A New Stability Approach for Positivity-Preserving Patankar-type Schemes

Davide Torlo[*], Philipp Öffner[†], Hendrik Ranocha[‡]

August 18, 2021

Patankar-type schemes are linearly implicit time integration methods designed to be unconditionally positivity-preserving by going outside of the class of general linear methods. Thus, classical stability concepts cannot be applied and there is no satisfying stability theory for these schemes. We develop a new approach to study stability properties of Patankar-type methods. In particular, we demonstrate problematic behavior of these methods that can lead to undesired oscillations or order reduction. Extreme cases of the latter manifest as spurious steady states. We investigate various classes of Patankar-type schemes based on classical Runge-Kutta methods, strong stability preserving Runge-Kutta methods, and deferred correction schemes using our approach. Finally, we strengthen our analysis with challenging applications including stiff nonlinear problems.

Key words. positivity preservation, Patankar-type methods, Runge–Kutta methods, deferred correction methods, implicit-explicit methods, semi-implicit methods

AMS subject classification. 65L06, 65L20, 65L04

1. Introduction

Many differential equations in biology, chemistry, physics, and engineering are naturally equipped with constraints such as the positivity of certain solution components (e.g., density, energy, pressure) and conservation (e.g., total mass, momentum, energy). In particular, reaction equations are often of this form. Typically, such reaction systems can also be stiff. We consider such ordinary differential equations (ODEs)

$$u'(t) = f(u(t)) \quad u(0) = u_0, \tag{1}$$

that can be written as a production destruction system (PDS) [5]

$$f_i(u) = \sum_j (p_{ij}(u) - d_{ij}(u)), \tag{2}$$

where $p_{ij}, d_{ij} \geq 0$ are the production and destruction terms, respectively. Sometimes, these terms are conveniently written as matrices $p(u) = (p_{ij}(u))_{i,j}$ and $d(u) = (d_{ij}(u))_{i,j}$.

Definition 1.1. An ODE (1) is *positive*, if positive initial data $u_0 > 0$ result in positive solutions $u(t) > 0, \forall t$. Here, inequalities for vectors are interpreted componentwise, i.e., $u(t) > 0$ means $\forall i \colon u_i(t) > 0$. A production destruction system (2) is *conservative*, if $\forall i, j, u \colon p_{ij}(u) = d_{ji}(u)$.

[*] davide.torlo@inria.fr, Inria Bordeaux - Sud-Ouest, 200 avenue de la Vieille Tour 33405 Talence cedex, France.

[†] poeffner@uni-mainz.de, Institut für Mathematik, Johannes Gutenberg Universität, Staudingerweg 9, 55099 Mainz, Germany

[‡] mail@ranocha.de, Applied Mathematics, University of Münster, Orléans-Ring 10, 48149 Münster, Germany.

A slight generalization of the PDS (2) is given by the production destruction rest system (PDRS)

$$f_i(u) = r_i(u) + \sum_j (p_{ij}(u) - d_{ij}(u)), \tag{3}$$

where p_{ij}, d_{ij} are as before and additional rest terms r_i are introduced. These can of course violate the conservative nature of a PDS but can still result in a positive solution if $r_i \geq 0$. The rest term can be interpreted as some additional force (source term) and generalizes the above considered problems.

To ensure physically meaningful and robust numerical approximations, we would like to preserve positivity and conservation discretely.

Definition 1.2. A numerical method computing $u^{n+1} \approx u(t_{n+1})$ given $u^n \approx u(t_n)$ is called *conservative*, if $\sum_i u_i^{n+1} = \sum_i u_i^n$. It is called *unconditionally positive*, if $u^n > 0$ implies $u^{n+1} > 0$.

There are several ways to study positivity of numerical methods [8], e.g., based on the concept of strong stability preserving (SSP) methods [10] or adaptive Runge–Kutta (RK) methods [26]. However, such general linear methods methods are restricted to conditional positivity if they are at least second order accurate [4]. One way to circumvent such order restrictions is given by diagonally split RK methods, which can be unconditionally positive [2, 12, 13]. However, they are less accurate than the unconditionally positive implicit Euler method for large step sizes in practice [21].

Another approach to unconditionally positivity-preserving methods is based on the so-called Patankar trick [28, Section 7.2-2]. First- and second-order accurate conservative methods based thereon were introduced in [5]. Later, these were extended to families of second- and third-order accurate modified Patankar–Runge–Kutta (MPRK) methods based on the Butcher coefficients [17, 19] and the Shu–Osher form [14, 15]. Related deferred correction (DeC) methods were proposed recently [27]. Positive but not conservative methods using the Patankar trick have been proposed and studied in [6], although the connection to Patankar methods seems to be unknown up to now. Other related numerical schemes are inflow-implicit/outflow-explicit methods [9, 24, 25]. Ideas from Patankar-type methods have also been used in numerical methods based on limiters [20].

The methods mentioned above are based on explicit RK methods. To guarantee positivity, the schemes are modified to be linearly implicit, which seems to introduce some stabilization mechanism. In fact, Patankar-type methods have been applied successfully to some stiff systems [6, 16, 17, 19]. However, up to the authors' knowledge, there have been no stability investigations of Patankar-type methods which are applicable to systems of positive equations.

1.1. Motivating example

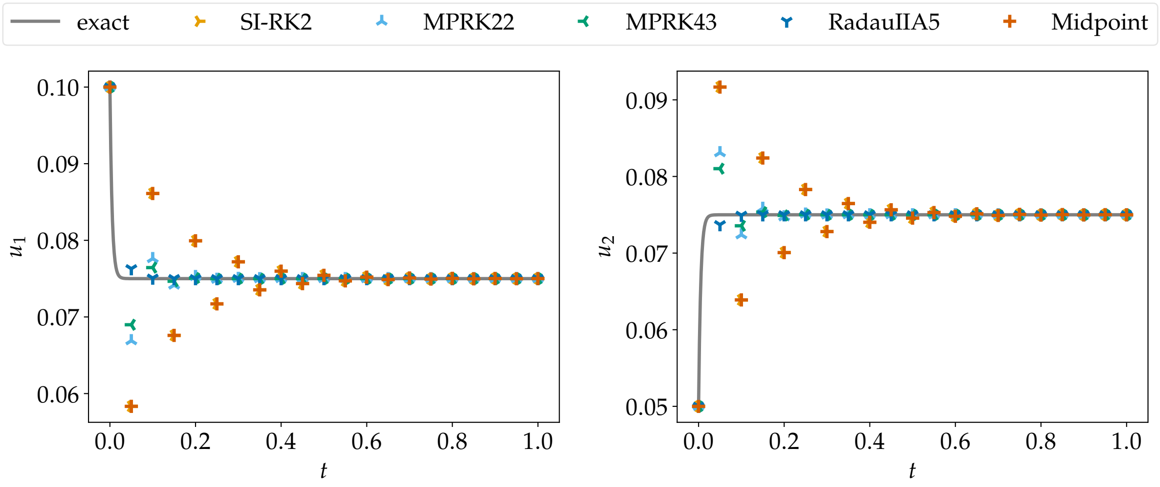
Consider the normal linear system

$$u'(t) = 10^2 \begin{pmatrix} -1 & 1 \\ 1 & -1 \end{pmatrix} u(t), \quad u(0) = u_0 = \begin{pmatrix} 0.1 \\ 0.05 \end{pmatrix}, \tag{4}$$

which can be written as a production destruction system with

$$p(u) = \begin{pmatrix} 0 & 10^2 u_2 \\ 10^2 u_1 & 0 \end{pmatrix}, \quad d(u) = \begin{pmatrix} 0 & 10^2 u_1 \\ 10^2 u_2 & 0 \end{pmatrix}. \tag{5}$$

The eigenvalues of the normal matrix in (4) are 0 and -200. The second order method SI-RK2 of [6] has been shown to be $A(\alpha)$-stable with $\alpha = \pi/4$ and stiff decay [6, Theorem 2.3]. Hence, one would expect that this scheme results in non-oscillating numerical solutions for a normal linear system such as (4). Similar investigations for scalar equations (3) can be done for the second- and third-order accurate modified Patankar–Runge–Kutta schemes MPRK22 and MPRK43 [17, 19].



(a) Component u_1.

(b) Component u_2.

Figure 1.: Numerical solutions (time step $\Delta t = 0.05$) of the normal linear system (4) with real and non-positive eigenvalues obtained using different Patankar-type schemes as well as two implicit Runge–Kutta methods.

While being linearly implicit, Patankar-type methods are nonlinear in the sense that they fall outside of the class of general linear methods. In particular, the application of Patankar-type methods to linear ODEs does not commute with diagonalization, invalidating the common eigenvalue analysis. In contrast to expectations based on scalar test problems, numerical solutions of (4) are oscillating for all three Patankar-type schemes SI-RK2, MPRK22, and MPRK43, as visualized in Figure 1. Of course, A-, B-, and L-stable Runge–Kutta methods such as the fifth-order, three stage RadauIIA5 scheme [11] implemented in DifferentialEquations.jl [29] in Julia [3] perform as expected and do not result in oscillations, since they are compatible with the eigenvalue analysis. While the implicit midpoint method is also compatible with the eigenvalue analysis, it is only A-stable but not L-stable and yields oscillations. This example demonstrates that classical scalar test problems are useless to classify the stability of Patankar-type schemes and motivates the forthcoming study.

1.2. Scope of the article

Positive and conservative schemes naturally satisfy some bounds on the maximum norm of the numerical solution, which is loosely related to the classical concept of A-stability. Motivated by the numerical example above, we are interested in a stability concept excluding the dominant appearance of spurious oscillations, which is loosely related to L-stability. Since Patankar-type methods are not compatible with an eigenvalue analysis, we propose to use a simple and generic linear system as test problem (instead of the classical scalar linear test equation).

We have focused on different types of systems (stiff, dissipative ones, etc.) and considered several quantities like the dissipation of some norms or Lyapunov functionals, cf. [30–34]. However, theses results have not been sufficient to describe the properties of the schemes in an adequate way instead we will measure the amount of spurious oscillations directly.

The rest of the article is structured as follows. The numerical schemes studied in this article are introduced in Section 2. Thereafter, we introduce the oscillation measure and the generic linear system in Section 3. We continue with an analytical investigation of specific schemes, followed by a numerical study for all introduced schemes in Section 4. Next, we derive time step restrictions which ensure the oscillation-free behavior in the linear case. In Section 5, we extend the numerical study to nonlinear and stiff problems. Finally, we summarize and discuss our results in Section 6.

2. Numerical schemes

Here, we introduce Patankar-type methods proposed in the literature that we will investigate later. In addition, we propose a new MPRK method and give a heuristic how to construct such schemes in general.

2.1. Modified Patankar–Euler method

The explicit Euler method $u^{n+1} = u^n + \Delta t f(u^n)$ can be modified by the Patankar trick [28, Section 7.2-2] for a production destruction rest system (3) to get the positive Patankar–Euler method

$$u_i^{n+1} = u_i^n + \Delta t r_i(u^n) + \Delta t \sum_j \left(p_{ij}(u^n) - d_{ij}(u^n) \frac{u_i^{n+1}}{u_i^n} \right). \tag{6}$$

Indeed, given $r, p, d \geq 0$, the new numerical solution u^{n+1} is obtained by solving a linear system with positive diagonal entries, vanishing off-diagonal entries, and a positive right-hand side.

Since the Patankar–Euler method (6) is not conservative, the modified Patankar–Euler method

$$u_i^{n+1} = u_i^n + \Delta t r_i(u^n) + \Delta t \sum_j \left(p_{ij}(u^n) \frac{u_j^{n+1}}{u_j^n} - d_{ij}(u^n) \frac{u_i^{n+1}}{u_i^n} \right), \tag{MPE}$$

has been introduced in [5] (with additional rest terms r here). The modification of the production terms makes the method conservative if the rest terms r vanish. Nevertheless, the method is still positive, because the arising linear systems has positive diagonal entries, negative off-diagonal entries, and is strictly diagonally dominant. Hence, the system matrix is an M matrix and the solution u^{n+1} given a positive right-hand side is positive [1, Section 6.1]. We observe that, when dealing with the scalar linear test problem $u' = \lambda u$ with $\lambda < 0$, the Patankar–Euler method coincides with the implicit Euler method.

Also MPE coincide with the implicit Euler method if we deal with positive and conservative production–destruction linear systems. Indeed, the PDS destruction terms $d_i(u)$ must go to 0 if $u_i \to 0$ [5]. Since the system is linear, $d_{ij}(u^n) = \tilde{d}_{ij} u_i^n$ with $\tilde{d}_{ij} \in \mathbb{R}^+$. Exploiting the conservation properties, we have $p_{ji}(u^n) = \tilde{d}_{ij} u_i^n$. Substituting these formulae in MPE leads to the implicit Euler method.

2.2. MPRK methods using Butcher coefficients

A one-parameter family of MPRK schemes based on the Butcher coefficients of a two stage, second-order RK method was introduced in [17]. Given a parameter $\alpha \in [1/2, \infty)$, the method is

$$\begin{aligned}
y^1 &= u^n, \\
y_i^2 &= u_i^n + \alpha \Delta t r_i(y^1) + \alpha \Delta t \sum_j \left(p_{ij}(y^1) \frac{y_j^2}{y_j^1} - d_{ij}(y^1) \frac{y_i^2}{y_i^1} \right), \\
u_i^{n+1} &= u_i^n + \Delta t \left(\frac{2\alpha - 1}{2\alpha} r_i(y^1) + \frac{1}{2\alpha} r_i(y^2) \right) \\
&\quad + \Delta t \sum_j \Bigg(\left(\frac{2\alpha - 1}{2\alpha} p_{ij}(y^1) + \frac{1}{2\alpha} p_{ij}(y^2) \right) \frac{u_j^{n+1}}{(y_j^2)^{1/\alpha} (y_j^1)^{1-1/\alpha}} \\
&\qquad - \left(\frac{2\alpha - 1}{2\alpha} d_{ij}(y^1) + \frac{1}{2\alpha} d_{ij}(y^2) \right) \frac{u_i^{n+1}}{(y_i^2)^{1/\alpha} (y_i^1)^{1-1/\alpha}} \Bigg).
\end{aligned} \tag{MPRK(2,2,α)}$$

The scheme for the choice $\alpha = 1$ is based on Heun's method, also known as SSPRK(2,2), and has been proposed already in [5].

A similar two-parameter family of four stage, third-order accurate schemes MPRK(4,3,α,β) was introduced and studied in [18, 19]. Since the numerical stability properties of these schemes will be analyzed with respect to the two parameters (α, β), the family under consideration can be found in the Appendix A for completeness.

2.3. MPRK methods using Shu–Osher coefficients

A two-parameter family of MPRK schemes based on the Shu–Osher coefficients of a two stage, second-order RK method was introduced in [14]. Given parameters α, β, the method is

$$\begin{aligned}
y^1 &= u^n, \\
y_i^2 &= y_i^1 + \beta\Delta t r_i\big(y^1\big) + \beta\Delta t \sum_j \left(p_{ij}\big(y^1\big)\frac{y_j^2}{y_j^1} - d_{ij}\big(y^1\big)\frac{y_i^2}{y_i^1} \right), \\
u_i^{n+1} &= (1-\alpha)y_i^1 + \alpha y_i^2 + \Delta t \left((1 - \frac{1}{2\beta} - \alpha\beta) r_i\big(y^1\big) + \frac{1}{2\beta} r_i\big(y^2\big) \right) \\
&\quad + \Delta t \sum_j \Bigg(\left((1 - \frac{1}{2\beta} - \alpha\beta) p_{ij}\big(y^1\big) + \frac{1}{2\beta} p_{ij}\big(y^2\big) \right) \frac{u_j^{n+1}}{(y_j^2)^\gamma (y_j^1)^{1-\gamma}} \\
&\qquad - \left((1 - \frac{1}{2\beta} - \alpha\beta) d_{ij}\big(y^1\big) + \frac{1}{2\beta} d_{ij}\big(y^2\big) \right) \frac{u_i^{n+1}}{(y_i^2)^\gamma (y_i^1)^{1-\gamma}} \Bigg),
\end{aligned} \tag{MPRKSO(2,2,α,β)}$$

where the parameters are restricted to $\alpha \in [0, 1]$, $\beta \in (0, \infty)$, $\alpha\beta + \frac{1}{2\beta} \leq 1$, and

$$\gamma = \frac{1 - \alpha\beta + \alpha\beta^2}{\beta(1 - \alpha\beta)}. \tag{7}$$

An extension to four stage, third-order accurate methods MPRKSO(4,3) was developed in [15] and can be found in the Appendix A.

2.4. Modified Patankar deferred correction schemes

Arbitrarily high-order conservative and positive modified Patankar deferred correction schemes (mPDeC) were introduced in [27]. A time step $[t^n, t^{n+1}]$ is divided into M subintervals, where $t^{n,0} = t^n$ and $t^{n,M} = t^{n+1}$. For every subinterval, the Picard-Lindelöf theorem is mimicked as follows. At each subtimestep $t^{n,m}$, an approximation y^m is calculated. An iterative procedure of K correction steps improves the approximation by one order of accuracy at each iteration. The modified Patankar trick is introduced inside the basic scheme to guarantee positivity and conservation of the intermediate approximations. Using the fact that initial states $y_i^{0,(k)} = u_i^n$ are identical for any correction k, the mPDeC correction steps can be rewritten for $k = 1, \dots, K$, $m = 1, \dots, M$ and $\forall i \in I$ as

$$y_i^{m,(k)} - y_i^0 - \sum_{l=0}^{M} \theta_l^m \Delta t r_i\big(y^{l,(k-1)}\big) - \sum_{l=0}^{M} \theta_l^m \Delta t \sum_{j=1} \left(p_{ij}(y^{l,(k-1)}) \frac{y^{m,(k)}_{\gamma(j,i,\theta_l^m)}}{y^{m,(k-1)}_{\gamma(j,i,\theta_l^m)}} - d_{ij}(y^{l,(k-1)}) \frac{y^{m,(k)}_{\gamma(i,j,\theta_l^m)}}{c^{m,(k-1)}_{\gamma(i,j,\theta_l^m)}} \right) = 0, \tag{mPDeC}$$

where θ_r^m are the correction weights and the $\gamma(j, i, \theta_r^m)$ are the indicator functions depending on θ if the values are positive or negative, see [27] for details. Finally, the new numerical solution is $u_i^{n+1} = y^{M,(K)}$.

The choice of the distribution and the number of subtimesteps M and the number of iterations K determines the order of accuracy of the scheme. In the following, we will use equispaced and Gauss–Lobatto points. To reach order d, we use $M = d - 1$ subintervals and $K = d$ corrections. We will denote the pth-order mPDeC method as mPDeCp. Note that mPDeC1 is equivalent to MPE and mPDeC2 is equivalent to MPRK(2,2,1).

