(y^{r,(K)}), f_{r,se} \right\rangle \right) - \eta(y_0) - \Delta t \gamma_n \sum_{r=0}^M \theta_r^M \left\langle \eta'_{se}(y^{r,(K)}), f_{r,se} \right\rangle \stackrel{!}{=} 0. \quad (42)$$

In the comparison simulations, we use analogously to the work [38] the Brent algorithm or the Marquard-Levenberg variation, but also a simple Newton method can also be applied to solve (42).

4 A Fully Entropy Conservative/Dissipative DeC-RD Approach

In the following section, we will describe how we can combine the relaxationDeC approach together with the RD framework and how the relaxation parameter γ is in this case calculated. We had for DeC in the RD framework the following update formula for the final step (34), denoting with $\mathbf{U}^l = \mathbf{U}^{l,(K-1)}$, with $\mathbf{U}^{end} = \mathbf{U}^{n+1}$ and $\mathbf{U}^0 = \mathbf{U}^n$, we know that the last step will be

$$\mathbf{U}^{end} = \mathbf{U}^0 + \Delta t \left\{ (\mathbf{I} - \mathbf{D}^{-1}\mathbf{M}) \frac{\mathbf{U}^M - \mathbf{U}^0}{\Delta t} - \sum_{r=0}^M \theta_r^M \mathbf{D}^{-1} \Phi_x(\mathbf{U}^r) \right\}, \quad (43)$$

We assume further that due to the usage of the entropy correction term (22) (plus the addition of diffusion terms) our space residual is already entropy/energy conservative/dissipative i.e.,

$$\sum_{K|\sigma \in K} |C_\sigma|^{-1} \langle V_\sigma(\mathbf{U}^r), \Phi_{\sigma,x}(\mathbf{U}^r) \rangle \stackrel{(\geq)}{=} \oint_{\partial K} g(\mathbf{U}^r) \cdot \mathbf{n} \, d\Gamma, \quad (44)$$

where V_σ are the entropy variables, g is the entropy flux and Φ_x denotes the space residual. We remark that we consider continuous RD distribution approximation. One can substitute in the previous equation the entropy flux g with a numerical flux $g^{num}(\cdot, \cdot)$ in case of discontinuous approximations.

Now, we want to apply the relaxation approach to obtain a fully discrete entropy conservative/dissipative scheme. Here, we have to make slight modifications.

Actually, the main idea of the relaxation approach is to decrease the update time-step with respect to the entropy production of the fully discrete scheme.

In the RD framework, we cannot simple apply this term since by focusing on (43), we realize that we have additional terms given by the lumping of the mass matrix in \mathcal{L}^1 . The sign of the entropy contribution of this term is unknown.

With an abuse of notation, we define a scalar product onto the whole space \mathcal{V} in the following way,

$$\langle \mathbf{U}, \mathbf{W} \rangle := \sum_K \sum_{\sigma \in K} \int_K \langle U_\sigma, W_\sigma \rangle \varphi_\sigma(x) dx = \sum_{\sigma \in \Omega} |C_\sigma| \langle U_\sigma, W_\sigma \rangle \quad (45)$$

for consistency reasons. Indeed, summing up all the cells K in (44) we obtain the inequality

$$\langle \mathbf{V}(\mathbf{U}^r), \mathbf{D}^{-1} \Phi_x(\mathbf{U}^r) \rangle \stackrel{(\geq)}{=} \oint_{\partial \Omega} g(\mathbf{U}^r) \cdot \mathbf{n} \, d\Gamma. \quad (46)$$

Let us develop the equation for entropy that we would like to preserve for the relaxed value

$$\mathbf{U}_{\gamma_n}^{end} := \mathbf{U}^0 + \gamma_n \Delta \mathbf{U}, \quad (47)$$

defining

$$\Delta \mathbf{U} := \Delta t \left\{ (\mathbf{I} - \mathbf{D}^{-1}\mathbf{M}) \frac{\mathbf{U}^M - \mathbf{U}^0}{\Delta t} - \sum_{r=0}^M \theta_r^M \mathbf{D}^{-1} \Phi_x(\mathbf{U}^r) \right\}. \quad (48)$$

The entropy reads and can be expanded as

$$\eta(\mathbf{U}_{\gamma_n}^{end}) = \eta(\mathbf{U}^0) + \gamma_n \Delta t \left\langle \eta'(\mathbf{U}^0), \underbrace{(\mathbf{I} - \mathbf{D}^{-1}\mathbf{M})}_{\mathcal{O}(\Delta x)} \underbrace{\frac{\mathbf{U}^M - \mathbf{U}^0}{\Delta t}}_{\mathcal{O}(1)} - \sum_{r=0}^M \theta_r^M \mathbf{D}^{-1} \Phi_x(\mathbf{U}^r) \right\rangle + \mathcal{O}(\Delta t^2) \quad (49)$$

$$= \eta(\mathbf{U}^0) - \gamma_n \Delta t \left\langle \mathbf{V}(\mathbf{U}^0), \mathbf{D}^{-1} \left[\sum_{r=0}^M \theta_r^M \Phi_x(\mathbf{U}^r) \right] \right\rangle + \mathcal{O}(\Delta t^2) \quad (50)$$

$$= \eta(\mathbf{U}^0) - \gamma_n \Delta t \sum_{r=0}^M \theta_r^M \underbrace{\langle \mathbf{V}(\mathbf{U}^r), \mathbf{D}^{-1} \Phi_x(\mathbf{U}^r) \rangle}_{\substack{(\geq) - \int_{\partial\Omega} g(\mathbf{U}^r) \cdot \mathbf{n} \, d\Gamma}} + \mathcal{O}(\Delta t^2). \quad (51)$$

We remind that for DeC with order less than 9 all the $\{\theta_r^M\}_r$ are positive, while for higher orders one has to stick to Gauss–Lobatto subimestep distribution to have positive coefficients. Motivated by this expansion, we impose that

$$\eta(\mathbf{U}_{\gamma_n}^{end}) - \eta(\mathbf{U}^0) + \gamma_n \Delta t \sum_{r=0}^M \theta_r^M \langle \mathbf{V}(\mathbf{U}^r), \mathbf{D}^{-1} \Phi_x(\mathbf{U}^r) \rangle \stackrel{!}{=} 0, \quad (52)$$

by solving this scalar equation for γ_n . Then, we have that

$$\begin{aligned} \eta(\mathbf{U}_{\gamma_n}^{end}) &= \eta(\mathbf{U}^0) - \gamma_n \Delta t \sum_{r=0}^M \theta_r^M \langle \mathbf{V}(\mathbf{U}^r), \mathbf{D}^{-1} \Phi_x(\mathbf{U}^r) \rangle \\ &\stackrel{(\leq)}{=} \eta(\mathbf{U}^0) - \gamma_n \Delta t \sum_{r=0}^M \theta_r^M \int_{\partial\Omega} g(\mathbf{U}^r) \cdot \mathbf{n} \, d\Gamma. \end{aligned} \quad (53)$$

A priori, the scalar equation (52) that we want to solve is a nonlinear equation that we can solve, for instance, with a Newton method, i.e.,

$$\eta(\mathbf{U}^0 + \gamma_n \Delta \mathbf{U}) - \eta(\mathbf{U}^0) + \gamma_n \Delta t \sum_{r=0}^M \theta_r^M \langle \mathbf{V}(\mathbf{U}^r), \mathbf{D}^{-1} \Phi_x(\mathbf{U}^r) \rangle = 0. \quad (54)$$

In case of energy entropy, $\eta(\mathbf{U}) := \frac{1}{2} \langle \mathbf{U}, \mathbf{U} \rangle$ and $\mathbf{V}(\mathbf{U}) = \mathbf{U}$, (52) becomes

$$\frac{\langle \mathbf{U}^0, \mathbf{U}^0 \rangle}{2} + \gamma_n \langle \mathbf{U}^0, \Delta \mathbf{U} \rangle + \gamma_n^2 \langle \Delta \mathbf{U}, \Delta \mathbf{U} \rangle - \frac{\langle \mathbf{U}^0, \mathbf{U}^0 \rangle}{2} + \gamma_n \Delta t \sum_{r=0}^M \theta_r^M \langle \mathbf{U}^r, \mathbf{D}^{-1} \Phi_x(\mathbf{U}^r) \rangle = 0, \quad (55)$$

hence, we can solve explicitly the equation for γ_n with

$$\gamma_n = -2 \frac{\Delta t \sum_{r=0}^M \langle \mathbf{U}^r, \mathbf{D}^{-1} \Phi_x(\mathbf{U}^r) \rangle + \langle \mathbf{U}^0, \Delta \mathbf{U} \rangle}{\langle \Delta \mathbf{U}, \Delta \mathbf{U} \rangle}. \quad (56)$$

Remark 4.1. We recall that the entropy corrections presented above can either be dissipative, i.e.,

$$\langle \mathbf{V}(\mathbf{U}^r), \mathbf{D}^{-1} \Phi_x(\mathbf{U}^r) \rangle > \oint_{\partial\Omega} g(\mathbf{U}^r) \cdot \mathbf{n} \, d\Gamma,$$

or conservative, i.e.,

$$\langle \mathbf{V}(\mathbf{U}^r), \mathbf{D}^{-1} \Phi_x(\mathbf{U}^r) \rangle = \oint_{\partial\Omega} g(\mathbf{U}^r) \cdot \mathbf{n} \, d\Gamma.$$

This does not detect when the problem switches between an entropy conservative regime to an entropy dissipative regime, as often happens in hyperbolic problems. Nevertheless, we know that the advantages of the relaxation schemes are remarkable when we conserve the entropy, hence in the simulations for hyperbolic problems, we will focus on the entropy conservative case, where we will impose $\eta(\mathbf{U}^0 + \gamma_n \Delta \mathbf{U}) = \eta(\mathbf{U}^0)$, solving directly, in the energy case,

$$\gamma_n = \frac{-2 \langle \mathbf{U}^0, \Delta \mathbf{U} \rangle}{\langle \Delta \mathbf{U}, \Delta \mathbf{U} \rangle}. \quad (57)$$

Remark 4.2. In Appendix A we perform similar computation weighting not the whole $\Delta \mathbf{U}$ term, but only the Δt which comes from the \mathcal{L}^2 formulation. This leads to a more complicated formulation, but anyway viable with the previously presented techniques.

