# Arbitrary high-order, conservative and positive preserving Patankar-type deferred correction schemes

Philipp Öffner* and Davide Torlo †
Institute of Mathematics, University of Zurich, Switzerland

August 6th, 2019

**Abstract**

Production-destruction systems (PDS) of ordinary differential equations (ODEs) are used to describe physical and biological reactions in nature. The considered quantities are subject to natural laws. Therefore, they preserve positivity and conservation of mass at the analytical level.

In order to maintain these properties at the discrete level, the so-called modified Patankar-Runge-Kutta (MPRK) schemes are often used in this context. However, up to our knowledge, the family of MPRK has been only developed up to third order of accuracy. In this work, we propose a method to solve PDS problems, but using the Deferred Correction (DeC) process as a time integration method. Applying the modified Patankar approach to the DeC scheme results in provable conservative and positivity preserving methods. Furthermore, we demonstrate that these modified Patankar DeC schemes can be constructed up to arbitrarily high order. Finally, we validate our theoretical analysis through numerical simulations.

## 1 Introduction

The modelling of geobiochemical processes or ecosystems leads often to systems of ordinary differential equations (ODEs) which can be formulated in the so-called **production-destruction** systems (PDS) as described in [5, 10] for example. To guarantee the physical and chemical laws, the quantities have to fulfil several conditions like positivity and conservation.

The applied numerical method should not violate these conditions and big efforts have been devoted to designing conservative and positivity preserving schemes, since classical approaches like Runge Kutta (RK) schemes do not guarantee these properties.

In [4] the authors suggest modified Patankar-type methods of first and second order which verify the desired properties, i.e., conservation and positivity. Recently, further extensions were done to construct modified Patankar-Runge-Kutta (MPRK) schemes of second and third order [14, 16, 15, 12, 11]. As the name suggests, all these schemes use, as a basic procedure, the Runge-Kutta method, which has been modified by weighting the production and destruction terms as suggested in [20]. Thanks to these weighting coefficients, the schemes are forced to maintain positivity of the variables and to conserve some quantities of interest. However, the described and constructed schemes are, up to our knowledge, at most third order accurate.

In this paper, we present a way to construct **arbitrary high-order, positivity preserving, numerically robust and conservative** schemes for PDS. Differently from previous schemes, we do not start building our schemes on RK methods. We consider the Deferred Correction (DeC) procedure, a high order time integration technique, and we modify it, in order to obtain a positivity preserving, conservative and arbitrary high-order scheme. Moreover, we provide a proof of the desired properties.

[*] Corresponding author: P. Öffner, philipp.oeffner@math.uzh.ch
[†] Corresponding author: D. Torlo, davide.torlo@math.uzh.ch

The paper is organised as follows.
In section 2 we introduce the production-destruction systems and we give a short introduction about the so-called Patankar trick and how it was applied in [5] to construct a modified Patankar-type scheme starting from the explicit Euler method. Afterwards, in section 3, we introduce the Deferred Correction (DeC) method and we discuss conservation and positivity for this classical formulation. In section 4, we build the main core of this work, explaining our modification of DeC through the Patankar trick (mPDeC) and we prove that the obtained mPDeC schemes are positive preserving, conservative and arbitrary high-order accurate. In section 5, we validate our theoretical investigations, considering three different benchmark problems, which are also discussed in different literature references, as [4, 14]. Finally, we give a summary and an outlook for possible extensions.

## 2 Production–Destruction Systems

In this paper we consider production-destruction systems (PDS) of the form

$$\begin{cases} d_t c_i = P_i(\mathbf{c}) - D_i(\mathbf{c}), \quad i = 1, \dots, I, \\ \mathbf{c}(t=0) = \mathbf{c}_0, \end{cases} \tag{1}$$

where $\mathbf{c} = (c_1, \dots, c_I)^T \in \mathbb{R}^I$ represents the vectors of $I$ constituents, $t$ denotes the time and $\mathbf{c}_0$ the initial condition. Moreover, $P_i(\mathbf{c})$ and $D_i(\mathbf{c})$ represent the production and destruction rates of the $i$-th constituent and both terms are assumed to be non-negative, i.e, $P_i,\ D_i \geq 0$ for $i = 1, \dots, I$. These systems rise naturally to describe geochemical processes as it is described in [4, 5] and we recapitulate their notations and definitions in this section.
The production and destruction terms can also be written in a *matrix form* as follows

$$P_i(\mathbf{c}) = \sum_{j=1}^{I} p_{i,j}(\mathbf{c}), \quad D_i(\mathbf{c}) = \sum_{j=1}^{I} d_{i,j}(\mathbf{c}), \tag{2}$$

where each term $p_{i,j} \geq 0$ and $d_{i,j} \geq 0$ are Lipschitz continuous functions and may depend linearly or non–linearly on $\mathbf{c}$. Furthermore, the term $d_{i,j}$ describes the rate of change from the $i$-th to the $j$-th constituent while $p_{i,j}$ is the rate at which the $j$-th constituent is transformed into the $i$-th.
We are interested in (fully) conservative and positive production–destruction systems. To clarify these expressions we repeat the definitions from [14].

**Definition 2.1.** *The PDS (1) is called* ***positive*** *if positive initial values $c_i(0) > 0$ for $i = 1, \dots, I$ imply positive solutions, $c_i(t) > 0$ for $i = 1, \cdots, I$ for all times $t > 0$.*
*The PDS (1) is called* ***conservative*** *if at any time $t \geq 0$, we have that*

$$\sum_{i=1}^{I} c_i(t) = \sum_{i=1}^{I} c_i(0) \tag{3}$$

*is fulfilled. In the analytic form (1), the conservation property (3) is equivalent to the following relation for the matrix representation (2)*

$$p_{i,j}(\mathbf{c}) = d_{j,i}(\mathbf{c}), \quad \forall i, j = 1, \dots, I. \tag{4}$$

*Moreover, the system is called* ***fully conservative*** *if additionally $p_{i,i}(\mathbf{c}) = d_{i,i}(\mathbf{c}) = 0$ holds for all $\mathbf{c} \geq 0$ and $i = 1, \dots, I$.*

As it is described in [14] every conservation PDS can be written in a fully conservative formulation. We can rewrite the two terms of (4) into one matrix of exchanging quantities $e(\mathbf{c})$ defined as

$$e_{i,j}(\mathbf{c}) := p_{i,j}(\mathbf{c}) - d_{i,j}(\mathbf{c}). \tag{5}$$

Clearly, from property (4), we have that $e_{i,i} = 0$. With this notation, let us define the total exchange rate for the $i$-th constituent as

$$E_i(\mathbf{c}) := P_i(\mathbf{c}) - D_i(\mathbf{c}). \tag{6}$$

A numerical method suited to solve a conservative and positive PDS (1) should mimic, at the discrete level, the continuous setting properties. For a one-step methods, we can introduce the discrete analogues of definitions (2.1).

**Definition 2.2.** *Let $\mathbf{c}^n$ denote the approximation of $\mathbf{c}(t^n)$ at the time level $t^n$. A one-step method*

$$\mathbf{c}^{n+1} = \mathbf{c}^n + \Delta t \Phi(t^n, \mathbf{c}^n, \mathbf{c}^{n+1}, \Delta t), \tag{7}$$

*with process function $\Phi$, is called*

- ***unconditionally conservative*** *if for all $n \in \mathbb{N}$ and $\Delta t > 0$*

$$\sum_{i=1}^{I} c_i^{n+1} = \sum_{i=1}^{I} c_i^n \tag{8}$$

  *holds;*

- ***unconditionally positive*** *if for all $\Delta t > 0$ and $\mathbf{c}^n > 0$, we have that $\mathbf{c}^{n+1} > 0$.*

**Example 2.3.** *Let us consider as an example the explicit Euler method. The method is defined by*

$$\mathbf{c}^{n+1} = \mathbf{c}^n + \Delta t E_i(\mathbf{c}^n). \tag{9}$$

*It is conservative since*

$$\sum_{i=1}^{I} \left(c_i^{n+1} - c_i^n\right) = \sum_{i=1}^{I} \left( c_i^n + \Delta t \sum_{i=1}^{I} (p_{i,j}(\mathbf{c}^n) - d_{i,j}(\mathbf{c}^n)) - c_i^n \right) = \Delta t \sum_{i=1}^{I} (p_{i,j}(\mathbf{c}^n) - d_{i,j}(\mathbf{c}^n)) = 0 \tag{10}$$

*holds. Conversely, the explicit Euler method is not unconditionally positive. Consider a conservative and positive PDS (1) where we assume that the right hand side is not identical zero. Then, there exists a $\mathbf{c}^n \geq 0$ such that $\mathbf{P}(\mathbf{c^n}) - \mathbf{D}(\mathbf{c^n}) \neq \mathbf{0}$. Since the PDS is conservative, we can at least find one constituent $i \in \{1, \dots, I\}$, where $D_i(\mathbf{c}^n) > P_i(\mathbf{c}^n) \geq 0$. Choosing*

$$\Delta t > \frac{c_i^n}{D_i(\mathbf{c}^n) - P_i(\mathbf{c}^n)} > 0, \tag{11}$$

*we obtain*

$$c_i^{n+1} = c_i^n + \Delta t \left(P_i(\mathbf{c}^n) - D_i(\mathbf{c}^n)\right) < c_i^n + \frac{\mathbf{c}_i^n}{D_i(\mathbf{c}^n) - P_i(\mathbf{c}^n)} \left(P_i(\mathbf{c}^n) - D_i(\mathbf{c}^n)\right) = c_i^n - c_i^n = 0. \tag{12}$$

*This demonstrates the violation of the positivity for the explicit Euler method for unbounded timesteps $\Delta t$.*

To build an unconditionally positive numerical scheme, Patankar had the idea in [20] of weighting the destruction term in the original explicit Euler methods with the following coefficient

$$c_i^{n+1} = c_i^n + \Delta t \left( \sum_{j=1}^{I} p_{i,j}(\mathbf{c}^n) - \sum_{j=1}^{I} d_{i,j}(\mathbf{c}^n) \frac{c_i^{n+1}}{c_i^n} \right), \quad i = 1, \dots, I. \tag{13}$$

Hence, the scheme (13) is unconditionally positive, but the conservation relation is violated. In [4] a modification of the Patankar scheme (13) was presented, resulting in an unconditionally positive and conservative method. It is defined as follows.

$$c_i^{n+1} := c_i^n + \Delta t \left( \sum_{j=1}^{I} p_{i,j}(\mathbf{c}^n) \frac{c_j^{n+1}}{c_j^n} - \sum_{j=1}^{I} d_{i,j}(\mathbf{c}^n) \frac{c_i^{n+1}}{c_i^n} \right), \quad i = 1, \dots, I. \tag{14}$$

The scheme is implicit and can be solved inverting the mass matrix M in the system $\mathrm{M}\mathbf{c}^{n+1} = \mathbf{c}^n$ where M is

$$\begin{cases} m_{i,i}(\mathbf{c}^n) = 1 + \Delta t \sum_{k=1}^{I} \frac{d_{i,k}(\mathbf{c}^n)}{c_i^n}, \quad i = 1, \dots, I, \\ m_{i,j}(\mathbf{c}^n) = -\Delta t \frac{p_{i,j}(\mathbf{c}^n)}{c_j^n}, \quad i, j = 1, \dots, I, \, i \neq j. \end{cases} \tag{15}$$

The construction of the mass matrix M must follow substantial prescriptions in order to preserve the positivity of the scheme, as suggested in [15].

**Remark 2.4.** *Extensions of the modified Patankar scheme* (14) *to Runge-Kutta schemes were proposed in [14, 15] and further developed in [11, 12]. Special focus lies in the weighting of the production and destruction terms as it is investigated for example in [16] and references therein. Families of second and third order modified Patankar-Runge-Kutta (MPRK) schemes can be found in the mentioned literature. We do not provide the definition of MPRK because the modified Patankar scheme* (14) *already gives us the basic idea for the new methods we want to propose. We will prove that these methods are positivity preserving, conservative and arbitrary high-order.*

# 3 Deferred Correction Methods

There are various approaches to solve numerically an ODE. A first approach is given by finite differences, where the derivative in time is replaced by differences of states in different timesteps. Backward and forward 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 Runge-Kutta methods (explicit and implicit ones), see [9, 24] for details. However, we follow a different approach in this paper.