2.5. A new MPRK method

We proposed the following new three stage, second-order MPRK method based on SSPRK(3,3):

$$
\begin{aligned}
y^1 &= u^n, \\
y_i^2 &= u^n + \Delta t r_i(y^1) + \Delta t \sum_j \left(p_{ij}(y^1) \frac{y_j^2}{y_j^1} - d_{ij}(y^1) \frac{y_i^2}{y_i^1} \right), \\
y^3 &= u^n + \Delta t \frac{r_i(y^1) + r_i(y^2)}{4} + \Delta t \sum_j \left(\frac{p_{ij}(y^1) + p_{ij}(y^2)}{4} \frac{y_j^3}{y_j^2} - \frac{d_{ij}(y^1) + d_{ij}(y^2)}{4} \frac{y_i^3}{y_i^2} \right), \\
u^{n+1} &= u^n + \Delta t \frac{r_i(y^1) + r_i(y^2) + 4 r_i(y^3)}{6} \\
&\quad + \Delta t \sum_j \left(\frac{p_{ij}(y^1) + p_{ij}(y^2) + 4 p_{ij}(y^3)}{6} \frac{u_j^{n+1}}{y_j^2} - \frac{d_{ij}(y^1) + d_{ij}(y^2) + 4 d_{ij}(y^3)}{6} \frac{u_i^{n+1}}{y_i^2} \right).
\end{aligned}
\tag{MPRK(3,2)}
$$

For explicitly time-dependent problems, the abscissae are the ones of SSPRK(3,3), i.e., $c = (0, 1, 0.5)$. As will be seen later, this scheme has some desirable stability properties. MPRK(3,2) is second-order accurate. This can be seen through the the following observation. The second stage y_i^2 is an approximation of order one. Next, the midpoint rule is applied as the quadrature which is second-order accurate. In the final stage, the Simpson rule is applied, where we get only second order accuracy since we use the first-order approximation which is multiplied by Δt. At the end, the scheme is second-order accurate.

Remark 2.1. The construction of higher-order MPRK schemes can be done in a similar way. The basic idea is to create a method with increasing stage order, similar to the construction of mPDeC. Starting from a high-order RK scheme, by applying the modified Patankar trick in the substeps in combination with quadrature rules should lead to high-order modified Patankar RK schemes. Essential in the construction is the fact that more stages have to be applied compared to classical RK schemes. This is in accordance with the result of [18] on the existence of third-order, three stages MPRK schemes. There is work in progress to describe a general recipe to construct MPRK schemes of arbitrary order and to study the properties of these schemes.

2.6. Semi-implicit methods

The semi-implicit methods of [6] are also based on the Shu–Osher representation of SSP RK methods, which can be decomposed into convex combinations of the previous step value and explicit Euler steps. Instead of introducing Patankar weights multiplying all destruction terms for a step/stage update, a Patankar weight is introduced for the destruction terms of each Euler stage which is used to compute the new value. Since this procedure limits the order of accuracy of the resulting scheme to first order, an additional function evaluation is used to correct the final solution and get second order of accuracy.

The two methods proposed in [6] are

$$
\begin{aligned}
y^1 &= u^n, \\
y_i^2 &= \frac{u_i^n + \Delta t r_i(y^1) + \Delta t \sum_j p_{ij}(y^1)}{1 + \Delta t \sum_j d_{ij}(y^1)/y_i^1}, \\
y_i^3 &= \frac{1}{2} u_i^n + \frac{1}{2} \frac{y_i^2 + \Delta t r_i(y^2) + \Delta t \sum_j p_{ij}(y^2)}{1 + \Delta t \sum_j d_{ij}(y^2)/y_i^2}, \\
u_i^{n+1} &= \frac{y_i^3 + \Delta t^2 \big(r_i(y^3) + \sum_j p_{ij}(y^3)\big) \sum_j d_{ij}(y^3)/y_i^3}{1 + \big(\Delta t \sum_j d_{ij}(y^3)/y_i^3\big)^2},
\end{aligned}
\tag{SI-RK2}
$$

which uses three stages and is based on SSPRK(2,2), and

$$
\begin{aligned}
y^1 &= u^n, \\
y_i^2 &= \frac{u_i^n + \Delta t r_i(y^1) + \Delta t \sum_j p_{ij}(y^1)}{1 + \Delta t \sum_j d_{ij}(y^1)/y_i^1}, \\
y_i^3 &= \frac{3}{4} u_i^n + \frac{1}{4} \frac{y_i^2 + \Delta t r_i(y^2) + \Delta t \sum_j p_{ij}(y^2)}{1 + \Delta t \sum_j d_{ij}(y^2)/y_i^2}, \\
y_i^4 &= \frac{1}{3} u_i^n + \frac{2}{3} \frac{y_i^3 + \Delta t r_i(y^3) + \Delta t \sum_j p_{ij}(y^3)}{1 + \Delta t \sum_j d_{ij}(y^3)/y_i^3}, \\
u_i^{n+1} &= \frac{y_i^4 + \Delta t^2 \big(r_i(y^4) + \sum_j p_{ij}(y^4)\big) \sum_j d_{ij}(y^4)/y_i^4}{1 + \big(\Delta t \sum_j d_{ij}(y^4)/y_i^4\big)^2},
\end{aligned}
\tag{SI-RK3}
$$

which uses four stages and is based on SSPRK(3,3).

The relation to Patankar schemes becomes obvious by rewriting the computation of the stage y^2 of (SI-RK2) as

$$
y_i^2 = u_i^n + \Delta t r_i(y^1) + \Delta t \sum_j \left(p_{ij}(y^1) - d_{ij}(y^1) \frac{y_i^2}{y_i^1} \right), \tag{8}
$$

which is the Patankar–Euler method (6). As for the Patankar–Euler method, the semi-implicit methods of [6] are not conservative, i.e., it is not guaranteed that $\sum_i u_i^n = \sum_i u_i^{n+1}$ when the system is conservative.

2.7. Steady state preservation

Motivated by the investigations of [6], steady state preservation for (modified) Patankar–Runge–Kutta methods will be studied here. Except for the SI-RK2 and SI-RK3 methods [6], such investigations cannot be found in the literature.

Definition 2.2. A method is steady state preserving if, given a time step Δt and $u^n = u^*$ with $r_i(u^*) + \sum_j p_{ij}(u^*) - d_{ij}(u^*) = 0$, then $u^{n+1} = u^n = u^*$.

Proposition 2.3. *All (modified) Patankar methods described above are steady state preserving.*

Proof. The solution to each stage and the new step value are unique. If the initial condition is a steady state, this steady state is also a valid solution to all stage and step equations. Hence, the steady state is preserved. □

This theorem is important, since some related modifications of explicit Runge–Kutta methods such as IMEX methods are not necessarily steady state preserving [6]. For (stiff) systems with an initial condition near a steady state, the ability to preserve this steady state exactly is desirable and usually results in a better approximation of solutions nearby or decaying to steady state. This result is also interesting since we will observe spurious steady states of Patankar-type methods in the following section.

3. Stability of Patankar-type schemes for linear problems

Here, we introduce a new stability approach for Patankar-type schemes. Recall that a classical stability analysis using scalar problems does not generalize to systems for Patankar-type methods, since these do not commute with diagonalization. Thus, we propose to use a system of the form (4) as test problem.

The basic idea behind the presented approach is that there should be no overshoots/undershoots around the asymptotic (steady state) solution. In particular, the first step should be above the asymptotic solution $u^* = (u_1^*, u_2^*)^T$ for the u_1 component and below for the u_2 component, as RadauIIA5 is doing in Figure 1. Thus, we consider the *oscillation measure*

$$\max \left\{ |u^{n+1} - u^n| - |u^n - u^*|, 0 \right\} \tag{9}$$

to detect whenever a scheme is not behaving as expected. This oscillation measure vanishes for non-oscillatory schemes and increases with the amplitude of oscillations.

3.1. General linear system

We study the oscillation measure (9) for a general 2×2 production-destruction linear system

$$\begin{pmatrix} u_1' \\ u_2' \end{pmatrix} = \begin{pmatrix} -a & b \\ a & -b \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix}. \tag{10}$$

Rescaling the time, we can simplify this system to a one parameter system setting $a + b = 1$ and $0 \le \theta = a \le 1$, i.e.,

$$\begin{pmatrix} u_1' \\ u_2' \end{pmatrix} = \begin{pmatrix} -\theta & (1-\theta) \\ \theta & -(1-\theta) \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix}. \tag{11}$$

We can also rescale any initial condition $u^0 = (u_1^0, u_2^0)^T$ to sum up to one (scaling by a factor $u_1^0 + u_2^0$). Thus, we consider the initial condition

$$\begin{pmatrix} u_1^0 \\ u_2^0 \end{pmatrix} = \begin{pmatrix} 1-\varepsilon \\ \varepsilon \end{pmatrix}, \tag{12}$$

with $0 < \varepsilon < 1$. The steady state of the system is $u^* = (1-\theta, \theta)^T$. Since we are interested in stability properties leading to non-oscillatory behavior, we need to check whether

$$|u_1^1 - u_1^0| \le |u_1^0 - u_1^*| \tag{13}$$

for every initial condition (IC) $0 < \varepsilon < 1$ and for every system $0 \le \theta \le 1$. We can simplify the search exploiting the symmetry of the system, for example considering only $0 < \varepsilon \le 0.5$. It suffices to check the initial step, since every other step will fall back in another IVP (11) with a different IC.

Remark 3.1 (Equivalent stability condition)**.** We can rewrite the previous condition as a positivity condition for the diagonalized system. Rewriting it into a matrix formulation

$$u' = Au = \begin{pmatrix} -\theta & (1-\theta) \\ \theta & -(1-\theta) \end{pmatrix} u,$$

we can obtain the diagonal form $A = R\Lambda R^{-1}$ of the system, i.e.,

$$\Lambda = \begin{pmatrix} -1 & 0 \\ 0 & 0 \end{pmatrix}; \quad R = \begin{pmatrix} 1 & 1-\theta \\ -1 & \theta \end{pmatrix}; \quad R^{-1} = \begin{pmatrix} \theta & -(1-\theta) \\ 1 & 1 \end{pmatrix}.$$

So for $y = R^{-1}u$ we can write an always positive (or always negative) exact solution for the first component

$$y_1 = \theta u_1 - (1-\theta)u_2; \qquad y_1' = -y_1; \quad y_1 = e^{-t}y_1^0.$$

The second equation corresponds to the total mass conservation. This tells us also what we want to preserve: the sign of y_1. If it starts positive, it should stay positive; if it starts negative, it should stay negative.

For a general linear method such as RK methods, this sign condition and the *oscillation-free* condition are equivalent, as we can always pass to the diagonal form. For example, the implicit–Euler is unconditionally positive and thus also unconditionally oscillations-free. Since Patankar-type methods are not general linear methods, they are not necessarily oscillations-free, even if they are unconditionally positive.

3.1.1. Stability restrictions of MPRK(2,2,1)

The method MPRK(2,2,α) with $\alpha = 1$ is equivalent to mPDeC2. Since it is simple enough, a detailed analysis for general linear systems (11) is feasible.

Theorem 3.2 (Time restriction for mPDeC2 for 2×2 linear systems)**.** *Consider the system* (11) *with the initial conditions* (12)*. mPDeC2 is stable in the sense of* (13) *for any initial condition* $0 < \varepsilon < 1$ *and any system* $0 \le \theta \le 1$ *under the time step restriction* $\Delta t \le 2$*.*

Proof. First of all, the cases $\theta = 0$ and $\theta = 1$ are trivially verified as the limit solutions are $(1,0)^T$ and $(0,1)^T$, respectively. Since the scheme is positive, $0 < u_1^n, u_2^n < 1$ holds for any possible initial condition and timestep, verifying the *oscillation-free* condition.

Secondly, the case $\varepsilon = \theta$ implies that the initial condition is the steady state. Since all modified Patankar schemes are able to unconditionally preserve the steady state, the solution will be steady.

In the general case, we can write the solution at the first time step as ratio of polynomials that are of degree one in Δt and θ and of degree two in ε. We refer to the computations inside `MPRK_2_2_1_generalSystem.nb` in the reproducibility repository [35]. The condition (13) simplifies to $u_2^1 \ge \theta$ in the case $\varepsilon > \theta$ and to $u_2^1 \le \theta$ if $\varepsilon < \theta$. In `MPRK_2_2_1_generalSystem.nb` [35], we show how both of these conditions lead to the condition $\Delta t \le z$, where z is the only positive zero of the polynomial $p_{\varepsilon,\theta}(x)$

$$p_{\varepsilon,\theta}(x) = x^3 - x^2 - 2\left(\frac{\varepsilon}{\theta} + \frac{1-\varepsilon}{1-\theta}\right)x - 2\frac{\varepsilon(1-\varepsilon)}{\theta(1-\theta)}. \tag{14}$$

Denoting with $y \le w \le z$ the three zeros of $p_{\varepsilon,\theta}(x)$, we see that they have to satisfy

$$\begin{cases} y + w + z = 1, \\ yz + wz + yw = -2\left(\frac{\varepsilon}{\theta} + \frac{1-\varepsilon}{1-\theta}\right) < -2, \\ ywz = 2\frac{\varepsilon(1-\varepsilon)}{\theta(1-\theta)} > 0. \end{cases} \tag{15}$$

Since ywz is positive and $yz + wz + yw$ is negative, it is clear that only one root is positive, while the other two are negative, w.l.o.g. $y \le w < 0 < z$. From the second equation of (15), we see that

$$z(w+y) < z(w+y) + wy = yz + wz + yw < -2, \tag{16}$$

$$w + y < -\frac{2}{z}. \tag{17}$$

Using then the first equation of (15), we have that

$$0 = z + y + w - 1 < z - \frac{2}{z} - 1, \quad 0 < z^2 - z - 2, \tag{18}$$

which has positive solutions only for $z > 2$. Hence, $\Delta t \leq 2$ in order to avoid oscillations for all systems (11). The bound is sharp in the sense that it can be reached for the limit polynomial $\lim_{\theta \to 0} \lim_{\varepsilon \to 0} p_{\varepsilon,\theta}(x)$. We can observe that when $\varepsilon \longrightarrow 0$, the third equation in (15) tells us that $w \longrightarrow 0^-$. Hence, from the second equation we can see that $y \to -2\frac{1}{(1-\theta)z}$. Finally, the third zero will converge to

$$z \to \frac{1 + \sqrt{1 + \frac{8}{1-\theta}}}{2}.$$

For $\theta \longrightarrow 0$, z goes to 2. □

Unfortunately, the computational complexity increases significantly for all other schemes considered in this article. Thus, we will study simplified problems for general schemes. In addition, we will perform numerical studies for all methods, using different initial conditions (ε), systems (θ), and step sizes (Δt) to find the largest possible timestep without oscillations.

3.2. Symmetric systems

To make an extended analysis feasible, we reduce the number of free parameters in this section. Thus, we consider the linear initial value problem (11) with $\theta = 0.5$, i.e.,

$$u'(t) = f(u(t)) = \frac{1}{2}\begin{pmatrix} -1 & 1 \\ 1 & -1 \end{pmatrix} u(t), \quad u(0) = u^0 = \begin{pmatrix} 1-\varepsilon \\ \varepsilon \end{pmatrix}. \tag{19}$$

To use the modified Patankar schemes with a generic implementation, ε must be strictly positive to avoid division by zero. As recommended in the literature [17], we set ε to the smallest positive number that can be represented as floating point number with given precision (usually 64 bit) whenever we are interested in the limit $\varepsilon \longrightarrow 0$. In the following we first study the behavior of the previously presented scheme for $\varepsilon \longrightarrow 0$, then we show where this study is meaningful and where it is less. In order to explain the kind of computations used in the following, we start with a simple example using MPRK(2,2,α) with $\alpha = 1$. Recall that the system (19) conserves the total mass $u_1 + u_2 = 1$ and can be formulated as

$$\begin{aligned} u_1'(t) &= \frac{-u_1 + u_2}{2} = \frac{1}{2} - u_1, \\ u_2'(t) &= \frac{u_1 - u_2}{2} = \frac{1}{2} - u_2, \end{aligned}$$

with asymptotic solution $u_1 = u_2 = \frac{1}{2}$.

Example 3.3 (Stability of MPRK(2,2,1))**.** We investigate the behavior of MPRK(2,2,1) applied to (19). Due to the conservation property and the symmetry of the problem, it suffices to focus on the first component u_1. For the first non-trivial stage, we obtain

$$y_1^2 = u_1^n + \frac{\Delta t}{2}\left(u_2^n \frac{y_2^2}{u_2^n} - u_1^n \frac{y_1^2}{u_1^n}\right) = u_1^n + \frac{\Delta t}{2}\left(1 - 2y_1^2\right) \quad \Longleftrightarrow \quad y_1^2 = \frac{u_1^n + \frac{\Delta t}{2}}{1 + \Delta t}. \tag{20}$$

Using this expression for y_1^2, the new numerical solution satisfies

$$\begin{aligned} u_1^{n+1} &= u_1^n + \frac{\Delta t}{4}\left(\left((1 - u_1^n) + (1 - y_1^2)\right)\frac{1 - u_1^{n+1}}{1 - y_1^2} - \left(u_1^n + y_1^2\right)\frac{u_1^{n+1}}{y_1^2}\right) \\ &= u_1^n + \frac{\Delta t}{4}\left(\frac{(1 - u_1^n)(1 - u_1^{n+1})}{1 - y_1^2} + 1 - u_1^{n+1} - \frac{u_1^n u_1^{n+1}}{y_1^2} - u_1^{n+1}\right). \end{aligned} \tag{21}$$

This can be reformulated as

$$\left(1+\frac{\Delta t}{4}\frac{1-u_1^n}{1-y_1^2}+\frac{\Delta t}{2}+\frac{\Delta t}{4}\frac{u_1^n}{y_1^2}\right)u_1^{n+1}=u_1^n+\frac{\Delta t}{4}\frac{1-u_1^n}{1-y_1^2}+\frac{\Delta t}{4}. \tag{22}$$

Passing to the limit $\varepsilon \to 0$ with $u_1^n = 1$ and $\lim_{\varepsilon\to 0} y_1^2 = (1+\Delta t/2)/(1+\Delta t)$ yields

$$\begin{aligned} &\left(1+\frac{\Delta t}{2}+\frac{\Delta t}{4}\frac{(1+\Delta t)}{1+\Delta t/2}\right)\lim_{\varepsilon\to 0} u_1^{n+1} = 1+\frac{\Delta t}{4} \\ \iff \qquad &\lim_{\varepsilon\to 0} u_1^{n+1} = \frac{8+6\Delta t+\Delta t^2}{8+10\Delta t+4\Delta t^2}. \end{aligned} \tag{23}$$

Since no oscillations should appear, one demands that $u_1^{n+1} \geq 1/2$. This is guaranteed by a CFL-like condition on Δt; in this case, $\Delta t \leq (1+\sqrt{17})/2 \approx 2.56$. We see that this is only slightly larger than the bound found in the previous study for general systems.

3.2.1. Stability and spurious steady states of MPRK(2,2,α)

Applying MPRK(2,2,α) to the symmetric system (19) results in the following observations.

- For $\alpha > 1$, taking $\varepsilon \to 0$ results in $u^1 = u^0$ and subsequently $u^n \equiv u^0$. This can also be observed numerically if sufficiently high accuracy is used, e.g. `BigFloat` in Julia. For `Float64`, the first few steps do nearly nothing and later steps result in the desired behavior. The number of steps necessary to actually do something increases for $\alpha \gg 1$.
- For $\alpha \in [1/2, 1]$, there is a maximal timestep resulting in oscillation-free solutions. $\alpha = 1$ allows the largest time step.

These are analyzed in detail in the following.

Theorem 3.4. *The test problem* (19) *becomes a spurious steady state for MPRK(2,2,α) with* $\alpha > 1$ *in the limit* $\varepsilon \to 0$.

Proof. This proof makes use of explicit calculations using Mathematica [36]. All calculations can be found in the notebook `MPRK_2_2_alpha.nb` in the accompanying reproducibility repository [35].