5 Numerical Simulations

In this part, we validate our RDeC methods and compare it also with the RRK method given and investigated in [22, 38]. We focus on similar examples, first we apply the pure time integration RDeC³ implementation, and we also extend the provided RRK code [22, 33] with RDeC schemes (arbitrarily high order using Gauss-Lobatto and equidistant nodes) for the ODE case and simple PDEs. Finally, we apply the RDeC approach together with the semidiscrete entropy conservative / dissipative RD method resulting in a fully discrete, explicit, entropy conservative / dissipative finite element based scheme. The θ nodes are calculated using $\theta_r^m = \int_{t^0}^{t^m} \varphi_r dt$. In the comparison part with RRK methods, we restrict ourself to the SSPRK(2,2), the SSPRK(3,3) and finally the classical fourth order RK method with four stages (RK(4,4)). As seen in example 2.7 the second order DeC approach is equivalent to SSPRK(2,2) and the results coincide. Therefore, we renounce to plot both methods and apply the SSPRK(2,2) to describe both SSPRK(2,2) and DeC2. Additionally to the investigation in [22, 38], we analyze also how the number of time steps changes among the different methods.

5.1 Numerical test in the ODE case

In this section we support our theoretical findings and validate the ideas for systems of ODEs

$$\partial_t U + F(U) = 0, \quad t \in [0, T], \quad (58)$$

where $U(t) \in \mathbb{R}^D$ and $F : \mathbb{R}^D \rightarrow \mathbb{R}^D$ is a Lipschitz continuous function. In this context, the residual Φ_x in section 4 become simply the function F and the mass matrix and the diagonal matrix are trivially the identity matrix.

5.1.1 Nonlinear Oscillator

The first problem is the nonlinear oscillator described in [22] through the system

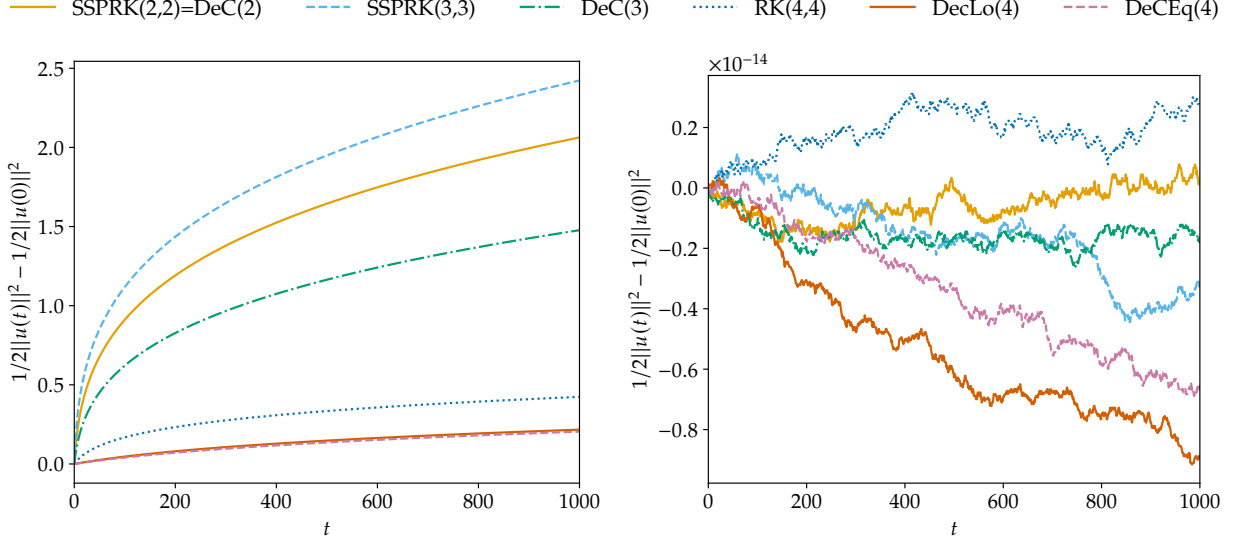
$$\begin{cases} \partial_t \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} \frac{-u_2}{n} \\ \frac{u_1}{n} \end{pmatrix}, & \text{with } n := \sqrt{u_1^2 + u_2^2}, \\ u_1(0) = u_1^0, \\ u_2(0) = u_2^0. \end{cases} \quad (59)$$

The system verifies the exact solution

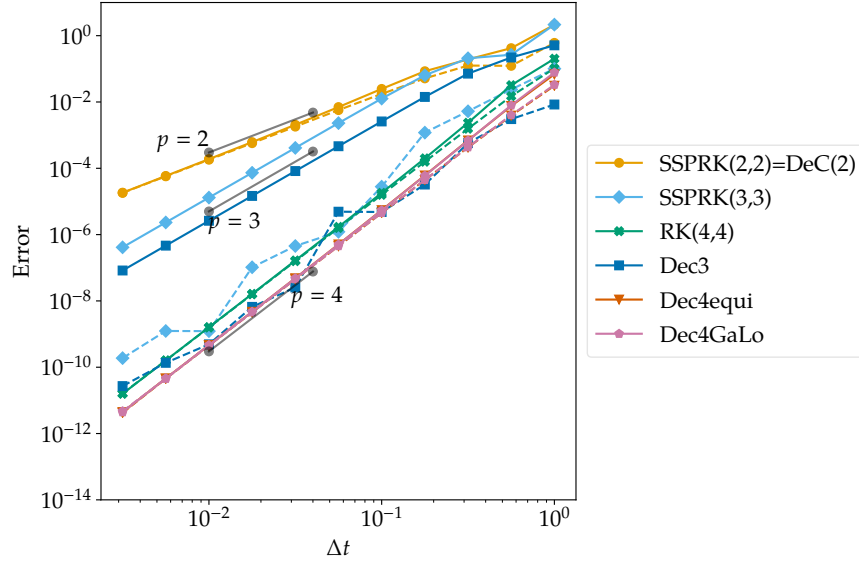
$$\begin{pmatrix} u_1(t) \\ u_2(t) \end{pmatrix} = \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix} \begin{pmatrix} u_1^0 \\ u_2^0 \end{pmatrix}, \quad \theta(t) := \frac{t}{n}. \quad (60)$$

³git.math.uzh.ch/abgrall-group/relaxation-dec-code.git

We consider the energy (L^2 -entropy) $\eta(u) = \frac{n^2}{2}$. In our tests, we took $\Delta t = 0.9$ and $T = 1000$. In fig. 2, we plot the progression of the energy. All the schemes gain energy over time when not using the relaxation term but the DeC schemes of order three and four (with equidistant points (DeCEq) and with Gauss-Lobatto points (DecLo)) behave better compared to the classical RK methods of the same order.



(a) Square entropy progress without modification with $\Delta t = 0.9$ (b) Square entropy progress with relaxation modification with $\Delta t = 0.9$



(c) Convergence for some RK and DeC methods, classical mehtods with continuous lines, relaxation methods with dashed lines

Figure 2: Nonlinear oscillator with $\Delta t = 0.9$

By using the relaxation approach, all schemes are energy conservative up to machine precision. Additionally, in fig. 2c we compared the convergence at $t = 10$ between the unmodified and the relaxed schemes. In general, the relaxation improved the accuracy for every scheme or at least kept it at the same level as with

the unmodified version. However, it should be pointed out that the time step is changed now in every step and we need more steps to reach $t = 10$.

Thanks to the RDeC formulation we can test the properties of our schemes in the very high order regime. In fig. 3 the simulations for the nonlinear oscillator problem are depicted for DeC and RDeC of order 2 and 6 for different number of timesteps. In the high order solutions in fig. 3a we can not observe qualitatively a difference between DeC and RDeC, but for the second order method in fig. 3b it is evident that the DeC increases the energy of the system violating the physical bounds.

Computing the norm of the energy and its evolution in fig. 3c we observe a zero deviation of the total energy of the RDeC schemes, while the DeC are producing more energy with respect to the exact values, with the respective order of accuracy.