We start our investigation with the **Deferred Correction (DeC)** method introduced in [7]. In its original formulation, it is an explicit, arbitrary high order method for ODEs. Further extensions of DeC can be found in the literature, including semi-implicit approaches as in [19]. However, in this work we will not consider the semi-implicit framework. Instead, we will focus on the **explicit** DeC approach used by Abgrall in [1]. In our opinion, his notation describes DeC in a more compact way than in previous works [7, 6, 17].[1] Nevertheless, the main idea is always the same and it is based on the Picard-Lindelöf theorem in the continuous setting. The theorem states the existence and uniqueness of solutions for ODEs. The classical proof makes use of the so-called Picard iterations to minimize the error and to prove convergence. The foundation of DeC relies on mimicking the Picard iterations at the discrete level. The approximation error decreases with several iteration steps. For the description of DeC, Abgrall introduces two operators: $\mathcal{L}^1$ and $\mathcal{L}^2$.

Here, the $\mathcal{L}^1$ operator represents a low-order easy-to-solve numerical scheme, e.g. the explicit Euler method, and $\mathcal{L}^2$ is a high-order operator that can present difficulties in its practical solution, e.g. an implicit RK scheme. The DeC method can be written as a combination of these two operators.

Given a timeinterval $[t^n, t^{n+1}]$ we subdivide it into $M$ subintervals $\{[t^{n,m-1}, t^{n,m}]\}_{m=1}^{M}$, where $t^{n,0} = t^n$ and $t^{n,M} = t^{n+1}$ and we mimic for every subinterval $[t^0, t^m]$ the Picard–Lindelöf theorem for both operators $\mathcal{L}^1$ and $\mathcal{L}^2$. We drop the dependency on the timestep $n$ for subtimesteps $t^{n,m}$ and substates $\mathbf{c}^{m,n}$ as denoted in Figure 1.

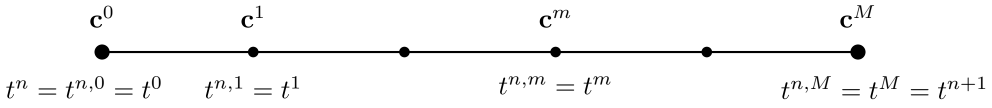

Figure 1: Figure: divided time interval

[1] We like to mention that Abgrall focused on DeC as a time integration scheme in the context of finite element methods. Applying a classical RK method, a dense mass matrix has to be inverted and Abgrall wanted to avoid this. By using a DeC scheme, instead, he showed that a mass matrix free approach is possible [1].

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

$$\mathcal{L}^2(\mathbf{c}^0,\dots,\mathbf{c}^M) := \begin{cases} \mathbf{c}^M - \mathbf{c}^0 - \int_{t^0}^{t^M} \mathcal{I}_M(E(\mathbf{c}^0),\dots,E(\mathbf{c}^M)) \\ \vdots \\ \mathbf{c}^1 - \mathbf{c}^0 - \int_{t^0}^{t^1} \mathcal{I}_M(E(\mathbf{c}^0),\dots,E(\mathbf{c}^M)) \end{cases}. \tag{16}$$

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

$$\mathcal{L}^2(\mathbf{c}^0,\dots,\mathbf{c}^M) = \begin{cases} \mathbf{c}^M - \mathbf{c}^0 - \sum_{r=0}^M \theta_r^M E(\mathbf{c}^r) \\ \vdots \\ \mathbf{c}^1 - \mathbf{c}^0 - \sum_{r=0}^M \theta_r^1 E(\mathbf{c}^r) \end{cases}. \tag{17}$$

The $\mathcal{L}^2$ operator represents an $(M+1)$ order numerical scheme if set equal to zero, i.e., $\mathcal{L}^2(\mathbf{c}^0,\dots,\mathbf{c}^M) = 0$. Unfortunately, the resulting scheme is implicit and, further, the terms $E$ may be non-linear. Because of this, the only $\mathcal{L}^2$ formulation is not explicit and more efforts have to be made to solve it.
For this purpose, we introduce a simplification of the $\mathcal{L}^2$ operator. Instead of using a quadrature formula at the points $\{t^m\}_{m=0}^M$ we evaluate the integral in equation (16) applying the left Riemann sum. The resulting operator $\mathcal{L}^1$ is given by the forward Euler discretization for each state $\mathbf{c}^m$ in the timeinterval, i.e.,

$$\mathcal{L}^1(\mathbf{c}^0,\dots,\mathbf{c}^M) := \begin{cases} \mathbf{c}^M - \mathbf{c}^0 - \beta^M \Delta t E(\mathbf{c}^0) \\ \vdots \\ \mathbf{c}^1 - \mathbf{c}^0 - \beta^1 \Delta t E(\mathbf{c}^0) \end{cases}. \tag{18}$$

with coefficients $\beta^m := \frac{t^m - t^0}{t^M - t^0}$.
To simplify the notation and to describe DeC, we introduce the vector of states for the variable $\mathbf{c}$ at all subtimesteps[2]

$$\underline{\mathbf{c}} := (\mathbf{c}^0,\dots,\mathbf{c}^M) \in \mathbb{R}^{M\times I}, \text{ such that} \tag{19}$$

$$\mathcal{L}^1(\underline{\mathbf{c}}) := \mathcal{L}^1(\mathbf{c}^0,\dots,\mathbf{c}^M) \text{ and } \mathcal{L}^2(\underline{\mathbf{c}}) := \mathcal{L}^2(\mathbf{c}^0,\dots,\mathbf{c}^M). \tag{20}$$

Now, the DeC algorithm uses a combination of the $\mathcal{L}^1$ and $\mathcal{L}^2$ operators to provide an iterative procedure. The aim is to recursively approximate $\underline{\mathbf{c}}^*$, the numerical solution of the $\mathcal{L}^2 = 0$ scheme, similarly to the Picard iterations in the continuous setting. The successive states of the iteration process will be denoted by the superscript $(k)$, where $k$ is the iteration index, e.g. $\underline{\mathbf{c}}^{(k)} \in \mathbb{R}^{M\times I}$. 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 $\mathbf{c}^{m,(k)} \in \mathbb{R}^I$. Finally, the DeC method can be written as

**DeC Algorithm**

$$\begin{aligned} &\mathbf{c}^{0,(k)} := \mathbf{c}(t^n), \quad k = 0,\dots,K, \\ &\mathbf{c}^{m,(0)} := \mathbf{c}(t^n), \quad m = 1,\dots,M \\ &\mathcal{L}^1(\underline{\mathbf{c}}^{(k)}) = \mathcal{L}^1(\underline{\mathbf{c}}^{(k-1)}) - \mathcal{L}^2(\underline{\mathbf{c}}^{(k-1)}) \text{ with } k = 1,\dots,K, \end{aligned} \tag{21}$$

[2] We provide a table with all definitions and notations in the appendix A.

where $K$ is the number of iterations that we want to compute. Using the procedure (21), we need, in particular, as many iterations as the desired order of accuracy, i.e., $K = d = M + 1$.
Notice that, in every step, we solve the equations for the unknown variables $\underline{\mathbf{c}}^{(k)}$ which appears only in the $\mathcal{L}^1$ formulation, the operator that can be easily inverted. Conversely, $\mathcal{L}^2$ is only applied to already computed predictions of the solution $\underline{\mathbf{c}}^{(k-1)}$. Therefore, the scheme 21 is completely explicit and arbitrary high order as stated in [1] with the following proposition.

**Proposition 3.1.** *Let $\mathcal{L}^1$ and $\mathcal{L}^2$ be two operators defined on $\mathbb{R}^M$, which depend on the discretization scale $\Delta = \Delta t$, such that*

- *$\mathcal{L}^1$ is coercive with respect to a norm, i.e., $\exists\, \alpha_1 > 0$ independent of $\Delta$, such that for any $\underline{\mathbf{c}}, \underline{\mathbf{d}}$ we have that*
$$\alpha_1 ||\underline{\mathbf{c}} - \underline{\mathbf{d}}|| \leq ||\mathcal{L}^1(\underline{\mathbf{c}}) - \mathcal{L}^1(\underline{\mathbf{d}})||,$$
- *$\mathcal{L}^1 - \mathcal{L}^2$ is Lipschitz with constant $\alpha_2 > 0$ uniformly with respect to $\Delta$, i.e., for any $\underline{\mathbf{c}}, \underline{\mathbf{d}}$*
$$||(\mathcal{L}^1(\underline{\mathbf{c}}) - \mathcal{L}^2(\underline{\mathbf{c}})) - (\mathcal{L}^1(\underline{\mathbf{d}}) - \mathcal{L}^2(\underline{\mathbf{d}}))|| \leq \alpha_2 \Delta ||\underline{\mathbf{c}} - \underline{\mathbf{d}}||.$$

*We also assume that there exists a unique $\underline{\mathbf{c}}^*_\Delta$ such that $\mathcal{L}^2(\underline{\mathbf{c}}^*_\Delta) = 0$. Then, if $\eta := \frac{\alpha_2}{\alpha_1}\Delta < 1$, the DeC is converging to $\underline{\mathbf{c}}^*$ and after $k$ iterations the error $||\underline{\mathbf{c}}^{(k)} - \underline{\mathbf{c}}^*||$ is smaller than $\eta^k ||\underline{\mathbf{c}}^{(0)} - \underline{\mathbf{c}}^*||$.*

**Remark 3.2.** *The DeC procedure is naturally conservative if $\mathcal{L}^1$ is conservative, but it is not positivity preserving if $\mathcal{L}^1$ is positivity preserving. Indeed, the coefficients $\theta_r^m$ of the operator $\mathcal{L}^2$ can be negative and spoil the positivity of the scheme. This is one of the points that make us modify the classical DeC into the scheme that we propose in this work.*

**Remark 3.3.** *Any DeC scheme can be interpreted as a RK scheme [6]. The main difference between RK and DeC is that the latter gives a general approach to the time discretization and does not require a specification of the coefficients for every order of accuracy. On the other side, by rewriting a DeC method as a RK scheme, it requires a number of stages equal to $K \times M = d \times (d-1)$, which is bigger than classical RK stages. However, one can notice that every subtimestep is independent of another, so one can compute sequentially the corrections and in parallel the subtimesteps, obtaining a computational cost of just $K = d$ corrections.*

**Example 3.4.** *For clarity, we provide here an example of a second order DeC scheme. To get this order of accuracy, we need $K = 2$ DeC iterations and one subtimestep $[t^n = t^{n,0}, t^{n,1} = t^{n+1}]$. Reminding that $\mathbf{c}^{0,(k)} = \mathbf{c}(t^n)\, \forall k$, the method (21) for the first step reads*

$$\begin{aligned}
&\mathcal{L}^1(\underline{\mathbf{c}}^{(1)}) \overset{!}{=} \mathcal{L}^1(\underline{\mathbf{c}}^{(0)}) - \mathcal{L}^2(\underline{\mathbf{c}}^{(0)}) \\
\Longleftrightarrow & c_i^{1,(1)} - c_i^{0,(0)} - \Delta t \beta^1 E_{i,j}(\mathbf{c}^{0,(1)}) = \\
& c_i^{1,(0)} - c_i^{0,(0)} - \Delta t \beta^1 E_i(\mathbf{c}^{0,(0)}) \\
& -c_i^{1,(0)} + c_i^{0,(0)} + \Delta t \sum_{r=0}^{M} \theta_r^1 E_i(\mathbf{c}^{r,(0)}) \\
\Longleftrightarrow & c_i^{1,(1)} = c_i^{0,(0)} + \Delta t E_i(\mathbf{c}^{0,(0)}) = c_i^{0,(0)} + \Delta t \sum_{j=1}^{I} \Big( p_{i,j}(\mathbf{c}^{0,(0)}) - d_{i,j}(\mathbf{c}^{0,(0)}) \Big)
\end{aligned}$$