For $\alpha > 1$ and $\varepsilon > 0$, the first step of (19) can be computed explicitly. The second component after the first step is of the form $u_2^1 = h_1(\varepsilon)/h_2(\varepsilon)$, where $\lim_{\varepsilon\to 0} h_1(\varepsilon) = \lim_{\varepsilon\to 0} h_2(\varepsilon) = 0$. Making use of l'Hôpital's rule,

$$\lim_{\varepsilon\to 0} h_1'(\varepsilon) = \big(\alpha\Delta t(1+\alpha\Delta t)/2\big)^{1/\alpha}\big(\Delta t + (\alpha - 1/4)\Delta t^2\big), \tag{24}$$

and

$$\lim_{\varepsilon\to 0} h_2'(\varepsilon) = \infty, \quad \alpha > 1, \tag{25}$$

results in $\lim_{\varepsilon\to 0} u_2^1 = 0 = \lim_{\varepsilon\to 0} u_2^0$ for $\alpha > 1$. Since the sum of all components of u is conserved, $\lim_{\varepsilon\to 0} u^0 = (1,0)$ is a spurious steady state. □

Theorem 3.5. *Consider the application of MPRK(2,2,α) with* $\alpha \in [0.5, 1]$ *to the test problem* (19) *in the limit* $\varepsilon \to 0$. *The first step does not result in oscillations if and only if*

$$\begin{cases} \dfrac{4\Delta t + 3\Delta t^2}{8+10\Delta t+4\Delta t^2} < \dfrac{1}{2}, & \alpha = 1, \\ \dfrac{4\Delta t + (4\alpha-1)\Delta t^2}{8+(4+8\alpha)\Delta t+(4\alpha-1)\Delta t^2} < \dfrac{1}{2}, & \alpha \in [0.5, 1). \end{cases} \tag{26}$$

Proof. This proof makes use of explicit calculations using Mathematica [36]. All calculations can be found in the notebook `MPRK_2_2_alpha.nb` in the accompanying reproducibility repository [35].

As in the proof of Theorem 3.4, we evaluate the limit $\varepsilon \to 0$ of $u_2^1 = h_1(\varepsilon)/h_2(\varepsilon)$ using l'Hôpital's rule. We have

$$\lim_{\varepsilon\to 0} h_2'(\varepsilon) = \begin{cases} \left(\Delta t + \frac{9}{4}\Delta t^2 + \frac{7}{4}\Delta t^3 + \frac{1}{2}\Delta t^4\right), & \alpha = 1, \\ \left(\alpha\Delta t(1+\alpha\Delta t)/2\right)^{1/\alpha}\left(2 + (1+2\alpha)\Delta t + (\alpha - \frac{1}{4})\Delta t^2\right), & \alpha \in [0.5, 1), \end{cases} \tag{27}$$

Using (24) from before,

$$\lim_{\varepsilon\to 0} u_2^1(\varepsilon) = \frac{\lim_{\varepsilon\to 0} h_1'(\varepsilon)}{\lim_{\varepsilon\to 0} h_2'(\varepsilon)} = \begin{cases} \dfrac{4\Delta t + 3\Delta t^2}{8 + 10\Delta t + 4\Delta t^2}, & \alpha = 1, \\ \dfrac{4\Delta t + (4\alpha - 1)\Delta t^2}{8 + (4+8\alpha)\Delta t + (4\alpha-1)\Delta t^2}, & \alpha \in [0.5, 1). \end{cases} \tag{28}$$

Note the discontinuity at $\alpha = 1$ of $\lim_{\varepsilon\to 0} u_2^1(\varepsilon)$. □

Theorem 3.6. *Consider the application of MPRK(2,2,α) with $\alpha \in [0.5, 1]$ to the test problem* (19) *in the limit $\varepsilon \to 0$. The timestep restriction preventing oscillations in the first step given in Theorem 3.5 is least restrictive for $\alpha = 1$.*

Proof. This proof makes use of explicit calculations using Mathematica [36]. All calculations can be found in the notebook `MPRK_2_2_alpha.nb` in the accompanying reproducibility repository [35]. □

Remark 3.7. This result does not demonstrate that there is an error in the proofs of the order of accuracy of MPRK(2,2,α) [17]. Indeed, studies of the order of accuracy focus on fixed $\varepsilon > 0$ and the limit $\Delta t \to 0$. Numerical experiments suggest that the numerical solutions stays approximately constant for a certain number of steps determined by ε (and with less sensitivity also by Δt) until small changes have accumulated and the exponential decay of u_1 becomes visible. In particular, the limits $\Delta t \to 0$ and $\varepsilon \to 0$ are not interchangeable.

Expanding Remark 3.7, a careful error analysis can be conduced by constructing Taylor expansions of the error after the first step for $\Delta t \to 0$ and $\varepsilon \to 0$ in both possible orders. Expanding at first around $\Delta t = 0$ shows that the leading order errors contain terms proportional to ε^{-1} for $\alpha \neq 1$, both for $\alpha < 1$ and for $\alpha > 1$. However, these leading order terms are in agreement with the analysis of [17], i.e., they are proportional to Δt^3.

More insights can be gained by studying the expansions first for $\varepsilon \to 0$, expanding around $\Delta t = 0$ afterwards. Then, the leading order terms in ε are $O(\Delta t^3)$ for $\alpha = 1$, $O(\Delta t^2)$ for $\alpha = 0.5$, and $O(\Delta t)$ for $\alpha = 2$. This can also be observed in numerical experiments using `BigFloat` in Julia [3] as shown in Figure 2.

3.3. Expansion for other MP algorithms

A similar analysis can be conducted for the mPDeC algorithm. However, the approach we used with Mathematica was only able to give results up to third order.

Since mPDeC2 is equivalent to MPRK(2,2,1), we get the same results as before. In particular,

$$\begin{aligned} \lim_{\varepsilon\to 0}\lim_{\Delta t\to 0} u_1^{n=1} - u_1(\Delta t) &= \left(\frac{2}{3} - \frac{4\varepsilon}{3} + O(\varepsilon^3)\right)\Delta t^3 + O(\Delta t^4) \\ \lim_{\Delta t\to 0}\lim_{\varepsilon\to 0} u_1^{n=1} - u_1(\Delta t) &= \left(\frac{\Delta t^3}{6} + O(\Delta t^4)\right) + \left(\frac{\Delta t^2}{2} - \frac{11\Delta t^3}{6} + O(\Delta t^4)\right)\varepsilon + \\ &\quad \left(-\frac{\Delta t}{2} + \frac{\Delta t^2}{4} + \frac{3\Delta t^3}{4} + O(\Delta t^4)\right)\varepsilon^2 + O(\varepsilon^3). \end{aligned}$$

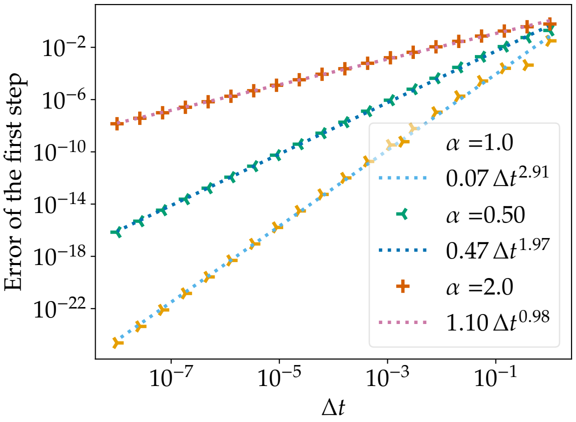


Figure 2.: Convergence study of the error of the first step for different members of the family MPRK(2,2,α) for the test problem (19) with ε = `eps(BigFloat)`.

Note that $\mathcal{O}(\varepsilon)$ terms can be ignored for the limit $\lim_{\Delta t\to 0}\lim_{\varepsilon\to 0}$ when evaluating the order of accuracy. Hence, we see that in both cases we have an error of $\mathcal{O}(\Delta t^3)$ for the first step, i.e., a second-order accurate method.

For the third-order algorithm mPDeC3, we have a different behavior and an order reduction for small ε:

$$\lim_{\varepsilon\to 0}\lim_{\Delta t\to 0} u_1^{n=1} - u_1(\Delta t) = \left(-\frac{1}{864\varepsilon^2} - \frac{5}{72\varepsilon} + \frac{1789}{864} - \frac{1697\varepsilon}{432} + \frac{7\varepsilon^2}{96} + \mathcal{O}(\varepsilon^3)\right)\Delta t^4 + \mathcal{O}(\Delta t^5)$$

$$\lim_{\Delta t\to 0}\lim_{\varepsilon\to 0} u_1^{n=1} - u_1(\Delta t) = \left(-\frac{2\Delta t^2}{3} + \mathcal{O}(\Delta t^3)\right) + \left(224\Delta t + \mathcal{O}(\Delta t^2)\right)\varepsilon - 74880\varepsilon^2 + \mathcal{O}(\Delta t\,\varepsilon^2) + \mathcal{O}(\varepsilon^3).$$

If we let $\varepsilon \to 0$ before $\Delta t \to 0$, we have a reduction to first order accuracy. The computations for these tests can be found in `MPDEC.nb` in the accompanying reproducibility repository [35].

For the second-order MPRK(3,2) proposed in this article, we observe a consistent second order accuracy in the limit case $\varepsilon \to 0$, i.e.,

$$\lim_{\varepsilon\to 0}\lim_{\Delta t\to 0} u_1^{n=1} - u_1(\Delta t) = \left(2 - 4\varepsilon + \mathcal{O}(\varepsilon^3)\right)\Delta t^3 + \mathcal{O}(\Delta t^4)$$

$$\lim_{\Delta t\to 0}\lim_{\varepsilon\to 0} u_1^{n=1} - u_1(\Delta t) = \left(\frac{4\Delta t^3}{3} + \mathcal{O}(\Delta t^4)\right) + \left(\frac{2\Delta t^2}{3} - \frac{71\Delta t^3}{12} + \mathcal{O}(\Delta t^4)\right)\varepsilon + \left(-\frac{2\Delta t}{3} + \frac{35\Delta t^2}{24} + \frac{139\Delta t^3}{144} + \mathcal{O}(\Delta t^4)\right)\varepsilon^2 + \mathcal{O}(\varepsilon^3).$$

This results are computed Mathematica and the related notebook `MPRK_3_2.nb` is in the accompanying reproducibility repository [35].

Remark 3.8. We have also analyzed MPRKSO$(2, 2, \alpha, \beta)$ with selected parameters α, β. We do not present these analyses here; in general, they all agree with the numerical studies presented in the following.

3.4. Need for a numerical study

In the previous studies, we observed two issues with the modified Patankar methods. First, there can be oscillations around the steady state, which can be avoided by a CFL-like restriction on Δt. Moreover, there can be spurious steady states at $u_1^0 \approx 0$, leading to a loss of order of accuracy or even an inconsistency.

We want to understand if such phenomena happen for all possible initial conditions and all test systems. The ultimate goal is to find conditions on the time step and schemes which do not produce oscillations but are also not stuck inside the spurious steady state. Since analytical approaches become infeasible with increasing number of parameters, we will resort to a numerical study in the following.

4. Numerical experiments for general linear systems

As described in Section 3.1, we consider the general 2×2 system (11) with initial condition $u^0 = (1 - \varepsilon, \varepsilon)^T$. The goal of this study is to find the largest timestep Δt for all possible systems parameterized by $0 \leq \theta \leq 1$ and initial conditions $0 < \varepsilon < 1$, such that the stability condition (13) is satisfied. We exploit the symmetry of the system studying only the $\varepsilon < 0.5$ case, as the other can be obtain substituting $\tilde{\varepsilon} = 1 - \varepsilon$ and $\tilde{\theta} = 1 - \theta$.

In the following tests, we compare different methods and families presented above: MPRK(2,2,α), MPRK(4,3,α, β), MPRKSO(2,2,α,β), MPRKSO(4,3), mPDeC both for equispaced and Gauss–Lobatto subtimesteps, MPRK(3,2), SI-RK2, and SI-RK3.

We apply all methods to a variety of $\varepsilon \in [0, 0.5]$ and $\theta \in [0, 0.5]$, which are uniformly distributed in a logarithmic scale. For θ, we also consider the symmetrized values for $[0.5, 1]$. We run the simulations for all these schemes and initial conditions for one time step Δt of varying size, uniformly distributed in a logarithmic scale between 2^{-6} and 2^6. The maximum Δt that gives stable results in the sense of (13) will be denoted as our bound.

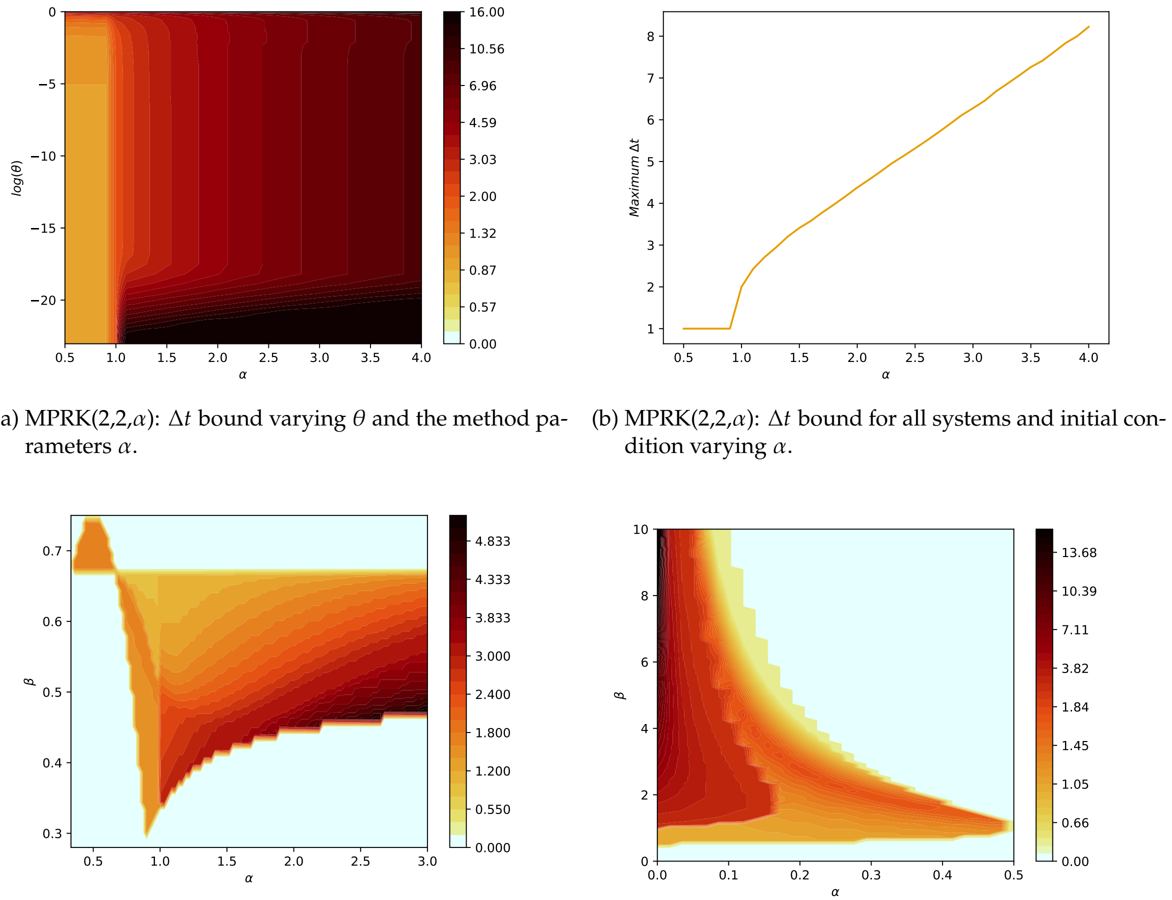
In Figure 3, we present the results for the all the modified Patankar methods and in Figure 4 for the semi-implicit Runge-Kutta methods. We highlight an important remark on the check of the stability condition (13). The evaluation of such condition is done with a tolerance of 5 machine epsilons. Some tests can be sensitive to this tolerance, in particular for (mPDeC) equispaced schemes with high odd order of accuracy, when the Δt bound is large. There the number of stages is large and the machine error can sum up to non-negligible errors.

For MPRK(2,2,α), we see in Figures 3a and 3b that the bound on Δt is 1 for $\alpha < 1$, 2 for $\alpha = 1$, and is increasing with $\alpha > 1$. Recall that the methods with $\alpha > 1$ become inconsistent for $\varepsilon \to 0$, preserving the initial condition as spurious steady state. This must be kept in mind when choosing the scheme one wants to use. Varying the system parameter θ influences the bound on the time step, as shown in Figure 3a.

For MPRK(4,3,α, β), outside the area where the scheme is not well defined, we observe areas where the Δt bound reaches very low values ($\ll 1$) and other areas where it is larger than one. It must be noted that in the areas where the Δt bound is large, we observe inconsistency problems as the one shown in Section 3.3. The precise area where this happens is denoted in brown in Figure 5a. It is noticeable that around the curve $\beta(6\alpha - 3) = 3\alpha - 2$, which is a boundary for nonnegative coefficients [19], the Δt bound is particularly large. Hence, in Figure 4d we plot the values for that specific curve, and indeed they are larger than other methods. On the other side, all the schemes given by these parameters show inconsistency when starting from their spurious steady state $u_1 \approx 0$.

For MPRKSO(2,2,α,β), we observe that a large area of the α, β plane has Δt bound around unity. The bounds increase close to the line $\alpha = 0$. For this family of methods, we also observe inconsistencies as $\varepsilon \to 0$ for large Δt. The precise area where this happens is denoted in brown in Figure 5b.

For mPDeC, we observe very different behaviors between equispaced and Gauss–Lobatto points. The two formulations coincide up to third order. The second order mPDeC shows the $\Delta t = 2$ bound that was derived analytically in Section 3.1. The methods based on Gauss–Lobatto nodes have a time step restriction of unity for orders four and higher. Moreover, we have no evidence of order reduction or inconsistency when $\varepsilon \to 0$. For equispaced nodes, we obtain larger Δt bounds, in particular for schemes with odd order of accuracy. In contrast to Gauss–Lobatto nodes, we often



(a) MPRK(2,2,α): Δt bound varying θ and the method parameters α.

(b) MPRK(2,2,α): Δt bound for all systems and initial condition varying α.

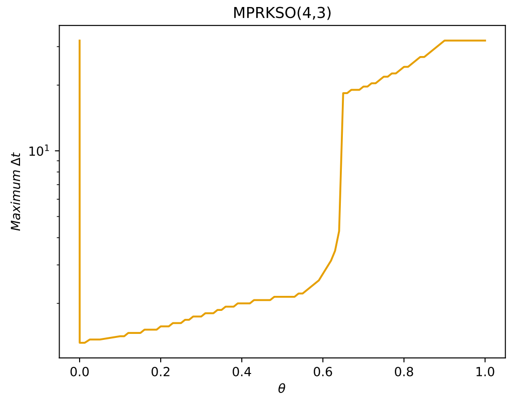
(c) Δt bound for MPRK(4,3,α,β) varying α and β. The zero area (light blue) is always unstable or not producing sensible results for any timestep.