Finally, we observe in fig. 4 that there are several phenomena of super convergence in this test. In particular in the case of equispaced points we fall back in the superconvergence of the odds orders as already shown in [33], while for Gauss Lobatto nodes we have even further phenomena of super convergence. Only order 2 is not affected by this.

5.1.2 Damped Nonlinear Oscillator

The second ODE we consider is a damped version of the previous nonlinear oscillator. It is defined by the initial value problem

$$\begin{cases} \partial_t \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} \frac{-u_2}{n} - \alpha u_1 \\ \frac{u_1}{n} - \alpha u_2 \end{pmatrix}, & \text{with } n := \sqrt{u_1^2 + u_2^2}, \\ u_1(0) = u_1^0, \\ u_2(0) = u_2^0. \end{cases} \quad (61)$$

The final time is set to $T = 100$. The system verifies the exact solution

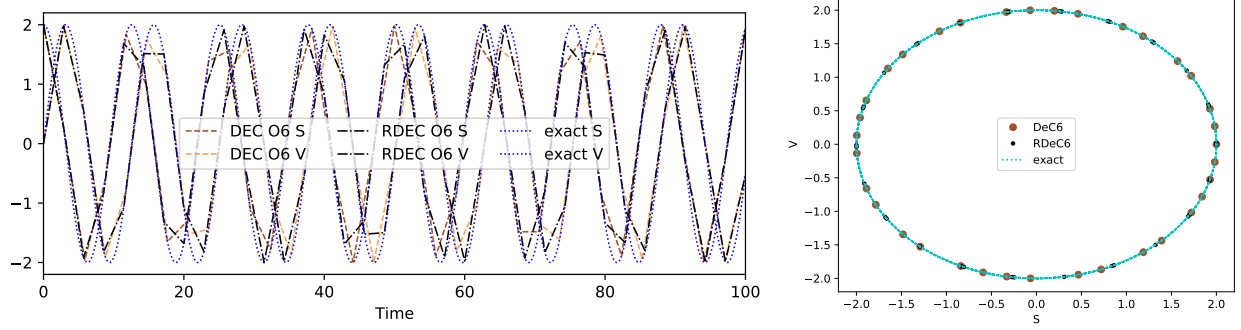
$$\begin{pmatrix} u_1(t) \\ u_2(t) \end{pmatrix} = n(t) \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix} \begin{pmatrix} u_1^0 \\ u_2^0 \end{pmatrix}, \quad n(t) = n(0)e^{-\alpha t}, \quad \theta(t) := \frac{1}{\alpha n(0)} (e^{\alpha t} - 1). \quad (62)$$

We take $\alpha = 0.01$. Here, we consider only the RDeC formulation and we do not compare it with RRK methods. However, similar results can be seen. This problem is much harder to be solved numerically as the oscillations have frequencies that increase with time and tends to infinity. So, we need to use many more timesteps. In figs. 5a and 5b we show the simulations and the phase values till time 100 for the RDeC and DeC of order 6 and 2 respectively. As before, qualitatively the high order methods are both precise, while the second order methods show large differences, DeC2 does not catch at all the decrease in time of the energy and hence the increase of frequency, while RDeC2 is much more accurate.

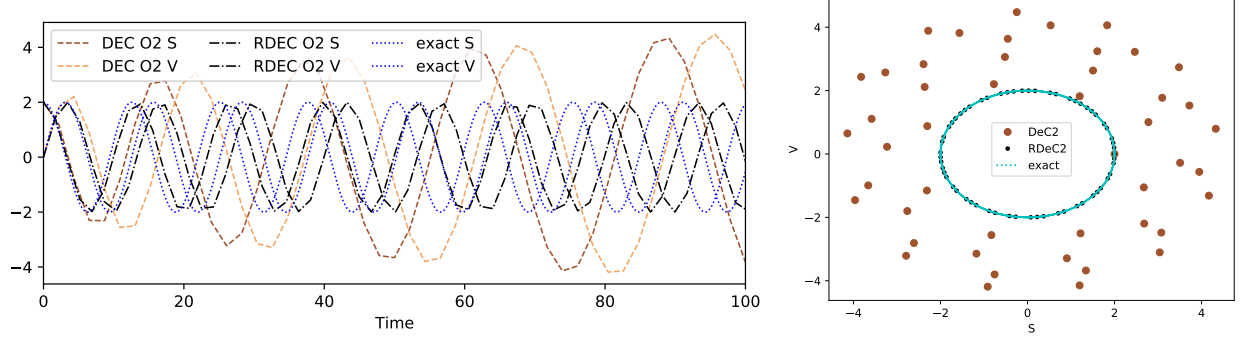
In fig. 5c we can observe the different errors of the energy from the exact one and clearly the high order methods are much more precise. Moreover, we can state that the relaxed DeCs are around one order of magnitude more precise in catching the energy level. In fig. 6 we observe similar superconvergence results as in the previous test.

5.1.3 Nonlinear Pendulum

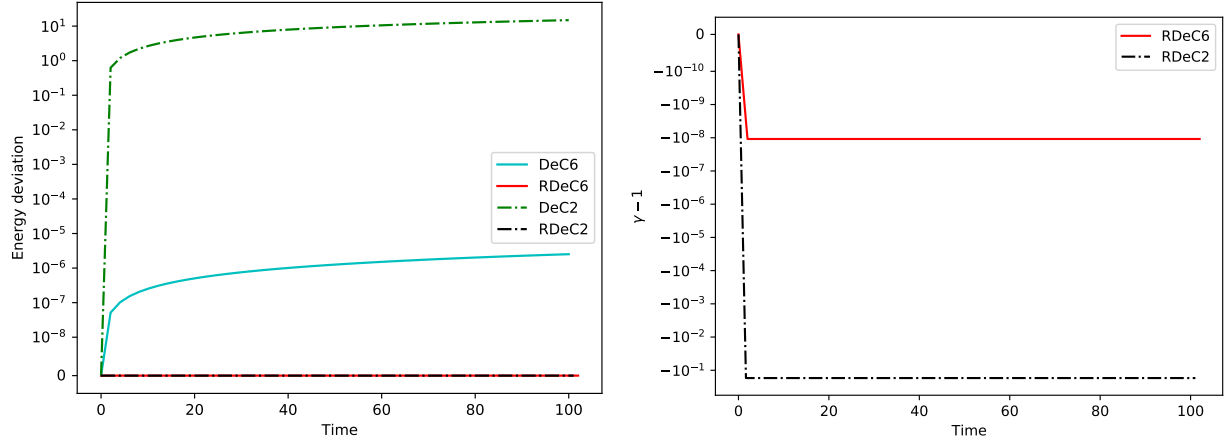
As another example from [38], we focus on the nonlinear pendulum. The trajectories of the pendulum are given by the system $\begin{bmatrix} u_1 \\ u_2 \end{bmatrix}' = \begin{bmatrix} -\sin(u_2) \\ u_1 \end{bmatrix}$ with the initial condition $\begin{bmatrix} u_1(0) \\ u_2(0) \end{bmatrix} = \begin{bmatrix} 1.5 \\ 0 \end{bmatrix}$. Here, we investigate the entropy $\eta(u) = \frac{1}{2}u_1^2 - \cos(u_2)$. Again the entropy progress over time is shown in fig. 7 for $\Delta t = 0.9$ and $T = 1000$. The entropy behaves as expected, namely it is not constant for non-relaxed schemes but zero up to machine precision for the relaxed schemes. Notice that without relaxation the entropy for SSPRK(3,3) is increasing whereas the entropy for DeC3 is decreasing and behaves similarly to the fourth order schemes. This explains also the different behaviors of the trajectories visible in fig. 8: the pendulum breaks out for SSPRK(3,3), goes to the center for DeC3 and stays within its path when the relaxation term is applied.



(a) Simulation for order 6 and $N = 35$: time evolution (left), phase space (right)



(b) Simulation for order 2 and $N = 47$: time evolution (left), phase space (right)



(c) Energy (left) and γ (right) comparison for RDeC orders 2 and 6 with $N = 50$

Figure 3: Simulation of the nonlinear oscillator for DeC and RDeC of orders 2 and 6