*Substituting this term into the first correction steps leads finally to*

$$\begin{aligned}
&\mathcal{L}^1(\mathbf{c}^{(2)}) = \mathcal{L}^1(\mathbf{c}^{(1)}) - \mathcal{L}^2(\mathbf{c}^{(1)}) \\
\Longleftrightarrow & c_i^{1,(2)} - c_i^{0,(2)} - \Delta t E_i(\mathbf{c}^{0,(2)}) \\
& = c_i^{1,(1)} - c_i^{0,(1)} - \Delta t E_i(\mathbf{c}^{0,(1)}) \\
& - c_i^{1,(1)} + c_i^{0,(1)} + \sum_{r=0}^{1} \theta_r^1 \Delta t E_i(\mathbf{c}^{r,(1)})
\end{aligned}$$

*The correction step is not modifying the initial subtimestep. Therefore, with $\mathbf{c}^{0,(1)} = \mathbf{c}^{0,(2)}$, we get*

$$c_i^{n+1} = c_i^{1,(2)} = c_i^{0,(0)} + \sum_{r=0}^{1} \theta_r^1 \Delta t \sum_{j=1}^{I} \left( p_{i,j}(\mathbf{c}^{r,(1)}) - d_{i,j}(\mathbf{c}^{r,(1)}) \right)$$

*where $\theta_0^1 = \theta_1^1 = \frac{1}{2}$. This scheme coincides with the strong stability preserving Runge-Kutta method of second order [8].*

**Remark 3.5.** *Before we modify our DeC framework, we want to give some final remarks.*
*The presented DeC approach is not the most general version. In our description we always include both endpoints in the point distribution of the subtimesteps, i.e., $t^0 = t^n$ and $t^M = t^{n+1}$. However, this is not necessary, as it is already described in [7], where also Gauss-Legendre nodes are applied. Then, the approximation at the endpoint is done via extrapolation. Nevertheless, we do not consider in this work this class of point distribution.*
*Secondly, instead of using the explicit Euler method in $\mathcal{L}^1$, explicit high-order RK methods can also be applied. In principle, this yields a faster increase of the order of accuracy in the iterative procedure, but it has been shown, that it leads also to some problems of smoothness of the error behaviour as it is described in [6], which results in a drop down of the expected accuracy order. However, we will consider this approach in future research.*

# 4 Modified Patankar Deferred Correction Scheme

In this section, we are going to propose a positivity preserving, conservative and arbitrary high-order scheme, that will be denoted as **modified Patankar Deferred Correction (mPDeC)**.
The DeC procedure (21) serves us as a starting point to construct this scheme, and, thanks to its structure, we will be able to prove the hypotheses of Proposition 3.1. This yield us directly the desired order condition for our modified DeC scheme without performing a specific Taylor expansion for every order of accuracy. We will adapt DeC in such a way to obtain all the properties we are interested in.
The conservation can be easily guaranteed by the consistency of the two operators, i.e., by the consistency of the two schemes described by $\mathcal{L}^1$ and by $\mathcal{L}^2$.
Conversely, more effort is required to produce a positivity preserving scheme. For this purpose, we follow the ideas of Patankar [20] and Burchard et al. [4] of weighting the destruction and production terms in the scheme. Their aim is to obtain a mass matrix shaped as in the modified Patankar scheme (14) where all of the positive terms are collected on the diagonal, while the negative terms are put in the non-diagonal entries. This will guarantee that the mass matrix is diagonally dominant by columns, with positive diagonal values, and, thus, its inverse will be positive. Therefore, we introduce some coefficients similar to the ones proposed in (14).
Finally, as we have seen in the example (2.3), an explicit scheme is not positivity preserving and the investigation in [4, 16, 11] support our decision to modify the DeC scheme in order to get a *fully* implicit method. Because of all the above mentioned considerations, we came to the conclusion of modifying the $\mathcal{L}^2$ operator, to make it fully implicit. In particular, it has to depend on both the previous and the current corrections of the DeC procedure. We redefine it as follows.

$$\begin{aligned}
&\mathcal{L}^2(\mathbf{c}^{0,(k-1)}, \dots, \mathbf{c}^{M,(k-1)}, \mathbf{c}^{0,(k)}, \dots, \mathbf{c}^{M,(k)}) = \mathcal{L}^2(\underline{\mathbf{c}}^{(k-1)}, \underline{\mathbf{c}}^{(k)}) := \\
&\begin{cases}
c_i^{M,(k-1)} - c_i^{0,(k-1)} - \sum\limits_{r=0}^{M} \theta_r^M \Delta t \sum\limits_{j=1}^{I} \left( p_{i,j}(\mathbf{c}^{r,(k-1)}) \frac{c^{M,(k)}_{\gamma(j,i,\theta_r^M)}}{c^{M,(k-1)}_{\gamma(j,i,\theta_r^M)}} - d_{i,j}(\mathbf{c}^{r,(k-1)}) \frac{c^{M,(k)}_{\gamma(i,j,\theta_r^M)}}{c^{M,(k-1)}_{\gamma(i,j,\theta_r^M)}} \right), \forall i = 1, \dots, I \\
\vdots \\
c_i^{1,(k-1)} - c_i^{0,(k-1)} - \sum\limits_{r=0}^{M} \theta_r^1 \Delta t \sum\limits_{j=1}^{I} \left( p_{i,j}(\mathbf{c}^{r,(k-1)}) \frac{c^{1,(k)}_{\gamma(j,i,\theta_r^1)}}{c^{1,(k-1)}_{\gamma(j,i,\theta_r^1)}} - d_{i,j}(\mathbf{c}^{r,(k-1)}) \frac{c^{1,(k)}_{\gamma(i,j,\theta_r^1)}}{c^{1,(k-1)}_{\gamma(i,j,\theta_r^1)}} \right), \forall i = 1, \dots, I
\end{cases}, \\
\end{aligned} \tag{22}$$

where $\gamma(a,b,\theta) = a$ if $\theta > 0$ and $\gamma(a,b,\theta) = b$ if $\theta < 0$.

**Remark 4.1.** *The modification of the scheme is done only through the coefficients* $\frac{c_j^{m,(k)}}{c_j^{m,(k-1)}}$ *on both the production and the destruction terms. The fact that these coefficients depend on the new correction* $(k)$ *means that we are modifying the mass matrix of the whole DeC correction step.*
*These coefficients allow to choose in which term of the mass matrix we want to put each term* $\theta_r^m p_{i,j}$ *and* $\theta_r^m d_{i,j}$*, according to the sign of the* $\theta$ *coefficient. The pseudo-algorithm 1 provides the construction steps of the mass matrix, see B. There, it is straightforward to see that the diagonal terms are all positive and the off–diagonal are all negative. The index* $\gamma$ *takes care of the sign of the destruction and production terms which are added in the mass matrix. It is inspired by the explanation given in [14, Remark 2.5], that states that, when negative entries in the Butcher Tableau of the RK scheme appear, one has to interchange the destruction terms with the production ones to guarantee the positivity preserving property. With the* $\gamma$ *function we are taking this into account. In our opinion, it is complicated and unclear to investigate higher order* $(> 3)$ *RK schemes properties because of these exchanges depending on the Butcher Tableau. While, with this DeC approach, we can in few lines generalize every order scheme.*
*Moreover, it is helpful to notice that the coefficients that we are using to modify the contributions, namely* $\frac{c_j^{m,(k)}}{c_j^{m,(k-1)}}$*, are converging to 1 as the iteration index of the DeC increases. In subsection 4.2 we will make this statement more precise and we will study how fast these coefficients converge to 1.*

Most of the terms in the $\mathcal{L}^1$ operator will cancel out through the iteration process, therefore we keep the $\mathcal{L}^1$ operator as presented in the original DeC (18).

$$\mathcal{L}^1(\mathbf{c}^{0,(k)},\dots,\mathbf{c}^{M,(k)}) = \begin{cases} c_i^{M,(k)} - c_i^{0,(k)} - \beta^M \Delta t \left( \sum_{j=1}^I p_{i,j}(\mathbf{c}^{0,(k)}) - \sum_{j=1}^I d_{i,j}(\mathbf{c}^{0,(k)}) \right), \forall i = 1,\dots,I \\ \vdots \\ c_i^{1,(k)} - c_i^{0,(k)} - \beta^1 \Delta t \left( \sum_{j=1}^I p_{i,j}(\mathbf{c}^{0,(k)}) - \sum_{j=1}^I d_{i,j}(\mathbf{c}^{0,(k)}) \right), \forall i = 1,\dots,I \end{cases}. \tag{23}$$

Now, we propose the modified Patankar DeC scheme as follows.

**mPDeC Algorithm**

$$\begin{aligned} &\mathbf{c}^{0,(k)} := \mathbf{c}(t^n), \quad k = 0,\dots,K, \\ &\mathbf{c}^{m,(0)} := \mathbf{c}(t^n), \quad m = 1,\dots,M \\ &\mathcal{L}^1(\underline{\mathbf{c}}^{(k)}) = \mathcal{L}^1(\underline{\mathbf{c}}^{(k-1)}) - \mathcal{L}^2(\underline{\mathbf{c}}^{(k-1)},\underline{\mathbf{c}}^{(k)}) \text{ with } k = 1,\dots,K. \end{aligned} \tag{24}$$

One can notice that, using the fact that initial states $c_i^{0,(k)}$ are identical for any correction $(k)$, the DeC correction step (24) can be rewritten for $k = 1,\dots,K$, $m = 1,\dots,M$ and $\forall i \in I$ into

$$c_i^{m,(k)} - c_i^0 - \sum_{r=0}^M \theta_r^m \Delta t \sum_{j=1}^I \left( p_{i,j}(\mathbf{c}^{r,(k-1)}) \frac{c^{m,(k)}_{\gamma(j,i,\theta_r^m)}}{c^{m,(k-1)}_{\gamma(j,i,\theta_r^m)}} - d_{i,j}(\mathbf{c}^{r,(k-1)}) \frac{c^{m,(k)}_{\gamma(i,j,\theta_r^m)}}{c^{m,(k-1)}_{\gamma(i,j,\theta_r^m)}} \right) = 0. \tag{25}$$

We keep both formulations (24) and (25) to prove different properties. The DeC formulation (24) will help us to demonstrate the accuracy order of the scheme whereas formulation (25) will be used to prove conservation and positivity. Before we start to prove these properties, we give a small example to get used to the formulation (24). Furthermore, we like to mention that although the algorithm (24) seems quite complex, it is actually easy to implement. We put a small pseudo-code in the B and we refer to the repository [3] for a Julia version of the code.

**Example 4.2.** *We give a small example of the constructed method, applying the DeC approach at second order of accuracy, as already considered in example 3.4, i.e., $K = 2$ DeC iterations and one subtimestep $[t^n = t^{n,0}, t^{n,1} = t^{n+1}]$. In this case, we recall that $\theta_0^1 = \theta_1^1 = \frac{1}{2}$ and that $\mathbf{c}^{0,(0)} = \mathbf{c}^{1,(0)}$.*
*The method (24) for the first step reads*

$$
\begin{aligned}
&\mathcal{L}^1(\underline{\mathbf{c}}^{(1)}) - \mathcal{L}^1(\underline{\mathbf{c}}^{(0)}) + \mathcal{L}^2(\underline{\mathbf{c}}^{(0)}, \underline{\mathbf{c}}^{(1)}) \overset{!}{=} 0\\
\Longleftrightarrow & c_i^{1,(1)} - c_i^{0,(1)} - \Delta t \sum_{j=1}^{I} \Big( p_{i,j}(\mathbf{c}^{0,(1)}) - d_{i,j}(\mathbf{c}^{0,(1)}) \Big) =\\
& c_i^{1,(0)} - c_i^{0,(0)} - \Delta t \sum_{j=1}^{I} \Big( p_{i,j}(\mathbf{c}^{0,(0)}) - d_{i,j}(\mathbf{c}^{0,(0)}) \Big)\\
& -c^{1,(0)} + c_i^{0,(0)} + \Delta t \sum_{r=0}^{1} \theta_r^1 \sum_{j=1}^{I} \left( p_{i,j}(\mathbf{c}^{r,(0)}) \frac{c_j^{1,(1)}}{c_j^{1,(0)}} - d_{i,j}(\mathbf{c}^{r,(0)}) \frac{c_i^{1,(1)}}{c_i^{1,(0)}} \right)\\
\Longleftrightarrow & c_i^{1,(1)} = c_i^{0,(0)} + \Delta t \sum_{j=1}^{I} \left( p_{i,j}(\mathbf{c}^{0,(0)}) \frac{c_j^{1,(1)}}{c_j^{1,(0)}} - d_{i,j}(\mathbf{c}^{0,(0)}) \frac{c_i^{1,(1)}}{c_i^{1,(0)}} \right),
\end{aligned}
$$