(d) Δt bound for MPRKSO(2,2,α,β) varying α and β. The light blue area is oscillating for any timestep.

mPDeC

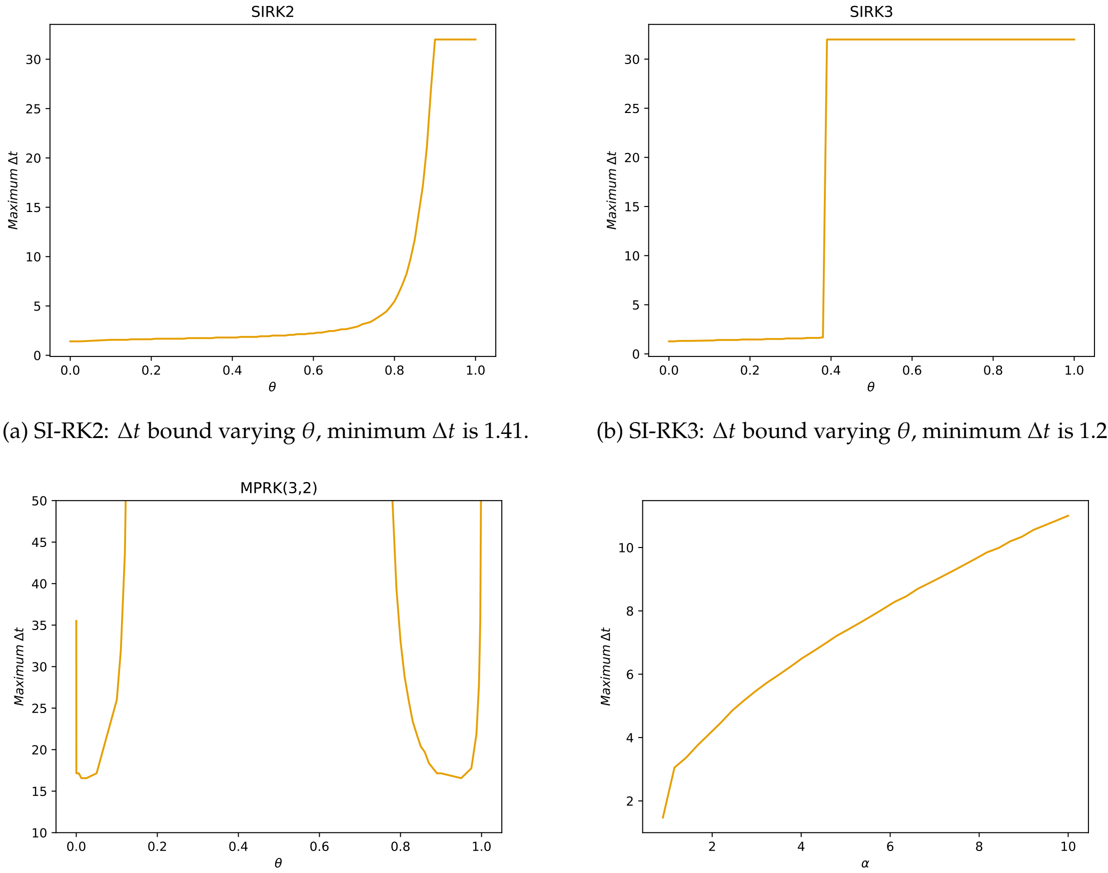
Equispaced points		Gauss-Lobatto points	
Order	Δt bound	Order	Δt bound
1	∞	1	∞
2	2.0	2	2.0
3	1.19	3	1.19
4	1.11	4	1.07
5	1.07	5	1.04
6	1.04	6	1.0
7	1.04	7	1.0
8	1.37	8	1.0
9	6.96	9	1.0
10	1.0	10	1.0
11	16.0	11	1.0
12	1.0	12	1.0
13	40.79	13	1.0
14	1.07	14	1.0
15	27.85	15	1.0
16	1.80	16	1.0

(e) Δt bound for mPDeC with equispaced and Gauss–Lobatto points. In red the schemes with spurious steady state.



(f) Δt bound for MPRKSO(4,3) varying the system through θ. Minimum Δt is 1.31.

Figure 3.: Numerical search of the Δt bound for having a stable first time step, in the sense of (13), for problem (11) varying IC and system parameter θ.



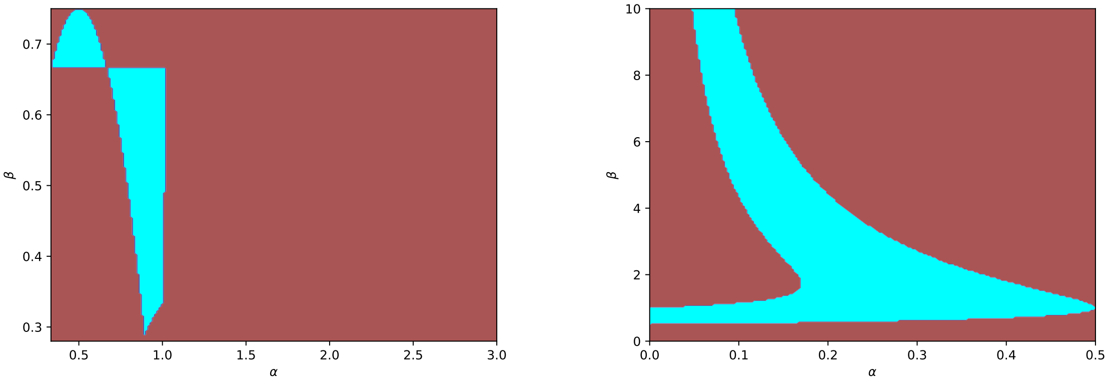
(a) SI-RK2: Δt bound varying θ, minimum Δt is 1.41.

(b) SI-RK3: Δt bound varying θ, minimum Δt is 1.27.

(c) MPRK(3,2): Δt bound varying θ. Minimum Δt is 16.56.

(d) Δt bound for MPRK(4,3,α,β) on the curve $\beta(6\alpha-3) = 3\alpha-2$ for all the system through θ.

Figure 4.: Numerical search of the Δt bound for having a stable first time step, in the sense of (13), for problem (11) varying IC and system parameter θ on the semi–implicit Runge Kutta methods (SI-RK2) and (SI-RK3).



(a) MPRK(4,3,α,β)

(b) MPRKSO(2,2,α,β)

Figure 5.: Inconsistency area for some schemes. Brown for inconsistent, light blue for consistent schemes.

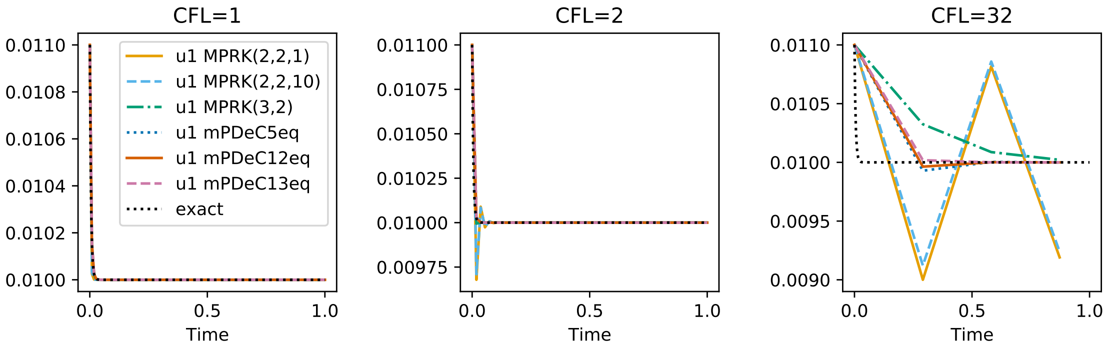


Figure 6.: Simulations of (29) at different CFLs for some schemes.

observe order reduction/inconsistency problems shown in Section 3.3 for high order schemes, more precisely for order 9 and order greater or equal to 11.

The MPRKSO(4,3) scheme has a Δt bound of 1.31, as shown in Figure 3f. Moreover, it does not show inconsistencies in the numerical tests. Indeed, MPRK(3,2) has maybe the best conditions of all the schemes, see Figure 4c. Its Δt bound is around 16 and it never shows inconsistencies.

Finally, in Figures 4a and 4b, the semi-implicit schemes are presented. Both show similar behaviors with Δt slightly larger than unity. For these methods, we do not observe the same inconsistency shown in previous methods.

5. Validation on nonlinear problems

5.1. Scalar nonlinear problem

The second problem on which we are testing our methods on is a scalar ODE with a source term [6]. Find $u : [0, 0.15] \to \mathbb{R}$, with $u(0) = 1.1\sqrt{1/k}$, where $k > 0$ is a coefficient of the problem, and

$$u' = -k|u|u + 1. \tag{29}$$

The solution for this problem is monotonically decreasing and converging to $u_\infty = \sqrt{1/k}$. The schemes can be applied to this problem following simple prescriptions.

- The source shall be integrated in time without considering the Patankar trick, simply using the coefficients of the original schemes.
- The productions and destruction terms must be rewritten as $d_{11} = k|u|u$ and $p_{11} = 0$.

We want to extend the linear analysis of the previous two sections, trying to understand if the linear Δt bound can be useful in the nonlinear case as well. Aiming at that, we check the first time step, which often shows overshoots with respect to the steady state, for different time steps.

In particular, we can observe that the (local) Lipschitz constant of the right-hand side of (29) is $C(k) := \max_u k|u| = k|u_0| = 1.1\sqrt{k}$. Hence, inspired by the theory for numerical PDEs, we define a CFL number in $\mathbb{R}^+$ and we set the Δt step as $\Delta t := \text{CFL}/C(k)$. Doing so, we essentially get rid of the dependence on k, through a rescaling factor both for time and amplitude on the solution. In this way, the CFL number should be comparable with the Δt bound found in the previous sections. We fix $k = 10^4$ for the following simulations; analogous results can be obtained for other k.

Figure 6 shows the simulations for different CFLs. For low CFLs, we observe no oscillations for essentially all methods. Increasing the CFL number, we observe that most of the schemes go below u_∞ for the first timestep.

Table 1.: Oscillation measure for problem (29) and selected methods.

CFL	0.5	1.0	2.0	4.0	8.0	16.0	32.0	64.0
MPDeC1eq	0	0	2.7e-04	5.3e-04	7.0e-04	8.0e-04	8.5e-04	8.8e-04
MPDeC2eq	0	0	3.2e-04	6.1e-04	8.1e-04	9.3e-04	1.0e-03	1.0e-03
MPDeC5GL	0	0	0	2.8e-06	6.2e-05	2.0e-04	3.4e-04	4.4e-04
MPDeC5eq	0	0	0	0	0	2.0e-05	7.1e-05	1.1e-04
MPDeC9eq	0	0	0	0	0	0	0	0
MPDeC9GL	0	0	0	0	0	0	0	0
MPDeC11eq	0	0	0	0	0	0	0	0
MPDeC11GL	0	0	0	0	1.4e-06	1.9e-05	9.0e-05	1.9e-04
MPRK(2,2,10.0)	0	0	2.9e-04	5.5e-04	7.2e-04	8.2e-04	8.8e-04	9.1e-04
MPRK(4,3,1.25,0.39)	0	0	0	0	0	0	0	0
MPRKSO(2,2,0.1,1.5)	0	0	3.6e-04	6.7e-04	8.8e-04	1.0e-03	1.1e-03	1.1e-03
MPRKSO(4,3)	0	0	5.1e-05	7.7e-05	0	0	0	0
MPRK(3,2)	0	0	5.2e-06	0	0	0	0	0
SIRK2	0	0	2.9e-04	5.4e-04	7.0e-04	8.0e-04	8.5e-04	8.8e-04
SIRK3	0	0	1.0e-04	3.3e-05	0	0	0	0

We analyze now all the methods at the first step. We list in Table 1 some representative methods and their oscillations (9) with different CFLs. In the supplementary material, we include more tables with many more schemes and parameters, which we summarize in the following.

For mPDeC methods with equispaced and Gauss–Lobatto subtimesteps, we notice that many schemes overshoot the steady values when increasing the CFL. In particular, whenever we are below the Δt bound of Figure 3e, we do not observe oscillations. In some cases, we also do not have oscillations above this bounds, but this might depend on the problem itself. Surprisingly, MPE is not so well performing as in the linear tests, where it was unconditionally not oscillating. For this nonlinear problem, it shows oscillations for CFL > 1.

We also observe oscillations for some methods of the family MPRK(2,2,α) even if the bound found in the linear tests is higher than the CFL tested in the nonlinear ones. Anyway, for CFL < 1 all the methods are not oscillating.

Testing MPRK(4,3,α,β), with some interesting parameters, we found oscillations according to the Δt bound found in Figure 3c almost everywhere, while, on the bottom curve $\beta(6\alpha-3) = 3\alpha-2$, we observe no oscillations for $\alpha \geq 1$, which is slightly better then expected, considering the (large but not so large) Δt bounds in Figure 4d.

Another disappointing result comes from the schemes MPRKSO(2,2,α,β) for which, even on the line $\alpha = 0$, we do not have oscillation-free simulations with large Δt as predicted by Figure 3d on linear problems. Conversely, for the other parameters we have, as expected, oscillations for almost all CFLs larger than 1.

For MPRK(3,2), MPRKSO(4,3), and SI-RK3, the oscillations appear for CFL neither too small nor too large. This is surprising, first of all for MPRK(3,2) of which we expected no oscillations up to CFL $\approx$ 16, which shows anyway a very small oscillation only for CFL=2. The amplitude of this oscillation is comparable only with ones produced by very high order schemes. For MPRKSO(4,3) and SI-RK3, we have slightly better results than expected for large CFLs. For SI-RK2, the results are exactly following the Δt bounds found in Figure 4a.

Conclusion 5.1. For this test, most of the schemes behaves as predicted based on the linear example, with few exceptions for second-order methods. The bounds of the linear case can mostly be transferred to the considered nonlinear problem. The linear analysis gives some meaningful results also for more challenging problems.

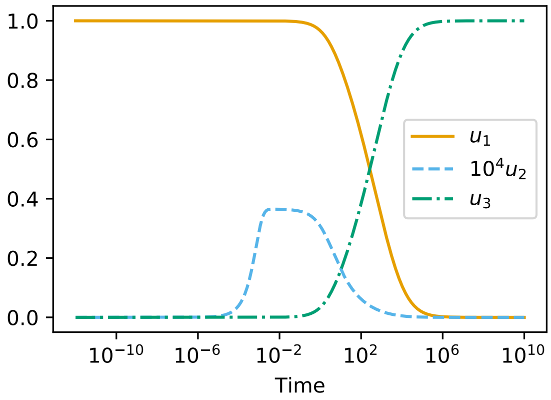


Figure 7.: Robertson problem: reference solution obtained with 3000 time steps with mPDeC2.

5.2. Robertson problem

The Robertson problem [22, Section II.10] with parameters $k_1 = 0.04$, $k_2 = 3 \cdot 10^7$, and $k_3 = 10^4$ is a stiff system of three nonlinear ODEs. It can be written as a PDS [17] with non-zero components

$$p_{12}(u) = d_{21}(u) = k_3 u_2 u_3, \quad p_{21}(u) = d_{12}(u) = k_1 u_1, \quad p_{32}(u) = d_{23}(u) = k_2 u_2^2. \tag{30}$$

Reactions in this problem scale with different orders of magnitudes. To reasonably capture the behavior of the solution, it is necessary to use exponentially increasing time steps [17]. A reference solution can be found in Figure 7. To apply generic modified Patankar schemes, we have to modify the initial condition u^0 slightly, replacing 0 by $\varepsilon > 0$; here, we use $\varepsilon = 10^{-180}$.

For this problem, oscillations are not so clear, because the steady state $u_\infty = (0, 0, 1)^T$ cannot be exceeded since all the schemes are positive (and conservative). Nevertheless, we might encounter the spurious steady state problem. In Figure 8, we observe that many methods do not catch the behavior of u_2 and remain zero. In some cases, even u_3 stays at zero. All these phenomena are in accordance with the results found for the linear problem. Indeed, among the computed tests we see that MPRK(2,2,α) for $\alpha > 1$, MPRK(4,3,10,0.5), MPRKSO(2,2,0.001,10) and mPDeC11 with equispaced subtimesteps lead to spurious steady states. Both semi-implicit methods SI-RK2 and SI-RK3 go to infinity as they cannot keep the conservation of the quantities. Hence, we are not showing their simulations.

5.3. HIRES

We consider the "High Irradiance RESponse" problem HIRES [11]. The original problem HIRES [22, Section II.1] can be rewritten as a nine-dimensional production–destruction system with

$$\begin{aligned}
&r_1(u) = \sigma, && p_{21}(u) = d_{12}(u) = k_1 u_1, && p_{12}(u) = d_{21}(u) = k_2 u_2,\\
&p_{42}(u) = d_{24}(u) = k_3 u_2, && p_{43}(u) = d_{34}(u) = k_1 u_3, && p_{13}(u) = d_{31}(u) = k_6 u_3,\\
&p_{34}(u) = d_{43}(u) = k_2 u_4, && p_{64}(u) = d_{46}(u) = k_4 u_4, && p_{65}(u) = d_{56}(u) = k_1 u_5,\\
&p_{35}(u) = d_{53}(u) = k_5 u_5, && p_{56}(u) = d_{65}(u) = k_2 u_6, && p_{57}(u) = d_{75}(u) = \frac{k_2}{2} u_7,\\
&p_{67}(u) = d_{76}(u) = \frac{k_-}{2} u_7, && p_{97}(u) = d_{79}(u) = \frac{k_*}{2} u_7, && p_{76}(u) = d_{67}(u) = k_+ u_6 u_8,\\
&p_{78}(u) = d_{87}(u) = k_+ u_6 u_8, && p_{87}(u) = d_{78}(u) = \frac{k_- + k_* + k_2}{2} u_7.
\end{aligned} \tag{31}$$

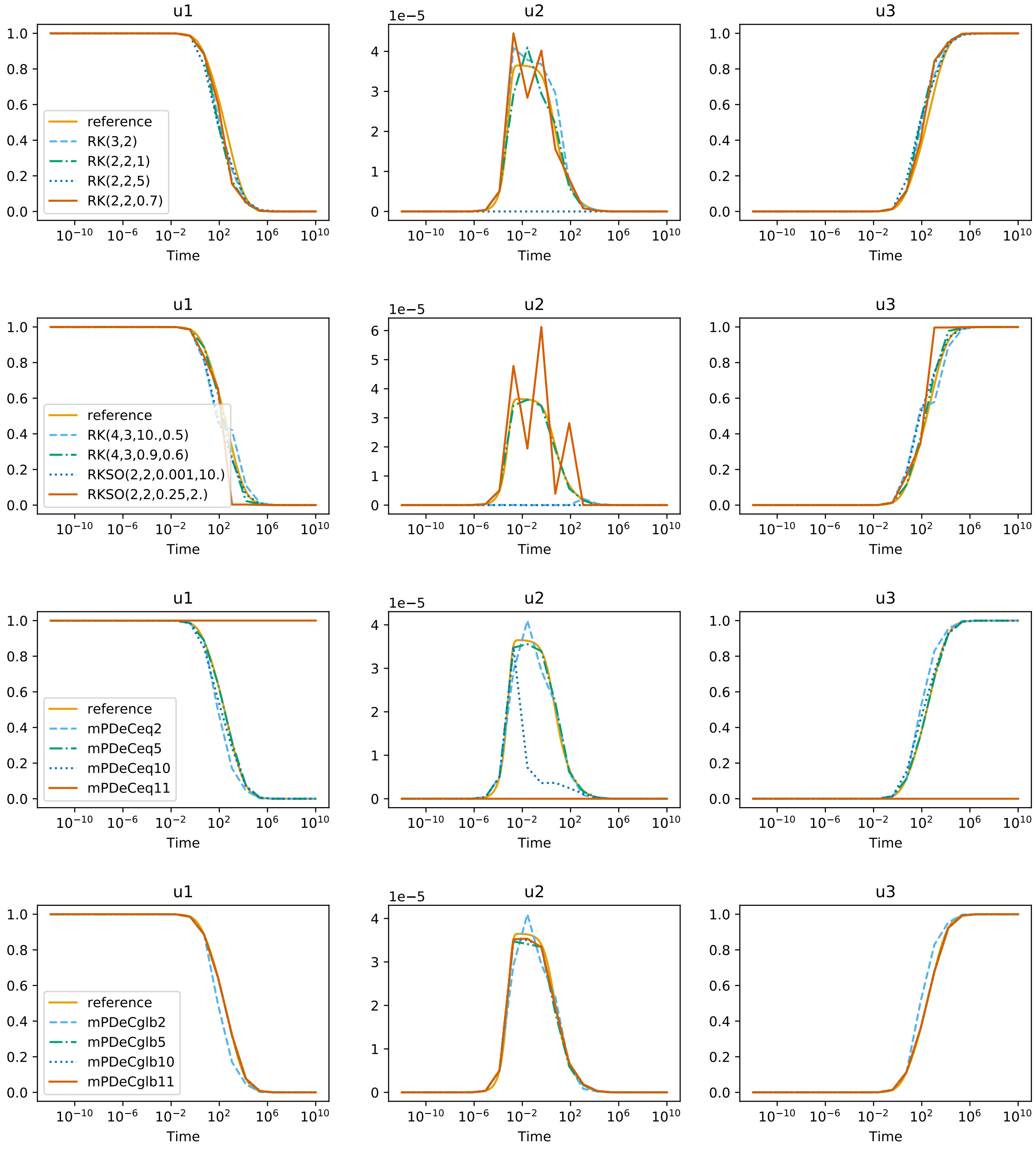


Figure 8.: Robertson problem with different methods and 20 time steps.

and parameters

$$\begin{aligned}&k_1 = 1.71, \quad k_2 = 0.43, \quad k_3 = 8.32, \quad k_4 = 0.69, \quad k_5 = 0.035,\\ &k_6 = 8.32, \quad k_+ = 280, \quad k_- = 0.69, \quad k_* = 0.69, \quad \sigma = 0.0007.\end{aligned} \tag{32}$$

The initial condition is $u(0) = (1, 0, 0, 0, 0, 0, 0, 0.0057, 0)^T$, where numerically we used 10^{-35} instead of zero for vanishing initial constituents. The time interval is $t \in [0, 321.8122]$.