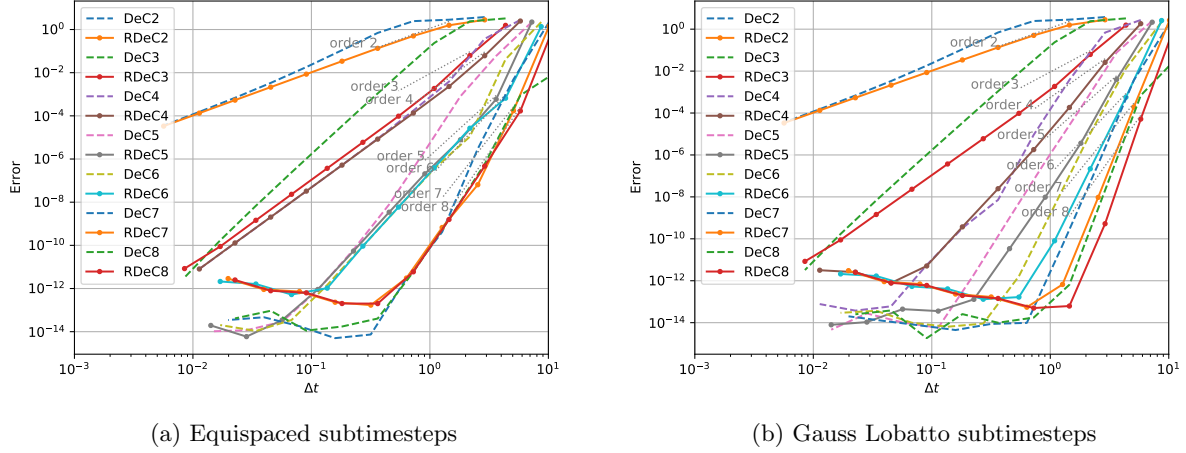


Figure 4: Convergence error of DeC and RDeC methods for nonlinear oscillator

Now, we look also at how the number of time steps are changing when applying the relaxation. If we define N as the amount of time steps needed to reach T_{end} we have $N = \frac{T_{end}}{\Delta t}$ when no relaxation is applied. However, when introducing the relaxation term, the time step size varies and so does N . This relation is shown in fig. 8. Without relaxation $N = \frac{1000}{0.9} \approx 1111$ and with relaxation we can see that this amount is increased if the average entropy residual term is smaller than 1 and is decreased if it is greater than 1 (DeC3). In fig. 8a, we use a boxplot to demonstrate the variation of the relaxation factors. We recognize the biggest amplitude for SSPRK(2,2)=DeC2. For higher order schemes, we notice that RK schemes has with respect to DeC schemes, in average, a larger variance in the γ_n . Moreover, DeC3 is the one which requires less time steps because of its γ_n often larger than 1, see fig. 8.

5.2 Numerical test in the PDE case

In this part, we validate the relaxed time integrators for hyperbolic conservation laws

$$\partial_t U + \text{div } F(U) = 0, \quad t \in [0, T], \quad (63)$$

where U are conservative variables and $F(U)$ is the flux function.

We proceed testing different approaches. First, we test the RDeC on an entropy conservative FV discretization for Burgers' equation, then we test the RDeC with the RD discretization. We check the accuracy of the RDeC-RD approach on a linear transport equation for the conservative case and we test on 2D advection and Burgers' equations.

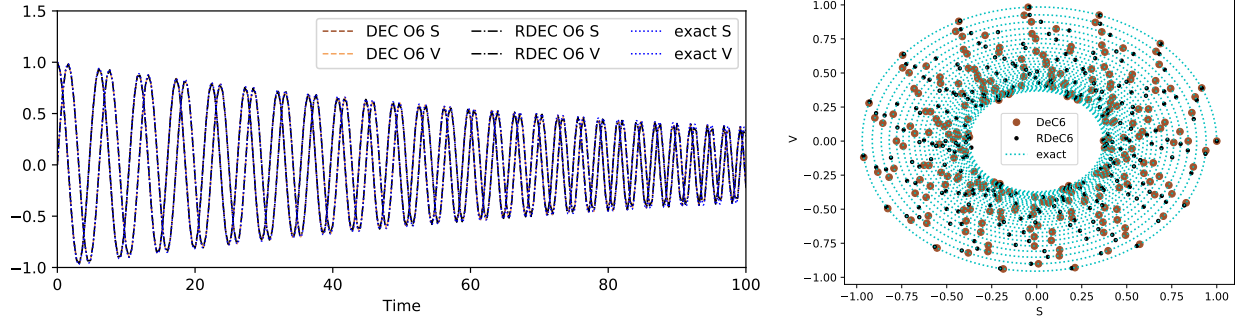
5.2.1 RDeC-FV for Burgers' Equation

In a first test case, we apply the RDeC just as a time integrator using a FV spatial discretization for Burgers' equation. This is an extension of the code by Ketcheson et al. [22, 33] with the RDeC approach. Here, we know that the spatial discretization is entropy conservative and, hence, we stop the simulation before shock formation.

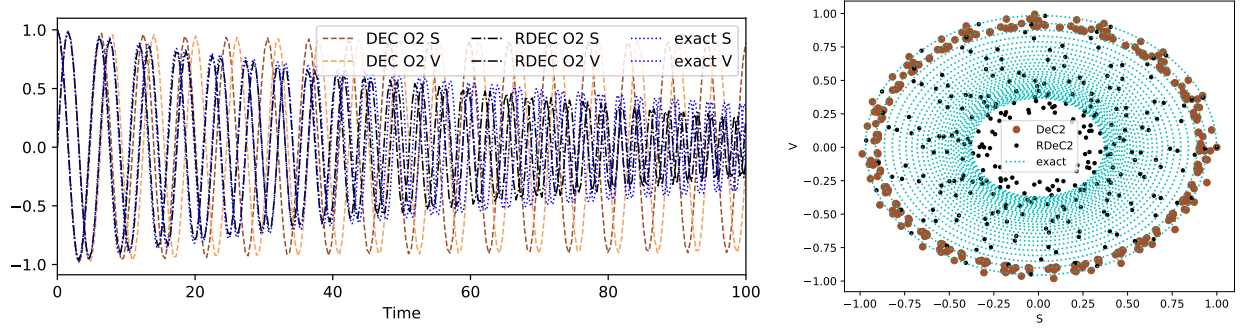
The inviscid Burgers' equation reads

$$U_t + \frac{1}{2}(U^2)_x = 0 \quad (64)$$

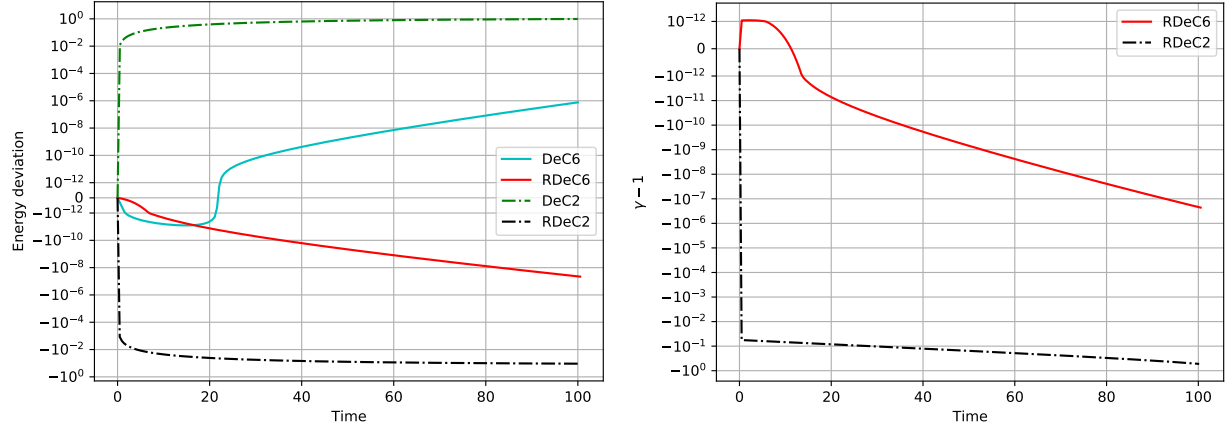
on the interval $-1 \leq x \leq 1$ with periodic boundary condition and the initial data $U(0, x) = \exp(-30x^2)$.



(a) Simulation for order 6 and $N = 250$: time evolution (left), phase space (right)



(b) Simulation for order 2 and $N = 250$: time evolution (left), phase space (right)



(c) Energy (left) and γ (right) comparison for RDeC orders 2 and 6 with $N = 50$

Figure 5: Simulation of the damped nonlinear oscillator for DeC and RDeC of orders 2 and 6

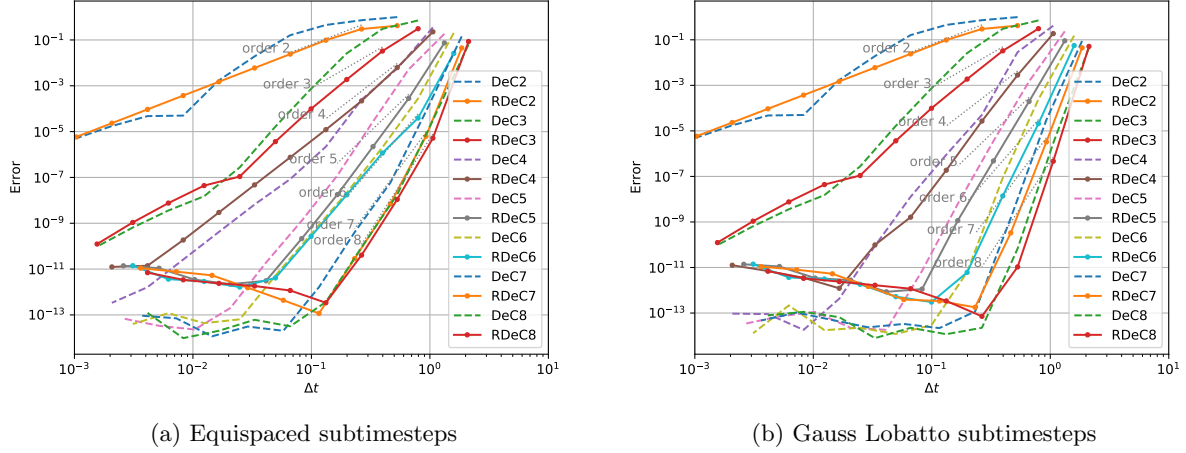


Figure 6: Convergence error of DeC and RDeC methods for nonlinear oscillator

For the space discretization, we use the flux differencing technique and obtain the semidiscretisation

