Relaxation Deferred Correction Methods and their Applications to Residual Distribution Schemes

R. Abgrall [(1)], E. Le Méledo[(1)], P. Öffner[(3)*] and D. Torlo[(2)]

(1): Institute of Mathematics, University of Zurich, Switzerland

(2): SISSA Mathlab, Trieste, Italy

(3): Institute of Mathematics, Johannes Gutenberg-University Mainz, Germany

March 28, 2022

Abstract

In [2] has been introduced a simplified Deferred Correction (DeC) approach, which, combined with the residual distribution (RD) approach, allows to construct high order continuous Galerkin (cG) methods. One of their main feature is to avoid the inversion of a mass matrix. Thanks to the entropy correction function proposed in [3], one can obtain an entropy conservative/dissipative spatial discretization. The relaxation approach introduced in [26] allows to construct entropy conservative/dissipative time discretization methods. We first study the relaxation technique applied to the DeC as an ODE solver, then we extend this combination to the residual distribution method, requiring more technical steps. The outcome is a class of cG methods that is fully entropy conservative/dissipative and where we can avoid the inversion of a mass matrix.

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 properties of the underlying problem. To keep the positivity, for example, Patankar approaches [36, 25, 30] or adaptive/limiting strategies [28, 33] can be found in the literature, while, recently, Ketcheson proposed relaxation Runge-Kutta (RRK) methods to guarantee conservation or dissipation of any inner-product norm, e.g. energy, entropy and Ljapunov functionals. 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. [26, 39, 38, 41]. Special attention has been given on entropy conservation/dissipation for hyperbolic PDEs in [41] where the relaxation approach is combined with a semidiscrete entropy conservative/dissipative scheme via a simple method of lines (MOL).

In this work, we also build an entropy conservative/dissipative scheme using the relaxation approach, but we apply it on the deferred correction (DeC) method. The DeC is based on the Picard-Lindelöf theorem and gives a simple algorithm to construct arbitrarily high order schemes. As time integration schemes DeC methods can be rewritten as RK schemes, and, therefore, the relaxation approach can be straightforwardly applied also to them. Nevertheless, in the context of time-dependent PDEs, DeC has been recently combined with the residual distribution (RD) method [2, 4, 10, 11] resulting in a high order explicit finite element method which avoids the inversion of a mass matrix. Moreover, in [3] it has been introduced a correction term that allows to preserve the total entropy in the spatial RD 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 character of the DeC RD scheme avoiding the inversion of a mass matrix.

*Corresponding author: mail@philippoeffner.de or p.oeffner@uni-mainz.de

Therefore, the paper is organized as follows: in section 2, we review the numerical methods on which the novel method is based. We start with the relaxation Runge–Kutta method introduced by Ketcheson [26], the DeC as time integration method [19] in the formulation presented by Abgrall [2] and its combination with RD for the solution of hyperbolic problems. We also recall the construction of entropy conservative/dissipative RD schemes for steady state problems. In section 3, we focus on DeC for ODEs and on how the application of the relaxation technique acts on it. Since the DeC can be interpreted as an RK scheme for ODE problems (or on PDE problems with the MOL approach), the relaxation approach can be directly transferred to the DeC method. Nevertheless, we need to prove analogous results also for the DeC framework of [2] as this should lead to a better understanding of the relaxation approach and to extend the algorithm to the RD-DeC method. Since the RD-DeC method is not a MOL, we need to prove the high order accuracy of the method in this different framework. In section 4, we demonstrate how to combine the relaxation DeC (RDeC) with the semidiscrete entropy conservative/dissipative RD method of [3] 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

In this work we consider numerical approximations of both ODEs of the type

$$\begin{cases} \frac{dy(t)}{dt} = f(t,y(t)), \quad t \in [0,T], \quad y : [0,T] \to \mathbb{R}^I, \quad f : [0,T] \times \mathbb{R}^I \to \mathbb{R}^I, \\ y(0) = y_0, \end{cases} \tag{2.1}$$

and of hyperbolic conservation laws, i.e.,

$$\begin{cases} \frac{\partial U}{\partial t} + \operatorname{div} F(U) = 0, & x \in \Omega \subset \mathbb{R}^d, \quad t \in [0,T], \quad U : \Omega \times [0,T] \to \mathbb{R}^I, \quad F : \mathbb{R}^I \to \mathbb{R}^{I \times d}, \\ U(x,0) = U_0(x), & x \in \Omega \end{cases} \tag{2.2}$$

where Ω is the spatial domain. In many numerical methods the temporal discretization of both ODEs and PDEs can be obtained in similar manners. When a numerical method for PDEs splits the spatial and the temporal discretization through the method of lines, we can use ODE solvers on the partially discretized PDE to obtain a global solver. This is not always the case. In particular, the DeC method, firstly introduced to solve ODEs [19, 32, 36], has been used to construct arbitrarily high order methods for hyperbolic equations in [2] combining spatial (residual distribution) and temporal discretization not resulting in a MOL but obtaining a continuous finite element formulation which avoids the inversion of a mass matrix.

The goal of this work is to obtain an entropy preserving/dissipative arbitrarily high order method for ODEs and hyperbolic PDEs. Hence, we will combine the relaxation approach [26] with the DeC and, for PDEs, with an entropy stable spatial discretization [3]. In this section, we review all the used methods and we highlight their key features: Relaxation Runge–Kutta, DeC, RD and DeC-RD.

2.1 Relaxation Runge–Kutta Methods

There are various approaches to solve numerically the ODE (2.1). 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 standard tools for solving ODEs.

Let us divide the time interval $[0,T]$ into N segments $T_n = [t^n, t^{n+1}]$ and the time steps are given by $\Delta t^n = t^{n+1} - t^n$.

A Runge-Kutta (RK) method applied to (2.1) approximates the variable in the time steps, i.e., $y^n \approx y(t^n)$, in the form

$$\begin{aligned} u_i &:= y^n + \Delta t \sum_{j=1}^{s} a_{ij} f(t^n + c_j \Delta t, u_j), \quad i = 1, \dots, s, \\ y(t^n + \Delta t) \approx y^{n+1} &:= y^n + \Delta t \sum_{j=1}^{s} b_j f(t^n + c_j \Delta t, u_j). \end{aligned} \tag{2.3}$$

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 derivative. The coefficients of the RK method can also be written into a Butcher tableau of the form

$$\begin{array}{c|c} c & A \\ \hline & b \end{array} \tag{2.4}$$

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

To explain the relaxation approach, we first follow the spirit of Ketcheson [26] and explain the basic framework. For simplicity, we concentrate again only on the initial value problem (2.1). We focus on problems which are *dissipative (conservative)* with respect to some inner-product norm, i.e.,

$$\frac{\mathrm{d}}{\mathrm{d}t} \|y(t)\|^2 = 2 \langle y, f(t,y) \rangle \overset{(=)}{\leq} 0, \tag{2.5}$$

where the equality sign is used for conservative problems. Here, $\langle \cdot, \cdot \rangle$ denotes an inner product and $\|\cdot\|$ the corresponding norm. For dissipative (conservative) problems, it is desirable that the numerical solution verifies (2.5) discretely, i.e.,

$$\left\|y^{n+1}\right\| \overset{(=)}{\leq} \|y^n\|. \tag{2.6}$$

A method is called *monotonicity preserving* if it guarantees (2.6) for all problems satisfying (2.5).

Remark 2.1. The term $\|y\|^2$ is will be called *energy* in the following as in [26] 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 [41], the relaxation approach is extended to general convex quantities, i.e., general entropies. We start the discussion focusing on the energy but we will extend it to general entropies afterwards.

As it is shown in [21, 26, 35] the change of the energy between two steps is given by

$$\begin{aligned} \left\|y^{n+1}\right\|^2 - \|y^n\|^2 &= 2\Delta t \sum_{j=1}^{s} b_j \langle u_j, f_j \rangle + 2\Delta t \sum_{j=1}^{s} b_j \langle y^n - u_j, f_j \rangle + \Delta t^2 \sum_{i,j=1}^{s} b_i b_j \langle f_i, f_j \rangle \\ &\overset{(2.3)}{=} 2\Delta t \sum_{j=1}^{s} b_j \langle u_j, f_j \rangle - 2\Delta t^2 \sum_{j,i=1}^{s} a_{ji} b_j \langle f_j, f_i \rangle + \Delta t^2 \sum_{i,j=1}^{s} b_i b_j \langle f_i, f_j \rangle \end{aligned}$$

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 $b_j \geq 0$ because of (2.5). 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 so that the remaining terms are canceling out. To obtain this, a relaxation coefficient γ_n is introduced as a factor of the time step only in the final update formula of y^{n+1}, so that the RK update reads

$$y_{\gamma_n}^{n+1} = y^n + \gamma_n \Delta t \sum_{j=1}^{s} b_j f(t_n + c_j \Delta t, u_j), \tag{2.7}$$

and the energy difference becomes

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

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

$$\gamma_n = \frac{2\sum_{j,i=1}^{s} a_{ji} b_j \langle f_j, f_i \rangle}{\sum_{i,j=1}^{s} b_i b_j \langle f_i, f_j \rangle}, \tag{2.8}$$

while the second root is $\gamma_n = 0$ and is not further considered. If the denominator of (2.8) vanishes, we already have that $y^{n+1} = y^n$ and we achieve our goal (i.e., conservation) by taking $\gamma_n = 1$. Thus, we define

$$\gamma_n := \begin{cases} \frac{2\sum_{j,i=1}^{s} a_{ji} b_j \langle f_j, f_i \rangle}{\sum_{i,j=1}^{s} b_i b_j \langle f_i, f_j \rangle}, & \text{if } ||\sum_{i=1}^{s} b_i f_i||^2 \neq 0, \\ 1, & \text{else.} \end{cases} \tag{2.9}$$

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 formulates the following lemma [26, Lemma 1]:

Lemma 2.2. *Let $\sum_{i,j} b_i a_{ij} > 0$, let f be sufficiently smooth, and let γ_n be defined by* (2.9)*. 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 $\sum_{i,j} b_i a_{ij} = \frac{1}{2}$.

In case of a general entropy we can proceed analogously. Let us denote with $\eta : \mathbb{R}^I \to \mathbb{R}$ an entropy for (2.1) and $w : \mathbb{R}^I \to \mathbb{R}^I$ being $w(y) = \partial_y \eta(y)$ such that

$$\langle w(y), f(t,y) \rangle \overset{(=)}{\leq} 0. \tag{2.10}$$

Hence, for the analytical model, we know that $\partial_t \eta(y(t)) \overset{(=)}{\leq} 0$. Hence, we can impose for the last step of a RK method, the (nonlinear) relaxation equation for γ_n, i.e.,

$$r(\gamma_n) := \eta(y_{\gamma_n}^{n+1}) - \eta(y^n) - \gamma_n \Delta t \sum_{i=1}^{s} b_i \underbrace{\langle w(u_i), f_i \rangle}_{\overset{(=)}{\leq} 0} \overset{!}{=} 0. \tag{2.11}$$

It can be solved with some iterative methods as bisection, Newton or similar. In practice the computations of most of the ingredients of r can be done only once and then the scalar equation can be quickly assembled. Let

$$\begin{cases} \Delta y^n := y^{n+1} - y^n = \Delta t \sum_{i=1}^{s} b_i f_i, \\ \Delta \eta^n := \Delta t \sum_{i=1}^{s} b_i \langle w(u_i), f_i \rangle, \end{cases} \tag{2.12}$$

then the relaxation scalar equation becomes

$$r(\gamma_n) := \eta(y^n + \gamma_n \Delta y^n) - \eta(y^n) - \gamma_n \Delta \eta^n \overset{!}{=} 0. \tag{2.13}$$

Here we use the symbol $\overset{!}{=}$ to mean that we impose the equation to be equal to 0 and that we aim to solve it for γ_n.

Remark 2.3. Further results about relaxation RK methods can be found in [26, 41, 39] including also extension to multistep methods. 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 $\sum_j \gamma_n b_j \neq 1$. However, to shorten this part, we stress out that $\gamma_n = 1 + \mathcal{O}(\Delta t^{p-1})$ holds with p denoting the order of the RK method (proofs can be found in the above literature).

Remark 2.4. The relaxation methods are very useful in conservative tests, as they allow to preserve the exact entropy/energy level. In the dissipative case, they are reliably 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.

2.2 Deferred Correction Methods

The idea of DeC schemes as introduced in [19] 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 to prove convergence. The foundation of DeC relies on mimicking these Picard iterations at the discrete level, decreasing the approximation error in several iterative steps. To describe the DeC, we follow the approach presented in [2]. For the description, two operators are introduced: $\mathscr{L}^1$ and $\mathscr{L}^2$. Here, the $\mathscr{L}^1$ operator represents a low-order easy-to-solve numerical scheme, e.g. the explicit Euler method, and $\mathscr{L}^2$ 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.

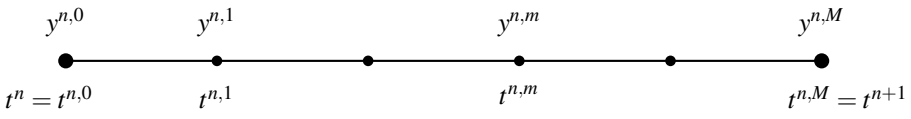


Figure 1: Time interval divided into subintervals

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 $[t^{n,0}, t^{n,m}]$ the Picard–Lindelöf Theorem for both operators $\mathscr{L}^1$ and $\mathscr{L}^2$. We also denote with $y^{n,m}$ the approximation at the time point $t^{n,m}$ as denoted in Figure 1. Without loss of generality, we will consider autonomous systems from here on, not including the dependence on t for the evolution operator f, i.e., $f(t,y) = f(y)$.