For this test, the concept of oscillation is not clear. Nevertheless, we can observe spurious steady states also for this problem. We compute the reference solution with 10^5 uniform time steps using mPDeC5 with equispaced subtimesteps, which is in accordance with the reference solution [22] up to the fourth significant digits for all constituents.

Testing with $N = 10^3$ uniform time steps, we spot troubles with the *inconsistent* methods. We test the problem with many schemes presented above and we include the relative plots in the supplementary material. For brevity, we plot in Figure 9 just a sample.

For mPDeC, we observe the inconsistency problem only for equispaced time steps for high odd orders $(9, 11, 13$ and so on). In Figure 9, we see the simulation for mPDeC6 with Gauss–Lobatto points. We observe that the high accuracy helps in obtaining a good result at the end of the simulation, when u_7 and u_8 react. The moment at which this change happens is hard to catch and only high order methods are able to obtain it with this number of time steps.

We run the MPRK(2,2,α) with $\alpha \in \{1, 5\}$. As for the linear case, we observe inconsistencies only for $\alpha > 1$. This is demonstrated in Figure 9 for $\alpha = 5$, where the evolution of some constituents is completely missed, e.g., u_2, u_3, u_5, u_9, while for $\alpha = 1$ we obtain consistent results.

We test MPRKSO(2,2,α,β) with $\alpha = 0.3$, $\beta = 2$ and $\alpha = 0$, $\beta = 8$. As expected, the second one shows the inconsistent spurious steady state. An oscillatory behavior can be observed, though, in the first simulation, which is shown in Figure 9. This is probably due to the CFL condition; refining the time discretization, the oscillations disappear.

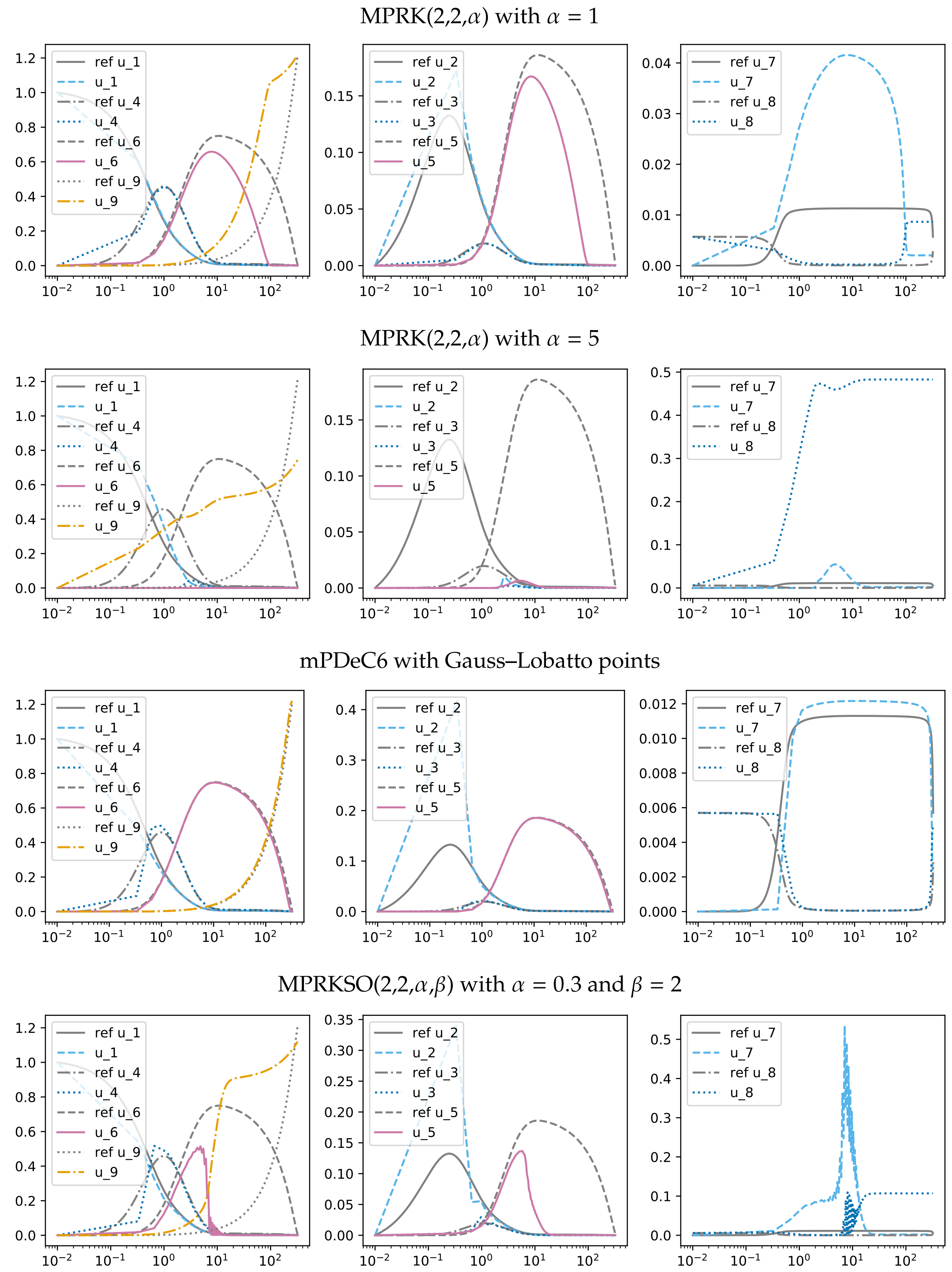
For MPRK(4,3,α, β), we test $\alpha = 0.9$, $\beta = 0.6$ and $\alpha = 5$, $\beta = 0.5$, observing inconsistencies only for the second one, in accordance with the linear tests. For MPRKSO(4,3), MPRK(3,2), SI-RK2 and SI-RK3, we do not observe inconsistencies, as in the linear test, nor other particular behaviors.

6. Summary and discussion

We proposed a novel stability analysis for Patankar-type schemes. Focusing on a generic 2×2 linear test problem, we introduced an oscillation measure to quantify stability properties. Based thereon, we derived a CFL-like time step restriction for all methods under consideration, either analytically (whenever feasible) or numerically. Moreover, we investigated these methods near vanishing components, discovering spurious behavior including order reduction and artificial steady states. Additionally, we applied the methods to more challenging problems including stiff nonlinear ones. We observed that our proposed stability and consistency analysis generalizes reasonably well to these other problems.

From our point of view, this is a first step toward stability investigations of Patankar-type schemes. Extensions and further analyses could be based on various Lyapunov functionals instead of our oscillation measure. Moreover, different test systems could be considered. Nevertheless, we would like to stress that our current approach seems promising and generalizes well to other demanding problems.

In the future, our stability investigation should be extended to hyperbolic conservation laws. Using the structure of corresponding spatial semidiscretizations, the resulting ODE can be written as a production-destruction-rest system [7, 15, 23]. Here, the relation between the time step restrictions derived in this work and classical CFL conditions will be the major focus of research.



MPRK(2,2,α) with $\alpha = 1$

MPRK(2,2,α) with $\alpha = 5$

mPDeC6 with Gauss–Lobatto points

MPRKSO(2,2,α,β) with $\alpha = 0.3$ and $\beta = 2$

Figure 9.: Simulations run with different schemes with $N = 10^3$ time steps, plot in logarithmic scale in time.

A. Third order modified Patankar Runge–Kutta methods

In the following part, the third order accurate MPRK(4,3,α,β) from [18, 19] is repeated for completeness. Please note that the investigated version is called $MPRK43I(\alpha,\beta)$ in their papers. It is given by

$$\begin{aligned}
y^1 &= u^n,\\
y_i^2 &= u_i^n + a_{21}\Delta t r_i(y^1) + a_{21}\Delta t \sum_j \left(p_{ij}(y^1)\frac{y_j^2}{y_j^1} - d_{ij}(y^1)\frac{y_i^2}{y_i^1} \right),\\
y_i^3 &= u_i^n + \Delta t \left(a_{31} r_i(y^1) + a_{32} r_i(y^2) \right)\\
&\quad + \Delta t \sum_j \Bigg(\left(a_{31} p_{ij}(y^1) + a_{32} p_{ij}(y^2) \right) \frac{y_j^3}{\left(y_j^2\right)^{1/p}\left(y_j^1\right)^{1/p-1}}\\
&\qquad - \left(a_{31} d_{ij}(y^1) + a_{32} d_{ij}(y^2) \right) \frac{y_i^3}{\left(y_i^2\right)^{1/p}\left(y_i^1\right)^{1/p-1}} \Bigg),\\
\sigma_i &= u_i^n + \Delta t \sum_j \Bigg(\left(\beta_1 p_{ij}(y^1) + \beta_2 p_{ij}(y^2) \right) \frac{\sigma_j}{\left(y_j^2\right)^{1/q}\left(y_j^1\right)^{1/q-1}}\\
&\quad - \left(\beta_1 d_{ij}(y^1) + \beta_2 d_{ij}(y^2) \right) \frac{\sigma_i}{\left(y_i^2\right)^{1/q}\left(y_i^1\right)^{1/q-1}} \Bigg)\\
u_i^{n+1} &= u_i^n + \Delta t \left(b_1 r_i(y^1) + b_2 r_i(y^2) + b_3 r_i(y^3) \right)\\
&\quad + \Delta t \sum_j \Bigg(\left(b_1 p_{ij}(y^1) + b_2 p_{ij}(y^2) + b_3 p_{ij}(y^3) \right) \frac{u_j^{n+1}}{\sigma_j}\\
&\qquad - \left(b_1 d_{ij}(y^1) + b_2 d_{ij}(y^2) + b_3 d_{ij}(y^3) \right) \frac{u_i^{n+1}}{\sigma_i} \Bigg),
\end{aligned} \tag{MPRK(4,3,α,β)}$$

where $p = 3a_{21}\left(a_{31} + a_{32}\right)b_3,\ q = a_{21},\ \beta_2 = \frac{1}{2a_{21}}$ and $\beta_1 = 1 - \beta_2$. The Butcher tableaus in respect to the two parameters

$$\begin{array}{c|ccc}
0 & & & \\
\alpha & \alpha & & \\
\beta & \frac{3\alpha\beta(1-\alpha)-\beta^2}{\alpha(2-3\alpha)} & \frac{\beta(\beta-\alpha)}{\alpha(2-3\alpha)} & \\
\hline
 & 1 + \frac{2-3(\alpha+\beta)}{6\alpha\beta} & \frac{3\beta-2}{6\alpha(\beta-\alpha)} & \frac{2-3\alpha}{6\beta(\beta-\alpha)}
\end{array} \tag{33}$$

with

$$\left.\begin{array}{l}
2/3 \leq \beta \leq 3\alpha(1-\alpha)\\
3\alpha(1-\alpha) \leq \beta \leq 2/3\\
(3\alpha-2)/(6\alpha-3) \leq \beta \leq 2/3
\end{array}\right\} \text{ for } \begin{cases}
1/3 \leq \alpha < \frac{2}{3},\\
2/3 \leq \alpha < \alpha_0,\\
\alpha > \alpha_0,
\end{cases}$$

and $\alpha_0 \approx 0.89255$.
Next, also the MPRKSO(4,3) from [15] is repeated. It is given by

$$
\begin{aligned}
y^1 &= u^n,\\
y_i^2 &= y_i^1 + a_{10}\Delta t r_i(y^1) + \Delta t \sum_j b_{10}\left(p_{ij}(y^1)\frac{y_j^2}{y_j^1} - d_{ij}(y^1)\frac{y_i^2}{y_i^1}\right),\\
\varrho_i &= n_1 y_i^2 + n_2 y_i^1 \left(\frac{y_i^2}{y_i^1}\right)^2\\
y_i^3 &= (a_{20}y_i^1 + a_{21}y_i^2) + \Delta t \left(b_{20} r_i(y^1) + b_{21} r_i(y^2)\right)\\
&\quad + \Delta t \sum_j \left(\left(b_{20}p_{ij}(y^1) + b_{21}p_{ij}(y^2)\right)\frac{y_j^2}{\varrho_j} - \left(b_{20}d_{ij}(y^1) + b_{21}d_{ij}(y^2)\right)\frac{y_i^2}{\varrho_i}\right),\\
\mu_i &= y_i^1\left(\frac{y_i^2}{y_i^1}\right)^s \qquad \text{(MPRKSO(4,3))}\\
\tilde{a}_i &= \eta_1 y_i^1 + \eta_2 y_i^2 + \Delta t \sum_j \left(\left(\eta_3 p_{ij}(y^1) + \eta_4 p_{ij}(y^2)\right)\frac{\tilde{a}_j}{\mu_j} - \left(\eta_3 d_{ij}(y^1) + \eta_4 d_{ij}(y^2)\right)\frac{\tilde{a}_i}{\mu_i}\right)\\
\sigma_i &= \tilde{a}_i + z y_i^1 \frac{y_i^2}{\varrho_i}\\
u_i^{n+1} &= \left(a_{30}y_i^1 + a_{31}y_i^2 + a_{32}y_i^3\right) + \Delta t \left(b_{30}r_i(y^1) + b_{31}r_i(y^2) + b_{32}r_i(y^3)\right)\\
&\quad + \Delta t \sum_j \Bigg(\left(b_{30}p_{ij}(y^1) + b_{31}p_{ij}(y^2) + b_{32}p_{ij}(y^2)\right)\frac{u_j^{n+1}}{\sigma_j}\\
&\qquad - \left(b_{30}d_{ij}(y^1) + b_{31}d_{ij}(y^2) + b_{32}d_{ij}(y^2)\right)\frac{u_i^{n+1}}{\sigma_i}\Bigg).
\end{aligned}
$$

Here, the optimal SSP coefficients determined in [15] will be used. They are given by

$$
\begin{aligned}
n_1 &= 2.569046025732011E-01, & n_2 &= 7.430953974267989E-01,\\
a_{10} &= 1, & a_{20} &= 9.2600312554031827E-01,\\
a_{21} &= 7.3996874459681783E-02, & a_{31} &= 2.0662904223744017E-10,\\
b_{10} &= 4.7620819268131703E-01, & a_{30} &= 7.0439040373427619E-01,\\
a_{32} &= 2.9560959605909481E-01, & b_{20} &= 7.7545442722396801E-02,\\
b_{21} &= 5.9197500149679749E-01, & b_{31} &= 6.8214380786704851E-10,\\
b_{30} &= 2.0044747790361456E-01, & b_{32} &= 5.9121918658514827E-01,\\
\eta_1 &= 3.777285888379173E-02, & \eta_2 &= 1/3,\\
\eta_3 &= 1.868649805549811E-01, & \eta_3 &= 2.224876040351123,\\
z &= 6.288938077828750E-01, & s &= 5.721964308755304.
\end{aligned}
$$

Acknowledgment

D. T. is funded by Team CARDAMOM in Inria–Bordeaux Sud–Ouest, France. P.Ö. gratefully acknowledge support of the Gutenberg Research College, JGU Mainz and the UZH Postdoc Scholarship (Number FK-19-104). H. R. was supported by the German Research Foundation (DFG, Deutsche Forschungsgemeinschaft) under Germany's Excellence Strategy EXC 2044-390685587, Mathematics Münster: Dynamics-Geometry-Structure.

We would like to thank Stefan Kopecz and David Ketcheson for fruitful discussion at the beginning of this project. This project has started with the visit by H.R. in Zurich in 2019 which was supported by the SNF project (Number 175784) and the King Abdullah University of Science and Technology (KAUST).

References

[1] O. Axelsson. *Iterative Solution Methods*. Cambridge: Cambridge University Press, 1996. DOI: 10.1017/CBO9780511624100.

[2] A Bellen and L. Torelli. "Unconditional Contractivity in the Maximum Norm of Diagonally Split Runge–Kutta Methods." In: *SIAM Journal on Numerical Analysis* 34.2 (1997), pp. 528–543. DOI: 10.1137/S0036142994267576.

[3] J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah. "Julia: A Fresh Approach to Numerical Computing." In: *SIAM Review* 59.1 (2017), pp. 65–98. DOI: 10.1137/141000671. arXiv: 1411.1607 [cs.MS].

[4] C. Bolley and M. Crouzeix. "Conservation de la positivité lors de la discrétisation des problèmes d'évolution paraboliques." In: *RAIRO. Analyse numérique* 12.3 (1978), pp. 237–245.

[5] H. Burchard, E. Deleersnijder, and A. Meister. "A high-order conservative Patankar-type discretisation for stiff systems of production–destruction equations." In: *Applied Numerical Mathematics* 47.1 (2003), pp. 1–30. DOI: 10.1016/S0168-9274(03)00101-6.

[6] A. Chertock, S. Cui, A. Kurganov, and T. Wu. "Steady state and sign preserving semi-implicit Runge–Kutta methods for ODEs with stiff damping term." In: *SIAM Journal on Numerical Analysis* 53.4 (2015), pp. 2008–2029. DOI: 10.1137/151005798.

[7] M. Ciallella, L. Micalizzi, P. Öffner, and D. Torlo. *Application of mPDeC Schemes in the Shallow Water Equation*. 2021. in prepration: inprepration.