$$u'_i(t) = \frac{1}{\Delta x}(F_{i+1/2} - F_{i-1/2}) \quad (65)$$

with the two-point numerical flux $F_{i+1/2} = \frac{u_i^2 + u_i u_{i+1} + u_{i+1}^2}{2}$. One can easily check that the spatial discretization is energy conservative, i.e.,

$$\sum_i u_i (F_{i+1/2}(u) - F_{i-1/2}(u)) = 0,$$

using periodic boundary conditions. The spatial domain is discretized with 100 equally-spaced points and the CFL number is set to 0.3. Again, for this test case the method of lines have been used and we are splitting between time and space discretization. In fig. 9, we plot the entropy evolution up to $t = 0.2$ for different time integration methods with and without relaxation. We notice that for SSPRK(2,2)=DeC2 entropy is produced in time whereas the rest the schemes are entropy dissipative. By applying the relaxation technique, the change of entropy for every scheme is of the order of machine precision. Finally, we like to point out that DeC methods have still better performance than classical RK methods in our simulations. This is due to the fact that DeC uses more stages than the considered RK methods. By increasing the number of stages in RK methods, we will obtain similar results.

Here, we only proved that the relaxation approach can be used in DeC methods together with classical space discretization methods like DG or FV approaches using the method of lines. In the next section, we apply the RDeC with RD where no method of lines is applicable.

5.2.2 Entropy Conservative/ Dissipative RDeC-RD Schemes - Linear Case

Now, we test the RDeC with the RD spatial discretization. The first test we take into consideration is the linear transport equation

$$\partial_t u + \partial_x u = 0, \quad (66)$$

on the domain $[0, 2]$ with periodic boundary condition and initial condition $u(t = 0, x) = 0.1 \sin(\pi x)$. Here, we aim at getting an entropy conservative scheme, as the problem is energy conservative. So, we simply impose (57) to find γ_n at each time step. The spatial discretization is obtained with cubature elements (Lagrangian polynomials on Gauss-Lobatto points) that we denote with \mathbb{C}^p with p the polynomial degree and the residuals are obtained with Galerkin discretization plus the jump stabilization term (31). In particular,

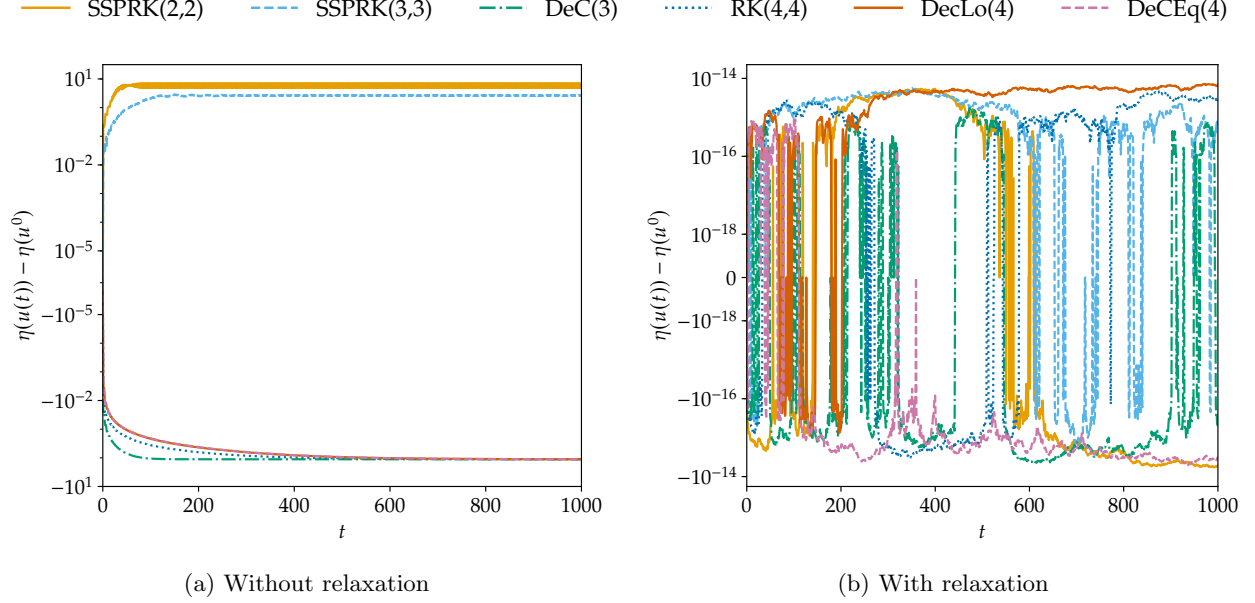


Figure 7: Entropy progression for nonlinear pendulum with $\Delta t = 0.9$

we use DeC^{p+1} in combination with \mathbb{C}^p polynomials to obtain a $p+1$ th order accurate scheme. In fig. 10a we observe that the error of the DeC does not decrease adding the relaxation procedure, while in fig. 10b the energy of the relaxation methods is of the order of machine precision, much lower with respect to classical methods where the entropy error is of the order of the spatial discretization.

In order to understand the size of the correction of the time step γ , in figure fig. 10c we plot $\gamma - 1$ as a function of time for the different methods. For second and third order schemes we increase the time step, while for fourth order we decrease it. Anyway, the order of $\gamma - 1$ is proportional to the accuracy of the scheme itself. Overall, we proved that the RDeC-RD method obtains the desired result.

After the one-dimensional set up, we extend our investigation to a two-dimensional rotation problem as it is also investigated in [6, 8]. We have the following problem:

$$\begin{aligned} \partial_t u(t, x, y) + \partial_x(2\pi y u(t, x, y)) + \partial_y(2\pi x u(t, x, y)) &= 0, \quad (x, y) \in D, t \in (0, 1), \\ u(0, x, y) &= u_0(x, y) = \exp(-40(x^2 + (y - 0.5)^2)), \quad (x, y) \in D, \end{aligned} \quad (67)$$

where D is the unit disk in \mathbb{R}^2 . For the boundary, outflow conditions are considered. For time integration, the second order DeC methods is used in the RD framework with CFL number 0.8. From the previous simulations, we deduce that the highest effect of the relaxation approach will be seen for low order methods. Therefore, we limit ourselves at DeC2 in the following. A continuous Galerkin scheme with entropy correction and jump stabilization (i.e. $\nu = 0.05$) of second order with Bernstein polynomials are applied as basis functions, see [3]. We want to remark that even if the residual is defined to be diffusive and entropy dissipative, thanks to its high order character, we were able to use the relaxation approach (57) gaining the exact entropy behavior⁴.

In this test, a small bump centered in $(0, 0.5)$ with radius 0.25 is moving around $(0, 0)$ in the circle D . The rotation is completed at $t = 1$. The mesh contains 3576 triangular elements. In table 1, the change in the energy is given after approximately half rotation and after one full rotation. We apply the relaxation

⁴We apply the same test also without jump stabilization. The results were quite similar to the ones presented. The only different was in the change of energy which has been closer to zero due to the semidiscrete entropy conservativity of the scheme (between $\approx 10^{-8} - 10^{-12}$).

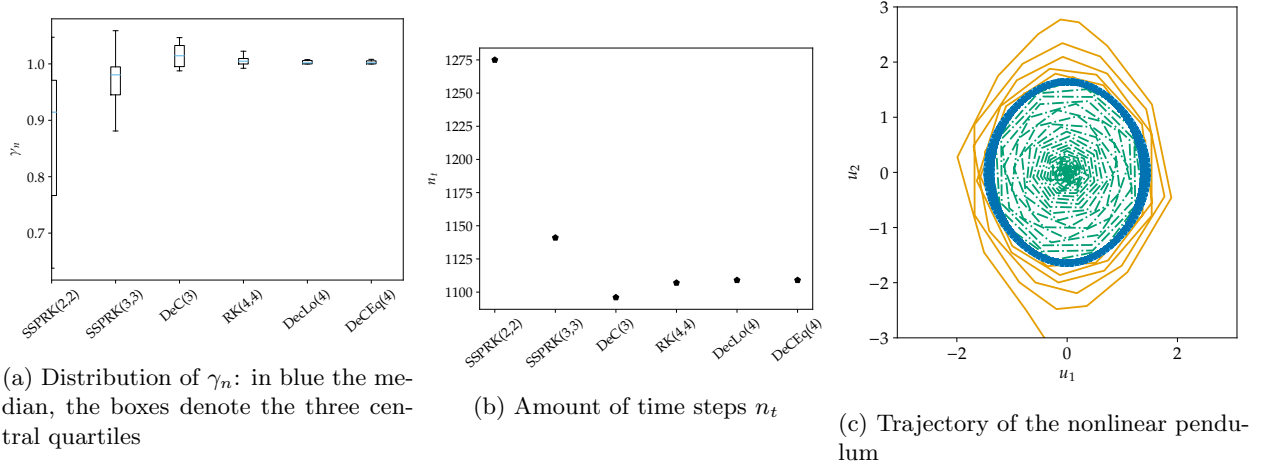


Figure 8: Several information for the nonlinear pendulum. $\Delta t = 0.9, T = 1000$.