Then, the $\mathscr{L}^2$ operator is given by

$$\mathscr{L}^2(y^{n,0},\dots,y^{n,M}) := \begin{cases} y^{n,M} - y^{n,0} - \int_{t^{n,0}}^{t^{n,M}} \mathscr{I}_M(f(y^{n,0}),\dots,f(y^{n,M}))\mathrm{d}t \\ \vdots \\ y^{n,1} - y^{n,0} - \int_{t^{n,0}}^{t^{n,1}} \mathscr{I}_M(f(y^{n,0}),\dots,f(y^{n,M}))\mathrm{d}t \end{cases}. \tag{2.14}$$

Here, the term $\mathscr{I}_M$ denotes an interpolation polynomial of order M evaluated at the points $\{t^{n,r}\}_{r=0}^{M}$. In particular, we use Lagrange polynomials $\{\varphi_r\}_{r=0}^{M}$, which fulfills $\varphi_r(t^{n,m}) = \delta_{r,m}$ and satisfy the property $\sum_{r=0}^{M} \varphi_r(s) = 1$ for any $s \in [t^{n,0}, t^{n,M}]$. In practice the interpolant is defined as

$$\mathscr{I}_M((y^{n,0}),\dots,f(y^{n,M})) := \sum_{r=0}^{M} \varphi_r(s) f(y^{n,r}).$$

Using these properties, we can actually compute the integral of the interpolants thanks to a high order quadrature rule, obtaining weights

$$\theta_r^m := \frac{1}{\Delta t} \int_{t^{n,0}}^{t^{n,m}} \varphi_r(s)\mathrm{d}s,$$

resulting in

$$\mathscr{L}^2(y^{n,0},\dots,y^{n,M}) = \begin{cases} y^{n,M} - y^{n,0} - \Delta t \sum_{r=0}^{M} \theta_r^M f(y^{n,r}) \\ \vdots \\ y^{n,1} - y^{n,0} - \Delta t \sum_{r=0}^{M} \theta_r^1 f(y^{n,r}) \end{cases}. \tag{2.15}$$

The $\mathscr{L}^2$ operator represents an $(M+1)$th order numerical scheme if set equal to zero, i. e., $\mathscr{L}^2(y^{n,0},\dots,y^{n,M}) = 0$. Unfortunately, the resulting scheme is implicit and, further, the terms f may be nonlinear. It can thought as an implicit Runge–Kutta scheme, which requires some techniques to be solved. For this purpose, we introduce a simplification of the $\mathscr{L}^2$ operator. Instead of using a high order accurate quadrature formula at the points $\{t^{n,m}\}_{m=0}^{M}$ we evaluate the integral in equation (2.14) applying the left Riemann sum approximation. The resulting operator $\mathscr{L}^1$ is given by the

forward Euler discretization for each state $y^{n,m}$ in the time interval, i. e.,

$$\mathscr{L}^1(y^{n,0},\dots,y^{n,M}) := \begin{cases} y^{n,M} - y^{n,0} - \beta^M \Delta t f(y^{n,0}) \\ \vdots \\ y^{n,1} - y^{n,0} - \beta^1 \Delta t f(y^{n,0}) \end{cases} \tag{2.16}$$

with coefficients $\beta^m := \frac{t^{n,m}-t^{n,0}}{t^{n,M}-t^{n,0}}$.
To simplify the notation and to describe DeC, we introduce the matrix of states for the variable y at all subtimesteps.

$$\mathbf{y} := (y^{n,0},\dots,y^{n,M}) \in \mathbb{R}^{M\times I}, \text{ such that} \tag{2.17}$$

$$\mathscr{L}^1(\mathbf{y}) := \mathscr{L}^1(y^{n,0},\dots,y^{n,M}) \text{ and } \mathscr{L}^2(\mathbf{y}) := \mathscr{L}^2(y^{n,0},\dots,y^{n,M}). \tag{2.18}$$

The DeC algorithm consists of an iterative procedure that combines of the $\mathscr{L}^1$ and $\mathscr{L}^2$ operators. The aim is to recursively approximate $\mathbf{y}^*$, the numerical solution of the $\mathscr{L}^2 = 0$ 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 subtimestep and the k-th iteration of the DeC algorithm. We will indicate the variable by $y^{n,m,(k)} \in \mathbb{R}^I$. Finally, the DeC method can be written for every time step $[t^n, t^{n+1}]$ as

DeC Algorithm

$$\begin{aligned} &y^{n,0,(k)} := y^n, \quad k = 0,\dots,K, \\ &y^{n,m,(0)} := y^n, \quad m = 1,\dots,M \\ &\mathscr{L}^1(\mathbf{y}^{(k)}) = \mathscr{L}^1(\mathbf{y}^{(k-1)}) - \mathscr{L}^2(\mathbf{y}^{(k-1)}) \text{ with } k = 1,\dots,K, \\ &y^{n+1} = y^{n,M,(K)}, \end{aligned} \tag{2.19}$$

where K is the number of iterations that we want to compute. Using the procedure (2.19), it has been proven [2, 10, 36] that with as many iterations as the order of accuracy of the $\mathscr{L}^2$ operator d the DeC results in a d-th order accurate scheme, i. e., we set $K = d$. Recall that the number of subtimesteps M and their distribution define the order of the $\mathscr{L}^2$ operator, e.g. for equispaced subtimesteps $d = M + 1$, for Gauss–Lobatto $d = 2M$.

We quickly go through the theorem and proof of the order of accuracy of the DeC algorithm as it will be preparatory for the following sections.

Theorem 2.5 (DeC)**.** *Let $\mathscr{L}^1, \mathscr{L}^2 : \mathbb{R}^{I\times M} \to \mathbb{R}^{I\times M}$ be operators depending on Δt such that $\mathscr{L}^1$ is coercive with constant $C_1 > 0$ independent of Δt, i. e.,*

$$||\mathscr{L}^1(\mathbf{y}) - \mathscr{L}^1(\mathbf{w})|| \geq C_1||\mathbf{y} - \mathbf{w}|| \tag{2.20}$$

and $\mathscr{L}^1 - \mathscr{L}^2$ is Lipschitz continuous with constant $C_2\Delta t > 0$, i. e.,

$$||\mathscr{L}^1(\mathbf{y}) - \mathscr{L}^2(\mathbf{y}) - \mathscr{L}^1(\mathbf{w}) + \mathscr{L}^2(\mathbf{w})|| \leq C_2\Delta t||\mathbf{y} - \mathbf{w}|| \tag{2.21}$$

and let $\mathbf{y}^*$ be the unique solution of the system $\mathscr{L}^2(\mathbf{y}^*) = 0$. Then for algorithm (2.19) *we have*

$$||\mathbf{y}^{(K)} - \mathbf{y}^*|| \leq \left(\frac{C_2}{C_1}\Delta t\right)^K ||\mathbf{y}^{(0)} - \mathbf{y}^*||. \tag{2.22}$$

Proof. The proof is fairly simple. It consist in a series of inequalities where we exploit in this order: coercivity, definition of DeC algorithm, $\mathbf{y}^*$ being the solution of $\mathscr{L}^2 = 0$ and Lipschitz continuity. It works by induction as

$$\begin{aligned} &||\mathbf{y}^{(k)} - \mathbf{y}^*|| \leq \frac{1}{C_1}||\mathscr{L}^1(\mathbf{y}^{(k)}) - \mathscr{L}^1(\mathbf{y}^*)|| \\ &= \frac{1}{C_1}||\mathscr{L}^1(\mathbf{y}^{(k-1)}) - \mathscr{L}^2(\mathbf{y}^{(k-1)}) - \mathscr{L}^1(\mathbf{y}^*) + \mathscr{L}^2(\mathbf{y}^*)|| \leq \frac{C_2}{C_1}\Delta t||\mathbf{y}^{(k-1)} - \mathbf{y}^*||. \end{aligned} \tag{2.23}$$

□

The hypotheses of the theorem (coercivity, Lipschitz continuity and existence of the high order accurate solution of the $\mathscr{L}^2$ operator) are satisfied by the operators defined in (2.16) and (2.15). A proof of this can be found in [2, 10] and it uses the smoothness of the solution y and the Lipschitz continuity of f.

Example 2.6. Second and third order DeC

- For the second order case, we have only the two sub time points $t^{n,0} = t^n$ and $t^{n,1} = t^n + \Delta t$. We calculate the first approximation at $t^{n,1} = 1$ using the explicit Euler method. Afterwards, the iteration step (2.19) is used and it reads

$$y^{n,1,(1)} = y^{n,0} + \Delta t f(t^{n,0}, y^{n,0}), \quad y^{n,1,(2)} = y^{n,0} + \frac{1}{2}\Delta t \left(f(t^{n,0}, y^{n,0}) + f(t^{n,1}, y^{n,1,(1)}) \right),$$

  which is equivalent to the SSPRK(2,2) given by the following Butcher tableau: $\begin{array}{c|cc} 0 & & \\ 1 & 1 & \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array}$.

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

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

  First Correction :

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

  Second Correction :

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

  Here, we have ignored the update for $y^{n,1,(3)}$ since it does not effect y^{n+1}. This is different for the classical DeC as described in [19] and investigated in [29].
  However, as before, we can interpret DeC(3) in form of a RK method. The Butcher tableau reads

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

2.3 Semidiscrete Entropy Conservative Residual Distribution Methods

In this section we provide a spatial discretization for hyperbolic equations provided by RD. Historically the RD has been developed in the FE context for steady state equations and only later they have been extended to time evolution problems. RD can be seen as an (arbitrarily high order) finite element discretization which does not require solution of linear systems. Exactly for this reason, the generalization to time dependent problems cannot be performed with the classical method of lines that would require the inversion of the mass matrix and decrease the order [1]. Firstly some correction terms [42], then DeC [2] have been used to overcome those issues.

To better catch the principles of RD, we start explaining it for a steady state problem

$$\operatorname{div} F(U) = 0. \tag{2.24}$$

We define the space of globally continuous, piecewise polynomial functions of degree p:

$$\mathscr{V}_h := \left\{ U \in (C^0(\Omega_h))^I, U_{|\kappa} \in (\mathbb{P}^p)^I, \forall \kappa \in \Omega_h \right\}, \tag{2.25}$$

where Ω_h is a discretization of Ω into elements, e.g. triangles. $\mathbb{P}^p$ denotes the space of polynomials with degree p. With this definition an approximation of a solution of (2.24) can be written as a linear combination of a suitable choice of basis functions of $\mathscr{V}_h$, which are denoted by φ_σ:

$$U(x,t^n) \approx U^{h,n} = \sum_{\sigma \in \Omega_h} U_\sigma^n \varphi_\sigma(x),\ x \in \Omega. \tag{2.26}$$

In general, the choice of basis functions is arbitrary as long as $\operatorname{span}\{\varphi_{\sigma|\kappa}\} = \mathbb{P}^p$ for all $\kappa \in \Omega_h$. However, it has been shown in [4] that there are additional constraints on the basis functions when RD is combined with DeC. Here, the condition $\int_\kappa \varphi_\sigma \mathrm{d}x > 0$ must be also fulfilled [4]. Therefore, Bernstein polynomials [4] or cubature elements [17] (Lagrangian polynomials defined on quadrature points) will be used in our case for the space discretization. Once the setup is done, the approach works in three steps:

1. Define $\Phi^\kappa(U) := \int_\kappa \nabla \mathbf{F}(U) \mathrm{d}x$ which is called the total residual of an element κ;
2. Define Φ_σ^κ as the contribution of a DOF σ to the total residual of the element κ and it will be denoted by local residual. 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 κ and any $U^h \in \mathscr{V}_h$ the conservation relation holds:
$$\sum_{\sigma \in \kappa} \Phi_\sigma^K(U^h) = \Phi^\kappa(U^h) = \int_K \nabla \mathbf{F}(U^h) \mathrm{d}x; \tag{2.27}$$
3. Finally, all local residuals belonging to one DOF σ are collected and summed up. This gives the equation for that DOF U_σ, i.e.,
$$\sum_{\kappa | \sigma \in \kappa} \Phi_\sigma^\kappa = 0, \quad \forall \sigma \in \Omega_h. \tag{2.28}$$

When needed, we can also include the boundary elements Γ in (2.28) and then the update scheme reads

$$\sum_{\kappa | \sigma \in \kappa} \Phi_\sigma^\kappa + \sum_{\Gamma | \sigma \in \Gamma} \Phi_\sigma^\Gamma = 0, \quad \forall \sigma \in \Omega_h. \tag{2.29}$$

This can be done for different purposes, e.g. to add diffusion on the boundaries [4, 31] or in a discontinuous Galerkin setting to introduce the numerical fluxes [9, 3]. As highlighted before, the choice of Φ_σ^κ fully determines the scheme. The order of accuracy of the scheme is given, hence, by the chosen FE space and by the type of discretization of the nodal residual (2.27). Here, we present a definition of nodal residuals which introduces small dissipation levels guaranteeing the L^2 stability of the scheme, and not changing the order of accuracy of the scheme.

Example 2.7. A pure Galerkin discretization can be written in residual form as $\Phi_\sigma^\kappa(U^h) = \int_\kappa \varphi_\sigma \nabla \cdot \mathbf{F}(U^h) \mathrm{d}x$. In our simulations, we add additional stabilization on the jump of the derivatives of the solution. This stabilization term is known as continuous interior penalty (CIP) method and it was proposed in a series of articles [12, 13, 18]. The resulting local residual definition is the following:

$$\Phi_\sigma^\kappa(U^h) = \int_\kappa \varphi_\sigma \nabla \cdot \mathbf{F}(U^h) \mathrm{d}x + \sum_{e \in \kappa} \lambda h_e^2 \oint_e \left[\nabla U^h \right] \cdot [\nabla \varphi_\sigma] \, \mathrm{d}\gamma \tag{2.30}$$

with the jump defined as $\left[\nabla U^h\right] := \nabla U^h_{|_\kappa} - \nabla U^h_{|_{\kappa^+}}$, with κ^+ the neighbouring element sharing the edge e and λ being a stabilization coefficient [31]. This formulation leads to an arbitrarily high order accurate spatial discretization with the corresponding choice of the polynomial degree, i.e., for polynomials of degree p we obtain accuracy order $d = p+1$. More RD formulations for different schemes including discontinuous Galerkin, flux reconstruction, etc. can be found in [5, 6, 8].