*where the last step is obtained considering, again the fact that for the iteration (0) all the states coincide. Collecting the mass matrix terms as in (15), one can solve the previous equation for $\mathbf{c}^{1,(1)}$. Substituting this term into the second iteration step leads finally to*

$$
\begin{aligned}
&\mathcal{L}^1(\underline{\mathbf{c}}^{(2)}) = \mathcal{L}^1(\underline{\mathbf{c}}^{(1)}) - \mathcal{L}^2(\underline{\mathbf{c}}^{(1)}, \underline{\mathbf{c}}^{(2)})\\
\Longleftrightarrow & c_i^{1,(2)} - c_i^{0,(2)} - \Delta t \sum_{j=1}^{I} p_{i,j}(\mathbf{c}^{0,(2)}) + \sum_{j=1}^{I} d_{i,j}(\mathbf{c}^{0,(2)})\\
& = c_i^{1,(1)} - c_i^{0,(1)} - \Delta t \sum_{j=1}^{I} p_{i,j}(\mathbf{c}^{0,(1)}) + \sum_{j=1}^{I} d_{i,j}(\mathbf{c}^{0,(1)})\\
& - c_i^{1,(1)} + c_i^{0,(1)} + \sum_{r=0}^{1} \theta_r^1 \Delta t \left( \sum_{j=1}^{I} p_{i,j}(\mathbf{c}^{r,(1)}) \frac{c_j^{1,(2)}}{c_j^{1,(1)}} - \sum_{j=1}^{I} d_{i,j}(\mathbf{c}^{r,(1)}) \frac{c_i^{1,(2)}}{c_i^{1,(1)}} \right).
\end{aligned}
$$

*The correction step has no effect on the initial subtimestep. Therefore, we get with $\mathbf{c}^{0,(1)} = \mathbf{c}^{0,(2)}$:*

$$
c_i^{n+1} = c_i^{1,(2)} = c_i^{0,(0)} + \sum_{r=0}^{1} \theta_r^1 \Delta t \left( \sum_{j=1}^{I} p_{i,j}(\mathbf{c}^{r,(1)}) \frac{c_j^{1,(2)}}{c_j^{1,(1)}} - \sum_{j=1}^{I} d_{i,j}(\mathbf{c}^{r,(1)}) \frac{c_i^{1,(2)}}{c_i^{1,(1)}} \right)
$$

*where $\theta_0^1 = \theta_1^1 = \frac{1}{2}$. This scheme coincides with a modified Runge Kutta Patankar scheme of second order as it is presented in [14] .*

## 4.1 Conservation and positivity of modified Patankar DeC

In this section, we are proving that the proposed scheme is unconditionally conservative and positivity preserving.

[3] https://git.math.uzh.ch/abgrall_group/deferred-correction-patankar-scheme

**Theorem 4.3.** *The mPDeC scheme in (25) is unconditionally conservative for all substages, i.e.,*

$$\sum_{i=1}^{I} c_i^{m,(k)} = \sum_{i=1}^{I} c_i^0,$$

*for all $k = 1, \dots, K$ and $m = 0, \dots, M$.*

*Proof.* Using formulation (25), we can easily see that $\forall k, m$

$$\sum_{i \in I} c_i^{m,(k)} - \sum_{i \in I} c_i^0 = \tag{26}$$

$$= \Delta t \sum_{i,j=1}^{I} \sum_{r=0}^{M} \theta_r^m \left( p_{i,j}(\mathbf{c}^{r,(k-1)}) \frac{c^{m,(k)}_{\gamma(j,i,\theta_r^m)}}{c^{m,(k-1)}_{\gamma(j,i,\theta_r^m)}} - d_{i,j}(\mathbf{c}^{r,(k-1)}) \frac{c^{m,(k)}_{\gamma(i,j,\theta_r^m)}}{c^{m,(k-1)}_{\gamma(i,j,\theta_r^m)}} \right) = \tag{27}$$

$$= \Delta t \sum_{i,j=1}^{I} \sum_{r=0}^{M} \theta_r^m \left( d_{j,i}(\mathbf{c}^{r,(k-1)}) \frac{c^{m,(k)}_{\gamma(j,i,\theta_r^m)}}{c^{m,(k-1)}_{\gamma(j,i,\theta_r^m)}} - d_{i,j}(\mathbf{c}^{r,(k-1)}) \frac{c^{m,(k)}_{\gamma(i,j,\theta_r^m)}}{c^{m,(k-1)}_{\gamma(i,j,\theta_r^m)}} \right) = \tag{28}$$

$$= \Delta t \sum_{r=0}^{M} \theta_r^m \left( \sum_{i,j=1}^{I} d_{j,i}(\mathbf{c}^{r,(k-1)}) \frac{c^{m,(k)}_{\gamma(j,i,\theta_r^m)}}{c^{m,(k-1)}_{\gamma(j,i,\theta_r^m)}} - \sum_{i,j=1}^{I} d_{i,j}(\mathbf{c}^{r,(k-1)}) \frac{c^{m,(k)}_{\gamma(i,j,\theta_r^m)}}{c^{m,(k-1)}_{\gamma(i,j,\theta_r^m)}} \right) = 0. \tag{29}$$

To get this result, we have just used the definition of the scheme (25) in (27) and the property (4) of the production and destruction operators $d_{i,j} = p_{j,i}$ in (28). In the last step, we have exchanged the sums over $j$ and $i$.

□

To demonstrate the positivity of the scheme, we introduce some preliminary results.

**Lemma 4.4.** *The mass matrix of every correction step of the mPDeC scheme described in (25) is diagonal dominant by columns.*

*Proof.* At each step $(m, k)$ we are solving an implicit linear system where the mass matrix is given by

$$\mathrm{M}(\mathbf{c}^{m,(k-1)})_{ij} = \begin{cases} 1 + \Delta t \sum\limits_{r=0}^{M} \sum\limits_{l=1}^{I} \frac{\theta_r^m}{c_i^{m,(k-1)}} \left( d_{i,l}(\mathbf{c}^{r,(k-1)}) \mathbb{1}_{\{\theta_r^m > 0\}} - p_{i,l}(\mathbf{c}^{r,(k-1)}) \mathbb{1}_{\{\theta_r^m < 0\}} \right) & \text{for } i = j \\ -\Delta t \sum\limits_{r=0}^{M} \frac{\theta_r^m}{c_j^{m,(k-1)}} \left( p_{i,j}(\mathbf{c}^{r,(k-1)}) \mathbb{1}_{\{\theta_r^m > 0\}} - d_{i,j}(\mathbf{c}^{r,(k-1)}) \mathbb{1}_{\{\theta_r^m < 0\}} \right) & \text{for } i \neq j \end{cases}. \tag{30}$$

Under the assumption that $p_{i,j}$ and $d_{i,j}$ are always positive, it is straightforward to see that all the terms of the sum of $\mathrm{M}(\mathbf{c}^{m,(k-1)})_{ii}$ are positive by construction and that all the terms of the sum of the non-diagonal terms $\mathrm{M}(\mathbf{c}^{m,(k-1)})_{ij}$ for $i \neq j$ are negative. Moreover, we can demonstrate that

$$|\mathrm{M}(\mathbf{c}^{m,(k-1)})_{ii}| = \mathrm{M}(\mathbf{c}^{m,(k-1)})_{ii} > \sum_{j=1, j\neq i}^{I} -\mathrm{M}(\mathbf{c}^{m,(k-1)})_{ji} = \sum_{j=1, j \neq i}^{I} |\mathrm{M}(\mathbf{c}^{m,(k-1)})_{ji}|, \tag{31}$$

by showing

$$\begin{aligned} \mathrm{M}(\mathbf{c}^{m,(k-1)})_{ii} = 1 + \Delta t \sum_{r=0}^{M} \sum_{j=1}^{I} \frac{\theta_r^m}{c_i^{m,(k-1)}} \left( d_{i,j}(\mathbf{c}^{r,(k-1)}) \mathbb{1}_{\{\theta_r^m > 0\}} - p_{i,j}(\mathbf{c}^{r,(k-1)}) \mathbb{1}_{\{\theta_r^m < 0\}} \right) > \\ > \Delta t \sum_{r=0}^{M} \sum_{j=1}^{I} \frac{\theta_r^m}{c_i^{m,(k-1)}} \left( p_{j,i}(\mathbf{c}^{r,(k-1)}) \mathbb{1}_{\{\theta_r^m > 0\}} - d_{j,i}(\mathbf{c}^{r,(k-1)}) \mathbb{1}_{\{\theta_r^m < 0\}} \right) = \\ = - \sum_{j=1, j\neq i}^{I} \mathrm{M}(\mathbf{c}^{m,(k-1)})_{ji} = \sum_{j=1, j \neq i}^{I} |\mathrm{M}(\mathbf{c}^{m,(k-1)})_{ji}|, \end{aligned} \tag{32}$$

where we have used the property of the $p$ and $d$ matrices to obtain the previous computation. Finally, this proves that the mass matrix is diagonally dominant by columns. □

Using Lemma 4.4 we prove the following theorem.

**Theorem 4.5.** *The mPDeC scheme defined in (25) is positivity preserving, i.e., if* $\mathbf{c}^0 > 0$ *then* $\mathbf{c}^{m,(k)} > 0$, *for all* $m = 1, \dots, M$ *and* $k = 1, \dots, K$.

*Proof.* Using lemma 4.4, we can prove that the inverse of any mass matrix obtained from the DeC iterations is positive, i.e., $(\mathrm{M}^{-1})_{ij} \geq 0$, $\forall i, j$. The proof follows the path of what was proposed in [14]. Using the Jacobi method, we can converge to $\mathrm{M}^{-1}$ with iterative matrices $Z^{(s)}$ for $s \in \mathbb{N}$, where

$$Z^{(s+1)} := (I - D^{-1}\mathrm{M})Z^{(s)} + D^{-1}, \text{ with } Z^{(0)} = I. \tag{33}$$

Here, $I$ is the identity and $D$ is the diagonal of M. If we denote the iteration matrix as $B := I - D^{-1}\mathrm{M}$, we can see that it has spectral radius smaller than one, since M is diagonally dominant. This means that the Jacobi method is convergent to $\mathrm{M}^{-1}$. Now, since $B > 0$ and $D^{-1} > 0$ from previous lemma 4.4 and, by induction, also $Z^{(s)}$ is positive, we can say that $\mathrm{M}^{-1} = \lim\limits_{s \to \infty} Z^{(s)}$ will be positive.

□

## 4.2 Convergence order

To prove that the solution of the mPDeC procedure is high-order accurate, we mimic the proof of the original DeC convergence as in [1]. We denote by $\underline{\mathbf{c}}^*$ the solution of the $\mathcal{L}^2$ operator, i.e., $\mathcal{L}^2(\underline{\mathbf{c}}^*, \underline{\mathbf{c}}^*) = 0$. This solution $\underline{\mathbf{c}}^*$ coincides with the solution of the classical $\mathcal{L}^2$ operator defined in (16).
We want to prove that for each iteration step the following inequalities are fulfilled:

$$\begin{aligned} ||\underline{\mathbf{c}}^{(k)} - \underline{\mathbf{c}}^*|| \leq & C_0 ||\mathcal{L}^1(\underline{\mathbf{c}}^{(k)}) - \mathcal{L}^1(\underline{\mathbf{c}}^*)|| = && (34)\\ = & C_0 ||\mathcal{L}^1(\underline{\mathbf{c}}^{(k-1)}) - \mathcal{L}^2(\underline{\mathbf{c}}^{(k-1)}, \underline{\mathbf{c}}^{(k)}) - \mathcal{L}^1(\underline{\mathbf{c}}^*) + \mathcal{L}^2(\underline{\mathbf{c}}^*, \underline{\mathbf{c}}^*)|| \leq && (35)\\ \leq & C\Delta t ||\underline{\mathbf{c}}^{(k-1)} - \underline{\mathbf{c}}^*|| && (36) \end{aligned}$$

which implies that for each iteration step we obtain one order of accuracy more than the previous iteration. After $K$ iterations we, finally, get

$$||\underline{\mathbf{c}}^{(K)} - \underline{\mathbf{c}}^*|| \leq C^K \Delta t^K ||\underline{\mathbf{c}}^0 - \underline{\mathbf{c}}^*||. \tag{37}$$

To prove that the inequalities (34) and (36) are valid, we have to demonstrate the following

1. the coercivity of the operator $\mathcal{L}^1$ (as in the inequality (34))
2. the Lipschitz inequality for operator $\mathcal{L}^1 - \mathcal{L}^2$ used in (36)
3. the high-order accuracy of the operator $\mathcal{L}^2$, i.e., $||\underline{\mathbf{c}}^* - \underline{\mathbf{c}}^{exact}|| \leq C_d \Delta t^d$.