[8] I. Fekete, D. I. Ketcheson, and L. Lóczi. "Positivity for convective semi-discretizations." In: *Journal of Scientific Computing* 74.1 (2018), pp. 244–266. DOI: 10.1007/s10915-017-0432-9.

[9] P. Frolkovic. "Semi-implicit methods based on inflow implicit and outflow explicit time discretization of advection." In: *Proceedings of ALGORITMY*. 2016, pp. 165–174.

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

[11] E. Hairer and G. Wanner. "Stiff differential equations solved by Radau methods." In: *Journal of Computational and Applied Mathematics* 111.1-2 (1999), pp. 93–111. DOI: 10.1016/S0377-0427(99)00134-X.

[12] Z. Horváth. "Positivity of Runge–Kutta and diagonally split Runge–Kutta methods." In: *Applied Numerical Mathematics* 28.2-4 (1998), pp. 309–326. DOI: 10.1016/S0168-9274(98)00050-6.

[13] K. in't Hout. "A note on unconditional maximum norm contractivity of diagonally split Runge–Kutta methods." In: *SIAM Journal on Numerical Analysis* 33.3 (1996), pp. 1125–1134. DOI: 10.1137/0733055.

[14] J. Huang and C.-W. Shu. "Positivity-Preserving Time Discretizations for Production–Destruction Equations with Applications to Non-equilibrium Flows." In: *Journal of Scientific Computing* 78.3 (2019), pp. 1811–1839. DOI: 10.1007/s10915-018-0852-1.

[15] 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." In: *Journal of Scientific Computing* 79.2 (2019), pp. 1015–1056. DOI: 10.1007/s10915-018-0881-9.

[16] S. Kopecz and A. Meister. "A comparison of numerical methods for conservative and positive advection–diffusion–production–destruction systems." In: *PAMM* 19.1 (2019). DOI: 10.1002/pamm.201900209.

[17] S. Kopecz and A. Meister. "On order conditions for modified Patankar–Runge–Kutta schemes." In: *Applied Numerical Mathematics* 123 (2018), pp. 159–179. DOI: 10.1016/j.apnum.2017.09.004.

[18] S. Kopecz and A. Meister. "On the existence of three-stage third-order modified Patankar–Runge–Kutta schemes." In: *Numerical Algorithms* (2019), pp. 1–12. doi: 10.1007/s11075-019-00680-3.

[19] S. Kopecz and A. Meister. "Unconditionally positive and conservative third order modified Patankar–Runge–Kutta discretizations of production–destruction systems." In: *BIT Numerical Mathematics* 58.3 (2018), pp. 691–728. doi: 10.1007/s10543-018-0705-1.

[20] D. Kuzmin. "Entropy stabilization and property-preserving limiters for $\mathbb{P}^1$ discontinuous Galerkin discretizations of scalar hyperbolic problems." In: *Journal of Numerical Mathematics* (2020).

[21] C. B. Macdonald, S. Gottlieb, and S. J. Ruuth. "A numerical study of diagonally split Runge–Kutta methods for PDEs with discontinuities." In: *Journal of Scientific Computing* 36.1 (2008), pp. 89–112. doi: 10.1007/s10915-007-9180-6.

[22] F. Mazzia and C. Magherini. *Test Set for Initial Value Problem Solvers*. Technical Report Release 2.4. Italy: Department of Mathematics, University of Bari, Feb. 2008.

[23] A. Meister and S. Ortleb. "A positivity preserving and well-balanced DG scheme using finite volume subcells in almost dry regions." In: *Applied Mathematics and Computation* 272 (2016), pp. 259–273.

[24] K. Mikula and M. Ohlberger. "Inflow-implicit/outflow-explicit scheme for solving advection equations." In: *Finite Volumes for Complex Applications VI Problems & Perspectives*. Vol. 4. Springer Proceedings in Mathematics. Berlin, Heidelberg: Springer, 2011, pp. 683–691. doi: 10.1007/978-3-642-20671-9_72.

[25] K. Mikula, M. Ohlberger, and J. Urbán. "Inflow-implicit/outflow-explicit finite volume methods for solving advection equations." In: *Applied Numerical Mathematics* 85 (2014), pp. 16–37. doi: 10.1016/j.apnum.2014.06.002.

[26] S. Nüßlein, H. Ranocha, and D. I. Ketcheson. *Positivity-Preserving Adaptive Runge-Kutta Methods*. May 2020. arXiv: 2005.06268 [math.NA].

[27] P. Öffner and D. Torlo. "Arbitrary high-order, conservative and positivity preserving Patankar-type deferred correction schemes." In: *Applied Numerical Mathematics* 153 (2020), pp. 15–34.

[28] S. V. Patankar. *Numerical Heat Transfer and Fluid Flow*. Washington: Hemisphere Publishing Corporation, 1980.

[29] C. Rackauckas and Q. Nie. "DifferentialEquations.jl – A Performant and Feature-Rich Ecosystem for Solving Differential Equations in Julia." In: *Journal of Open Research Software* 5.1 (2017), p. 15. doi: 10.5334/jors.151.

[30] H. Ranocha. "On strong stability of explicit Runge–Kutta methods for nonlinear semibounded operators." In: *IMA Journal of Numerical Analysis* 41.1 (2021), pp. 654–682.

[31] H. Ranocha and D. I. Ketcheson. "Energy Stability of Explicit Runge–Kutta Methods for Nonautonomous or Nonlinear Problems." In: *SIAM Journal on Numerical Analysis* 58.6 (2020), pp. 3382–3405.

[32] H. Ranocha and P. Öffner. "L_2 Stability of Explicit Runge–Kutta Schemes." In: *Journal of Scientific Computing* 75.2 (May 2018), pp. 1040–1056. doi: 10.1007/s10915-017-0595-4.

[33] Z. Sun and C.-W. Shu. "Stability of the fourth order Runge–Kutta method for time-dependent partial differential equations." In: *Annals of Mathematical Sciences and Applications* 2.2 (2017), pp. 255–284. doi: 10.4310/AMSA.2017.v2.n2.a3.

[34] Z. Sun and C.-W. Shu. "Strong Stability of Explicit Runge–Kutta Time Discretizations." In: *SIAM Journal on Numerical Analysis* 57.3 (2019), pp. 1158–1182. doi: 10.1137/18M122892X. arXiv: 1811.10680 [math.NA].

[35] D. Torlo, P. Öffner, and H. Ranocha. *Stability of Positivity Preserving Patankar-Type Schemes*. Git repository: `https://git.math.uzh.ch/abgrall_group/patankar-stability`. Aug. 2021.

[36] Wolfram Research, Inc. *Mathematica*. Version 12.0. 2019. URL: `https://www.wolfram.com`.

A New Stability Approach for Positivity-Preserving Patankar-type Schemes: Supplementary Material

Davide Torlo, Philipp Öffner, Hendrik Ranocha

August 18, 2021

1 Patankar Methods

In order to make the document self-contained, we list again the used methods.

- Modified Patankar Euler method [1]

$$u_i^{n+1} = u_i^n + \Delta t r_i(u^n) + \Delta t \sum_j \left(p_{ij}(u^n) \frac{u_j^{n+1}}{u_j^n} - d_{ij}(u^n) \frac{u_i^{n+1}}{u_i^n} \right), \tag{MPE}$$

- Modified Patankar Runge–Kutta(2,2,α) methods [1]

$$\begin{aligned}
y^1 &= u^n, \\
y_i^2 &= u_i^n + \alpha \Delta t r_i(y^1) + \alpha \Delta t \sum_j \left(p_{ij}(y^1) \frac{y_j^2}{y_j^1} - d_{ij}(y^1) \frac{y_i^2}{y_i^1} \right), \\
u_i^{n+1} &= u_i^n + \Delta t \left(\frac{2\alpha - 1}{2\alpha} r_i(y^1) + \frac{1}{2\alpha} r_i(y^2) \right) \\
&\quad + \Delta t \sum_j \left(\left(\frac{2\alpha - 1}{2\alpha} p_{ij}(y^1) + \frac{1}{2\alpha} p_{ij}(y^2) \right) \frac{u_j^{n+1}}{(y_j^2)^{1/\alpha} (y_j^1)^{1-1/\alpha}} \right. \\
&\qquad \left. - \left(\frac{2\alpha - 1}{2\alpha} d_{ij}(y^1) + \frac{1}{2\alpha} d_{ij}(y^2) \right) \frac{u_i^{n+1}}{(y_i^2)^{1/\alpha} (y_i^1)^{1-1/\alpha}} \right).
\end{aligned} \tag{MPRK(2,2,α)}$$

- Modified Patankar Shu–Osher Runge–Kutta(2,2,α,β) methods [4]

$$\begin{aligned}
y^1 &= u^n,\\
y_i^2 &= y_i^1 + \beta\Delta t r_i\left(y^1\right) + \beta\Delta t \sum_j \left(p_{ij}\left(y^1\right)\frac{y_j^2}{y_j^1} - d_{ij}\left(y^1\right)\frac{y_i^2}{y_i^1} \right),\\
u_i^{n+1} &= (1-\alpha)y_i^1 + \alpha y_i^2 + \Delta t \left((1 - 1/2\beta - \alpha\beta) r_i\left(y^1\right) + \frac{1}{2\beta} r_i\left(y^2\right) \right)\\
&\quad + \Delta t \sum_j \Bigg(\left((1 - 1/2\beta - \alpha\beta) p_{ij}\left(y^1\right) + \frac{1}{2\beta} p_{ij}\left(y^2\right) \right) \frac{u_j^{n+1}}{(y_j^2)^\gamma (y_j^1)^{1-\gamma}}\\
&\qquad - \left((1 - 1/2\beta - \alpha\beta) d_{ij}\left(y^1\right) + \frac{1}{2\beta} d_{ij}\left(y^2\right) \right) \frac{u_i^{n+1}}{(y_i^2)^\gamma (y_i^1)^{1-\gamma}} \Bigg),
\end{aligned} \tag{MPRKSO(2,2,α,β)}$$

where the parameters are restricted to $\alpha \in [0,1]$, $\beta \in (0,\infty)$, $\alpha\beta + 1/2\beta \leq 1$, and

$$\gamma = \frac{1 - \alpha\beta + \alpha\beta^2}{\beta(1-\alpha\beta)}. \tag{1}$$

- Modified Patankar Deferred Correction methods [8]

$$y_i^{m,(k)} - y_i^0 - \sum_{l=0}^{M} \theta_l^m \Delta t r_i\left(y^{l,(k-1)}\right) - \sum_{l=0}^{M} \theta_l^m \Delta t \sum_{j=1} \left(p_{ij}(y^{l,(k-1)}) \frac{y^{m,(k)}_{\gamma(j,i,\theta_l^m)}}{y^{m,(k-1)}_{\gamma(j,i,\theta_l^m)}} - d_{ij}(y^{l,(k-1)}) \frac{y^{m,(k)}_{\gamma(i,j,\theta_l^m)}}{c^{m,(k-1)}_{\gamma(i,j,\theta_l^m)}} \right) = 0, \tag{mPDeC}$$

where θ_r^m are the correction weights and the $\gamma(j,i,\theta_r^m)$ are the indicator functions depending on θ if the values are positive or negative, see [8] for details. Finally, the new numerical solution is $u_i^{n+1} = y^{M,(K)}$.

- The new Modified Patankar Runge–Kutta(3,2) method based on the SSPRK(3,3)

$$\begin{aligned}
y^1 &= u^n,\\
y_i^2 &= u^n + \Delta t r_i\left(y^1\right) + \Delta t \sum_j \left(p_{ij}\left(y^1\right)\frac{y_j^2}{y_j^1} - d_{ij}\left(y^1\right)\frac{y_i^2}{y_i^1} \right),\\
y^3 &= u^n + \Delta t \frac{r_i\left(y^1\right) + r_i\left(y^2\right)}{4} + \Delta t \sum_j \left(\frac{p_{ij}\left(y^1\right) + p_{ij}\left(y^2\right)}{4} \frac{y_j^3}{y_j^2} - \frac{d_{ij}\left(y^1\right) + d_{ij}\left(y^2\right)}{4} \frac{y_i^3}{y_i^2} \right),\\
u^{n+1} &= u^n + \Delta t \frac{r_i\left(y^1\right) + r_i\left(y^2\right) + 4 r_i\left(y^3\right)}{6}\\
&\quad + \Delta t \sum_j \left(\frac{p_{ij}\left(y^1\right) + p_{ij}\left(y^2\right) + 4 p_{ij}\left(y^3\right)}{6} \frac{u_j^{n+1}}{y_j^2} - \frac{d_{ij}\left(y^1\right) + d_{ij}\left(y^2\right) + 4 d_{ij}\left(y^3\right)}{6} \frac{u_i^{n+1}}{y_i^2} \right).
\end{aligned} \tag{MPRK(3,2)}$$

- (Patankar) Semi Implicit Runge–Kutta(2,2) methods [2]

$$\begin{aligned}
y^1 &= u^n,\\
y_i^2 &= \frac{u_i^n + \Delta t r_i(y^1) + \Delta t \sum_j p_{ij}(y^1)}{1 + \Delta t \sum_j d_{ij}(y^1)/y_i^1},\\
y_i^3 &= \frac{1}{2}u_i^n + \frac{1}{2}\frac{y_i^2 + \Delta t r_i(y^2) + \Delta t \sum_j p_{ij}(y^2)}{1 + \Delta t \sum_j d_{ij}(y^2)/y_i^2},\\
u_i^{n+1} &= \frac{y_i^3 + \Delta t^2 \big(r_i(y^3) + \Delta t \sum_j p_{ij}(y^3)\big) \sum_j d_{ij}(y^3)/y_i^3}{1 + \big(\Delta t \sum_j d_{ij}(y^3)/y_i^3\big)^2}.
\end{aligned} \tag{SI-RK2}$$

- Modified Patankar Runge–Kutta(4,3,α,β) methods [6]

$$\begin{aligned}
y^1 &= u^n,\\
y_i^2 &= \frac{u_i^n + \Delta t r_i(y^1) + \Delta t \sum_j p_{ij}(y^1)}{1 + \Delta t \sum_j d_{ij}(y^1)/y_i^1},\\
y_i^3 &= \frac{3}{4}u_i^n + \frac{1}{4}\frac{y_i^2 + \Delta t r_i(y^2) + \Delta t \sum_j p_{ij}(y^2)}{1 + \Delta t \sum_j d_{ij}(y^2)/y_i^2},\\
y_i^4 &= \frac{1}{3}u_i^n + \frac{2}{3}\frac{y_i^3 + \Delta t r_i(y^3) + \Delta t \sum_j p_{ij}(y^3)}{1 + \Delta t \sum_j d_{ij}(y^3)/y_i^3},\\
u_i^{n+1} &= \frac{y_i^4 + \Delta t^2 \big(r_i(y^4) + \Delta t \sum_j p_{ij}(y^4)\big) \sum_j d_{ij}(y^4)/y_i^4}{1 + \big(\Delta t \sum_j d_{ij}(y^4)/y_i^4\big)^2}.
\end{aligned} \tag{SI-RK3}$$

- (Patankar) Semi Implicit Runge–Kutta(2,2) methods [2]

$$\begin{aligned}
y^1 &= u^n,\\
y_i^2 &= u_i^n + a_{21}\Delta t r_i\left(y^1\right) + a_{21}\Delta t \sum_j \left(p_{ij}\left(y^1\right)\frac{y_j^2}{y_j^1} - d_{ij}\left(y^1\right)\frac{y_i^2}{y_i^1}\right),\\
y_i^3 &= u_i^n + \Delta t \left(a_{31} r_i\left(y^1\right) + a_{32} r_i\left(y^2\right)\right)\\
&\quad + \Delta t \sum_j \left(\left(a_{31} p_{ij}\left(y^1\right) + a_{32} p_{ij}\left(y^2\right)\right) \frac{y_j^3}{\left(y_j^2\right)^{1/p}\left(y_j^1\right)^{1/p-1}} \right.\\
&\qquad \left. - \left(a_{31} d_{ij}\left(y^1\right) + a_{32} d_{ij}\left(y^2\right)\right) \frac{y_i^3}{\left(y_i^2\right)^{1/p}\left(y_i^1\right)^{1/p-1}}\right),\\
\sigma_i &= u_i^n + \Delta t \sum_j \left(\beta_1 p_{ij}\left(y^1\right) + \beta_2 p_{ij}\left(y^2\right)\right) \frac{\sigma_j}{\left(y_j^2\right)^{1/q}\left(y_j^1\right)^{1/q-1}}\\
&\quad \left(\beta_1 d_{ij}\left(y^1\right) + \beta_2 d_{ij}\left(y^2\right)\right) \frac{\sigma_i}{\left(y_i^2\right)^{1/q}\left(y_i^1\right)^{1/q-1}}\\
u_i^{n+1} &= u_i^n + \Delta t \left(b_1 r_i\left(y^1\right) + b_2 r_i\left(y^2\right) + b_3 r_i\left(y^3\right)\right)\\
&\quad + \Delta t \sum_j \left(\left(b_1 p_{ij}\left(y^1\right) + b_2 p_{ij}\left(y^2\right) + b_3 p_{ij}\left(y^3\right)\right) \frac{u_j^{n+1}}{\sigma_j} \right.\\
&\qquad \left. - \left(b_1 d_{ij}\left(y^1\right) + b_2 d_{ij}\left(y^2\right) + b_3 d_{ij}\left(y^3\right)\right) \frac{u_i^{n+1}}{\sigma_i}\right),
\end{aligned} \tag{MPRK(4,3,α,β)}$$

where $p = 3a_{21}\left(a_{31} + a_{32}\right) b_3,\ q = a_{21},\ \beta_2 = \frac{1}{2a_{21}}$ and $\beta_1 = 1 - \beta_2$. The Butcher tableaus in respect to the two parameters

$$\begin{array}{c|ccc}
0 & & & \\
\alpha & \alpha & & \\
\beta & \frac{3\alpha\beta(1-\alpha)-\beta^2}{\alpha(2-3\alpha)} & \frac{\beta(\beta-\alpha)}{\alpha(2-3\alpha)} & \\
\hline
 & 1 + \frac{2-3(\alpha+\beta)}{6\alpha\beta} & \frac{3\beta-2}{6\alpha(\beta-\alpha)} & \frac{2-3\alpha}{6\beta(\beta-\alpha)}
\end{array} \tag{2}$$

with

$$\left.\begin{aligned} 2/3 \le \beta \le 3\alpha(1-\alpha) \\ 3\alpha(1-\alpha) \le \beta \le 2/3 \\ (3\alpha-2)/(6\alpha-3) \le \beta \le 2/3 \end{aligned}\right\} \text{ for } \begin{cases} 1/3 \le \alpha < \frac{2}{3}, \\ 2/3 \le \alpha < \alpha_0, \\ \alpha > \alpha_0, \end{cases}$$

and $\alpha_0 \approx 0.89255$.