Entropy Correction Term

Since we want to construct fully entropy conservative/dissipative RD schemes, we follow the approach presented in [3, 9] and we add an entropy correction term to our steady-state space residual. Let $\eta : \mathbb{R}^I \to \mathbb{R}$ be an entropy, $g : \mathbb{R}^I \to \mathbb{R}^d$ the corresponding entropy flux and $\partial_u \eta(u) = v \in \mathbb{R}^I$ is the entropy variable [24] such that $\langle \eta'(u), F'(u) \rangle = g'(u)$. Let $V^h \in \mathscr{V}_h$ be the discretization of the entropy variable v. The entropy equality in the conservative case using the RD framework reads

$$\sum_{\sigma \in \kappa} \left\langle V_\sigma, \tilde{\Phi}_\sigma^K \right\rangle = \int_{\partial \kappa} g\left(V^h\right) \cdot \mathbf{n} \mathrm{d}\gamma, \tag{2.31}$$

where $\tilde{\Phi}_\sigma^K$ is a modification of the previously presented residuals. Since (2.31) is not fulfilled for general Φ_σ^K, the entropy correction terms r_σ^κ is added to the residuals Φ_σ^K to guarantee (2.31). 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

$$\tilde{\Phi}_\sigma^K = \Phi_\sigma^K + r_\sigma^\kappa \tag{2.32}$$

with the goal of fulfilling the discrete entropy condition (2.31). In [3], the following correction terms are presented

$$r_\sigma^\kappa := \alpha(V_\sigma - \overline{V}), \quad \text{with } \overline{V} = \frac{1}{\#\kappa} \sum_{\sigma \in \kappa} V_\sigma, \tag{2.33}$$

$$\alpha = \frac{\mathscr{E}}{\sum_{\sigma \in \kappa} \left(V_\sigma - \overline{V}\right)^2}, \quad \mathscr{E} := \int_{\partial \kappa} g\left(V^h\right) \cdot \mathbf{n} \mathrm{d}\gamma - \sum_{\sigma \in \kappa} \left\langle V_\sigma, \Phi_\sigma^K \right\rangle, \tag{2.34}$$

where $\#\kappa$ denotes the number of DOFs belonging to κ. In addition, extensions and re-interpretations of the terms can be found in [9] where the following theorem is also proven:

Theorem 2.8. *The correction term* (2.33) *with* (2.34) *satisfies*

$$\sum_{\sigma \in \kappa} r_\sigma^\kappa = 0, \qquad \sum_{\sigma \in \kappa} \left\langle V_\sigma, r_\sigma^\kappa \right\rangle = \mathscr{E}. \tag{2.35}$$

By adding (2.33) *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.a. shocks), the scheme should not just fulfill the equality (2.31) but rather an inequality

$$\sum_{\sigma \in \kappa} \left\langle V_\sigma, \hat{\Phi}_\sigma^K \right\rangle \geq \int_{\partial \kappa} g\left(V^h\right) \cdot \mathbf{n} \mathrm{d}\gamma. \tag{2.36}$$

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

$$\hat{\Phi}_\sigma^K = \Phi_\sigma^K + r_\sigma^\kappa + \Psi_\sigma^\kappa, \tag{2.37}$$

where r_σ^k are defined by (2.33). The Ψ_σ^κ satisfy

$$\sum_{\sigma \in \kappa} \Psi_\sigma^\kappa = 0 \text{ and } \sum_{\sigma \in \kappa} \left\langle V_\sigma, \Psi_\sigma^\kappa \right\rangle \geq 0. \tag{2.38}$$

Two expressions for Ψ_σ^κ 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 similarly to (2.30), by

$$\Psi_\sigma^\kappa := \lambda h_\kappa^2 \int_{\partial \kappa} [\nabla \varphi_\sigma] \cdot [\nabla V^h] \mathrm{d}\gamma, \tag{2.39}$$

which ensures that $\sum_{\sigma \in \kappa} \langle V_\sigma, \Psi_\sigma^\kappa \rangle = \lambda h_\kappa^2 \int_{\partial \kappa} \cdot [\nabla V^h]^2 \mathrm{d}\gamma \geq 0$ for any $\lambda > 0$. We apply this correction term in the nonlinear case to guarantee the inequality in presence of shocks.

Remark 2.9. The presented entropy corrections for residual distribution schemes must be chosen entropy conservative, as in (2.35), or entropy dissipative, as in (2.38). This choice must be done a priori knowing the behavior of the problem.

2.4 Residual Distribution for Hyperbolic Problems

After describing the general construction of RD and DeC schemes, they can be coupled to form an explicit space–time FE scheme for the initial value problem (2.2) as described in [2]. We like to point out that this approach has several similarities and connections to the modern ADER approach [23].

The combination of RD and DeC needs a further modification of the $\mathscr{L}^1$ operator of the DeC algorithm (2.16), in order to avoid the inversion of a mass matrix in the combined scheme. This does not decrease the order of accuracy of the scheme [2]. Following [2, 4], for given local residuals, e.g. (2.30) or (2.37), we define the DeC $\mathscr{L}^1$ and $\mathscr{L}^2$ operators as

$$\mathscr{L}^1_\sigma(U^{(k)}) = \begin{pmatrix} |C_\sigma|(U_\sigma^{n,M,(k)} - U_\sigma^{n,0}) + \Delta t \beta_M \sum\limits_{\kappa|\sigma\in\kappa} \Phi_\sigma^\kappa(U^{n,0}) \\ \vdots \\ |C_\sigma|(U_\sigma^{n,1,(k)} - U_\sigma^{n,0}) + \Delta t \beta_1 \sum\limits_{\kappa|\sigma\in\kappa} \Phi_\sigma^\kappa(U^{n,0}) \end{pmatrix},$$

$$\mathscr{L}^2_\sigma(U^{(k)}) = \begin{pmatrix} \sum\limits_{\kappa|\sigma\in\kappa} \left(\int_\kappa \varphi_\sigma \left(U^{n,M,(k)} - U^{n,0} \right) \mathrm{d}x + \Delta t \sum\limits_{r=0}^{M} \theta_r^M \Phi_\sigma^\kappa(U^{n,r,(k)}) \right) \\ \vdots \\ \sum\limits_{\kappa|\sigma\in\kappa} \left(\int_\kappa \varphi_\sigma \left(U^{n,1,(k)} - U^{n,0} \right) \mathrm{d}x + \Delta t \sum\limits_{r=0}^{M} \theta_r^1 \Phi_\sigma^\kappa(U^{n,r,(k)}) \right) \end{pmatrix},$$

where β_i, θ_r^i are the quadrature weights for the time integration in (2.15) and (2.16) and $|C_\sigma| := \int_\kappa \varphi_\sigma$. The term $|C_\sigma|$ can be seen as some introduced mass lumping effect and should be positive. The m-th line of the DeC iterative procedure (2.19) simply becomes

$$U_\sigma^{n,m,(k)} = U_\sigma^{n,m,(k-1)} - |C_\sigma|^{-1} \sum_{\kappa|\sigma\in\kappa} \left(\int_\kappa \varphi_\sigma (U^{n,m,(k-1)} - U^{n,0}) + \Delta t \sum_{r=0}^{M} \theta_r^m \Phi_\sigma^\kappa(U^{n,r,(k-1)}) \right), \tag{2.40}$$

that we will also use in its vector formulation. Denoting with $\mathbf{M}_{ij} = \int_\Omega \varphi_i \varphi_j \mathrm{d}x$ the mass matrix, with $\mathbf{D}_{ii} = |C_i| = \int_\Omega \varphi_\sigma \mathrm{d}x$ the lumped diagonal matrix, with $\mathbf{I}$ the identity matrix and with $\Phi(U)$ the vector of the residuals $\Phi_\sigma(U) := \sum_{\kappa|\sigma\in\kappa} \Phi_\sigma^\kappa(U)$, we can write it as

$$\begin{aligned} U^{n,m,(k)} =& U^{n,m,(k-1)} - \mathbf{D}^{-1} \left(\mathbf{M}(U^{n,m,(k-1)} - U^{n,0}) + \Delta t \sum_{r=0}^{M} \theta_r^m \Phi(U^{n,r,(k-1)}) \right) && (2.41) \\ =& U^{n,0} + (\mathbf{I} - \mathbf{D}^{-1}\mathbf{M})(U^{n,m,(k-1)} - U^{n,0}) - \Delta t \sum_{r=0}^{M} \theta_r^m \mathbf{D}^{-1} \Phi(U^{n,r,(k-1)}). && (2.42) \end{aligned}$$

Remark 2.10. First of all, we remark that the obtained scheme is arbitrarily high order and avoids the inversion of the mass matrix [2]. However, we have to deal with further terms in the semidiscretization, which makes the scheme not a MOL. This has to be taken into account when the energy/entropy production is discussed. Indeed, we have now two ways of applying the entropy correction terms (2.35) or (2.38) in (2.42). Either, we apply it on the space residual $\Phi(U^{n,r,(k-1)})$ resulting in a semidiscrete entropy conservative or dissipative RD scheme. Or, we apply it to the whole space-time residual

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

to obtain a fully discrete conservative scheme. In [9], experiments on the second approach were already done. The authors noticed only small differences but a rather complex implementation strategy. Therefore, in this work, we aim at combining the entropy correction approach (2.37) with the relaxation framework of section 2.1, in order to simplify the implementation. Hence, we apply the correction term (2.38) only in the semidiscrete setting together with the proposed relaxation approach. In the future, a numerical comparison between the two different approaches will be considered extending also the application to different FE based schemes. However, this is not the purpose of this work.

3 Deferred Correction Methods – Connection to RK and Relaxation Technique

In this section, we highlight the connection between DeC and RK schemes and how to construct DeC Butcher tableaux. Therefore, we focus again only on the simple ODE case (2.1). Furthermore, we apply the relaxation technique to DeC. Using the RK interpretation we demonstrate the results valid for RRK. We will also follow the step of the proof of the order of RRK for relaxation DeC (RDeC) with the formalism of [2]. This will be preparatory to the proof in the fully discretized case. In the following sections we will numerically compare the RDeC to the classical relaxed RK methods both for ODEs and PDEs using the MOL.

In the hyperbolic community the DeC is mostly known for its application with RD, hence, not as a simple RK method applied to a semidiscretization. Similarly also the ADER method has been presented as space time discretization [23]. Here, we want to highlight that such methods can be applied to ODEs and the resulting discretization is a classical RK method as noticed, *inter alia*, in [22, Section 5] and [27].

As we have seen already in example 2.6, the first two DeC approaches can be directly interpreted as RK schemes where the quadrature weights θ and β define the coefficients of the Butcher tableau. However, this is not new at all. Already in [22, 15, 27] 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 of DeC of [2]. As it has been presented in [34, 44], DeC has the advantage that one does not need to specify the coefficients for every order of accuracy as usually necessary in classical RK methods, but arbitrarily high order schemes can be automatically written starting form classical quadrature formulae.

On the other side, rewriting a DeC method as a RK scheme requires a number of stages equal to $K \times M = d \times (d-1)$ for equispaced sub time points and $K \times M = d \times \lceil d/2 \rceil$ for Gauss Lobatto sub time points, which is larger than classical RK stages. Nevertheless, in our case we can even delete additional stages since all the intermediate values of the last correction step are not needed anymore due to the simplification introduced in [2] with respect to the original DeC [19, 29], see example 2.6. Therefore, we get a number of stages equal to $(K-1) \times M + 1 = (d-1) \times (d-1) + 1$ in the equispaced case and $(d-1) \times \lceil d/2 \rceil + 1$ in the Gauss–Lobatto case. The number of flux evaluations coincide with the number of stages.

Moreover, one can notice that every subtimestep is independent of another, so one can compute sequentially the corrections and in parallel the subtimesteps and the flux evaluations, obtaining a computational cost of just $K = d$ corrections and $K = d$ flux evaluations. This procedure makes sense in particular when complex problems are taken into consideration and the flux evaluation is the dominant cost of the time discretization procedure. Furthermore, the resulting Runge–Kutta scheme is a low storage one, as it requires only the storage of $M+1$ stages (both variables and flux evaluations). This has been noticed also in [27] for DeC scheme with $\theta = 0$ in the DeC formulation of [29] where Ketcheson et al. already point out that this formulation allows huge parallelization and that some stages can be skipped. If not parallelized, the DeC is disadvantaged with respect to RK methods in terms of computational costs. For example a DeC with Gauss-Lobatto points with order 4 consists in 10 stages as we will explain below, while there exist RK schemes with 4 stages. Nevertheless, it is straightforward to obtain arbitrarily high order DeC schemes, while to write a very high order RK method with optimized number of stages is not a trivial procedure. Let us consider our version of DeC and rewrite it in a Butcher tableau

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

To this purpose, let us define some block matrices that we use to build up A, b and c. Let $\underline{\beta} \in \mathbb{R}^M$ be a column vector with entries $\underline{\beta}^m = \beta^m$. Then, let us split $\underline{\underline{\theta}} := (\theta_r^m) \in \mathbb{R}^{(M+1)\times M}$, where r spans the columns and m the rows, into $\underline{\underline{\theta}} = (\underline{\theta}_0 | \underline{\underline{\tilde{\theta}}})$, with $\underline{\theta}_0 \in \mathbb{R}^M$ and $\underline{\underline{\tilde{\theta}}} \in \mathbb{R}^{M\times M}$. Now, $A \in \mathbb{R}^{((K-1)\times M+1)\times((K-1)\times M+1)}$ is block diagonal after the first column. The first column of A includes all the coefficients referring to $y^{n,0}$, which are the β^i coefficients for the first iteration and $\underline{\theta}_0$ for all the other corrections. For all the corrections for $k = 2, \dots, K-1$, we insert the matrix $\underline{\underline{\tilde{\theta}}}$ in the (k) row block and $(k-1)$ column block, since to compute the (k) iteration, we need information only from the $(k-1)$ one. For the last correction, we can neglect all the intermediate values write the final update in the row of b. We obtain $b_1 = \theta_0^M$, then there are $K-2$ blocks of zeros, followed by θ_r^M with $r \in 1, \dots, M$, denoting the vector

$\underline{\theta}_r^{M,T} = (\theta_1^M, \ldots, \theta_M^M)$. Hence, the Butcher tableau for an arbitrary DeC approach can be written as

$$\begin{array}{c|cccccc} 0 & 0 & & & & & \\ \underline{\beta} & \underline{\beta} & & & & & \\ \underline{\beta} & \underline{\theta}_0 & \underline{\tilde{\theta}} & & & & \\ \vdots & \underline{\theta}_0 & \underline{0} & \underline{\tilde{\theta}} & & & \\ \vdots & \underline{\theta}_0 & \underline{0} & \underline{0} & \underline{\tilde{\theta}} & & \\ \vdots & \vdots & \vdots & \vdots & \ddots & \ddots & \\ \underline{\beta} & \underline{\theta}_0 & \underline{0} & \ldots & \ldots & \underline{0} & \underline{\tilde{\theta}} \\ \hline & \theta_0^M & \underline{0}^T & \ldots & & \ldots & \underline{0}^T & \underline{\theta}_r^{M,T} \end{array} . \tag{3.1}$$

A comparison to the presented example 2.6 demonstrated that the interpretations coincide. Please note that this description is from [34] where the different y^0 values have been set directly equal to the starting point. In [44], these values have been considered inside the RK methods yielding to a slight different Butcher tableau which is equivalent to the presented one (3.1).