Let us start with the coercivity lemma.

**Lemma 4.6** (Coercivity of $\mathcal{L}^1$)**.** *Given any* $\underline{\mathbf{c}}^{(k)}, \underline{\mathbf{c}}^* \in \mathbb{R}^{M \times I}$, *there exists a positive* $C_0$, *such that, the operator* $\mathcal{L}^1$ *verifies*

$$||\mathcal{L}^1(\underline{\mathbf{c}}^{(k)}) - \mathcal{L}^1(\underline{\mathbf{c}}^*)|| \geq C_0 ||\underline{\mathbf{c}}^{(k)} - \underline{\mathbf{c}}^{(*)}||. \tag{38}$$

*Proof.* We remind that the beginning states coincide for all the variables, i.e., $\mathbf{c}^{0,(k)} = \mathbf{c}^{0,*} = \mathbf{c}^0$. So, the evolution part simplifies in the two operators and we get the following relation

$$\mathcal{L}^1(\underline{\mathbf{c}}^{(k)}) - \mathcal{L}^1(\underline{\mathbf{c}}^*) = (\underline{\mathbf{c}}^{(k)} - \underline{\mathbf{c}}^{(*)}). \tag{39}$$

This proves that with constant $C_0 = 1$ the equation (38) holds.

□

Before proving Lipschitz continuity, we need two lemmas. The first one proves that each stage of the scheme is a first order approximation of the previous timestep.

**Lemma 4.7.** *For every subtimestep $m = 1, \dots, M$ and correction $k = 1, \dots, K$, there exists a matrix $G$, such that*

$$\mathbf{c}^{m,(k)} = \mathbf{c}^0 + \Delta t G(\mathbf{c}^{m,(k-1)})\mathbf{c}^0 \tag{40}$$

*holds. Moreover, $G(\mathbf{c}^{m,(k-1)}) = W(\mathbf{c}^{m,(k-1)}) + \mathcal{O}(\Delta t)$, where $W$ does not depend on $\Delta t$.*

*Proof.* For any $m = 0, \dots, M$ and $k = 0, \dots, K$, the equation (25) tells us that the mass matrix $\mathrm{M}(\mathbf{c}^{m,(k-1)})$ can be written as $\mathrm{M}(\mathbf{c}^{m,(k-1)}) = I - \Delta t W(\mathbf{c}^{m,(k-1)})$ where $W$ does not depend on $\Delta t$, but only on $\mathbf{c}^{m,(k-1)}$ and the production–destruction functions. It is defined as

$$W(\mathbf{c}^{m,(k-1)})_{ij} = \begin{cases} -\sum\limits_{r=0}^{M}\sum\limits_{l=1}^{I} \frac{\theta_r^m}{c_i^{m,(k-1)}} \left( d_{i,l}(\mathbf{c}^{r,(k-1)})\mathbb{1}_{\{\theta_r^m>0\}} - p_{i,l}(\mathbf{c}^{r,(k-1)})\mathbb{1}_{\{\theta_r^m<0\}} \right) & \text{for } i = j \\ +\sum\limits_{r=0}^{M} \frac{\theta_r^m}{c_j^{m,(k-1)}} \left( p_{i,j}(\mathbf{c}^{r,(k-1)})\mathbb{1}_{\{\theta_r^m>0\}} - d_{i,j}(\mathbf{c}^{r,(k-1)})\mathbb{1}_{\{\theta_r^m<0\}} \right) & \text{for } i \neq j \end{cases}. \tag{41}$$

This leads to an inverse

$$(\mathrm{M}(\mathbf{c}^{m,(k-1)}))^{-1} = I + \Delta t W(\mathbf{c}^{m,(k-1)}) + \mathcal{O}(\Delta t^2).$$

Now, we can define $G$ by

$$G(\mathbf{c}^{m,(k-1)}) := \frac{1}{\Delta t}\left( (\mathrm{M}(\mathbf{c}^{m,(k-1)}))^{-1} - I \right) = W(\mathbf{c}^{m,(k-1)}) + \mathcal{O}(\Delta t).$$

So, we can write

$$\mathbf{c}^{m,(k)} = (\mathrm{M}(\mathbf{c}^{m,(k-1)}))^{-1}\mathbf{c}^0 = \mathbf{c}^0 + \Delta t G(\mathbf{c}^{m,(k-1)})\mathbf{c}^0. \tag{42}$$

□

With the following lemma, we prove that the mPDeC process generates a Cauchy sequence similar to the continuous Picard iterations. Moreover, at each iteration, we differ from the previous step by an error of one order of accuracy more. We will drop the dependency on the subtimestep $m$, as all the relations hold for all of them.

**Lemma 4.8.** *Let $\mathbf{c}^{(k)}$ and $\mathbf{c}^{(k-1)} \in \mathbb{R}^I$ verifying Lemma 4.7, then*

$$\frac{c_i^{(k)}}{c_i^{(k-1)}} = 1 + \Delta t^{k-1} g_i + \mathcal{O}(\Delta t^k) \tag{43}$$

*holds where $g_i$ are constants independent from $\Delta t$.*

*Proof.* We prove the lemma by induction.
For $k = 1$, equation (43) follows directly from Lemma 4.7, i.e., $\frac{c_i^{(1)}}{c_i^{(0)}} = 1 + \mathcal{O}(\Delta t)$.
Given $k \in \mathbb{N}$, as induction hypothesis, (43) holds for $k$, i.e.,

$$c_i^{(k)} = c_i^{(k-1)}\left(1 + \Delta t^{k-1} g_i\right) + \mathcal{O}(\Delta t^k), \tag{44}$$

where $g_i = G_i(\underline{\mathbf{c}}^{(k-1)})\mathbf{c}^0$ and $G_i$ denotes the $i$th row of the matrix $G$. We can prove that (43) is verified also for $k+1$. Using Lemma (4.7), we obtain

$$\begin{aligned}
\frac{c_i^{(k+1)}}{c_i^{(k)}} =& \frac{c_i^{(0)} + \Delta t G_i(\underline{\mathbf{c}}^{(k)})\mathbf{c}^{(0)}}{c_i^{(0)} + \Delta t G_i(\underline{\mathbf{c}}^{(k-1)})\mathbf{c}^{(0)}} = \\
=& \frac{\left(c_i^{(0)} + \Delta t G_i(\underline{\mathbf{c}}^{(k)})\mathbf{c}^{(0)}\right)\left(c_i^{(0)} - \Delta t G_i(\underline{\mathbf{c}}^{(k-1)})\mathbf{c}^{(0)}\right)}{\left(c_i^{(0)} + \Delta t G_i(\underline{\mathbf{c}}^{(k-1)})\mathbf{c}^{(0)}\right)\left(c_i^{(0)} - \Delta t G_i(\underline{\mathbf{c}}^{(k-1)})\mathbf{c}^{(0)}\right)} = \\
=& \frac{\left(c_i^{(0)}\right)^2 + \Delta t c_i^{(0)} G_i(\underline{\mathbf{c}}^{(k)})\mathbf{c}^{(0)} - \Delta t c_i^{(0)} G_i(\underline{\mathbf{c}}^{(k-1)})\mathbf{c}^{(0)}}{\left(c_i^{(0)}\right)^2 - \left(\Delta t G_i(\underline{\mathbf{c}}^{(k-1)})\mathbf{c}^{(0)}\right)^2} + \\
& - \frac{\left(\Delta t G_i(\underline{\mathbf{c}}^{(k-1)})\mathbf{c}^{(0)}\right)\left(\Delta t G_i(\underline{\mathbf{c}}^{(k)})\mathbf{c}^{(0)}\right)}{\left(c_i^{(0)}\right)^2 - \left(\Delta t G_i(\underline{\mathbf{c}}^{(k-1)})\mathbf{c}^{(0)}\right)^2}.
\end{aligned}$$

Inserting the induction step (44) we get

$$\begin{aligned}
\frac{c_i^{(k+1)}}{c_i^{(k)}} =& \frac{\left(c_i^{(0)}\right)^2 + \Delta t c_i^{(0)} \left(G_i\left(\mathbf{c}^{(k-1)} \bullet \left(\mathbf{1} + \Delta t^{k-1}\mathbf{g}\right) + \mathcal{O}(\Delta t^k)\right) - G_i(\mathbf{c}^{(k-1)})\right)\mathbf{c}^{(0)}}{\left(c_i^{(0)}\right)^2 - \left(\Delta t G_i(\mathbf{c}^{(k-1)})\mathbf{c}^{(0)}\right)^2} + \\
& - \frac{\left(\Delta t G_i(\mathbf{c}^{(k-1)})\mathbf{c}^{(0)}\right)\left(\Delta t G_i\left(\mathbf{c}^{(k-1)} \bullet \left(\mathbf{1} + \Delta t^{k-1}\mathbf{g}\right) + \mathcal{O}(\Delta t^k)\right)\mathbf{c}^{(0)}\right)}{\left(c_i^{(0)}\right)^2 - \left(\Delta t G_i(\mathbf{c}^{(k-1)})\mathbf{c}^{(0)}\right)^2}
\end{aligned}$$

Here, $\bullet$ denotes the Hadamard product and $\mathbf{1} := (1, \dots, 1)^T \in \mathbb{R}^I$. The induction step is evaluated for every entry $i$. Using the regularity of $G_i$, we expand its Taylor series in $\mathbf{c}^{(k-1)}$ for every constituent $i$. Thanks again to the result of Lemma (4.7), we can write

$$\begin{aligned}
\frac{c_i^{(k+1)}}{c_i^{(k)}} =& \frac{\left(c_i^{(0)}\right)^2 + \Delta t c_i^{(0)} G_i\left(\mathbf{c}^{(k-1)}\right)\mathbf{c}^{(0)} + \Delta t^k c_i^{(0)} \nabla G_i(\overline{\mathbf{c}})\mathbf{g}\mathbf{c}^{(0)} - \Delta t c_i^{(0)} G_i(\mathbf{c}^{(k-1)})\mathbf{c}^{(0)}}{\left(c_i^{(0)}\right)^2 - \left(\Delta t G_i(\mathbf{c}^{(k-1)})\mathbf{c}^{(0)}\right)^2} + \\
& - \frac{\left(\Delta t G_i(\mathbf{c}^{(k-1)})\mathbf{c}^{(0)}\right)\left(\Delta t G_i\left(\mathbf{c}^{(k-1)}\right)\mathbf{c}^{(0)} + \Delta t^k \nabla G_i(\overline{\mathbf{c}})\mathbf{g}\mathbf{c}^{(0)} + \mathcal{O}(\Delta t^k)\right)}{\left(c_i^{(0)}\right)^2 - \left(\Delta t G_i(\mathbf{c}^{(k-1)})\mathbf{c}^{(0)}\right)^2}
\end{aligned}$$

where $\overline{\mathbf{c}}$ is the point of the Lagrange form of the remainder of the Taylor expansion. Hence, we can proceed as follows

$$\begin{aligned}
\frac{c_i^{(k+1)}}{c_i^{(k)}} =& \frac{\left(c_i^{(0)}\right)^2 + \Delta t c_i^{(0)} G_i\left(\mathbf{c}^{(k-1)}\right)\ \mathbf{c}^{(0)} + \Delta t^k c_i^{(0)} \nabla G_i(\overline{\mathbf{c}})\mathbf{g}\mathbf{c}^{(0)} - \Delta t c_i^{(0)} G_i(\mathbf{c}^{(k-1)})\mathbf{c}^{(0)}}{\left(c_i^{(0)}\right)^2 - \left(\Delta t G_i(\mathbf{c}^{(k-1)})\mathbf{c}^{(0)}\right)^2} + \\
& - \frac{\left(\Delta t G_i(\mathbf{c}^{(k-1)})\mathbf{c}^{(0)}\right)^2 + \mathcal{O}(\Delta t^{k+1})}{\left(c_i^{(0)}\right)^2 - \left(\Delta t G_i(\mathbf{c}^{(k-1)})\mathbf{c}^{(0)}\right)^2} = \\
=& \frac{\left(c_i^{(0)}\right)^2 - \left(\Delta t G_i(\mathbf{c}^{(k-1)})\mathbf{c}^{(0)}\right)^2 + \Delta t^k c_i^{(0)} \nabla G_i(\overline{\mathbf{c}})\mathbf{g}\mathbf{c}^{(0)} + \mathcal{O}(\Delta t^{k+1})}{\left(c_i^{(0)}\right)^2 - \left(\Delta t G_i(\mathbf{c}^{(k-1)})\mathbf{c}^{(0)}\right)^2} = \\
=& 1 + \Delta t^k \hat{g}_i + \mathcal{O}(\Delta t^{k+1})
\end{aligned}$$

which finally proves equation (43) for $k+1$. □

Now, let us prove on the Lipschitz continuity of the operator $\mathcal{L}^1 - \mathcal{L}^2$.