Rotation	without relaxation	with relaxation
1/2	$-5.5864522681 \cdot 10^{-4}$	$1.7347234760 \cdot 10^{-17}$
1	$-1.0268559191 \cdot 10^{-3}$	$1.7961196366 \cdot 10^{-17}$

Table 1: Total energy change $\int_{\Omega} 0.5U^2(t) - \int_{\Omega} 0.5U_0^2$ of numerical solutions using a continuous Galerkin scheme for the linear test problem (67).

approach in every step and adapt the time step in respect to the entropy production/destruction. In case that the relaxation approach is used we need less steps to obtain the full rotation (505 to 544 time steps). Finally, we would like to remark that similar results have been observed using higher order approximations but, as mentioned before, the biggest effect on the entropy can be observed on low order approximations.

5.2.3 Entropy Conservative/ Dissipative RDeC-RD Schemes - Nonlinear Case

After this smooth test cases, we finally apply the relaxation approach on a nonlinear problem where actually a shock will appear. We test the two dimensional Burgers' equation

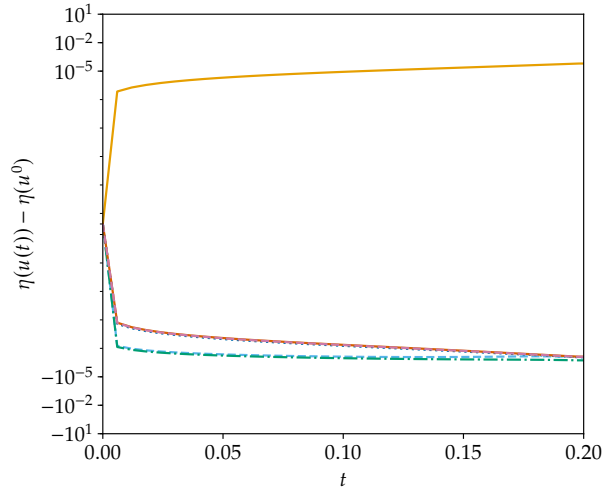
$$\begin{aligned} \partial_t u(t, x, y) + \partial_x(0.5u^2(t, x, y)) + \partial_y(u(t, x, y)) &= 0, \quad (x, y) \in D, t \in (0, 1), \\ u(0, x, y) &= u_0(x, y) = \exp(-40(x^2 + (y)^2)), \quad (x, y) \in D, \end{aligned} \quad (68)$$

where D is the unit disk in \mathbb{R}^2 and outflow boundary conditions are considered.

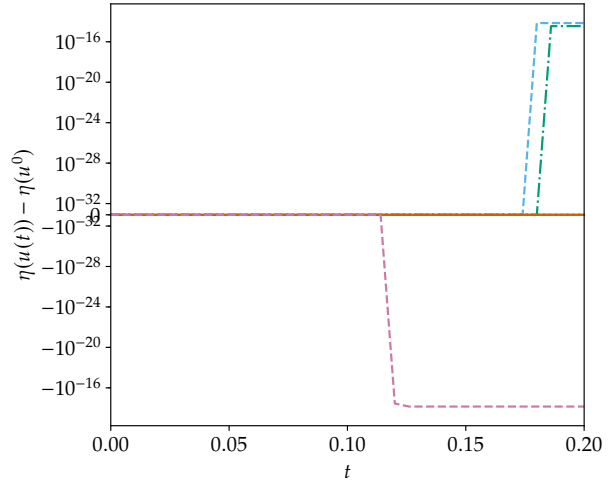
The DeC2 method is used with CFL number 0.35. We apply again a continuous Galerkin scheme with entropy correction and jump stabilization (i.e. $\nu = 0.1$) of second order with Bernstein polynomials on the same grid as before. We run our simulation until 0.5: after the shock formation. As explained before, we can either decide to be entropy dissipative in the definition of the residuals or entropy conservative, with the entropy correction. This problem has a dissipative nature. The entropy correction with the jump stabilization already has a dissipative nature. We test the relaxation algorithm starting both from the entropy correction (27) and the dissipative entropy correction (30). For the square entropy $U = u^2/2$, the differences $\int_{\Omega} U^2(0.5) - \int_{\Omega} U_0^2$ of the entropies with and without relaxation are given in table 2.

The results of the simulation can be seen in fig. 11. The left pictures demonstrated the result without relaxation while in the right picture relaxation has been used. We would like to point out that for this test case, we need not even half of the number of steps to get to endpoint when the relaxation approach has been used but this comes with some disadvantage. Before the shock forms the relaxation schemes are much more accurate, as they keep the total energy conserved as the entropy solution should, while the

— SSPRK(2,2)
 - - - SSPRK(3,3)
 - · - DeC(3)
 ····· RK(4,4)
 — DecLo(4)
 - - - DeCEq(4)



(a) Without relaxation



(b) With relaxation

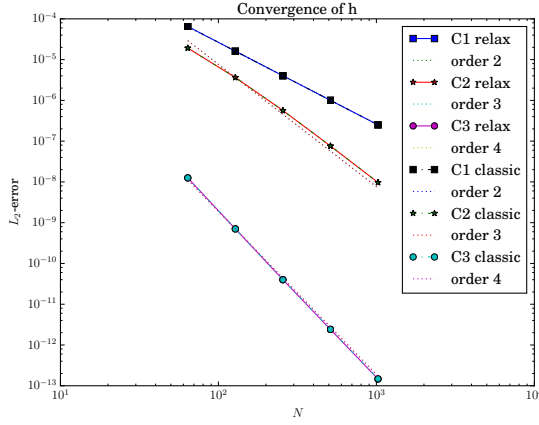
Figure 9: Entropy for Burgers with $\Delta t = 0.2$

dissipative approach of the entropy correction with jump stabilization already decreases widely this quantity, see table 2. Also the maximum value, which should stay constant before the shock formation, is decreased in the dissipative simulations thanks to the diffusive terms, while it is a bit increased in the entropy conservative simulation, as possible some dissipation is transformed into dispersion through the relaxation.

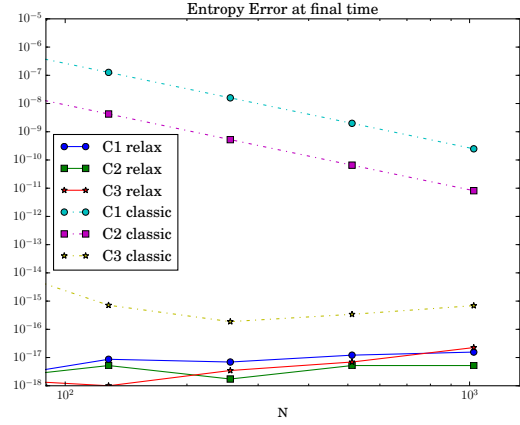
After the shock, the scheme with relaxation is more smeared and the shock profile is not sharp, as it does not converge to the entropy solution, forcing it to stay at the initial level. On the other side, the dissipative scheme is quite clear and correctly catches the shock structure. This is not surprising at all. Due the presence of a shock, a strict inequality is needed in the energy(entropy) equation, while, in the energy conservative approximation, the equality is enforced, violating the physics behind this test.

All in all, we can conclude that the relaxation DeC approach is working fine combined with the RD approach but special care has to be taken when shock appears.

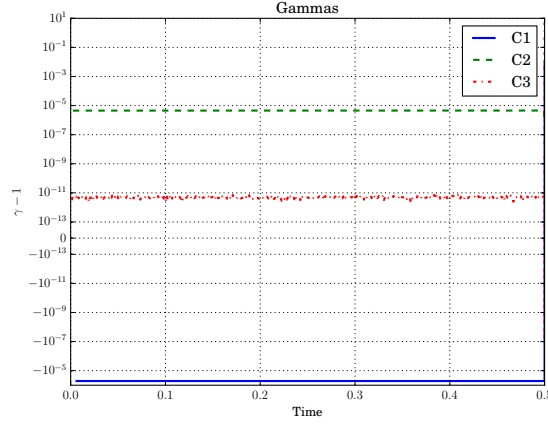
Remark 5.1 (Comparison to [8]). *Up to this point, we have considered the relaxation approach only for the square entropy to avoid the usage of a nonlinear solver in our RD code. An extension to the Euler equations will be considered in the future as well as a comparison with the approach presented in [8] where the entropy*



(a) Error convergence



(b) Error of the entropy



(c) $\gamma_n - 1$ as a function of time for RDeC-RD with $N = 256$ spatial cell discretization

Figure 10: Errors convergence for linear transport problem with DeC/RDeC-RD approach

correction term was used not only for the space discretization but for the whole space-time residual. From first numerical results, we do not want to hide that the approach using the entropy correction term completely seems more promising and completely general as a combination of entropy correction plus relaxation. Also no optimization problem has to be solved when general entropies are considered. A deeper analysis including more challenging tests will be considered in the future to investigate this topic. This is beyond the scope of this paper where we concentrate on the relaxation DeC approach and the application in RD.