3.1 Relaxation DeC

Due to the RK interpretation, all the theoretical findings of [26, 41, 39] will transfer directly to the DeC approach if one uses DeC as an ODE solver. Therefore, for such problems, we do not demonstrate new results in this part, rather a different technique to prove the accuracy of RDeC using the DeC framework. This should lead to a better understanding for the relaxation approach also for the DeC-RD context in section 4. There, we need the following results to guarantee that the RDeC-RD methods remain high order accurate. Again, in that context, a classical splitting technique through the method of lines has to be avoided for RD methods not to violate the high order property nor the mass–matrix free character.

3.1.1 Relaxation DeC

To adapt the DeC to the relaxation technique, we slightly modify the final update step that in the DeC reads

$$\mathscr{L}^{1,M}(\mathbf{y}^{(K)}) = \mathscr{L}^{1,M}(\mathbf{y}^{(K-1)}) - \mathscr{L}^{2,M}(\mathbf{y}^{(K-1)}). \tag{3.2}$$

with $\mathscr{L}^{1,m}(\mathbf{y}) = y^{n,m} - y^n - \Delta t \beta^m f(y^n)$ and $\mathscr{L}^{2,m}(\mathbf{y}) = y^{n,m} - y^n - \Delta t \sum_{r=0}^M \theta_r^m f(y^{n,r})$ for all $1 \leq m \leq M$. In particular, we modify the operators $\mathscr{L}^1$ and $\mathscr{L}^2$ to obtain two new operators that will be used just in this final step. We define

$$\begin{aligned} \mathscr{L}_{\gamma_n}^{1,M}(\mathbf{y}) := \frac{1}{\gamma_n}\left(y_{\gamma_n}^{n,M} - y^n - \gamma_n \Delta t \beta^M f(y^n)\right) &\approx \frac{1}{\gamma_n}(y(t^n + \gamma_n \Delta t) - y(t^n) - \gamma_n \Delta t f(y(t^n))) \\ &\approx \Delta t(\partial_t y(t^n) - f(y(t^n))) + \mathscr{O}(\Delta t^2), \end{aligned} \tag{3.3}$$

$$\mathscr{L}_{\gamma_n}^{2,M}(\mathbf{y}) := \frac{1}{\gamma_n}\left(y_{\gamma_n}^{n,M} - y^n - \gamma_n \Delta t \sum_{r=0}^M \theta_r^M F(y^{n,r})\right) \approx \Delta t(\partial_t y(t^n) - F(y(t^n))) + \mathscr{O}(\Delta t^{d+2}), \tag{3.4}$$

and then the scheme for the relaxed final step reads

$$\mathscr{L}_{\gamma_n}^{1,M}(\mathbf{y}_{\gamma_n}^{(K)}) = \mathscr{L}^{1,M}(\mathbf{y}^{(K-1)}) - \mathscr{L}^{2,M}(\mathbf{y}^{(K-1)}). \tag{3.5}$$

This sum up to the following formulations for the final update step:

$$\frac{1}{\gamma_n}\left(y_{\gamma_n}^{n+1} - y^n - \gamma_n \Delta t \beta^M f(y^n)\right) = \left(y^{n,M,(K-1)} - y^n - \Delta t \beta^M f(y^n)\right) - \left(y^{n,M,(K-1)} - y^n - \Delta t \theta_r^M f(y^{n,r,(K-1)})\right) \tag{3.6}$$

$$y_{\gamma_n}^{n+1} - y^n = \gamma_n \Delta t \theta_r^M f(y^{n,r,(K-1)}). \tag{3.7}$$

As before, γ_n can be found solving (2.13) with

$$\begin{cases} \Delta y^n := y^{n+1} - y^n = \Delta t \sum_{r=0}^{M} \theta_r^M f(y^{n,r,(K-1)}), \\ \Delta \varepsilon^n := \Delta t \sum_{r=0}^{M} \theta_r^M \left\langle w(y^{n,r,(K-1)}), f(y^{n,r,(K-1)}) \right\rangle. \end{cases} \tag{3.8}$$

In the quadratic energy case, i.e., $w(u) = u$, it leads to

$$\gamma_n = \frac{2\sum_{r=1}^{M} \theta_r^M \left\langle y^n - y^{n,r,(K-1)}, f(y^{n,r,(K-1)}) \right\rangle}{\Delta t \sum_{i,j=0}^{M} \theta_i^M \theta_j^M \left\langle f(y^{n,i,(K-1)}), f(y^{n,j,(K-1)}) \right\rangle}. \tag{3.9}$$

This formulation is equivalent to the relaxation Runge–Kutta (2.7) for ODEs or PDEs solved with the method of lines, but in case of DeC-RD it already defines a different method. In the following we prove the accuracy of the RDeC in a way that the proof can be used identically for the DeC-RD algorithm.

3.1.2 Accuracy of RDeC

We point out that y_γ^{n+1} approximates $y(t^n + \gamma_n \Delta t)$ and not $y(t^n + \Delta t)$ and this guarantees the accuracy of the solution. As it is formulated in [26], the approach that approximates $y(t^{n+1})$ corresponds to the *incremental direction technique* (IDT) and it was proven that the scheme yields to a $d-1$ order method if the corresponding RK method is of order d. This is also the case using the DeC approach. In this paper, we focus on the other interpretation, i.e., $y_\gamma^{n+1} \approx y(t^n + \gamma_n \Delta t)$, which gives a d order method.

Lemma 3.1 (Accuracy of RDeC)**.** *Let f be sufficiently smooth and consider the DeC algorithm* (2.19) *of order $d > 1$ followed by a last step given by* (3.5)*. The so defined RDeC method has order of accuracy d if $\gamma_n = 1 + \mathcal{O}(\Delta t^{d-1})$.*

Proof. Following the proof of the DeC algorithm, we can prove the following inequalities. We remark that K is chosen at least equal to d. Let $\mathbf{y}^*_{\gamma_n}$ be the solution of $\mathscr{L}^2_{\gamma_n}(\mathbf{y}^*) = 0$. Then, we have to prove the following inequalities

$$\left\| y_{\gamma_n}^{n,M,(K)} - y_{\gamma_n}^{*,M} \right\| \leq \frac{1}{C_1} \left\| \mathscr{L}^{1,M}_{\gamma_n}(\mathbf{y}^{(K)}_{\gamma_n}) - \mathscr{L}^{1,M}_{\gamma_n}(\mathbf{y}^*_{\gamma_n}) \right\| \tag{3.10}$$

$$= \frac{1}{C_1} \left\| \mathscr{L}^{1,M}(\mathbf{y}^{(K-1)}) - \mathscr{L}^{2,M}(\mathbf{y}^{(K-1)}) - \mathscr{L}^{1,M}_{\gamma_n}(\mathbf{y}^*_{\gamma_n}) + \mathscr{L}^{2,M}_{\gamma_n}(\mathbf{y}^*_{\gamma_n}) \right\| \tag{3.11}$$

$$\leq \frac{\Delta t C_2}{C_1} \left\| \mathbf{y}^{(K-1)} - \mathbf{y}^* \right\| + \mathcal{O}(\Delta t^d). \tag{3.12}$$

Equation (3.10) is easily proven as in the normal DeC case, using the definition of RDeC (3.3) and of $\mathbf{y}^*$. Equation (3.11) is given by the RDeC step (3.5), while (3.12) requires more developments. Using the triangular inequality we can split it into three terms

$$\left\| \mathscr{L}^{1,M}(\mathbf{y}^{(K-1)}) - \mathscr{L}^{2,M}(\mathbf{y}^{(K-1)}) - \mathscr{L}^{1,M}_{\gamma_n}(\mathbf{y}^*_{\gamma_n}) + \mathscr{L}^{2,M}_{\gamma_n}(\mathbf{y}^*_{\gamma_n}) \right\| \leq A + B + C, \tag{3.13}$$

with

$$\begin{cases} A := \left\| \mathscr{L}^{1,M}(\mathbf{y}^{(K-1)}) - \mathscr{L}^{2,M}(\mathbf{y}^{(K-1)}) - \mathscr{L}^{1,M}(\mathbf{y}^*) + \mathscr{L}^{2,M}(\mathbf{y}^*) \right\|, \\ B := \left\| \mathscr{L}^{1,M}(\mathbf{y}^*) - \mathscr{L}^{2,M}(\mathbf{y}^*) - \mathscr{L}^{1,M}_{\gamma_n}(\mathbf{y}^*) + \mathscr{L}^{2,M}_{\gamma_n}(\mathbf{y}^*) \right\|, \\ C := \left\| \mathscr{L}^{1,M}_{\gamma_n}(\mathbf{y}^*) - \mathscr{L}^{2,M}_{\gamma_n}(\mathbf{y}^*) - \mathscr{L}^{1,M}_{\gamma_n}(\mathbf{y}^*_{\gamma_n}) + \mathscr{L}^{2,M}_{\gamma_n}(\mathbf{y}^*_{\gamma_n}) \right\|, \end{cases} \tag{3.14}$$

where $A \leq C_2 \Delta t \left\| \mathbf{y}^{(K-1)} - \mathbf{y}^* \right\|$ as in the classical DeC,

$$B \leq \left\| \left(\frac{1}{\gamma_n} - 1 \right) (y^{*,M} - y^n) - \left(\frac{1}{\gamma_n} - 1 \right) (y^{*,M} - y^n) \right\| = 0 \tag{3.15}$$

and

$$C = \left\| \frac{\Delta t}{\gamma_n} \sum_{r=0}^{M} \theta_r^M f(y_{\gamma_n}^{*,r}) - f(y^{*,r}) \right\| \leq \frac{\Delta t C_2}{\gamma_n} \left\| \mathbf{y}^* - \mathbf{y}_{\gamma_n}^* \right\| \leq \frac{\Delta t^d \tilde{C}_2}{\gamma_n} = \mathscr{O}(\Delta t^d), \tag{3.16}$$

where, as for the Lipschitz continuity of the operator $\mathscr{L}^1 - \mathscr{L}^2$, we exploit the Lipschitz continuity of f and we know that $\mathbf{y}^*$ and $\mathbf{y}_{\gamma_n}^*$ approximate with order of accuracy d the solution y at times $y(t^n + \Delta t)$ and $y(t^n + \gamma_n \Delta t)$, where $\gamma_n = 1 + \mathscr{O}(\Delta t^{d-1})$. So, using the regularity of the solution, one obtains the aimed result.
Finally, using these results, the original DeC result and the fact that $\mathscr{L}_{\gamma_n}^2(\mathbf{y}_{\gamma_n}^*) = 0$ gives a d order accurate approximation of $y^{ex}(t^n + \gamma_n \Delta t)$ we can state

$$\left\| y_{\gamma_n}^{n,M,(K)} - y^{ex}(t^n + \gamma_n \Delta t) \right\| \leq \left\| y_{\gamma_n}^{n,M,(K)} - y_{\gamma_n}^{*,M} \right\| + \left\| y_{\gamma_n}^{*,M} - y^{ex}(t^n + \gamma_n \Delta t) \right\| \leq \bar{C} \Delta t^d. \tag{3.17}$$

□

Finally, we have to say that in lemma 3.1, we have assumed that $\gamma_n = 1 + \mathscr{O}(\Delta t^{d-1})$ is fulfilled. However, this is proven for the ODE case in [26, Lemma 3]:

Lemma 3.2. *Let a_{ij}, b_j denote the coefficients of a RK method of order d, let f be a sufficiently smooth function, and let γ_n be defined by* (2.9)*. Then $\gamma_n = 1 + \mathscr{O}(\Delta t^{d-1})$.*

Remark 3.3. An alternative proof of this result is shown in [41]. Since we can interpret DeC as a RK scheme, we know that this condition is always fulfilled.

Remark 3.4. These proof are based on the fact the original methods are high order methods and that the solution of the original method substituted in the residual entropy equation of the relaxed method gives an $\mathscr{O}(\Delta t^{d+1})$ [41]. Using then the convexity of the entropy and the Lipschitz continuity of all operators one can show that there must be a root of the relaxed method for γ_n which is close to one with an error which is as well of an $\mathscr{O}(\Delta t^{d-1})$ [41]. Same principles apply for the RDeC also in the PDE case.

Remark 3.5 (Extension to General Entropies). We have presented the RDeC formula for γ_n only for the quadratic energy in the ODE case. Those systems naturally develop from hyperbolic conservation/balance laws using energy (entropy) conservative/dissipative space discretizations, cf. [7, 14, 20, 40] 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. shallow water equations, Euler equations in gas dynamics. As presented in [41], the relaxation approach can be easily adapted to general entropy functions. We remark that the procedure for ODEs (or semidiscretized PDEs) a nonlinear equation must be solved to find γ_n. As for RRK, let us denote with $\varepsilon : \mathbb{R}^I \to \mathbb{R}$ an entropy for (2.1) and $w : \mathbb{R}^I \to \mathbb{R}^I$ being $w(y) = \varepsilon_y(y)$ such that

$$\langle w(y), f(y) \rangle \overset{(=)}{\leq} 0. \tag{3.18}$$

We then find γ_n at time step, solving (2.13) with (3.8). In the following section a complete entropy stable RDeC-RD approach will be developed.

4 A Fully Entropy Conservative/Dissipative DeC-RD Approach

In the following section, we will describe how we can combine the RDeC approach together with the RD framework and how the relaxation parameter γ is calculated in this case. We had for DeC in the RD framework the following update formula for the final step (2.42), so we can write the last step as

$$U^{n+1} = U^n + \Delta t \left\{ (\mathbf{I} - \mathbf{D}^{-1}\mathbf{M}) \frac{U^{n,M,(K-1)} - U^0}{\Delta t} - \sum_{r=0}^{M} \theta_r^M \mathbf{D}^{-1} \Phi(U^{n,r,(K-1)}) \right\}. \tag{4.1}$$

We assume further that due to the usage of the entropy correction term (2.29) (plus the addition of diffusion terms) our space residual is already entropy/energy conservative/dissipative, i.e.,

$$\sum_{\kappa | \sigma \in \kappa} \left\langle V_\sigma(U^{n,r,(K-1)}), \Phi_\sigma(U^{n,r,(K-1)}) \right\rangle \overset{(=)}{\geq} \int_{\partial \kappa} g(U^{n,r,(K-1)}) \cdot \mathbf{n} \, \mathrm{d}\Gamma, \tag{4.2}$$

where V_σ are the entropy variables, g is the entropy flux and Φ denotes the nodal 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. The equality can be obtained by the entropy fix (2.31) and the inequality (2.36) by adding extra diffusion (2.39). In particular, we know that physically the equality holds for smooth flows, while, in presence of discontinuities the inequality appears.