**Lemma 4.9** (Lipschitz continuity of $\mathcal{L}^1 - \mathcal{L}^2$)**.** *Let $\underline{\mathbf{c}}^{(k)}, \underline{\mathbf{c}}^{(k-1)}, \underline{\mathbf{c}}^* \in \mathbb{R}_+^{M \times I}$ fulfil Lemma 4.7. Then, the operator $\mathcal{L}^1 - \mathcal{L}^2$ is Lipschitz continuous with constant $\Delta t C_L$, i.e.,*

$$||\mathcal{L}^1(\underline{\mathbf{c}}^{(k-1)}) - \mathcal{L}^2(\underline{\mathbf{c}}^{(k-1)}, \underline{\mathbf{c}}^{(k)}) - \mathcal{L}^1(\underline{\mathbf{c}}^*) + \mathcal{L}^2(\underline{\mathbf{c}}^*, \underline{\mathbf{c}}^*)|| \leq C_L \Delta t ||\underline{\mathbf{c}}^{(k-1)} - \underline{\mathbf{c}}^*||. \tag{45}$$

*Proof.* Now, we apply Lemma (4.8) to substitute the new $\mathcal{L}^2$ operator (22) with the original one of the classical DeC (17) adding an error of order $\Delta t^{k-1}$ to the operator. We get another order from the time integration, such that

$$\mathcal{L}^2(\underline{\mathbf{c}}^{(k-1)}, \underline{\mathbf{c}}^{(k)}) = \mathcal{L}^2(\underline{\mathbf{c}}^{(k-1)}) + \mathcal{O}(\Delta t^k)$$

and, trivially, $\mathcal{L}^2(\underline{\mathbf{c}}^*, \underline{\mathbf{c}}^*) = \mathcal{L}^2(\underline{\mathbf{c}}^*)$ holds. Together, we obtain

$$\left|\left|\mathcal{L}^1(\underline{\mathbf{c}}^{(k-1)}) - \mathcal{L}^2(\underline{\mathbf{c}}^{(k-1)}, \underline{\mathbf{c}}^{(k)}) - \mathcal{L}^1(\underline{\mathbf{c}}^*) + \mathcal{L}^2(\underline{\mathbf{c}}^*, \underline{\mathbf{c}}^*)\right|\right| \leq \tag{46}$$

$$\leq \left|\left|\mathcal{L}^1(\underline{\mathbf{c}}^{(k-1)}) - \mathcal{L}^2(\underline{\mathbf{c}}^{(k-1)}) - \mathcal{L}^1(\underline{\mathbf{c}}^*) + \mathcal{L}^2(\underline{\mathbf{c}}^*)\right|\right| + \mathcal{O}(\Delta t^k). \tag{47}$$

Now, we have to take care about the different variables in the operators. Let us start studying the operator $\mathcal{L}^1 - \mathcal{L}^2$. We are focusing on each line of the schemes for an arbitrary subtimestep. The difference is given by

$$\begin{aligned}
&\mathcal{L}_i^{1,m}(\mathbf{c}^{(k-1)}) - \mathcal{L}_i^{2,m}(\mathbf{c}^{(k-1)}) = \\
&\int_{t^0}^{t^m} \mathcal{I}_M\left(\left\{E_i(\mathbf{c}^{r,(k-1)})\right\}_{r=0}^M\right) - \mathcal{I}_0\left(\left\{E_i(\mathbf{c}^{r,(k-1)})\right\}_{r=0}^M\right) dt \\
&= \int_{t^0}^{t^m} (\mathcal{I}_M - \mathcal{I}_0)\left(\left\{E_i(\mathbf{c}^{r,(k-1)})\right\}_{r=0}^M\right) dt.
\end{aligned} \tag{48}$$

Now, we can compute the difference of the two terms

$$\begin{aligned}
&||\mathcal{L}_i^{1,m}(\mathbf{c}^{(k-1)}) - \mathcal{L}_i^{2,m}(\mathbf{c}^{(k-1)}) - \mathcal{L}_i^{1,m}(\mathbf{c}^*) + \mathcal{L}_i^{2,m}(\mathbf{c}^*)|| = \\
=&\left|\left|\int_{t^0}^{t^m} (\mathcal{I}_M - \mathcal{I}_0)\left(\left\{E_i(\mathbf{c}^{r,(k-1)}) - E_i(\mathbf{c}^{r,*})\right\}_{r=0}^M\right) dt\right|\right| \leq \\
\leq&\Delta t C_1 ||E_i(\underline{\mathbf{c}}^{(k-1)}) - E_i(\underline{\mathbf{c}}^*)|| \leq \\
\leq&\Delta t C_L ||\mathbf{c}^{(k-1)} - \mathbf{c}^*||.
\end{aligned} \tag{49}$$

In last step, we have used the regularity of the solutions $\underline{\mathbf{c}}^{(k-1)}$ and $\underline{\mathbf{c}}^*$ and the fact that $\mathcal{I}_M - \mathcal{I}_0$ brings an error of order zero $\mathcal{O}(1)$ times $\Delta t$ given by the time integration. Then, we have used the Lipschitz continuity of the functions $E_i$.
Overall, the constant $C_L$ depends on the operators $p$ and $d$ and the Lemma is proven.

□

Finally, we need to show that the solution $\mathbf{c}^*$ of the operator $\mathcal{L}^2(\mathbf{c}^*, \mathbf{c}^*) = 0$ is an $(M+1)$-order accurate solution. This is given directly by the definition of the operator (22), since it is an $(M+1)$-order accurate approximation of the original problem (1) when the two input coincide and, thus, the modification coefficients become 1 and the operator becomes the original one (17).

**Theorem 4.10** (Convergence of mPDeC)**.** *Let $\mathcal{L}^1(\cdot)$ and $\mathcal{L}^2(\cdot,\cdot)$ be the operators defined in* (23) *and* (22) *respectively. The mPDeC procedure* (24) *gives an approximation solution with order of accuracy equal to* $\min(M+1, K)$.

*Proof.* With Lemma 4.6 we proved the coercivity of the operator $\mathcal{L}^1$, which verifies the inequality in (34). The definition of the mPDeC scheme (24) gives us the equality (35) and the Lipschitz continuity lemma 4.9 proves the inequality (36). Moreover, we know that $\underline{\mathbf{c}}^*$ is an $(M+1)$-accurate approximation of the $\underline{\mathbf{c}}^{ex}$ exact solution.
So, overall, we have

$$||\underline{\mathbf{c}}^{(K)} - \underline{\mathbf{c}}^{ex}|| \leq ||\underline{\mathbf{c}}^* - \underline{\mathbf{c}}^{ex}|| + (C\Delta t)^K ||\underline{\mathbf{c}}^* - \underline{\mathbf{c}}^{(0)}|| \leq C^* \Delta t^{M+1} + (C\Delta t)^K. \tag{50}$$

□

All the desired properties (unconditionally positivity, unconditionally conservation and high-order accuracy) are fulfilled by the proposed scheme.

# 5 Numerics

In this section, we validate our theoretical investigation of section 4 considering some test cases from [14, 4]. We focus here only on systems of ordinary differential equations (ODE) (stiff and non-stiff). However, the mPDeC schemes can be in general used as time-integration methods for a semidiscrete formulation of partial differential equations, where the spatial discretization is already provided by RD, DG, FR, (c.f. [2, 3, 23]) or your favourite space discretization method.
As part of future research, we will consider these schemes in real applications like non-equilibrium flows or shallow water equations as it was already done, for example, for MPRK together with a WENO approach in [11] or a DG one in [18]. In this work we focus on systems of ODEs. In all the numerical tests, we applied the mPDeC approach on equidistant subtimestep points distributions.

## 5.1 Linear Model

We start by considering a simple linear test case proposed in [4, 18]. The initial value problem for the PDS is given by

$$\begin{aligned} c_1'(t) &= c_2(t) - 5c_1(t), \qquad & c_2'(t) &= 5c_1(t) - c_2(t), \\ c_1(0) &= c_1^0 = 0.9, & c_2(0) &= c_2^0 = 0.1\,. \end{aligned} \tag{51}$$

The initial values of (51) are positive and we can rewrite the right hand side of the ODE system in a PDS format as follows

$$p_{1,2}(\mathbf{c}) = d_{2,1}(\mathbf{c}) = c_2, \quad p_{2,1}(\mathbf{c}) = d_{1,2}(\mathbf{c}) = 5c_1$$

and $p_{i,i}(\mathbf{c}) = d_{i,i}(\mathbf{c}) = 0$ for $i = 1, 2$. The system describes the exchange of mass between two constituents. The analytical solution is given by

$$c_1(t) = \frac{1}{6}\left(1 + \frac{13}{5}\exp(-6t)\right) \text{ and } c_2(t) = 1 - c_1(t). \tag{52}$$

The problem is considered on the time interval $[0, 1.75]$ and, analogously to [4], we use $\Delta t = 0.25$ in the simulations. In Figure (2) we plot the analytical solution (dotted, blue line) and the approximated solutions using 2-nd (solid line, green) and 5-th (dash-dotted line, black) order mPDeC methods. The purple lines represent the sum of the constituents and they are constantly equal to 1, since the methods are conservative. Qualitatively, we see that the 5-th order method approximates better the analytical solutions. Furthermore, to verify the order of convergence of the methods, we consider also the error behavior of the different order