6 Conclusion

In this work, we extended the relaxation technique [22] to the arbitrarily high order DeC time integration method, in particular in its applications to RD where the MOL is not applicable. In this context, a spatial entropy preserving discretization is available [3] and its combination with the relaxation algorithm allows to obtain a global entropy conservative or dissipative scheme. The whole procedure require some choices, for example in the construction of the equation that we want to set equal to 0 to find the relaxation parameter γ_n . Putting together all these ingredients one can obtain an entropy conservative or entropy dissipative

Final time	Entropy Correction	+ Jump	+ Relaxation	+ Relaxation + Jump
0.22	$-1.97 \cdot 10^{-4}$	$-1.51 \cdot 10^{-3}$	$-2.43 \cdot 10^{-17}$	$-3.46 \cdot 10^{-18}$
0.5	$-1.51 \cdot 10^{-3}$	$-3.64 \cdot 10^{-3}$	$1.39 \cdot 10^{-17}$	$-3.12 \cdot 10^{-17}$

Table 2: Entropy variation before and after the shock

Final time	Entropy Correction	+ Jump	+ Relaxation	+ Relaxation + Jump
0.22	≈ 1.1	≈ 0.93	≈ 1.2	≈ 1.0
0.5	≈ 1.3	≈ 0.83	≈ 1.4	≈ 1.1

Table 3: Maximum of u before and after the shock

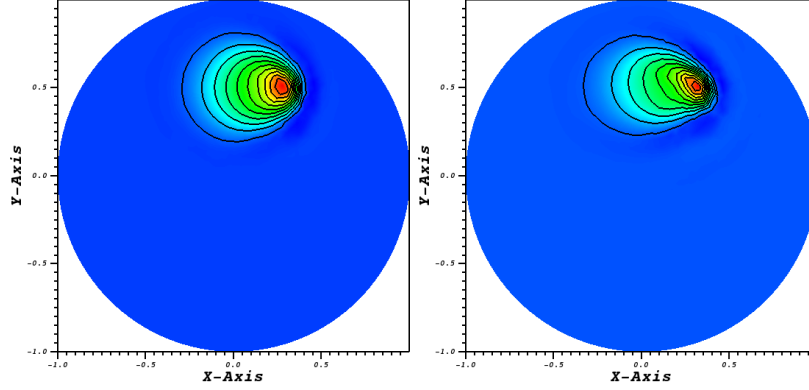


Figure 11: $t = 0.5$, DeC(2), Bernstein polynomials, 3576 elements, CFL=0.35, left entropy correction + jump, right relaxation + entropy correction + jump

arbitrarily high order matrix-free method to solve general hyperbolic PDEs.

This topic can be expanded in different directions, first of all more general tests with nonlinear entropies could be studied, for example in Euler's equations, or a transition between the conservative and dissipative regime could be thought and implemented in order to be used on more general cases where a priori it is unknown the nature of the problem. Finally, a deeper analysis between the presented approach and the one from [8] is also desirable.

Acknowledgements

E. Le Méleto has been funded by the the SNF project (Number 175784). P. Öffner has also been funded by the UZH Postdoc Forschungskredit (Number FK-19-104), the SNF project (Number 175784) and the Gutenberg Fellowship (University Mainz). D. Torlo has been funded by Team CARDAMOM in INRIA Bordeaux - Sud-Ouest. P. Öffner likes to thank Vinzenz Muser for some early discussions about the topic and H. Ranocha for fruitful conversations about the relaxation technique. RA was partially funded by an International chair, Inria Bordeaux Sud-Ouest.

A Another possible relaxation formulation

The relaxation DeC RD presented before is not unique. There, different possibilities related to the weighting of Δt of \mathcal{L}^2 or both \mathcal{L}^1 and \mathcal{L}^2 . This does not affect the ODE case, but in the PDE case there are some differences. Here, we modify only \mathcal{L}^2 . Let us restart from the formulation for the final update step, i.e.,

$$U_{\sigma}^{n,l,(k)} = U_{\sigma}^{n,l,(k)} - |C_{\sigma}|^{-1} \sum_{K|\sigma \in K} \left(\oint_K \varphi_{\sigma}(U^{n,l,(k)} - U^{n,0}) dx + \Delta t \sum_{r=0}^M \theta_r^l \Phi_{\sigma,x}^K(U^{n,r,(k)}) \right). \quad (32)$$

Focusing on the energy for simplicity, the relaxation term has been given by (11) and was calculated by determining the energy production in RK schemes. In the RD framework, we cannot simply apply this term since by focusing on (32), we realize that we have additional terms in the update which are not even multiplied by the time step Δt . Therefore, we compare first the change of the energy between two time steps using (32). It is given by the following calculation on one degree of freedom⁵:

$$\left\| U_{\sigma}^{M,(K)} \right\|^2 - \left\| U_{\sigma}^0 \right\|^2 = \left(U_{\sigma}^{M,(K-1)} - |C_{\sigma}|^{-1} \sum_{K|\sigma \in K} \left(\int_K \varphi_{\sigma}(U_{\sigma}^{M,(K-1)}) - U^0 dx + \Delta t \sum_{r=0}^M \theta_r^M \Phi_{\sigma,x}^K(U_{\sigma}^{r,(K-1)}) \right) \right)^2 - (U_{\sigma}^0)^2.$$

We can reorder the equation and get

$$= \left(\left(|C_{\sigma}|^{-1} \sum_{K|\sigma \in K} \int_K \varphi_{\sigma} U^0 dx \right) + \left(U_{\sigma}^{M,(K-1)} - |C_{\sigma}|^{-1} \sum_{K|\sigma \in K} \int_K \varphi_{\sigma} U^{M,(K-1)} dx \right) + |C_{\sigma}|^{-1} \Delta t \sum_{K|\sigma \in K} \sum_{r=0}^M \theta_r^M \Phi_{\sigma,x}^K(U_{\sigma}^{r,(K-1)}) \right)^2 - (U_{\sigma}^0)^2.$$

Here, the first term describes an approximation of U^0 , the second term is some approximation of $U_{\sigma}^{M,(K-1)}$ and the rest is the update scheme. We apply in the following the abbreviations

$$A = \left(|C_{\sigma}|^{-1} \sum_{K|\sigma \in K} \int_K \varphi_{\sigma} U^0 dx \right) \quad (69)$$

$$B := \left(U_{\sigma}^{M,(K-1)} - |C_{\sigma}|^{-1} \sum_{K|\sigma \in K} \int_K \varphi_{\sigma} U^{M,(K-1)} dx \right) \quad (70)$$

$$C = |C_{\sigma}|^{-1} \Delta t \sum_{K|\sigma \in K} \sum_{r=0}^M \theta_r^M \Phi_{\sigma,x}^K(U_{\sigma}^{r,(K-1)}) \quad (71)$$

We can now focus again on above equation and get

$$\begin{aligned} &= (A + B + C)^2 - (U_{\sigma}^0)^2 = (A^2 + 2AB + 2AC + B^2 + 2BC + C^2) - (U_{\sigma}^0)^2 \\ &= \underbrace{A^2 - (U_{\sigma}^0)^2 + 2AB + B^2 + 2BC + 2AC + C^2}_D. \end{aligned} \quad (72)$$

The term D does not depend on Δt but depends on the used quadrature formula in space. AC and BC depends on Δt through C and C^2 behaves with Δt^2 . We focus now on the $AC + BC$ and get

$$\begin{aligned} &|C_{\sigma}|^{-1} \left(\sum_{K|\sigma \in K} \sum_{r=0}^M \theta_r^M \Phi_{\sigma,x}^K(U_{\sigma}^{r,(K-1)}) \right) \left(U_{\sigma}^{M,(K-1)} - |C_{\sigma}|^{-1} \sum_{K|\sigma \in K} \int_K \varphi_{\sigma} (U_{\sigma}^{M,(K-1)} - U^0) dx \right) \\ &= |C_{\sigma}|^{-1} \sum_{r=0}^M \theta_r^M \sum_{K|\sigma_1 \in K} \left\langle \Phi_{\sigma_1}^K(U_{\sigma}^{r,(K-1)}), \left(U_{\sigma}^{M,(K-1)} - |C_{\sigma}|^{-1} \sum_{K|\sigma \in K} \int_K \varphi_{\sigma} (U_{\sigma}^{M,(K-1)} - U^0) \right) + (U_{\sigma_1}^{r,(K-1)} - U_{\sigma_1}^{r,(K-1)}) \right\rangle \\ &= |C_{\sigma}|^{-1} \sum_{r=0}^M \theta_r^M \left(\sum_{K|\sigma_1 \in K} \left\langle \Phi_{\sigma_1}^K(U_{\sigma}^{r,(K-1)}), U_{\sigma_1}^{r,(K-1)} \right\rangle \right) \\ &\quad + \underbrace{|C_{\sigma}|^{-1} \sum_{r=0}^M \theta_r^M \left(\sum_{K|\sigma_1 \in K} \left\langle \Phi_{\sigma_1}^K(U_{\sigma}^{r,(K-1)}), \left(U_{\sigma}^{r,(K-1)} - |C_{\sigma}|^{-1} \sum_{K|\sigma \in K} \int_K \varphi_{\sigma} (U_{\sigma}^{r,(K-1)} - U^0) \right) - U_{\sigma_1}^{r,(K-1)} \right\rangle \right)}_{0.5E} \end{aligned}$$

⁵For simplicity, we avoid the usage of n in the following.