- Modified Patankar Shu–Osher Runge–Kutta(4,3) method [5]

$$\begin{aligned}
y^1 &= u^n, \\
y_i^2 &= y_i^1 + a_{10}\Delta t r_i\left(y^1\right) + \Delta t \sum_j b_{10}\left(p_{ij}\left(y^1\right)\frac{y_j^2}{y_j^1} - d_{ij}\left(y^1\right)\frac{y_i^2}{y_i^1}\right), \\
\varrho_i &= n_1 y_i^2 + n_2 y_i^1 \left(\frac{y_i^2}{y_i^1}\right)^2 \\
y_i^3 &= (a_{20}y_i^1 + a_{21}y_i^2) + \Delta t\left(b_{20}r_i\left(y^1\right) + b_{21}r_i\left(y^2\right)\right) \\
&\quad + \Delta t \sum_j \left(\left(b_{20}p_{ij}\left(y^1\right) + b_{21}p_{ij}\left(y^2\right)\right)\frac{y_j^2}{\varrho_j} - \left(b_{20}d_{ij}\left(y^1\right) + b_{21}d_{ij}\left(y^2\right)\right)\frac{y_i^2}{\varrho_i}\right), \\
\mu_i &= y_i^1\left(\frac{y_i^2}{y_i^1}\right)^s \\
\tilde{a}_i &= \eta_1 y_i^1 + \eta_2 y_i^2 + \Delta t \sum_j \left(\left(\eta_3 p_{ij}\left(y^1\right) + \eta_4 p_{ij}\left(y^2\right)\right)\frac{\tilde{a}_j}{\mu_j} - \left(\eta_3 d_{ij}\left(y^1\right) + \eta_4 d_{ij}\left(y^2\right)\right)\frac{\tilde{a}_i}{\mu_i}\right) \\
\sigma_i &= \tilde{a}_i + z y_i^1 \frac{y_i^2}{\varrho_i} \\
u_i^{n+1} &= \left(a_{30}y_i^1 + a_{31}y_i^2 + a_{32}y_i^3\right) + \Delta t\left(b_{30}r_i\left(y^1\right) + b_{31}r_i\left(y^2\right) + b_{32}r_i\left(y^3\right)\right) \\
&\quad + \Delta t \sum_j \Bigg(\left(b_{30}p_{ij}\left(y^1\right) + b_{31}p_{ij}\left(y^2\right) + b_{32}p_{ij}\left(y^2\right)\right)\frac{u_j^{n+1}}{\sigma_j} \\
&\qquad - \left(b_{30}d_{ij}\left(y^1\right) + b_{31}d_{ij}\left(y^2\right) + b_{32}d_{ij}\left(y^2\right)\right)\frac{u_i^{n+1}}{\sigma_i}\Bigg).
\end{aligned} \tag{MPRKSO(4,3)}$$

Here, the optimal SSP coefficients determined in [5] will be used. They are given by

$$\begin{aligned}
n_1 &= 2.569046025732011E-01, & n_2 &= 7.430953974267989E-01, \\
a_{10} &= 1, & a_{20} &= 9.2600312554031827E-01, \\
a_{21} &= 7.3996874459681783E-02, & a_{31} &= 2.0662904223744017E-10,
\end{aligned}$$

$$\begin{aligned}
b_{10} &= 4.7620819268131703E-01, & a_{30} &= 7.0439040373427619E-01,\\
a_{32} &= 2.9560959605909481E-01, & b_{20} &= 7.7545442722396801E-02,\\
b_{21} &= 5.9197500149679749E-01, & b_{31} &= 6.8214380786704851E-10,\\
b_{30} &= 2.0044747790361456E-01, & b_{32} &= 5.9121918658514827E-01,\\
\eta_1 &= 3.777285888379173E-02, & \eta_2 &= 1/3,\\
\eta_3 &= 1.868649805549811E-01, & \eta_3 &= 2.224876040351123,\\
z &= 6.288938077828750E-01, & s &= 5.721964308755304.
\end{aligned}$$

2 Validation on nonlinear problems

2.1 Scalar nonlinear problem

The second problem on which we are testing our methods on is a scalar ODE with a source term [2]. Find $u : [0, 0.15] \to \mathbb{R}$, with $u(0) = 1.1\sqrt{1/k}$, where $k > 0$ is a coefficient of the problem, and

$$u' = -k|u|u + 1. \tag{3}$$

The solution for this problem is monotonically decreasing and converging to $u_\infty = \sqrt{1/k}$.

The schemes can be applied to this problem following simple prescriptions.

- The source shall be integrated in time without considering the Patankar trick, simply using the coefficients of the original schemes.
- The productions and destruction terms must be rewritten as $d_{11} = k|u|u$ and $p_{11} = 0$.

We can see oscillations around the steady state produced by the schemes.

In this section, we want to validate the analysis done in the linear case, trying to understand if the Δt bound we found in the previous section can be useful in the nonlinear case as well. Aiming at that, we check the first time step, which often shows overshoots with respect to the steady state, for different time steps.

In particular, we can observe that the Lipschitz constant of the right-hand side of (3) is $C(k) := \max_u k|u| = k|u_0| = 1.1\sqrt{k}$. Hence, inspired by the theory for numerical PDEs, we define a CFL number in $\mathbb{R}^+$ and we set the Δt step as

$$\Delta t := \frac{\text{CFL}}{C(k)}. \tag{4}$$

Doing so, we essentially get rid of the dependence on k, through a rescaling factor both for time and amplitude on the solution. In this way, the CFL number should be comparable with the Δt bound found in the previous sections. We fix $k = 10^4$ for the following simulations, but proportional results can be obtained for different k.

Figure 1 shows the simulations for different CFLs. For low CFLs, we observe no oscillations for essentially all methods. Increasing the CFL number, we observe that most of the schemes go below u_∞ for the first timestep.

In Tables 1 and 2, we list the oscillation measure for all mPDeC methods with equispaced and Gauss–Lobatto subtimesteps, respectively. Increasing the CFL, we see that many schemes overshoot the steady values. In particular, whenever we are below the Δt bound of the stability analysis, we do not observe oscillations. In some cases, also above this bound we do not have oscillations, but this might depend on the problem itself.

In Table 3, we show similar results for MPRK(2,2,α) f or some α. In contrast to the previous case, we observe oscillations even if the bound is higher than the CFL tested.

In Table 4, we test MPRK(4,3,α,β), with some interesting values and then on the curve $\beta(6\alpha-3) = 3\alpha - 2$. The first values show oscillations according to the Δt bound found in the stability analysis,

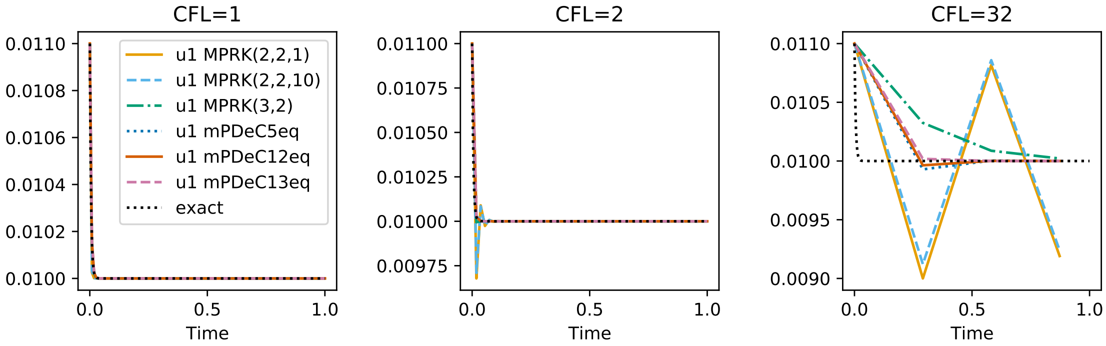


Figure 1: Simulations of (3) at different CFLs for some schemes.

Table 1: Oscillation measure for problem (3) with mPDeC schemes with equispaced subtimesteps.

CFL	0.5	1.0	2.0	4.0	8.0	16.0	32.0	64.0
MPDeC1eq	0	0	2.7e-04	5.3e-04	7.0e-04	8.0e-04	8.5e-04	8.8e-04
MPDeC2eq	0	0	3.2e-04	6.1e-04	8.1e-04	9.3e-04	1.0e-03	1.0e-03
MPDeC3eq	0	0	0	0	0	0	0	0
MPDeC4eq	0	0	0	0	0	2.8e-05	8.5e-05	1.3e-04
MPDeC5eq	0	0	0	0	0	2.0e-05	7.1e-05	1.1e-04
MPDeC6eq	0	0	0	2.6e-06	4.4e-05	1.1e-04	1.6e-04	1.7e-04
MPDeC7eq	0	0	0	0	0	0	0	0
MPDeC8eq	0	0	0	5.9e-07	9.5e-06	7.4e-06	5.0e-07	8.9e-06
MPDeC9eq	0	0	0	0	0	0	0	0
MPDeC10eq	0	0	0	0	1.2e-06	8.2e-06	5.4e-05	1.2e-04
MPDeC11eq	0	0	0	0	0	0	0	0
MPDeC12eq	0	0	0	0	0	7.5e-06	3.7e-05	5.7e-05
MPDeC13eq	0	0	0	0	0	0	0	0

Table 2: Oscillation measure for problem (3) with mPDeC schemes with Gauss–Lobatto subtimesteps.

CFL	0.5	1.0	2.0	4.0	8.0	16.0	32.0	64.0
MPDeC1GL	0	0	2.7e-04	5.3e-04	7.0e-04	8.0e-04	8.5e-04	8.8e-04
MPDeC2GL	0	0	3.2e-04	6.1e-04	8.1e-04	9.3e-04	1.0e-03	1.0e-03
MPDeC3GL	0	0	0	0	0	0	0	0
MPDeC4GL	0	0	0	0	0	0	0	0
MPDeC5GL	0	0	0	2.8e-06	6.2e-05	2.0e-04	3.4e-04	4.4e-04
MPDeC6GL	0	0	0	2.4e-05	1.3e-04	3.2e-04	5.0e-04	6.2e-04
MPDeC7GL	0	0	0	0	1.7e-05	2.8e-05	1.1e-05	0
MPDeC8GL	0	0	0	0	0	0	0	0
MPDeC9GL	0	0	0	0	0	0	0	0
MPDeC10GL	0	0	0	0	0	0	0	0
MPDeC11GL	0	0	0	0	1.4e-06	1.9e-05	9.0e-05	1.9e-04
MPDeC12GL	0	0	0	0	1.6e-06	2.5e-05	1.1e-04	2.1e-04
MPDeC13GL	0	0	0	0	0	2.2e-06	9.9e-06	3.2e-06

Table 3: Oscillation measure for problem (3) with MPRK(2,2,α) for few α.

CFL	0.5	1.0	2.0	4.0	8.0	16.0	32.0	64.0
MPRK(2,2,0.5)	0	0	2.7e-04	5.3e-04	7.0e-04	8.0e-04	8.5e-04	8.8e-04
MPRK(2,2,0.8)	0	0	3.1e-04	6.0e-04	8.0e-04	9.2e-04	9.9e-04	1.0e-03
MPRK(2,2,1.0)	0	0	3.2e-04	6.1e-04	8.1e-04	9.3e-04	1.0e-03	1.0e-03
MPRK(2,2,1.5)	0	0	3.3e-04	6.1e-04	8.0e-04	9.2e-04	9.8e-04	1.0e-03
MPRK(2,2,2.0)	0	0	3.2e-04	6.0e-04	7.9e-04	9.0e-04	9.6e-04	9.9e-04
MPRK(2,2,10.0)	0	0	2.9e-04	5.5e-04	7.2e-04	8.2e-04	8.8e-04	9.1e-04

Table 4: Oscillation measure for problem (3) with MPRK(4,3,α,β) for some interesting α,β. The second half of the table is on the curve $\beta(6\alpha - 3) = 3\alpha - 2$

CFL	0.5	1.0	2.0	4.0	8.0	16.0	32.0	64.0
MPRK(4,3,0.50,0.70)	0	0	6.4e-05	1.3e-04	1.1e-04	5.8e-05	1.8e-05	0
MPRK(4,3,0.90,0.65)	0	0	7.5e-05	1.6e-04	1.6e-04	1.4e-04	1.2e-04	1.1e-04
MPRK(4,3,1.00,0.67)	0	0	7.9e-05	1.7e-04	1.8e-04	1.6e-04	1.5e-04	1.3e-04
MPRK(4,3,1.00,0.33)	0	0	0	0	0	0	0	0
MPRK(4,3,1.25,0.60)	0	0	5.7e-05	1.2e-04	9.3e-05	4.6e-05	7.5e-06	0
MPRK(4,3,2.00,0.60)	0	0	4.5e-05	9.4e-05	5.7e-05	2.5e-07	0	0
MPRK(4,3,10.00,0.60)	0	0	2.1e-05	5.2e-05	0	0	0	0
MPRK(4,3,0.50,0.75)	0	0	6.8e-05	1.4e-04	1.3e-04	1.0e-04	7.0e-05	5.1e-05
MPRK(4,3,0.70,0.63)	0	0	1.0e-04	2.2e-04	2.7e-04	2.9e-04	2.9e-04	2.9e-04
MPRK(4,3,0.80,0.48)	0	0	8.4e-05	1.8e-04	1.9e-04	1.7e-04	1.5e-04	1.3e-04
MPRK(4,3,0.89,0.29)	0	0	1.8e-05	0	0	0	0	0
MPRK(4,3,0.90,0.29)	0	0	1.4e-05	0	0	0	0	0
MPRK(4,3,1.00,0.33)	0	0	0	0	0	0	0	0
MPRK(4,3,1.25,0.39)	0	0	0	0	0	0	0	0
MPRK(4,3,2.00,0.44)	0	0	0	0	0	0	0	0
MPRK(4,3,10.00,0.49)	0	0	0	0	0	0	0	0

Table 5: Oscillation measure for problem (3) with MPRKSO(2,2,α,β) for some interesting α,β

CFL	0.5	1.0	2.0	4.0	8.0	16.0	32.0	64.0
MPRKSO(2,2,0.0,0.5)	0	0	2.7e-04	5.3e-04	7.0e-04	8.0e-04	8.5e-04	8.8e-04
MPRKSO(2,2,0.0,1.0)	0	0	3.2e-04	6.1e-04	8.1e-04	9.3e-04	1.0e-03	1.0e-03
MPRKSO(2,2,0.0,2.0)	0	0	3.2e-04	6.0e-04	7.9e-04	9.0e-04	9.6e-04	9.9e-04
MPRKSO(2,2,0.0,5.0)	0	0	3.0e-04	5.7e-04	7.4e-04	8.5e-04	9.0e-04	9.3e-04
MPRKSO(2,2,0.0,10.0)	0	0	2.9e-04	5.5e-04	7.2e-04	8.2e-04	8.8e-04	9.1e-04
MPRKSO(2,2,0.1,1.5)	0	0	3.6e-04	6.7e-04	8.8e-04	1.0e-03	1.1e-03	1.1e-03
MPRKSO(2,2,0.1,6.0)	0	3.1e-04	9.7e-04	1.6e-03	2.1e-03	2.5e-03	2.7e-03	2.8e-03
MPRKSO(2,2,0.2,2.0)	0	5.9e-05	5.0e-04	9.1e-04	1.2e-03	1.4e-03	1.5e-03	1.6e-03
MPRKSO(2,2,0.3,1.5)	0	2.8e-05	4.5e-04	8.3e-04	1.1e-03	1.3e-03	1.4e-03	1.5e-03
MPRKSO(2,2,0.5,1.0)	0	0	2.7e-04	5.3e-04	7.0e-04	8.0e-04	8.5e-04	8.8e-04

Table 6: Oscillation measure for problem (3) with MPRK(3,2), MPRKSO(4,3), SI-RK2 and SI-RK3

CFL	0.5	1.0	2.0	4.0	8.0	16.0	32.0	64.0
MPRK(3,2)	0	0	5.2e-06	0	0	0	0	0
MPRKSO(4,3)	0	0	5.1e-05	7.7e-05	0	0	0	0
SIRK2	0	0	2.9e-04	5.4e-04	7.0e-04	8.0e-04	8.5e-04	8.8e-04
SIRK3	0	0	1.0e-04	3.3e-05	0	0	0	0

while, on the bottom curve, we observe no oscillations starting from $\alpha = 1$, which is slightly better then expected, considering the (large but not so large) Δt bounds in the stability analysis.

Another disappointing result comes from the schemes MPRKSO(2,2,α,β) in Figure 5, where even on the line $\alpha = 0$ we do not have oscillation-free simulations with large Δt as predicted by the stability analysis. Conversely, for the other parameters we expected the oscillations for almost all CFL larger than 1.

Finally, in Table 6, we have different behaviors, except for SI-RK2. The oscillations appear for CFL neither too small nor too large. This is surprising, first of all for MPRK(3,2) of which we expected no oscillations up to CFL ≈ 16, which shows anyway a very small oscillation (only very high order schemes have comparable oscillation amplitudes) only for CFL=2. For MPRKSO(4,3) and SI-RK3 we have slightly better results than expected for large CFLs and for SI-RK2 the results are exactly following the Δt bounds found in the stability analysis.

Conclusion 2.1. For this test, most of the schemes behaves as predicted based on the linear example, with few exceptions for second-order methods. The bounds of the linear case can mostly be transferred to the considered nonlinear problem. The linear analysis gives some meaningful results also for more challenging problems.

2.2 HIRES

This problem is called HIRES after Hairer and Wanner [3], referring to „High Irradiance RESponse“. The original problem HIRES [7, Section II.1] can be rewritten into a nine-dimensional production–destruction system with

$$\begin{aligned}
&r_1(u) = \sigma, && p_{21}(u) = d_{12}(u) = k_1 u_1, && p_{12}(u) = d_{21}(u) = k_2 u_2,\\
&p_{42}(u) = d_{24}(u) = k_3 u_2, && p_{43}(u) = d_{34}(u) = k_1 u_3, && p_{13}(u) = d_{31}(u) = k_6 u_3,\\
&p_{34}(u) = d_{43}(u) = k_2 u_4, && p_{64}(u) = d_{46}(u) = k_4 u_4, && p_{65}(u) = d_{56}(u) = k_1 u_5,\\
&p_{35}(u) = d_{53}(u) = k_5 u_5, && p_{56}(u) = d_{65}(u) = k_2 u_6, && p_{57}(u) = d_{75}(u) = \frac{k_2}{2} u_7,\\
&p_{67}(u) = d_{76}(u) = \frac{k_-}{2} u_7, && p_{97}(u) = d_{79}(u) = \frac{k_*}{2} u_7, && p_{76}(u) = d_{67}(u) = k_+ u_6 u_8,\\
&p_{78}(u) = d_{87}(u) = k_+ u_6 u_8, && p_{87}(u) = d_{78}(u) = \frac{k_- + k_* + k_2}{2} u_7.
\end{aligned} \tag{5}$$

with parameters

$$\begin{aligned}
&k_1 = 1.71, \quad k_2 = 0.43, \quad k_3 = 8.32, \quad k_4 = 0.69, \quad k_5 = 0.035,\\
&k_6 = 8.32, \quad k_+ = 280, \quad k_- = 0.69, \quad k_* = 0.69, \quad \sigma = 0.0007,
\end{aligned} \tag{6}$$

The time interval is $t \in [0, 321.8122]$.

For this test the concept of oscillation is not clear, nevertheless, we can observe the spurious steady states also for this problem. We compute the reference solution with 100,000 uniform timesteps. We use the mPDeC5 with equispaced subtimesteps to obtain this reference solution and we see that is in accordance with the reference solution [7] up to the fourth significant digits for all constituents.

Testing with $N = 1000$ uniform timesteps, we spot troubles with the *inconsistent* methods. We test the problem with many schemes presented above. For the mPDeC we spot the inconsistency problem only for equispaced timesteps for high odd orders $(9, 11, 13$ and so on). In Figure 18 we see the simulation for mPDeC6 with Gauss–Lobatto points. We observe that the high accuracy helps in obtaining a good result at the end of the simulation, when u_7 and u_8 react. The moment at which this change happens is really hard to catch and only high order methods are able to obtain it with this number of timesteps.