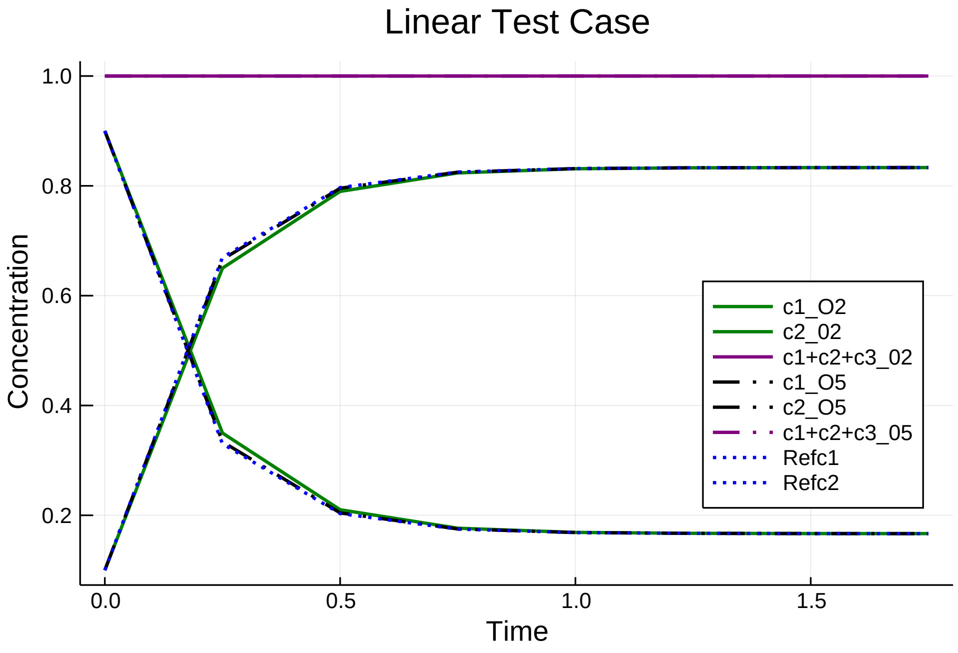

Figure 2: Second and fifth order methods together with the reference solution (52)

schemes. Differently from Kopecz and Meister [14, 15] instead of calculating the relative errors, we compute the absolute discrete $L^2$ errors taken over all the timesteps $\{t^n\}_{n=0}^N$ and all the constituents:

$$\mathrm{E} = \frac{1}{N}\sum_{n=1}^{N}\left(\frac{1}{I}\sum_{i=1}^{I}\left(c_i(t^n)-c_i^n\right)^2\right)^{\frac{1}{2}}. \tag{53}$$

After a comparison between the final time error and the one proposed (53), we do not observe much discrepancy. Therefore, we will provide only results obtained with (53).
In Figure 3, the left picture shows the error decay for mPDeC schemes at different discretization scales $\Delta t$. In the right picture, we plot the slope of the error decay for different orders of accuracy. These graphs demonstrate the high-order accuracy of the proposed methods and the expected convergence rates, validating the theoretical results. It is also possible to test the scheme with higher order of accuracy. However, we have notice a reduction of the order as we reach orders higher than 10, probably due to Runge phenomena. These are well known issues that arise also with the usual DeC methods [7] using equidistant points distribution in the subtimesteps. A possible solution of this problem can be the usage of Gauß-Lobatto nodes as point distributions. This and stability investigations will be part of future research.

## 5.2 Nonlinear test problem

In this next subsection, we consider the nonlinear test problem

$$\begin{aligned} c_1'(t) &= -\frac{c_1(t)c_2(t)}{c_1(t)+1}, \\ c_2'(t) &= \frac{c_1(t)c_2(t)}{c_1(t)+1} - 0.3c_2(t), \\ c_3'(t) &= 0.3c_2(t) \end{aligned} \tag{54}$$

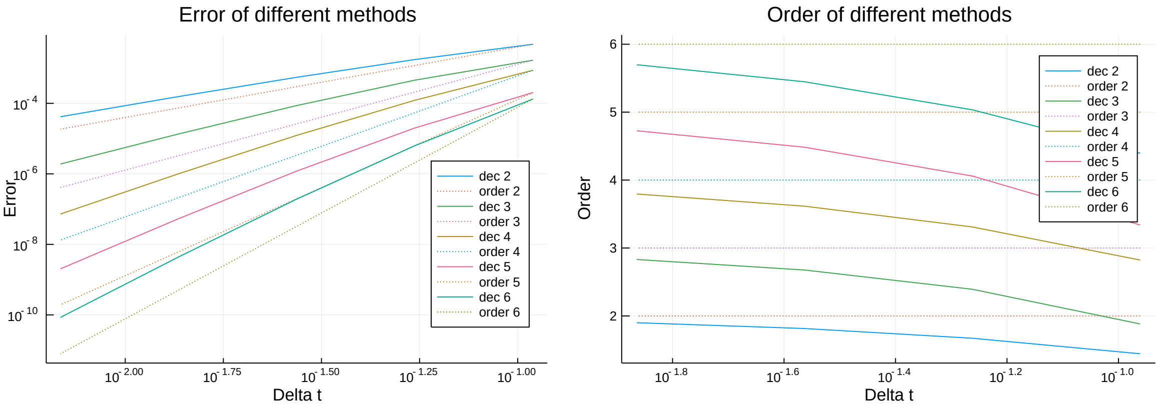

Figure 3: Second to sixth order error decay and slope of the errors

with initial condition $\mathbf{c}^0 = (9.98, 0.01, 0.01)^T$. As before, this problem was proposed in [14]. The PDS system in the matrix formulation can be expressed by

$$p_{2,1}(\mathbf{c}) = d_{1,2}(\mathbf{c}) = \frac{c_1(t)c_2(t)}{c_1(t)+1}, \quad p_{3,2}(\mathbf{c}) = d_{2,3}(\mathbf{c}) = 0.3c_2(t)$$

and $p_{i,j}(\mathbf{c}) = d_{i,j}(\mathbf{c}) = 0$ for all other combinations of $i$ and $j$. This system (54) is used to describe an algal bloom, that transforms nutrients $c_1$ via phytoplankton $c_2$ into detritus $c_3$. In our test, we consider the time interval $[0, 30]$ and $\Delta t = 0.5$. We calculate the reference solution with the strong stability preserving Runge-Kutta method 10 stages 4th order introduced by Ketcheson [13], further investigated in [22] and implemented in Julia, see [21] for details.
In Figure 4, the 6-th order mPDeC (black, dash-dotted lines) approximates very precisely the reference solution. The 2-nd order method (solid line, green) shows the same structure as the reference solution but it exhibits a severe error. However, the approximated second order solution is comparable with the results obtained in [14]. We see again that the conservation property is fulfilled in the purple lines.
Since we lack of an analytical solution, in the error plots, we compare successive errors between two refinements of the time mesh

$$\mathrm{E}_N = \frac{1}{N}\sum_{n=1}^{N}\left(\frac{1}{I}\sum_{i=1}^{I}\left(c_{i,N}^{n} - c_{i,2N}^{2n}\right)^2\right)^{\frac{1}{2}}. \tag{55}$$

Here, the subscript $N$ indicates the number of equispaced timesteps used to subdivide the total time interval. The results are presented in Figure 5. As for the linear case, we can see that the error decay fulfils the expected behavior and that the order of accuracy tends to the correct one. The slight decrease of the slope function in the right picture using sixth order can be explained by the fact that the error values are close to machine precision in that area and this causes the deprecation of the slope.

These plots verify our theoretical investigations from section 4.

## 5.3 Robertson Test case

In the last test case, we prove the robustness of the mPDeC schemes in presence of stiff problems. The proposed test is the Robertson problem for a chemical reaction system. It consists of

$$\begin{aligned} c_1'(t) &= 10^4 c_2(t)c_3(t) - 0.04c_1(t) \\ c_2'(t) &= 0.04c_1(t) - 10^4 c_2(t)c_3(t) - 3\cdot 10^7 c_2(t)^2 \\ c_3'(t) &= 3\cdot 10^7 c_2(t)^2 \end{aligned} \tag{56}$$

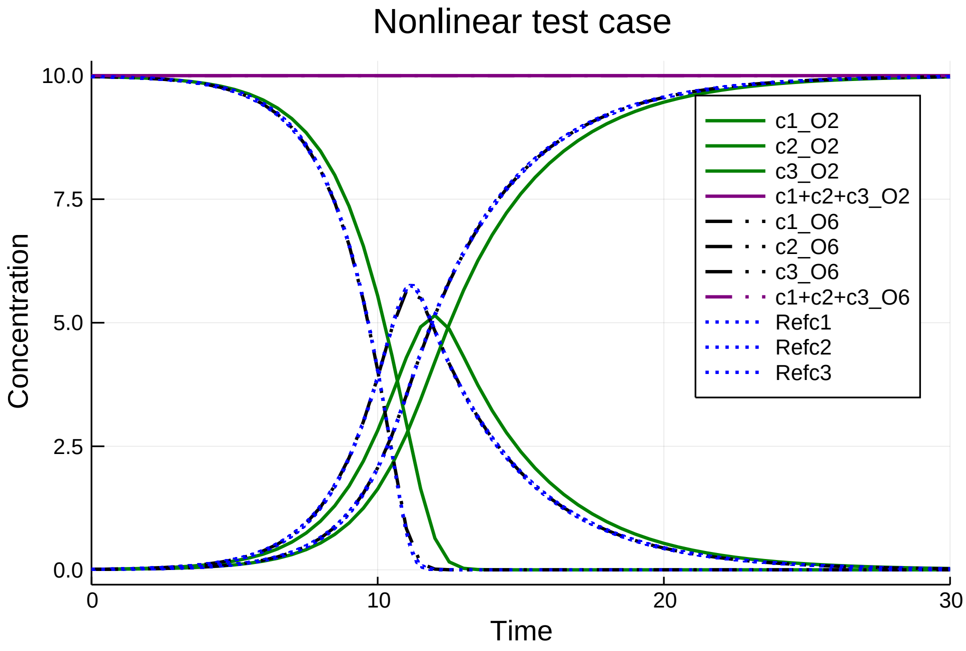

Figure 4: Second order and sixth order methods together with the reference solution (SSPRK104)

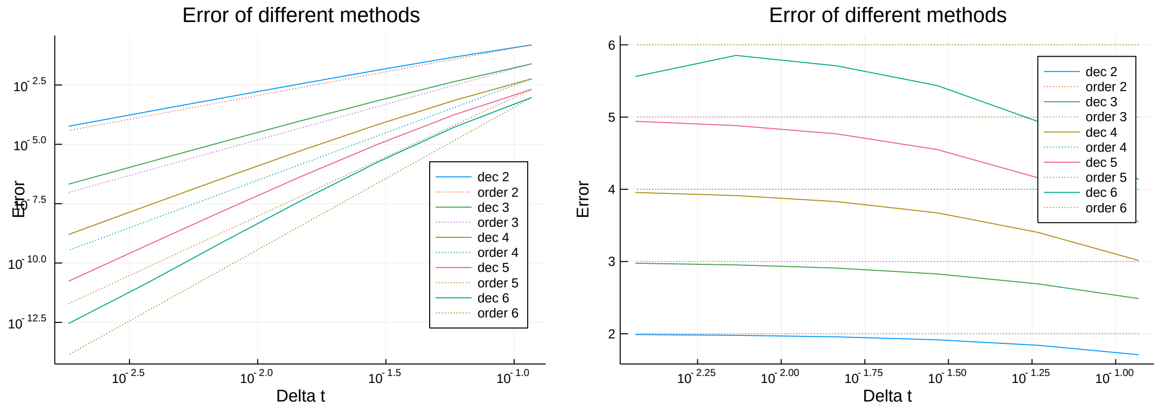

Figure 5: Second to sixth order error behaviors and slopes of the errors

with initial conditions $\mathbf{c}^0 = (1,0,0)$.[4] The time interval of interest is $[10^{-6}, 10^{10}]$. The PDS for (56) reads

$$p_{1,2}(\mathbf{c}) = d_{2,1}(\mathbf{c}) = 10^4 c_2(t)c_3(t), \quad p_{2,1}(\mathbf{c}) = d_{1,2}(\mathbf{c}) = 0.04c_1(t), \quad p_{3,2}(\mathbf{c}) = d_{2,3}(\mathbf{c}) = 3 \cdot 10^7 c_2(t)$$

and zero for the other combinations.
In the Robertson test case, the numerical scheme has to deal with several time scales. Therefore, a constant time step size is not suitable for this purpose. Following again the literature [14], we use increasing time steps $\Delta t_n = 2^{n-1}\Delta t_0$ with $\Delta t_0 = 10^{-6}$, where $n$ indicates the $n$-th timestep. To make the small $c_2$ values visible on the plot, we multiply it by $10^4$. As a comparison, we calculate the reference solution (dotted, blue line) using the function Rodas4[5] from Julia, where we split the time-interval into 55 subdomains and we solve it on every subdomain with relative tolerance $10^{-20}$ and absolute tolerance $10^{-20}$. We plot again a second order (green, solid lines) and fifth order (black, dashed-dotted lines) approximations generated by the mPDeC methods and, as it can be seen in figure 6, the designed methods produce reliable and robust results for this kind of stiff problems. As always, the conservation and the positivity properties are fulfilled. Finally, we can say that the simulations run in this section express the quality of the mPDeC schemes. Moreover, they show that all the targeted properties are obtained even for very problematic test cases.