The first term will cancel out if our space residual is energy conservative⁶ where the second term yields some rest to the equation. Here, the braces depend highly on the used quadrature rule. We can obtain a recurrence relation inserting the corrections for the terms. Nevertheless, the remaining term is at least $\mathcal{O}(\Delta t)$. Therefore, we have now in total for the energy production $D + E + C^2$. Using now the relaxation approach, we can multiply with a γ_n our θ . Here, γ_n will be the solution of the following equation

$$D + \gamma_n E + \gamma_n^2 C^2 = 0.$$

Actually, γ_n will be always positive and close to one if our quadrature rule is sufficiently accurate. With this approach we obtain that our DeC-RD approach is energy conservative (dissipative) in space and time.

References

- [1] R. Abgrall. A review of residual distribution schemes for hyperbolic and parabolic problems: the july 2010 state of the art. *Communications in Computational Physics*, 11(4):1043–1080, 2012.
- [2] R. Abgrall. High order schemes for hyperbolic problems using globally continuous approximation and avoiding mass matrices. *Journal of Scientific Computing*, 73(2):461–494, Dec 2017.
- [3] R. Abgrall. A general framework to construct schemes satisfying additional conservation relations. application to entropy conservative and entropy dissipative schemes. *Journal of Computational Physics*, 372:640–666, 2018.
- [4] R. Abgrall, P. Bacigaluppi, and S. Tokareva. High-order residual distribution scheme for the time-dependent Euler equations of fluid dynamics. *Computers & Mathematics with Applications*, 78(2):274–297, 07 2019.
- [5] R. Abgrall, E. I. Meledo, and P. Öffner. On the connection between residual distribution schemes and flux reconstruction. *arXiv preprint arXiv:1807.01261*, 2018.
- [6] R. Abgrall, J. Nordström, P. Öffner, and S. Tokareva. Analysis of the SBP-SAT stabilization for finite element methods part I: Linear problems. *Journal of Scientific Computing*, 85(2):1–29, 2020.
- [7] R. Abgrall, J. Nordström, P. Öffner, and S. Tokareva. Analysis of the SBP-SAT stabilization for finite element methods part II: Entropy stability. *Communications on Applied Mathematics and Computation*, pages 1–23, 2021.
- [8] R. Abgrall, P. Öffner, and H. Ranocha. Reinterpretation and extension of entropy correction terms for residual distribution and discontinuous Galerkin schemes. *arXiv:1908.04556*, 2019.
- [9] T. Chen and C.-W. Shu. Entropy stable high order discontinuous Galerkin methods with suitable quadrature rules for hyperbolic conservation laws. *Journal of Computational Physics*, 345:427–461, 2017.
- [10] A. Christlieb, B. Ong, and J.-M. Qiu. Integral deferred correction methods constructed with high order Runge-Kutta integrators. *Mathematics of Computation*, 79(270):761–783, 2010.
- [11] G. Cohen, P. Joly, J. Roberts, and N. Tordjman. Higher order triangular finite elements with mass lumping for the wave equation. *Siam Journal on Numerical Analysis - SIAM J NUMER ANAL*, 38, 01 2001.
- [12] A. Dutt, L. Greengard, and V. Rokhlin. Spectral Deferred Correction Methods for Ordinary Differential Equations. *BIT Numerical Mathematics*, 40(2):241–266, 2000.
- [13] L. Ferracina and M. Spijker. An extension and analysis of the Shu-Osher representation of Runge-Kutta methods. *Mathematics of Computation*, 74(249):201–219, 2005.
- [14] J. Glaubitz and P. Öffner. Stable discretisations of high-order discontinuous Galerkin methods on equidistant and scattered points. *Applied Numerical Mathematics*, 151:98–118, 2020.
- [15] J. Glaubitz, P. Öffner, H. Ranocha, and T. Sonar. Artificial viscosity for correction procedure via reconstruction using summation-by-parts operators. In *XVI International Conference on Hyperbolic Problems: Theory, Numerics, Applications*, pages 363–375. Springer, 2016.
- [16] S. Gottlieb, D. I. Ketcheson, and C.-W. Shu. *Strong stability preserving Runge-Kutta and multistep time discretizations*. World Scientific, 2011.
- [17] S. Gottlieb and C.-W. Shu. Total variation diminishing Runge-Kutta schemes. *Mathematics of computation of the American Mathematical Society*, 67(221):73–85, 1998.

⁶For an energy dissipative scheme, we have the right sign in this term and it has not further considered.

- [18] S. Gottlieb, C.-W. Shu, and E. Tadmor. Strong stability-preserving high-order time discretization methods. *SIAM review*, 43(1):89–112, 2001.
- [19] M. Han Veiga, P. Öffner, and D. Torlo. DeC and ADER: Similarities, Differences and a Unified Framework. *Journal of Scientific Computing*, 87(1):1–35, 2021.
- [20] A. Harten. On the symmetric form of systems of conservation laws with entropy. *Journal of computational physics*, 49:151–164, 1983.
- [21] J. Huang and C.-W. Shu. Positivity-preserving time discretizations for production–destruction equations with applications to non-equilibrium flows. *Journal of Scientific Computing*, 78(3):1811–1839, 2019.
- [22] D. Ketcheson. Relaxation Runge–Kutta methods: Conservation and stability for inner-product norms. *SIAM Journal on Numerical Analysis*, 57(6):2850–2870, 2019.
- [23] D. I. Ketcheson. Highly efficient strong stability-preserving Runge–Kutta methods with low-storage implementations. *SIAM Journal on Scientific Computing*, 30(4):2113–2136, 2008.
- [24] D. I. Ketcheson, C. B. Macdonald, and S. Gottlieb. Optimal implicit strong stability preserving Runge–Kutta methods. *Applied Numerical Mathematics*, 59(2):373–392, 2009.
- [25] D. Kuzmin, M. Q. de Luna, D. I. Ketcheson, and J. Grüll. Bound-preserving convex limiting for high-order Runge–Kutta time discretizations of hyperbolic conservation laws. *arXiv preprint arXiv:2009.01133*, 2020.
- [26] D. Levy and E. Tadmor. From semidiscrete to fully discrete: Stability of Runge–Kutta schemes by the energy method. *SIAM review*, 40(1):40–73, 1998.
- [27] Y. Liu, C.-W. Shu, and M. Zhang. Strong stability preserving property of the deferred correction time discretization. *Journal of Computational Mathematics*, pages 633–656, 2008.
- [28] A. Meister and S. Ortleb. On unconditionally positive implicit time integration for the DG scheme applied to shallow water flows. *International Journal for Numerical Methods in Fluids*, 76(2):69–94, 2014.
- [29] S. Nüßlein, H. Ranocha, and D. I. Ketcheson. Positivity-preserving adaptive Runge–Kutta methods. *arXiv preprint arXiv:2005.06268*, 2020.
- [30] P. Öffner. *Approximation and Stability properties of Numerical Methods for Hyperbolic Conservation laws*. habilitation, University Zurich, 2020.
- [31] P. Öffner, J. Glaubitz, and H. Ranocha. Analysis of artificial dissipation of explicit and implicit time-integration methods. *International Journal of Numerical Analysis and Modeling*, 17 (3):332–349, 12 2020.
- [32] P. Öffner and D. Torlo. Arbitrary high-order, conservative and positivity preserving patankar-type deferred correction schemes. *Applied Numerical Mathematics*, 153:15–34, 2020.
- [33] H. Ranocha and D. I. Ketcheson. Relaxation Runge–Kutta methods for inner-product norms. https://github.com/ketch/RRK_rr, 05 2019.
- [34] H. Ranocha and D. I. Ketcheson. Relaxation Runge–Kutta Methods for Hamiltonian Problems. *Journal of Scientific Computing*, 84(1):1–27, 2020.
- [35] H. Ranocha, L. Lóczi, and D. I. Ketcheson. General relaxation methods for initial-value problems with application to multistep schemes. *Numerische Mathematik*, 146(4):875–906, 2020.
- [36] H. Ranocha and P. Öffner. L_2 stability of explicit Runge–Kutta schemes. *Journal of Scientific Computing*, 75(2):1040–1056, 05 2018.
- [37] H. Ranocha, P. Öffner, and T. Sonar. Summation-by-parts operators for correction procedure via reconstruction. *Journal of Computational Physics*, 311:299–328, 2016.
- [38] H. Ranocha, M. Sayyari, L. Dalcin, M. Parsani, and D. I. Ketcheson. Relaxation Runge–Kutta methods: Fully discrete explicit entropy-stable schemes for the compressible Euler and Navier–Stokes equations. *SIAM Journal on Scientific Computing*, 42(2):A612–A638, 2020.
- [39] C.-W. Shu and S. Osher. Efficient implementation of essentially non-oscillatory shock-capturing schemes. *Journal of computational physics*, 77(2):439–471, 1988.
- [40] D. Torlo. *Hyperbolic Problems: High Order Methods and Model Order Reduction*. PhD thesis, University Zurich, 2020.