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 simply apply this term since by focusing on (4.1), we realize that we have additional terms given by the lumping of the mass matrix in $\mathscr{L}^1$. The sign of the entropy contribution of this term is unknown.

Let us recall that using the entropy correction term we obtain the relation (2.31) and (2.36). Defining the total entropy as

$$\varepsilon(U) := \int_\Omega \eta(U)dx \approx \sum_\kappa \sum_{\sigma\in\kappa} \eta_\sigma(U) \int_\kappa \varphi_\sigma(x)\mathrm{d}x \tag{4.3}$$

and noting that we want to have

$$\varepsilon(U^{n+1}) - \varepsilon(U^n) \overset{(\leq)}{=} -\int_{t^n}^{t^{n+1}} \int_{\partial\Omega} g(U(t,x))\cdot\mathbf{n}\,\mathrm{d}\Gamma\mathrm{d}t \approx -\Delta t \sum_{r=0}^{M} \theta_r^M \int_{\partial\Omega} g(U^{n,r,(K-1)}(x))\cdot\mathbf{n}\,\mathrm{d}\Gamma. \tag{4.4}$$

Also here, the equality makes sense physically only in case of smooth flows, while in presence of discontinuities, we have the inequality above.

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

$$U_{\gamma_n}^{n+1} := U^n + \gamma_n \Delta U, \tag{4.5}$$

defining

$$\Delta U := \Delta t \left\{ (\mathbf{I} - \mathbf{D}^{-1}\mathbf{M}) \frac{U^{n,M,(K-1)} - U^n}{\Delta t} - \sum_{r=0}^{M} \theta_r^M \mathbf{D}^{-1} \Phi(U^{n,r,(K-1)}) \right\}. \tag{4.6}$$

Using the summation by parts notation, we denote with 1^T an horizontal vector of ones. Then, the entropy reads and can be expanded as

$$\varepsilon(U_{\gamma_n}^{n+1}) = \int_\Omega \eta(U_{\gamma_n}^{n+1})\mathrm{d}x = \sum_{\kappa\in\Omega}\sum_{\sigma\in\kappa}\int_\kappa \varphi_\sigma \eta_\sigma(U_{\gamma_n}^{n+1})\mathrm{d}x = 1^T \mathbf{D}\eta(U_\gamma^{n+1}) \tag{4.7}$$

$$= 1^T\mathbf{D}\eta(U^n) + \gamma_n\Delta t \sum_{\kappa\in\Omega}\sum_{\sigma\in\kappa} \left\langle \eta'_\sigma(U^n), \underbrace{(\mathbf{D}-\mathbf{M})}_{\mathscr{O}(\Delta x^2)} \underbrace{\frac{U^{n,M,(K-1)} - U^n}{\Delta t}}_{\mathscr{O}(1)} - \sum_{r=0}^{M} \theta_r^M \Phi_\sigma^\kappa(U^{n,r,(K-1)}) \right\rangle + \mathscr{O}(\Delta t^2) \tag{4.8}$$

$$= 1^T\mathbf{D}\eta(U^n) - \gamma_n\Delta t \sum_{\kappa\in\Omega}\sum_{\sigma\in\kappa} \left\langle V_\sigma(U^n), \sum_{r=0}^{M} \theta_r^M \Phi_\sigma^\kappa(U^{n,r,(K-1)}) \right\rangle + \mathscr{O}(\Delta t^2) \tag{4.9}$$

$$= 1^T\mathbf{D}\eta(U^n) - \gamma_n\Delta t \sum_{r=0}^{M} \theta_r^M \underbrace{\sum_{\kappa\in\Omega}\sum_{\sigma\in\kappa} \left\langle V_\sigma(U^{n,r,(K-1)}), \Phi_\sigma^\kappa(U^{n,r,(K-1)}) \right\rangle}_{\overset{(=)}{\geq} \int_{\partial\Omega} g(U^{n,r,(K-1)})\cdot\mathbf{n}\,\mathrm{d}\Gamma} + \mathscr{O}(\Delta t^2). \tag{4.10}$$

We recall 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 subtimestep distribution to have positive coefficients. Motivated by this expansion, we impose that

$$\varepsilon(U_{\gamma_n}^{n+1}) - \varepsilon(U^n) + \gamma_n\Delta t \sum_{r=0}^{M} \theta_r^M \sum_{\kappa\in\Omega}\sum_{\sigma\in\kappa} \left\langle V_\sigma(U^{n,r,(K-1)}), \Phi_\sigma^\kappa(U^{n,r,(K-1)}) \right\rangle \overset{!}{=} 0, \tag{4.11}$$

by solving this scalar equation for γ_n. Again, the symbol $\stackrel{!}{=}$ means that we force the equality to be true, by solving the equation for γ_n. Then, we have that

$$\begin{aligned}\varepsilon(U^{n+1}_{\gamma_n}) =& \varepsilon(U^n) - \gamma_n \Delta t \sum_{r=0}^{M} \theta_r^M \sum_{\kappa\in\Omega}\sum_{\sigma\in\kappa} \left\langle V_\sigma(U^{n,r,(K-1)}), \Phi_\sigma^\kappa(U^{n,r,(K-1)}) \right\rangle \\ \stackrel{(=)}{\leq}& \varepsilon(U^n) - \gamma_n \Delta t \sum_{r=0}^{M} \theta_r^M \int_{\partial\Omega} g(U^{n,r,(K-1)}) \cdot \mathbf{n} \, \mathrm{d}\Gamma.\end{aligned} \tag{4.12}$$

A priori, the scalar equation (4.11) that we want to solve is a nonlinear equation that we can solve, for instance, with a Newton method. In case of quadratic entropy for scalar problems, i.e., $\eta(U) = U^2$, $\varepsilon(U) := \frac{1}{2} U^T \mathbf{D} U$ and $V(U) = U$, (4.11) becomes

$$\frac{U^{n,T}\mathbf{D}U^n}{2} + \gamma_n U^{n,T}\mathbf{D}\Delta U + \gamma_n^2 \frac{\Delta U^T \mathbf{D}\Delta U}{2} - \frac{U^{n,T}\mathbf{D}U^n}{2} + \gamma_n \Delta t \sum_{r=0}^{M} \theta_r^M U^{n,r,(K-1),T} \Phi(U^{n,r,(K-1)}) = 0, \tag{4.13}$$

hence, we can solve explicitly the equation for γ_n with

$$\gamma_n = -2 \frac{\Delta t \sum_{r=0}^{M} U^{n,r,(K-1),T} \Phi(U^{n,r,(K-1)}) + U^{n,T}\mathbf{D}\Delta U}{\Delta U^T \mathbf{D} \Delta U}. \tag{4.14}$$

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

$$\sum_{\kappa\in\Omega}\sum_{\sigma\in\kappa} \left\langle V_\sigma(U^{n,r,(K-1)}), \Phi_\sigma^\kappa(U^{n,r,(K-1)}) \right\rangle \geq \int_{\partial\Omega} g(U^{n,r,(K-1)}) \cdot \mathbf{n} \, \mathrm{d}\Gamma,$$

or conservative, i.e.,

$$\sum_{\kappa\in\Omega}\sum_{\sigma\in\kappa} \left\langle V_\sigma(U^{n,r,(K-1)}), \Phi_\sigma^\kappa(U^{n,r,(K-1)}) \right\rangle = \int_{\partial\Omega} g(U^{n,r,(K-1)}) \cdot \mathbf{n} \, \mathrm{d}\Gamma.$$

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 $\eta(U^n + \gamma_n \Delta U) = \eta(U^n)$, solving directly, in the energy case,

$$\gamma_n = \frac{-2U^{n,T}\mathbf{D}\Delta U}{\Delta U^T \mathbf{D}\Delta U}. \tag{4.15}$$

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

4.1 Accuracy of RDeC-RD

The accuracy of the RDeC-RD method follows directly from lemma 3.1. Indeed, the RDeC-RD method (4.5) with γ_n found in (4.11) can be written in the $\mathscr{L}^1$ and $\mathscr{L}^2$ setting of lemma 3.1 with

$$\begin{cases} \mathscr{L}^{1,m}(\mathbf{U}) := \left(U^{n,m} - U^n + \Delta t \beta^m \mathbf{D}^{-1} \Phi(U^n)\right), \\ \mathscr{L}^{2,m}(\mathbf{U}) := \frac{1}{\gamma_n}\mathbf{D}^{-1}\left(\mathbf{M}(U^{n,m} - U^n) + \Delta t \sum_{r=0}^{M} \theta_r^m \Phi(U^{n,r})\right), \\ \mathscr{L}^{1,m}_{\gamma_n}(\mathbf{U}) := \frac{1}{\gamma_n}\left(U^{n,m}_{\gamma_n} - U^n + \gamma_n \Delta t \beta^m \mathbf{D}^{-1} \Phi(U^n)\right), \\ \mathscr{L}^{2,m}_{\gamma_n}(\mathbf{U}) := \frac{1}{\gamma_n}\mathbf{D}^{-1}\left(\mathbf{M}(U^{n,m}_{\gamma_n} - U^n) + \gamma_n \Delta t \sum_{r=0}^{M} \theta_r^m \Phi(U^{n,r})\right). \end{cases} \tag{4.16}$$

There are minor modifications to the proof for ODEs which consists of the difference mass matrices of the two operators. In particular the term B of (3.14) is now

$$B = \left\| \underbrace{\left(\frac{1}{\gamma_n} - 1\right)}_{\mathscr{O}(\Delta t^{d-1})} \underbrace{(\mathbf{I} - \mathbf{D}^{-1}\mathbf{M})}_{\mathscr{O}(\Delta x)} \underbrace{(U^{*,M} - U^n)}_{\mathscr{O}(\Delta t)} \right\| \leq \mathscr{O}(\Delta t^{d+1}) \tag{4.17}$$

and in term C of (3.14) another term appears, which reads

$$C_{RD} = \left\| \frac{1}{\gamma_n} \underbrace{(\mathbf{I} - \mathbf{D}^{-1}\mathbf{M})}_{\mathscr{O}(\Delta x)} \underbrace{(U^{*,M} - U^{*,M}_{\gamma_n})}_{\mathscr{O}(\Delta t^d)} \right\| \leq \mathscr{O}(\Delta t^{d+1}). \tag{4.18}$$

Hence, the proof of lemma 3.1 can be applied very similarly also to RDeC-RD case. Arguments similar to the ODE case hold for estimating the order of $\gamma_n - 1$, see [41] for an analogous proof.

5 Numerical Simulations

In this part, we validate our RDeC methods and compare it also with the RRK method given and investigated in [26, 41]. We focus on similar examples, first we apply the pure time integration RDeC[1] implementation, and we also extend the provided RRK code [26, 37] 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. 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 2.6 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 [26, 41], 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 explore the simulations of RDeC for systems of ODEs

$$\partial_t y = f(y), \quad t \in [0, T], \tag{5.1}$$

where $f(t) \in \mathbb{R}^I$ and $F : \mathbb{R}^I \to \mathbb{R}^I$ is a Lipschitz continuous function.

5.1.1 Nonlinear Oscillator

The first problem is the nonlinear oscillator described in [26] through the system

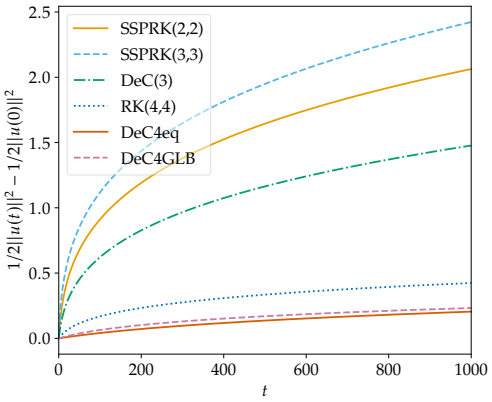
$$\begin{cases} \partial_t \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} \frac{-u_2}{|u|} \\ \frac{u_1}{|u|} \end{pmatrix}, & \text{with } |u| := \sqrt{u_1^2 + u_2^2}, \\ u_1(0) = u_1^0, \\ u_2(0) = u_2^0. \end{cases} \tag{5.2}$$

The system verifies the exact solution

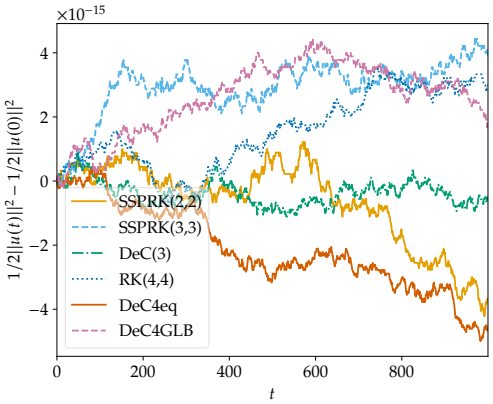
$$\begin{pmatrix} u_1(t) \\ u_2(t) \end{pmatrix} = \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix} \begin{pmatrix} u_1^0 \\ u_2^0 \end{pmatrix}, \quad \theta(t) := \frac{t}{|u(0)|}. \tag{5.3}$$

[1] git.math.uzh.ch/abgrall_group/relaxation-dec-code.git

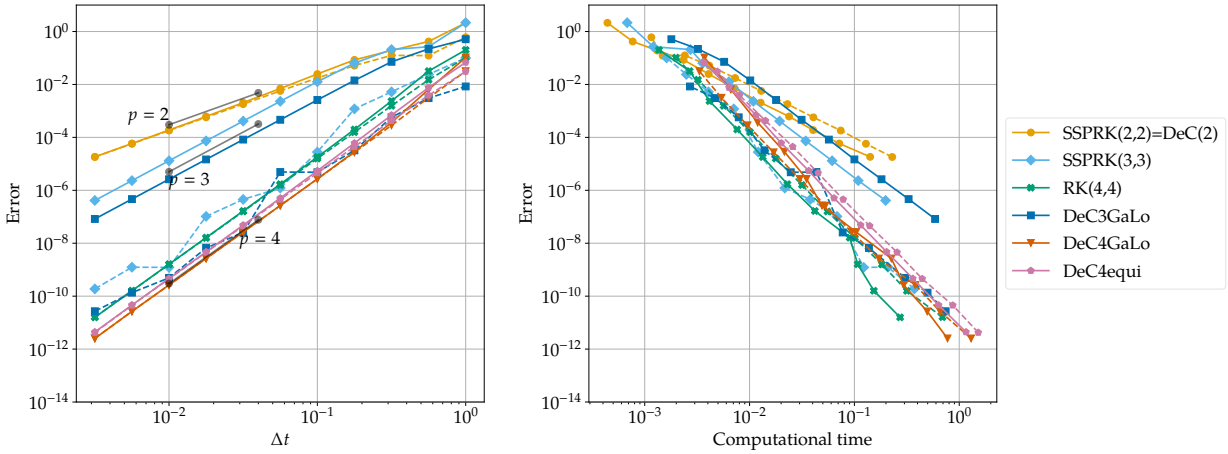
We consider the energy (L^2-entropy) $\varepsilon(u) = \frac{|u|^2}{2}$. In our tests, we took $\Delta t = 0.9$ and $T = 1000$. In fig. 2, we plot the progression of the energy. All the schemes gain energy over time when not using the relaxation approach but the DeC schemes of order three and four (with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