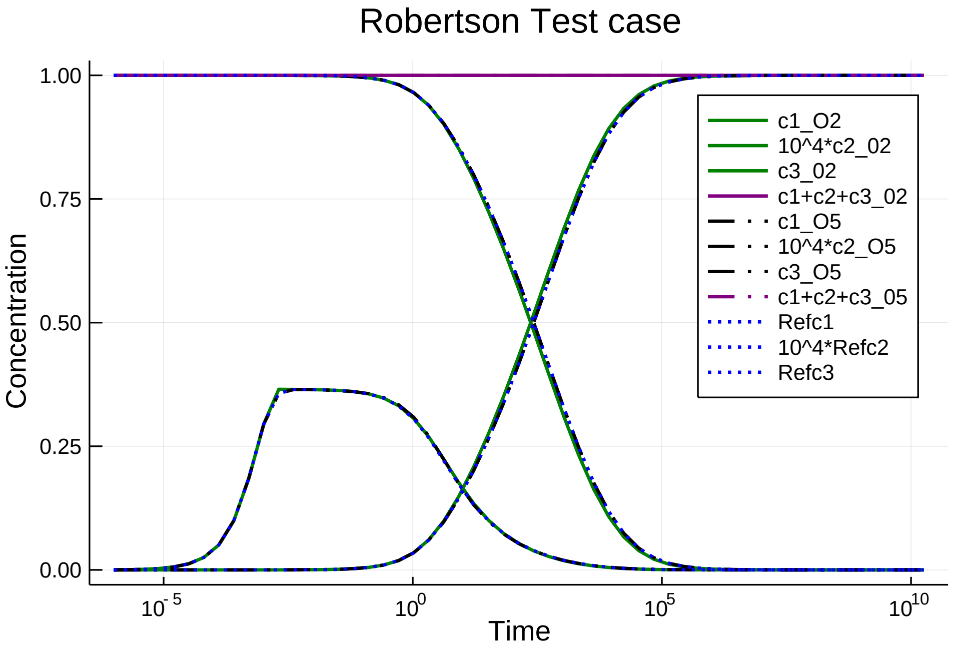

Figure 6: Second and fifth order solutions and references

## 6 Summary and Outlook

In this paper, we presented a way to build positivity preserving, conservative and arbitrary high-order numerical schemes for production-destructions systems of equations. We adapted the idea of [4] to build modified Patankar type schemes to the Deferred Correction method as an underlying scheme. By altering the $\mathcal{L}^1$ and $\mathcal{L}^2$ operators using the modified Patankar trick, we were able to obtain schemes with the desired properties. We proved that the proposed modified Patankar DeC (mPDeC) schemes are arbitrary high-order,

[4] To avoid the division by zero in the mPDeC scheme, we slightly modify the initial condition in the practical implementation, i.e., $\mathbf{c}^0 = (1-2eps, eps, eps)$ with $eps = 2.22 \cdot 10^{-16}$.

[5] A 4-th order A-stable stiffly stable Rosenbrock method with a stiff-aware 3rd order interpolant.

| Notation | Meaning |
|---:|:---|
| $I$ | Number of constituents and dimension of the ODE system |
| $i$ | Index for constituents |
| $c_i$ | Value of the $i$th constituents |
| $\mathbf{c}$ | Vector of all the constituents $\mathbf{c} = (c_1, \dots, c_I)^T$ |
| $N$ | Number of time intervals |
| $n$ | Index for a timestep |
| $t^n$ | Timestep |
| $\mathbf{c}^n$ | Variables at timestep $t^n$ |
| $M$ | Number of subtimeintervals in a timeinterval |
| $m$ | Index for subtimesteps |
| $t^{n,m} = t^m$ | Subtimestep |
| $\mathbf{c}^{n,m} = \mathbf{c}^m$ | Variable at subtimestep $m$ |
| $\underline{\mathbf{c}}$ | Vector of variables at all subtimestep $m = 0, \dots, M$ |
| $K$ | Number of iterations of the DeC procedure |
| $(k)$ | Index of the iteration |
| $\mathbf{c}^{n,m,(k)} = \mathbf{c}^{m,(k)}$ | Variables for timestep $n$ at the subtimestep $m$ and iteration $k$ |
| $\underline{\mathbf{c}}^{(k)}$ | Vector of variables for all subtimesteps $m = 0, \dots, M$ at the iteration $k$ |
| $\mathcal{L}^1(\cdot)$ | First order operator of DeC procedure |
| $\mathcal{L}^2(\cdot)$ | High order operator of DeC procedure |
| $\mathcal{L}^2(\cdot, \cdot)$ | High order operator of mPDeC procedure |
| $\underline{\mathbf{c}}^*$ | Solution of the system $\mathcal{L}^2(\underline{\mathbf{c}}^*) = 0$. |

Table 1: Notation table

conservative and positivity preserving. In numerical simulations, we confirmed our theoretical considerations with various test cases.

However, further research can be pursued in this direction. As it was investigated in [15, 14] for families of MPRK, it is possible to study the accuracy and the stability of the method varying the weightings of the production-destruction terms of the schemes. In the spirit of the work [14], a change of the weighting of the Patankar modification in the $\mathcal{L}^1$ and $\mathcal{L}^2$ operators should be easily applicable to the mPDeC schemes and theoretical investigations will be considered in future research, in particular regarding the stability conditions. Also the distribution of the subtimesteps between $t^n$ and $t^{n+1}$ plays a big role on stability and accuracy of the scheme. Many choices are valid and the possible influence of the properties of the method must be carefully analysed. This idea is already work in progress for the classical DeC approach and will be extended to the mPDeC version in the future. Finally, we want to apply and analyse this type of schemes in context of partial differential equations. Here, we focus on applications and problems as described in [11, 12, 18]. As one can see, there are still many open questions and tasks for the mPDeC schemes and we are looking forward to continue our work in this field.

# A Notation

We provide a small table 1 with the notation of symbols used along the paper. Even if some of the notations are ambiguous, the used indices should always clarify the referred meaning. We prefer to keep this notation to keep fluid the reading.

# B Algorithm

We present a pseudo-code for the creation of the mass matrix in Algorithm 1 and one for the mPDeC algorithm in box Algorithm 2. Both algorithms are very simple. The first one consists of 3 loops: 2 for the constituents $i, j = 1, \dots, I$ and one for the subtimesteps $r = 0, \dots, M$ and an if statement. The second one consists of 3 nested loops: one for timesteps $\{t^n\}_{n=0}^N$, one for corrections of the DeC algorithm $k = 1, \dots, K$ and one for the subtimesteps $m = 1, \dots, M$.

**Algorithm 1** Mass

```
Require: Production-destruction functions $p_{i,j}(\cdot)$, $d_{i,j}(\cdot)$, previous correction variables $\underline{\mathbf{c}}^{(k-1)}$, actual subtimestep $m$.
M := 0
for $i = 1$ to $I$ do
  for $j = 1$ to $I$ do
    for $r = 0$ to $M$ do
      if $\theta_r^m \geq 0$ then
        $\mathrm{M}_{i,j} = \mathrm{M}_{i,j} - \Delta t \theta_r^m \frac{p_{i,j}(\mathbf{c}^{r,(k-1)})}{c_j^{m,(k-1)}}$
        $\mathrm{M}_{i,i} = \mathrm{M}_{i,i} + \Delta t \theta_r^m \frac{d_{i,j}(\mathbf{c}^{r,(k-1)})}{c_i^{m,(k-1)}}$
      else
        $\mathrm{M}_{i,j} = \mathrm{M}_{i,j} + \Delta t \theta_r^m \frac{d_{i,j}(\mathbf{c}^{r,(k-1)})}{c_j^{m,(k-1)}}$
        $\mathrm{M}_{i,i} = \mathrm{M}_{i,i} - \Delta t \theta_r^m \frac{p_{i,j}(\mathbf{c}^{r,(k-1)})}{c_i^{m,(k-1)}}$
      end if
    end for
  end for
end for
```

**Algorithm 2** mPDeC

```
Require: Production-destruction functions $p_{i,j}(\cdot)$, $d_{i,j}(\cdot)$, timesteps $\{t^n\}_{n=0}^N$, initial condition $\mathbf{c}^0$.
for $n = 1$ to $N$ do
  for $k = 0$ to $K$ do
    Set $\mathbf{c}^{0,(k)} := \mathbf{c}^n$
  end for
  for $m = 1$ to $M$ do
    Set $\mathbf{c}^{m,(0)} := \mathbf{c}^n$
  end for
  for $k = 1$ to $K$ do
    for $m = 1$ to $M$ do
      Compute the mass matrix $\mathrm{M}(\mathbf{c}^{m,(k-1)})$ :=Mass($\underline{\mathbf{c}}^{(k-1)}, m$) using algorithm 1
      Compute $\mathbf{c}^{m,(k)}$ solving the linear system $\mathrm{M}(\mathbf{c}^{m,(k-1)})\mathbf{c}^{m,(k)} = \mathbf{c}^n$ given by (25)
    end for
  end for
  Set $\mathbf{c}^{n+1} := \mathbf{c}^{M,(K)}$
end for
```

# Acknowledgements

P. Öffner has been funded by the the SNF project (Number 175784).
Davide Torlo is supported by ITN ModCompShock project funded by the European Union's Horizon 2020 research and innovation program under the Marie Sklodowska-Curie grant agreement No 642768.

# References

[1] 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.

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

[3] R. Abgrall and D. Torlo. Asymptotic preserving deferred correction residual distribution schemes. *arXiv preprint arXiv:1811.09284*, 2018.

[4] H. Burchard, E. Deleersnijder, and A. Meister. A high-order conservative patankar-type discretisation for stiff systems of production–destruction equations. *Applied Numerical Mathematics*, 47(1):1–30, 2003.

[5] H. Burchard, E. Deleersnijder, and A. Meister. Application of modified patankar schemes to stiff biogeochemical models for the water column. *Ocean Dynamics*, 55(3-4):326–337, 2005.

[6] 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.

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

[8] S. Gottlieb, D. I. Ketcheson, and C.-W. Shu. *Strong stability preserving Runge-Kutta and multistep time discretizations*. World Scientific, 2011.

[9] E. Hairer, S. P. Nørsett, and G. Wanner. *Solving ordinary differential equations. 1, Nonstiff problems.* Springer-Vlg, 1991.

[10] I. Hense and A. Beckmann. The representation of cyanobacteria life cycle processes in aquatic ecosystem models. *Ecological Modelling*, 221(19):2330–2338, 2010.

[11] 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.

[12] J. Huang, W. Zhao, and C.-W. Shu. A third-order unconditionally positivity-preserving scheme for production–destruction equations with applications to non-equilibrium flows. *Journal of Scientific Computing*, pages 1–42, 2018.

[13] D. I. Ketcheson. Highly efficient strong stability-preserving Runge-Kutta methods with low-storage implementations. *SIAM Journal on Scientific Computing*, 30(4):2113–2136, 2008.

[14] S. Kopecz and A. Meister. On order conditions for modified patankar–runge–kutta schemes. *Applied Numerical Mathematics*, 123:159–179, 2018.

[15] S. Kopecz and A. Meister. Unconditionally positive and conservative third order modified patankar–runge–kutta discretizations of production–destruction systems. *BIT Numerical Mathematics*, pages 1–38, 2018.

[16] S. Kopecz and A. Meister. On the existence of three-stage third-order modified patankar–runge–kutta schemes. *Numerical Algorithms*, pages 1–12, 2019.

[17] 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.

[18] 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.

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

[20] S. Patankar. *Numerical heat transfer and fluid flow.* CRC press, 1980.

[21] C. Rackauckas and Q. Nie. Differentialequations. jl–a performant and feature-rich ecosystem for solving differential equations in julia. *Journal of Open Research Software*, 5(1), 2017.

[22] H. Ranocha and P. Öffner. $L_2$ stability of explicit Runge-Kutta schemes. *Journal of Scientific Computing*, 75(2):1040–1056, 05 2018.

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

[24] G. Wanner and E. Hairer. *Solving ordinary differential equations II.* Springer Berlin Heidelberg, 1996.