(a) Entropy progression without modification with $\Delta t = 0.9$



(b) Entropy progression with relaxation modification with $\Delta t = 0.9$

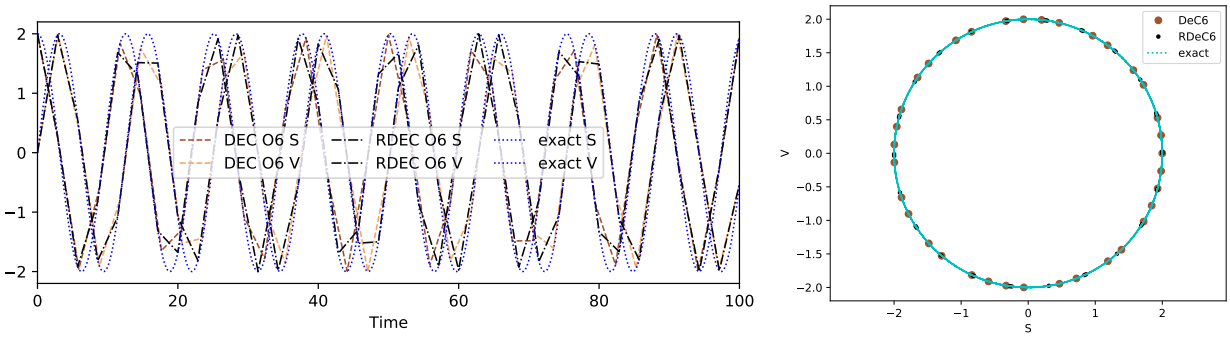


(c) Convergence with respect to timestep (left) and computational time (right) for some RK and DeC methods, classical methods with continuous lines, relaxation methods with dashed lines

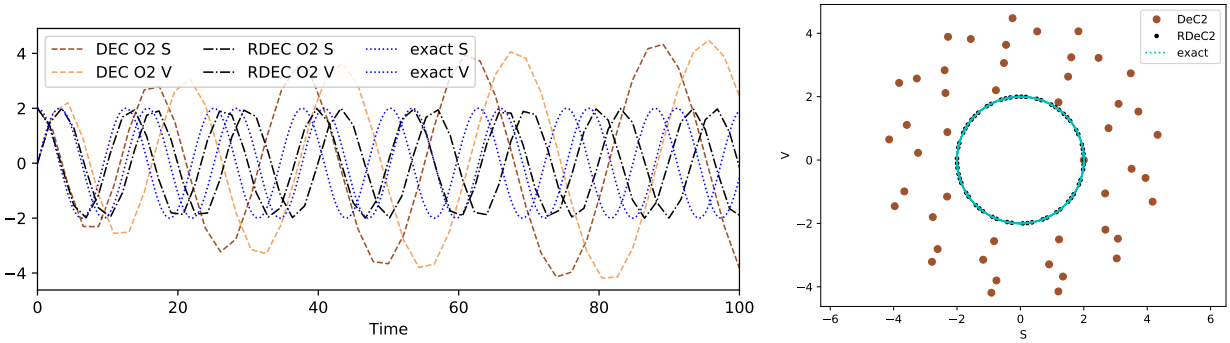
Figure 2: Nonlinear oscillator (5.2): error and entropy error

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 $t = 10$. We can observe it in computational times which increase a bit passing from classical methods to relaxation ones. Moreover, we can observe that also the computational time difference between RK and DeC methods of the same order of accuracy is not so large (factor 2 for order 4), with the possibility of easily obtain arbitrarily high order as shown in fig. 3c.

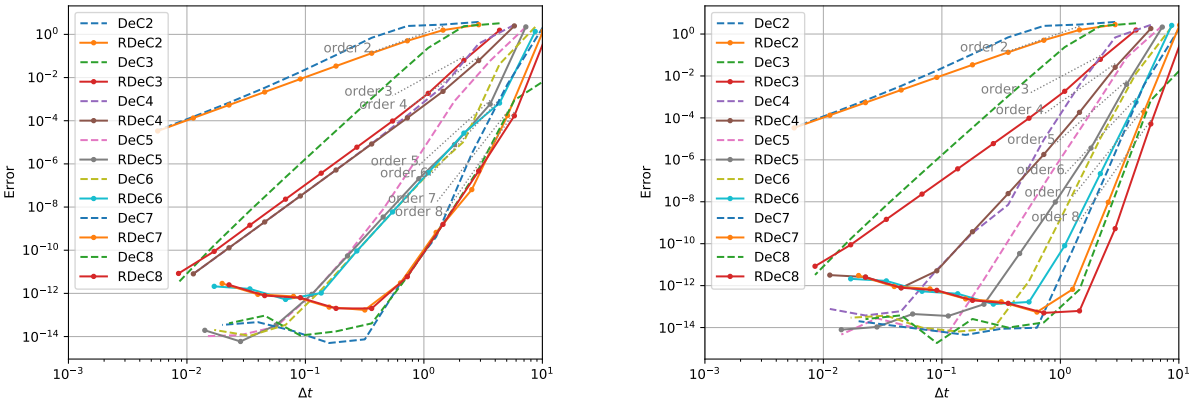
Thanks to the RDeC formulation we can test the properties of our schemes in the very high order regime. In figs. 3a and 3b the simulations for the nonlinear oscillator problem are depicted for DeC and RDeC of order 2 and 6 for different number of timesteps. In the high order solutions in fig. 3a we can 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



(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) Convergence error of DeC and RDeC methods for nonlinear oscillator: equispaced subtimesteps at left, Gauss Lobatto at right

Figure 3: Nonlinear oscillator (5.2) tests for DeC and RDeC: simulations for orders 2 and 6 and convergence for order up to 8

the system violating the physical bounds.

Computing the norm of the energy and its evolution in time, we observe that the classic DeC increase the energy up to an $\mathscr{O}(10)$ for DeC2 and up to $\mathscr{O}(10^{-6})$ for DeC6, while preserving it at machine precision for relaxed DeC. The values of γ_n are around $1-0.1$ for RDeC2 and $1-10^{-8}$ for RDeC6. This means that the relaxation is reducing the extra energy that the classical schemes are introducing, even if the effect for very high order is almost negligible.

Finally, we observe in fig. 3c 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 [37], while for Gauss Lobatto nodes we have even further phenomena of super convergence. Only order 2 is not affected by this. For very high order methods we observe that even for not so small Δt the truncation error becomes dominant and it is useless to further decrease the step size. This is clearly visible in the plateaux in the bottom of fig. 3c.

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

$$\begin{cases} \partial_t \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} \frac{-u_2}{|u|} - \alpha u_1 \\ \frac{u_1}{|u|} - \alpha u_2 \end{pmatrix}, \qquad \text{with } |u| := \sqrt{u_1^2 + u_2^2}, \\ u_1(0) = u_1^0, \\ u_2(0) = u_2^0. \end{cases} \tag{5.4}$$

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

$$\begin{pmatrix} u_1(t) \\ u_2(t) \end{pmatrix} = |u(t)| \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix} \begin{pmatrix} u_1^0 \\ u_2^0 \end{pmatrix}, \quad |u(t)| = |u(0)| e^{-\alpha t}, \quad \theta(t) := \frac{1}{\alpha |u(0)|} \left(e^{\alpha t} - 1 \right). \tag{5.5}$$

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. 4a and 4b 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. 4c 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.

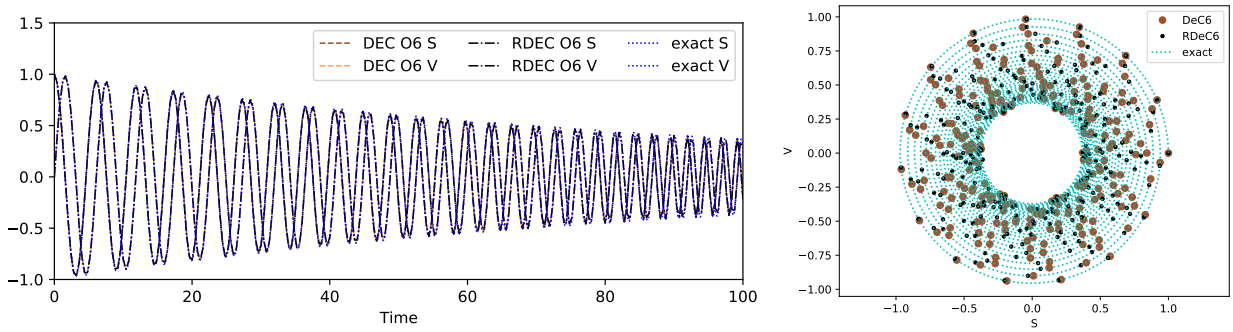
In fig. 5 we observe similar superconvergence results as in the previous test.

5.1.3 Nonlinear Pendulum

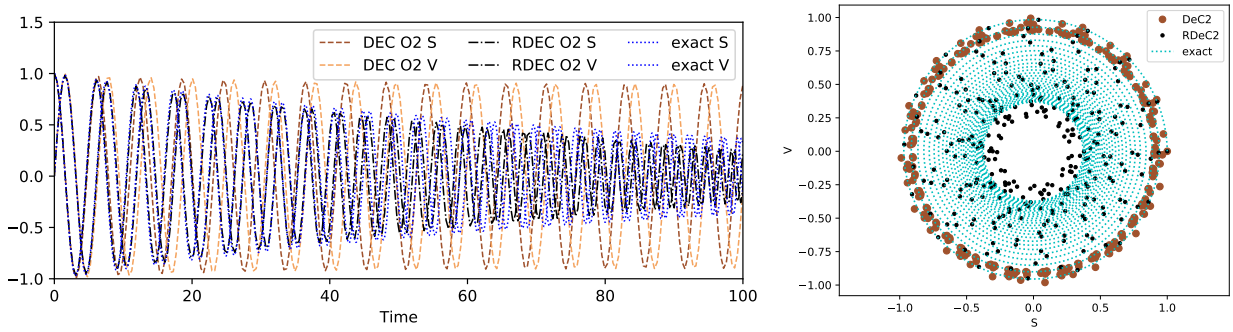
As another example from [41], we focus on the nonlinear pendulum. The trajectories of the pendulum are given by the system $\begin{bmatrix} u_1 \\ u_2 \end{bmatrix}' = \begin{bmatrix} -\sin(u_2) \\ u_1 \end{bmatrix}$ with the initial condition $\begin{bmatrix} u_1(0) \\ u_2(0) \end{bmatrix} = \begin{bmatrix} 1.5 \\ 0 \end{bmatrix}$. 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. 6 for $\Delta t = 0.9$ and $T = 1000$. The entropy behaves as expected, namely it is not constant for non-relaxed schemes but constant 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. 6: 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.

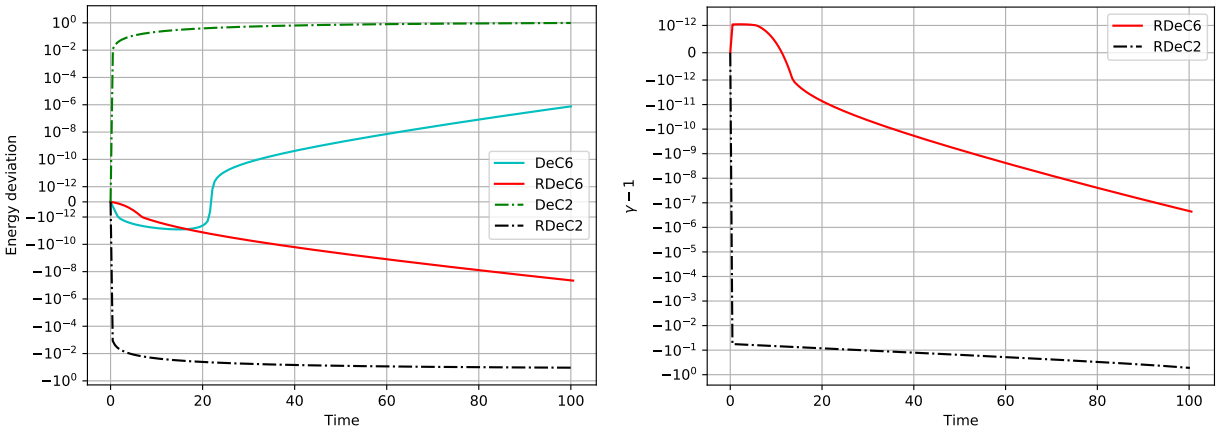
Now, we look also at how the number of time steps are changing when applying the relaxation.
If we define N as the amount of time steps needed to reach T_{end} we have $N = \frac{T_{end}}{\Delta t}$ when no relaxation is applied. However, when introducing the relaxation term, the time step size varies and so does N. This relation is shown in



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

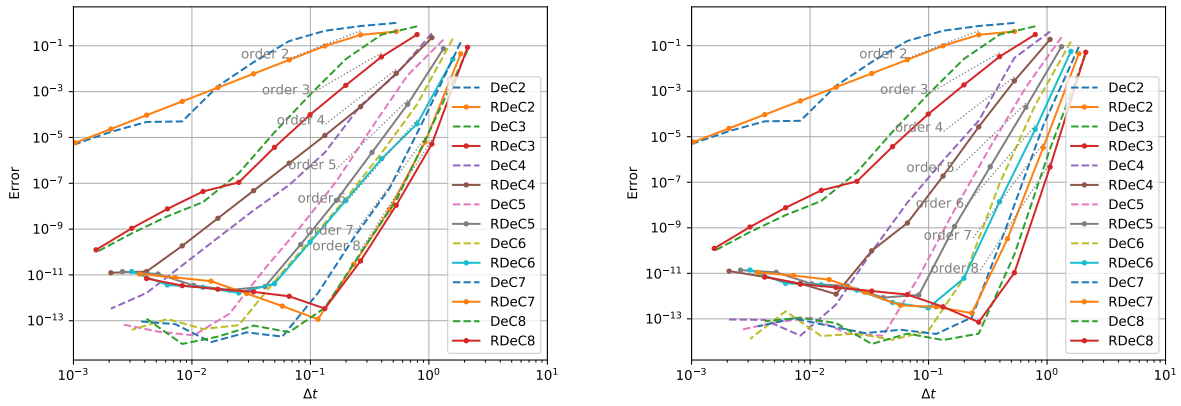


(b) Simulation for order 2 and $N = 250$: time evolution (left), phase space (right)



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

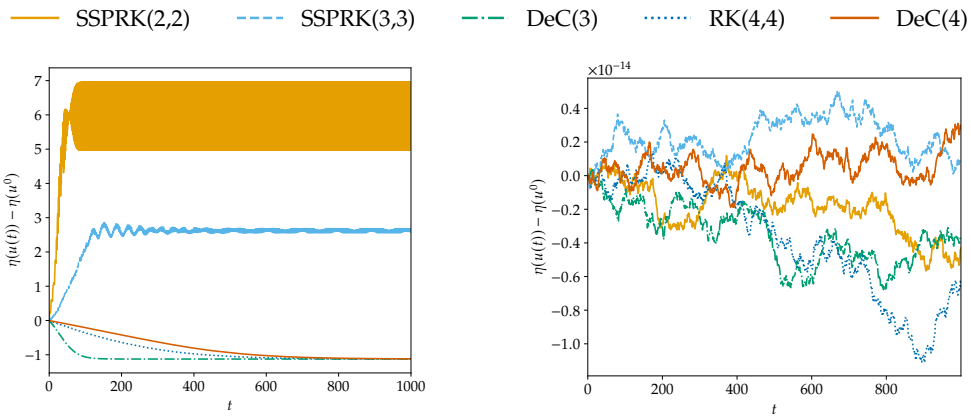
Figure 4: Simulation of the damped nonlinear oscillator (5.4) for DeC and RDeC of orders 2 and 6



(a) Equispaced subtimesteps

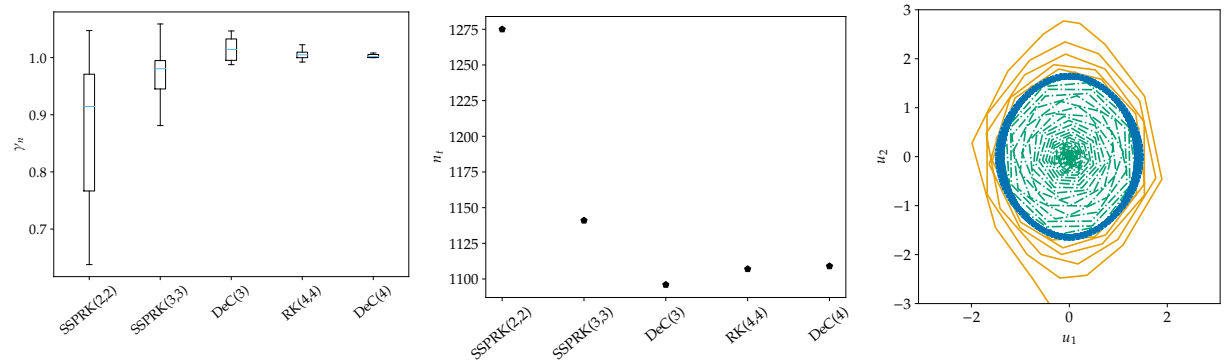
(b) Gauss Lobatto subtimesteps

Figure 5: Convergence error of DeC and RDeC methods for damped nonlinear oscillator (5.4)



(a) Entropy change without relaxation

(b) Entropy change with relaxation

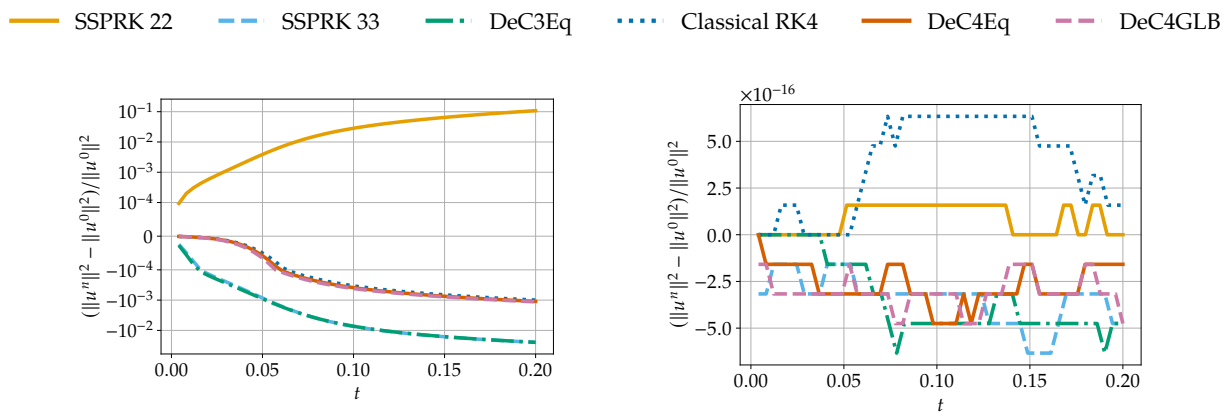


(c) Distribution of γ_n: in blue the median, the boxes denote the three central quartiles

(d) Amount of time steps n_t

(e) Trajectory of the nonlinear pendulum

Figure 6: Several information for the nonlinear pendulum with $\Delta t = 0.9, T = 1000$.



(a) Without relaxation

(b) With relaxation

Figure 7: Entropy for Burgers (5.7) with $\Delta t = 0.2$

fig. 6d. 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. 6c, 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 figs. 6c and 6d.

5.2 Numerical test in the PDE case in one dimension

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

$$\partial_t U + \operatorname{div} F(U) = 0, \quad t \in [0,T], \tag{5.6}$$

where $U \in \mathbb{R}^I$ are conservative variables and $F(U) : \mathbb{R}^I \to \mathbb{R}^{I\times d}$ 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 in the next section we test on 2D advection, Burgers-type and shallow water equations.

5.2.1 Burgers' Equation with method of line

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. [26, 37] 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

$$U_t + \frac{1}{2}(U^2)_x = 0 \tag{5.7}$$

on the interval $-1 \le x \le 1$ with periodic boundary condition and the initial data $U(0,x) = \exp(-30x^2)$. For the space discretization, we use the flux differencing technique and obtain the semidiscretisation

$$U_i'(t) = -\frac{1}{\Delta x}(F_{i+1/2} - F_{i-1/2}) \tag{5.8}$$

with the two-point numerical flux $F_{i+1/2} = \frac{U_i^2+U_iU_{i+1}+U_{i+1}^2}{6}$. 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. 7, we plot the entropy evolution up to $t = 0.2$, before the shock formation, 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 of 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 would 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.

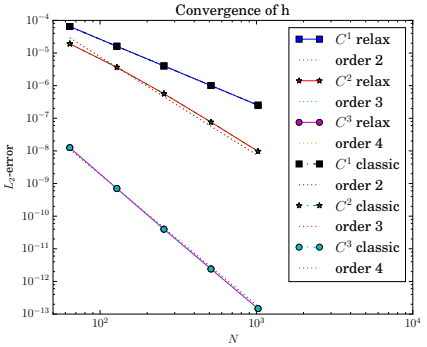
Here, we only proved that the relaxation approach can be used in DeC methods together with classical space discretization methods like DG or FV approaches using the method of lines. In the next section, we apply the RDeC with RD where no method of lines is applicable.

5.2.2 Linear Advection

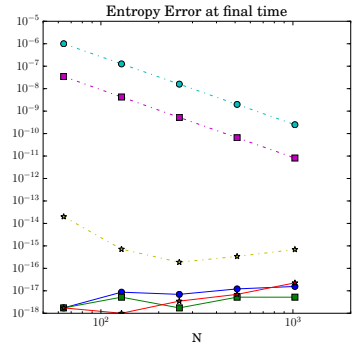
Now, we test the RDeC with the RD spatial discretization. The first test we take into consideration is the linear transport equation

$$\partial_t U + \partial_x U = 0, \tag{5.9}$$

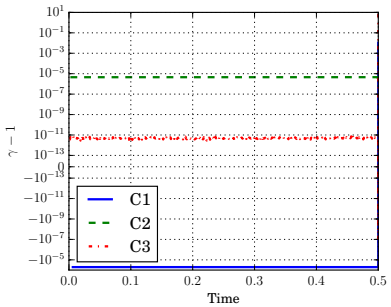
on the domain $[0,2]$ with periodic boundary condition and initial condition $U(t=0,x) = 0.1\sin(\pi x)$. Here, we aim at getting an entropy conservative scheme, as the problem is energy conservative. The spatial discretization is defined by Galerkin residual with CIP stabilization (2.30). For order 2,3 and 4 we have chosen CFL 0.8, 0.2 and 0.12 and CIP stabilization parameter λ equals to 0.12, 0.1 and 0.007, respectively. For details on the choice of CFL and stabilization parameters we refer to [31]. Then, we simply impose (4.15) to find γ_n at each time step. The spatial discretization is obtained with cubature elements [16, 31] (Lagrangian polynomials on Gauss–Lobatto points) that we denote with C^p with p the polynomial degree and the residuals are obtained with Galerkin discretization plus the CIP stabilization term (2.39). In particular, we use DeC$(p+1)$ in combination with C^p polynomials to obtain a $(p+1)$th order accurate scheme. In fig. 8a we observe that the error of the DeC does not decrease adding the relaxation procedure, while in



(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 8: Errors convergence for linear transport problem (5.9) with DeC/RDeC-RD approach

fig. 8b 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 fig. 8c we plot $\gamma - 1$ as a function of time for the different methods. For third and fourth order schemes we increase the time step, while for second 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.

5.3 Numerical test in the PDE case in two dimensions

5.3.1 Scalar rotation

After the one-dimensional set up, we extend our investigation to a two-dimensional rotation problem as it is also investigated in [7, 9]. We have the following problem:

$$\begin{aligned} \partial_t U(t,x,y) + \partial_x(2\pi y U(t,x,y)) + \partial_y(2\pi x U(t,x,y)) = 0, \quad (x,y) \in D, t \in (0,1), \\ U(0,x,y) = U_0(x,y) = \exp\big(-40(x^2 + (y-0.5)^2)\big), \quad (x,y) \in D, \end{aligned} \tag{5.10}$$

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 CIP stabilization with $\lambda = 0.05$ 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 (4.15) gaining the exact entropy behavior[2].

In this test, a small bump centered in $(0, 0.5)$ with radius 0.25 is moving around the center of the circle D, i.e., (0,0). 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 approach at every

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 $\int_\Omega \eta(t) - \int_\Omega \eta(t=0)$ of numerical solutions using a continuous Galerkin scheme for the linear test problem (5.10).

time step and adapt Δt with respect to the entropy production/destruction. In the relaxation case, we need less steps to obtain the full rotation: 505 with respect to 544 in the classical method.

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.

5.3.2 Burgers-type equation

After this smooth test cases, we apply the relaxation approach on a nonlinear problem where actually a shock appears. We test a two dimensional Burgers-type equation

$$\begin{aligned} \partial_t U(t,x,y) + \partial_x(0.5U^2(t,x,y)) + \partial_y(U(t,x,y)) = 0, \quad (x,y) \in D, t \in (0,1), \\ U(0,x,y) = U_0(x,y) = \exp\big(-40(x^2 + y^2)\big), \quad (x,y) \in D, \end{aligned} \tag{5.11}$$

where D is the unit disk in $\mathbb{R}^2$ and outflow boundary conditions are considered. This test is the scalar version of the two-dimensional Burgers' equation, where the term $v\partial_y U$ has been simplified considering a constant $v = 1$.
The DeC2 method is used with CFL number 0.35. We apply again a second order continuous Galerkin scheme with entropy correction and CIP stabilization with $\lambda = 0.1$ with first degree Bernstein polynomials on the same grid as in the previous test. We run our simulation until time $t = 0.5$: 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

[2]We apply the same test also without CIP stabilization. The results were quite similar to the ones presented. The only difference was in the change of energy which was closer to zero due to the semidiscrete entropy conservation of the scheme (between $\approx 10^{-8} - 10^{-12}$).

Final time	Entropy Correction	+ CIP	+ Relaxation	+ Relaxation + CIP
0.22	$-1.97\cdot10^{-4}$	$-1.51\cdot10^{-3}$	$-2.43\cdot10^{-17}$	$-3.46\cdot10^{-18}$
0.5	$-1.51\cdot10^{-3}$	$-3.64\cdot10^{-3}$	$1.39\cdot10^{-17}$	$-3.12\cdot10^{-17}$

Table 2: Entropy variation before and after the shock of Burgers'-type equation (5.11)

Final time	Entropy Correction	+ CIP	+ Relaxation	+ Relaxation + CIP
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 of Burgers'-type equation (5.11)

correction. This problem has a dissipative nature. The entropy correction with the CIP stabilization already has a dissipative nature. We test the relaxation algorithm starting both from the entropy correction (2.35) and the dissipative entropy correction (2.38). For the square entropy $\eta = u^2/2$, the differences $\int_\Omega \eta(t=0.5) - \int_\Omega \eta(t=0)$ of the entropies with and without relaxation are given in table 2.

The results of the simulation can be seen in fig. 9. 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 formation the relaxation schemes are much more accurate, as they keep the total energy conserved as the entropy solution should, while the dissipative approach of the entropy correction with CIP 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 (CIP). On the other side, the maximum increases a bit in the entropy conservative simulation. Observing the solution, we claim that 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.

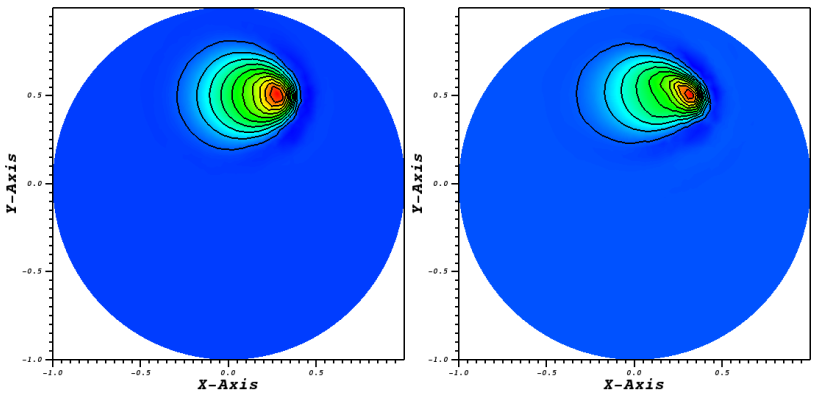


Figure 9: Simulation of Burgers-type equation (5.11) at $t = 0.5$ with DeC2, Bernstein polynomials, 3576 elements, CFL=0.35, left entropy correction + CIP, right relaxation + entropy correction + CIP

Overall, 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.

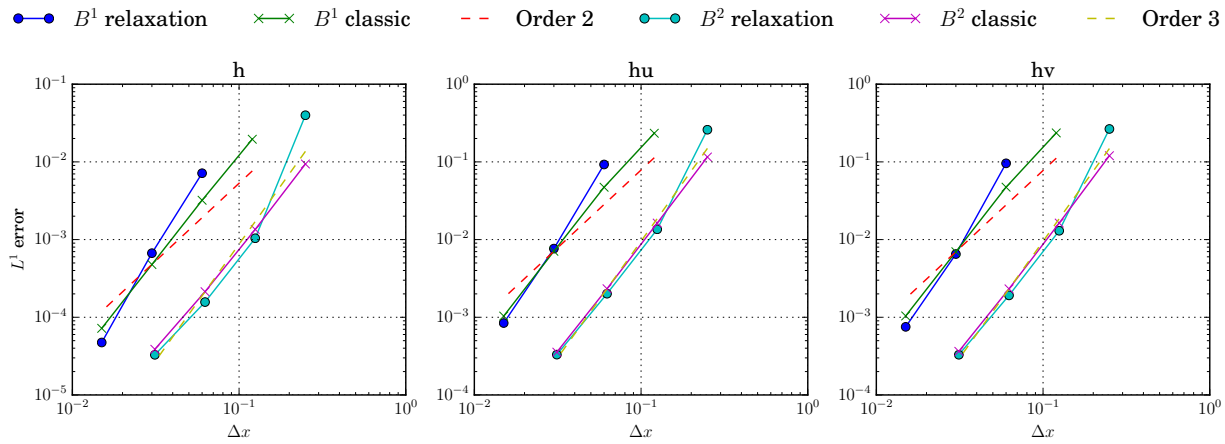


Figure 10: Convergence of the error for relaxation and classic DeC RD method on moving vortex (5.13)

5.3.3 Shallow water moving vortex

In this section, we simulate a moving vortex for the shallow water equations, which is a good benchmark problem to test the preservation of the entropy, as the solution is smooth. The shallow water equations in two dimensions read

$$\begin{cases} \partial_t h + \partial_x(hu) + \partial_y(hv) = 0, \\ \partial_t(hu) + \partial_x(\frac{gh^2}{2} + hu^2) + \partial_y(huv) = 0, \\ \partial_t(hv) + \partial_x(huv) + \partial_y(\frac{gh^2}{2} + hv^2) = 0, \end{cases} \tag{5.12}$$

where $h,u,v : [0,T] \times \Omega \subset \mathbb{R}^+ \times \mathbb{R}^2 \to \mathbb{R}$ are the unknowns. We consider one of the moving vortexes introduced in [43] for which we know the analytical solutions, so that we can compute the error. We take as domain $\Omega = [0,3]^2$ and final time $T = 0.7$. We consider a vortex with compact support, radius 1 and traveling from south-west towards north-east. Following [43], let us define the following auxiliary constants and functions $h_0 = 1$, $u_\infty = 1$, $v_\infty = 1$, $r_0 = 1$, $(x_0, y_0) = (1,1)$, $\Delta h = 0.1$, $g = 9.81$, $\Gamma = \frac{12\pi\sqrt{g\Delta h}}{r_0\sqrt{315\pi^2 - 2048}}$, $(x_c(t), y_c(t)) = (x_0 + u_\infty t, y_0 + v_\infty t)$, $r(x,y,t) := \sqrt{(x - x_c(t))^2 + (y - y_c(t))^2}$, the vortex that we consider is defined as

$$\begin{cases} h(x,y,t) = h_0 - \frac{1}{g}\left(\frac{\Gamma r_0}{\pi}\right)^2 \left(H_2(\pi) - H_2\left(\frac{\pi r(x,y,t)}{r_0}\right)\right), \\ u(x,y,t) = u_\infty - \Gamma\left(1 + \cos\left(\frac{\pi r(x,y,t)}{r_0}\right)\right)^2 (-(y - y_c(x,y,t))), \\ v(x,y,t) = v_\infty - \Gamma\left(1 + \cos\left(\frac{\pi r(x,y,t)}{r_0}\right)\right)^2 (x - x_c(x,y,t)), \end{cases} \tag{5.13}$$

when $r < r_0$, while it is $(h,u,v) = (h_0, u_\infty, v_\infty)$ else. The function H_2 is defined as

$$\begin{aligned} H_2(r) = &\frac{20}{3}\cos(r) + \frac{27}{16}\cos(r)^2 + \frac{4}{9}\cos(r)^3 + \frac{\cos(r)^4}{16} + \frac{20}{3}r\sin(r) + \frac{35}{16}r^2 + \\ &\frac{27}{8}r\cos(r)\sin(r) + \frac{4}{3}r\cos(r)^2\sin(r) + \frac{r}{4}\cos(r)^3\sin(r). \end{aligned}$$

For the spatial discretization we use the residual distribution defined by Bernstein polynomials B^p, a Galerkin projection and the continuous interior penalty (2.30) with $\lambda = 0.1$. The entropy considered here to apply the relaxation procedure is

$$\eta(h,hu,hv) := \frac{gh^2}{2} + \frac{hu^2 + hv^2}{2}.$$

Being the entropy nonlinear, we solve the nonlinear equation (4.11) through the bisection method with a tolerance close to machine precision (10^{-12}).

In fig. 10 we observe that the relaxation does not, again, spoil the order of accuracy. There is a superconvergent phenomenon for the second order method (DeC2 with $\mathbb{P}^1$ polynomials) and beside that, we can observe that the relaxation error for $\mathbb{P}^1$ is even decaying faster than the classical one. In these simulations we noticed that the relaxation parameter γ_n is larger than expected for the very coarse meshes used we have $\gamma_n \approx 2$ and it varies a lot with the chosen CFL. In particular, with small CFLs γ_n is larger than when choosing larger CFLs.

As for the previous simulations the total entropy is conserved numerically for the relaxation methods up to the imposed tolerance in the bisection method, while, for the classical method, it is conserved up to the error of the scheme, similarly to fig. 10. In this case we can compare the minimum value of the vortex at the end of the simulation, which, analytically, is 0.9. We observe that, for the second order method with a mesh with characteristic length of 0.015, the classic method has a minimum of 0.90018, while the relaxed scheme has a minimum of 0.90002, slightly better and less diffusive.

6 Conclusion

In this work, we extended the relaxation technique [26] 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 requires 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 arbitrarily high order method to solve general hyperbolic PDEs avoiding the inversion of the mass matrix.

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 [9] is also desirable.

Acknowledgements

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 Inria International Chair. Ph. Öffner and E. Le Mélédo were funded by an SNF grant (Number 175784). Ph. Öffner has also been funded by an UZH Postdoc Forschungskredit (Number FK-19-104) and the Gutenberg Fellowship (University Mainz). D. Torlo is funded by an Inria Postdoc.

A Another possible relaxation formulation

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

$$U_\sigma^{n,l,(K)} = U_\sigma^{n,l,(K-1)} - |C_\sigma|^{-1} \sum_{\kappa|\sigma\in\kappa} \left(\int_\kappa \varphi_\sigma (U^{n,l,(K-1)} - U^{n,0}) \mathrm{d}x + \Delta t \sum_{r=0}^{M} \theta_r^l \Phi_{\sigma,x}^\kappa (U^{n,r,(K-1)}) \right). \tag{2.40}$$

Focusing on the energy for simplicity, the relaxation term has been given by (2.9) and was calculated by determining the energy production in RK schemes. In the RD framework, we cannot simple apply this term since by focusing on (2.40), 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 (2.40).

It is given by the following calculation on one degree of freedom[3]:

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

We can reorder the equation and get

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

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

$$A = \left(|C_\sigma|^{-1}\sum_{\kappa|\sigma\in\kappa}\int_\kappa \varphi_\sigma U^0 dx\right) \tag{A.1}$$

$$B := \left(U_\sigma^{M,(K-1)} - |C_\sigma|^{-1}\sum_{\kappa|\sigma\in\kappa}\int_\kappa \varphi_\sigma U^{M,(K-1)}\mathrm{d}x\right) \tag{A.2}$$

$$C = |C_\sigma|^{-1}\Delta t\sum_{\kappa|\sigma\in\kappa}\sum_{r=0}^{M}\theta_r^M\Phi_{\sigma,x}^\kappa(U^{r,(K-1)}) \tag{A.3}$$

We can now focus again on above equation and get

$$\begin{aligned}&=(A+B+C)^2 - (U_\sigma^0)^2 = (A^2+2AB+2AC+B^2+2BC+C^2) - (U_\sigma^0)^2\\ &=\underbrace{A^2 - (U_\sigma^0)^2 + 2AB + B^2}_{D} + 2BC + 2AC + C^2.\end{aligned} \tag{A.4}$$

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

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

The first term will cancel out if our space residual is energy conservative[4] 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

[3]For simplicity, we avoid the usage of n in the following.

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

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. le Mélédo, P. Offner, and H. Ranocha. Error boundedness of correction procedure via reconstruction/flux reconstruction and the connection to residual distribution schemes. *Hyperbolic Problems: Theory, Numerics, Applications*, page 215, 2018.

[6] R. Abgrall, E. l. Meledo, and P. Öffner. On the connection between residual distribution schemes and flux reconstruction. *arXiv preprint arXiv:1807.01261*, 2018.

[7] 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.

[8] 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.

[9] R. Abgrall, P. Öffner, and H. Ranocha. Reinterpretation and extension of entropy correction terms for residual distribution and discontinuous Galerkin schemes: Application to structure preserving discretization. *Journal of Computational Physics*, page 110955, 2022.

[10] R. Abgrall and D. Torlo. High order asymptotic preserving deferred correction implicit-explicit schemes for kinetic models. *SIAM Journal on Scientific Computing*, 42(3):B816–B845, 2020.

[11] P. Bacigaluppi, R. Abgrall, and S. Tokareva. " a posteriori" limited high order and robust residual distribution schemes for transient simulations of fluid flows in gas dynamics. *arXiv preprint arXiv:1902.07773*, 2019.

[12] E. Burman and A. Ern. Continuous interior penalty *hp*-finite element methods for advection and advection-diffusion equations. *Mathematics of computation*, 76(259):1119–1140, 2007.

[13] E. Burman and P. Hansbo. The edge stabilization method for finite elements in cfd. In *Numerical mathematics and advanced applications*, pages 196–203. Springer, 2004.

[14] 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.

[15] 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.

[16] G. Cohen, X. Ferrieres, and S. Pernet. A spatial high-order hexahedral discontinuous Galerkin method to solve Maxwells equations in time domain. *Journal of Computational Physics*, 217(2):340–363, 2006.

[17] 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.

[18] J. Douglas and T. Dupont. *Interior Penalty Procedures for Elliptic and Parabolic Galerkin Method*, volume 58, pages 207–216. Springer, 08 2008.

[19] A. Dutt, L. Greengard, and V. Rokhlin. Spectral Deferred Correction Methods for Ordinary Differential Equations. *BIT Numerical Mathematics*, 40(2):241–266, 2000.

[20] 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.

[21] 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.

[22] S. Gottlieb, D. I. Ketcheson, and C.-W. Shu. High order strong stability preserving time discretizations. *Journal of Scientific Computing*, 38(3):251–289, 2009.

[23] 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.

[24] A. Harten. On the symmetric form of systems of conservation laws with entropy. *Journal of computational physics*, 49:151–164, 1983.

[25] 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.

[26] D. Ketcheson. Relaxation Runge–Kutta methods: Conservation and stability for inner-product norms. *SIAM Journal on Numerical Analysis*, 57(6):2850–2870, 2019.

[27] D. Ketcheson and U. bin Waheed. A comparison of high-order explicit Runge–Kutta, extrapolation, and deferred correction methods in serial and parallel. *Communications in Applied Mathematics and Computational Science*, 9(2):175–200, 2014.

[28] 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.

[29] 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.

[30] 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.

[31] S. Michel, D. Torlo, M. Ricchiuto, and R. Abgrall. Spectral analysis of continuous fem for hyperbolic PDEs: Influence of approximation, stabilization, and time-stepping. *Journal of Scientific Computing*, 89(31), 2021.

[32] M. L. Minion. Semi-implicit spectral deferred correction methods for ordinary differential equations. *Commun. Math. Sci.*, 1(3):471–500, 09 2003.

[33] S. Nüßlein, H. Ranocha, and D. I. Ketcheson. Positivity-preserving adaptive Runge–Kutta methods. *Communications in Applied Mathematics and Computational Science*, 16(2):155–179, 2021.

[34] P. Öffner. *Approximation and Stability properties of Numerical Methods for Hyperbolic Conservation Laws*. Habilitation thesis, University Zurich, 2020.

[35] 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.

[36] P. Öffner and D. Torlo. Arbitrary high-order, conservative and positivity preserving patankar-type deferred correction schemes. *Applied Numerical Mathematics*, 153:15–34, 2020.

[37] H. Ranocha and D. I. Ketcheson. Relaxation Runge–Kutta methods for inner-product norms. `https://github.com/ketch/RRK_rr`, 05 2019.

[38] H. Ranocha and D. I. Ketcheson. Relaxation Runge-Kutta Methods for Hamiltonian Problems. *Journal of Scientific Computing*, 84(1):1–27, 2020.

[39] 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.

[40] H. Ranocha, P. Öffner, and T. Sonar. Summation-by-parts operators for correction procedure via reconstruction. *Journal of Computational Physics*, 311:299–328, 2016.

[41] 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.

[42] M. Ricchiuto and R. Abgrall. Explicit Runge-Kutta residual distribution schemes for time dependent problems: second order case. *Journal of Computational Physics*, 229(16):5653–5691, 2010.

[43] M. Ricchiuto and D. Torlo. Analytical travelling vortex solutions of hyperbolic equations for validating very high order schemes. *arXiv preprint arXiv:2109.10183*, 2021.

[44] D. Torlo. *Hyperbolic Problems: High Order Methods and Model Order Reduction*. PhD thesis, University Zurich, 2020.