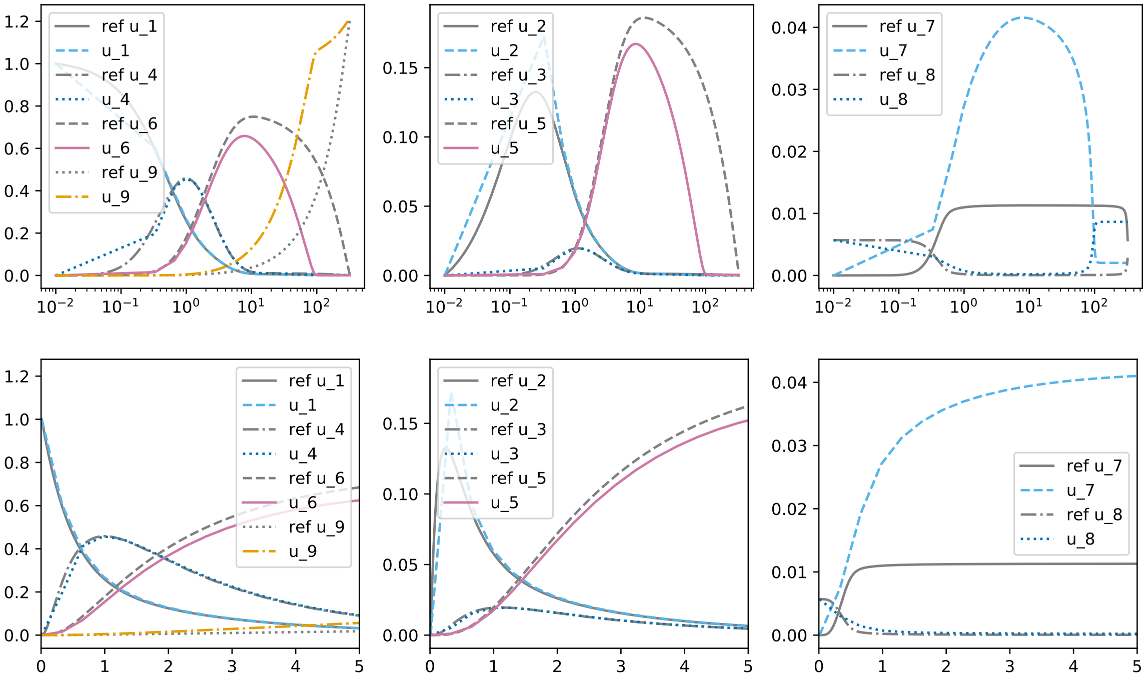


Figure 2: Simulations run with MPRK(2,2,α) with $\alpha = 1$ with $N = 1000$ timesteps, top logarithmic scale in time, bottom zoom on $t \in [0, 5]$

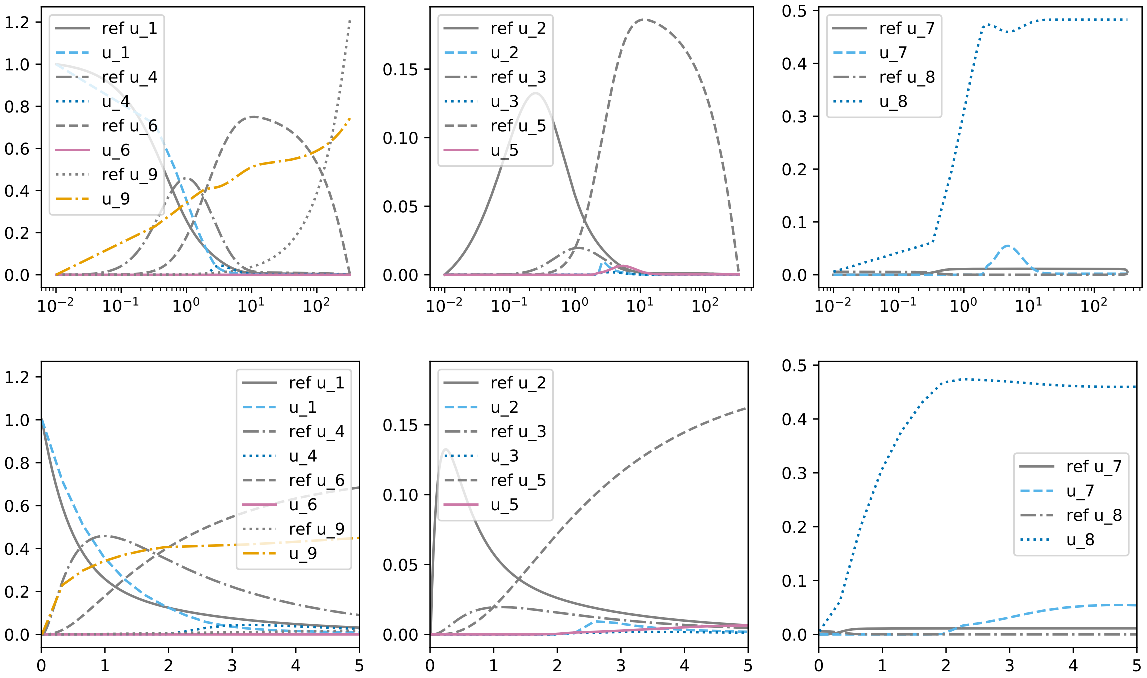


Figure 3: Simulations run with MPRK(2,2,α) with $\alpha = 5$ with $N = 1000$ timesteps, top logarithmic scale in time, bottom zoom on $t \in [0, 5]$

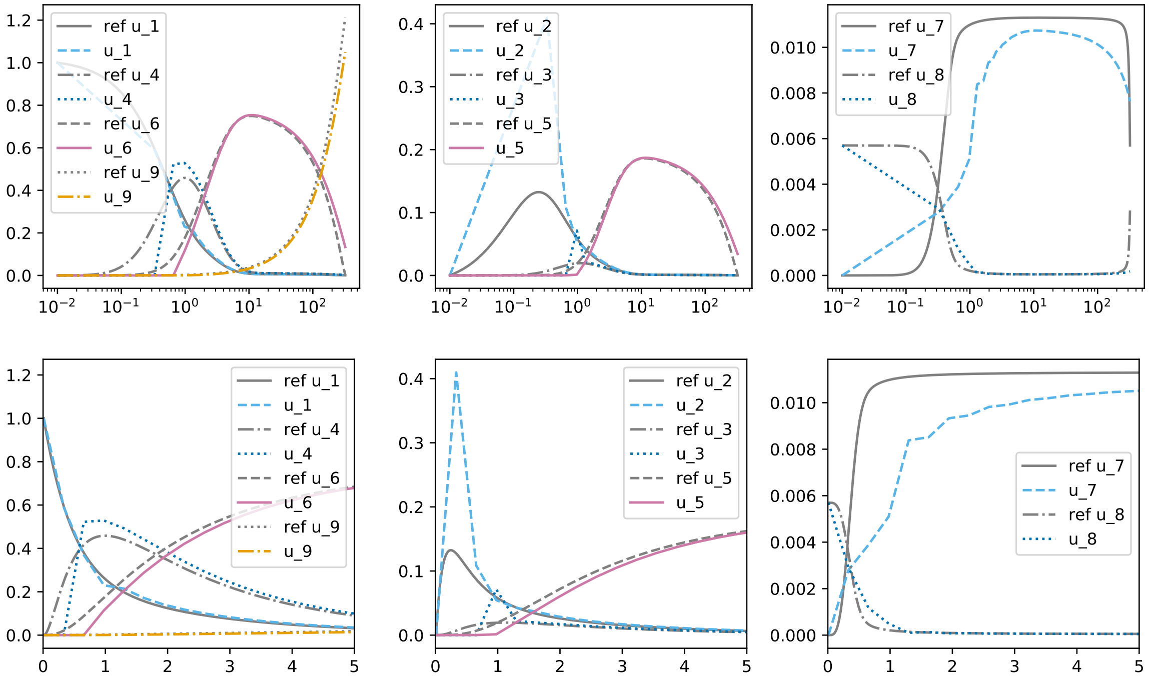


Figure 4: Simulations run with MPRK(2,2,α) with $\alpha = 0.7$ with $N = 1000$ timesteps, top logarithmic scale in time, bottom zoom on $t \in [0,5]$

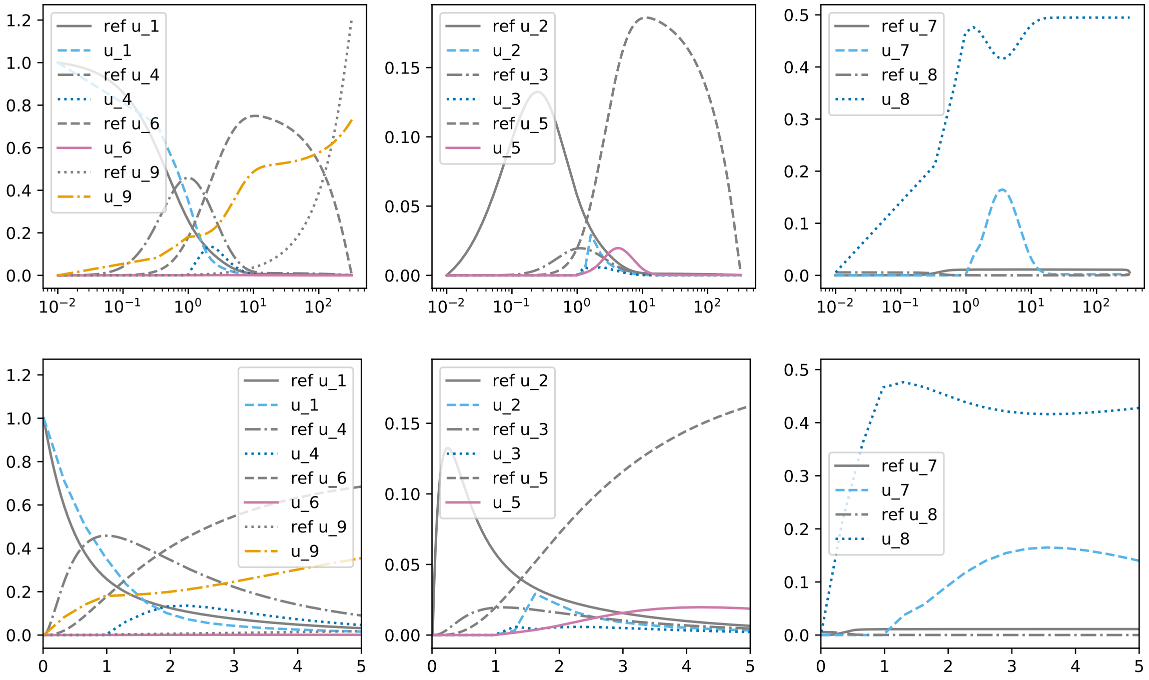


Figure 5: Simulations run with MPRK(4,3,α,β) with $\alpha = 5$ and $\beta = 0.5$ with $N = 1000$ timesteps, top logarithmic scale in time, bottom zoom on $t \in [0,5]$

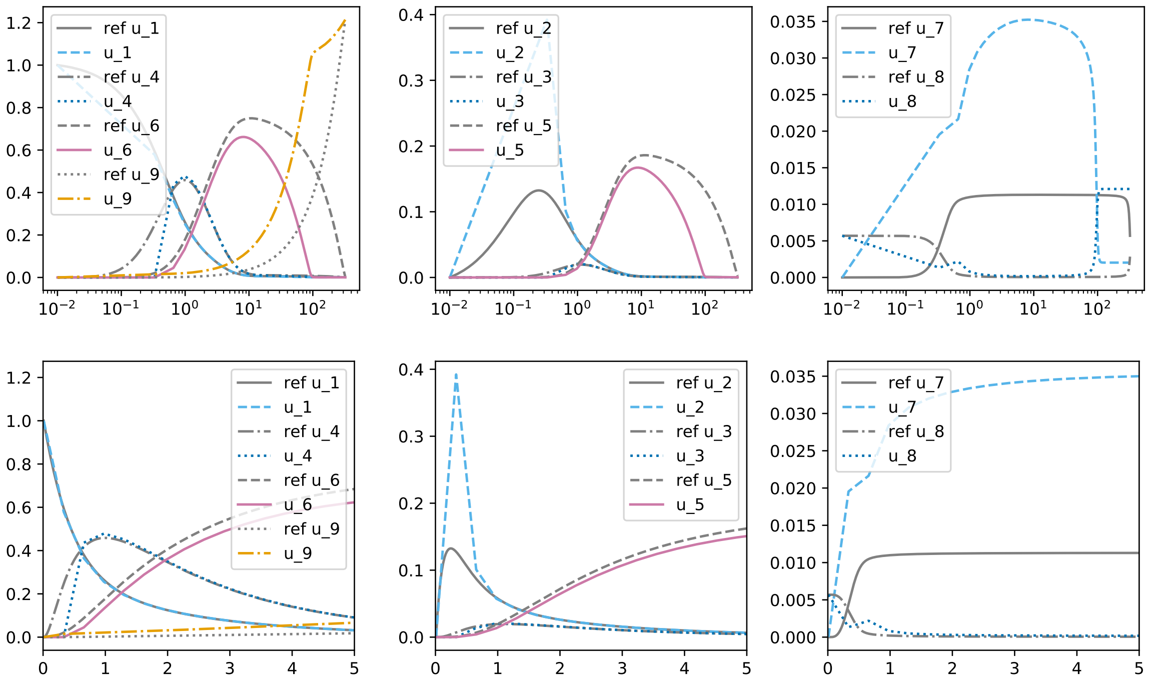


Figure 6: Simulations run with MPRK(4,3,α,β) with $\alpha = 0.9$ and $\beta = 0.6$ with $N = 1000$ timesteps, top logarithmic scale in time, bottom zoom on $t \in [0, 5]$

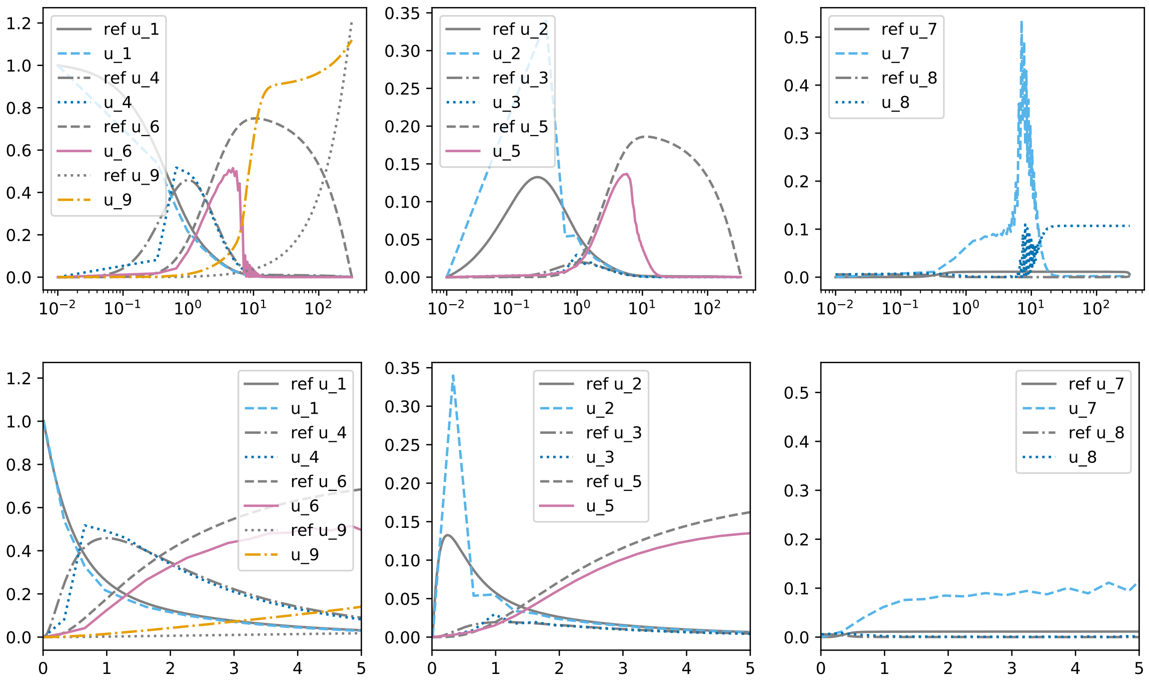


Figure 7: Simulations run with MPRKSO(2,2,α,β) with $\alpha = 0.3$ and $\beta = 2$ with $N = 1000$ timesteps, top logarithmic scale in time, bottom zoom on $t \in [0, 5]$

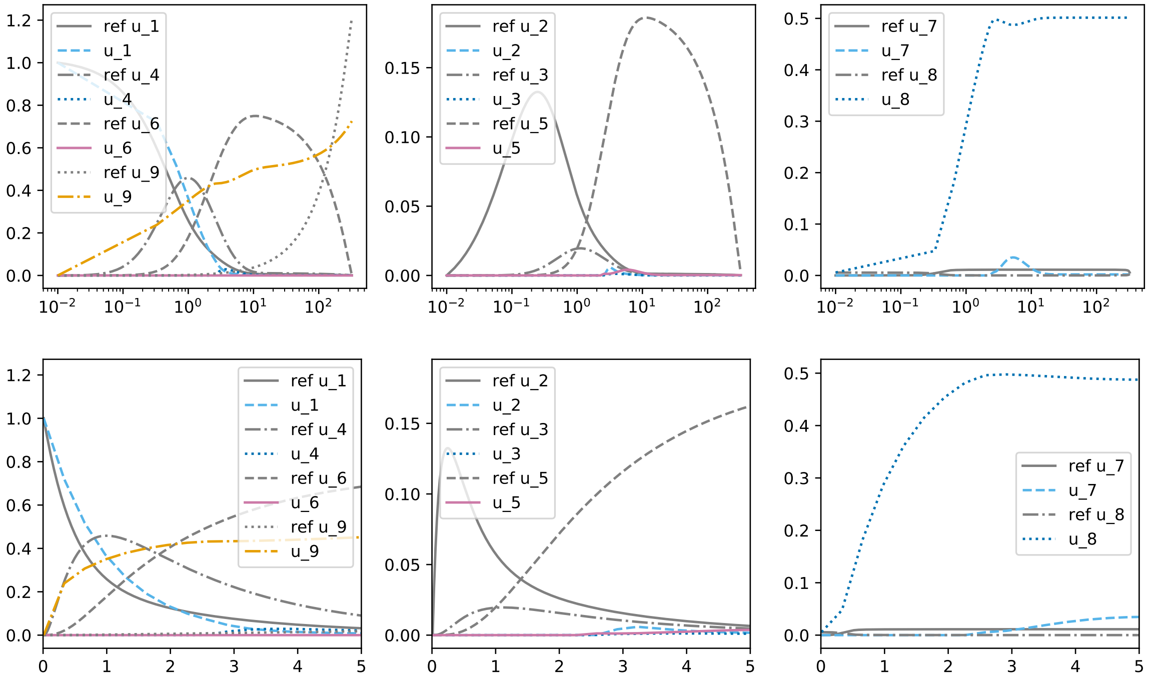


Figure 8: Simulations run with MPRKSO(2,2,α,β) with $\alpha = 0$ and $\beta = 8$ with $N = 1000$ timesteps, top logarithmic scale in time, bottom zoom on $t \in [0, 5]$

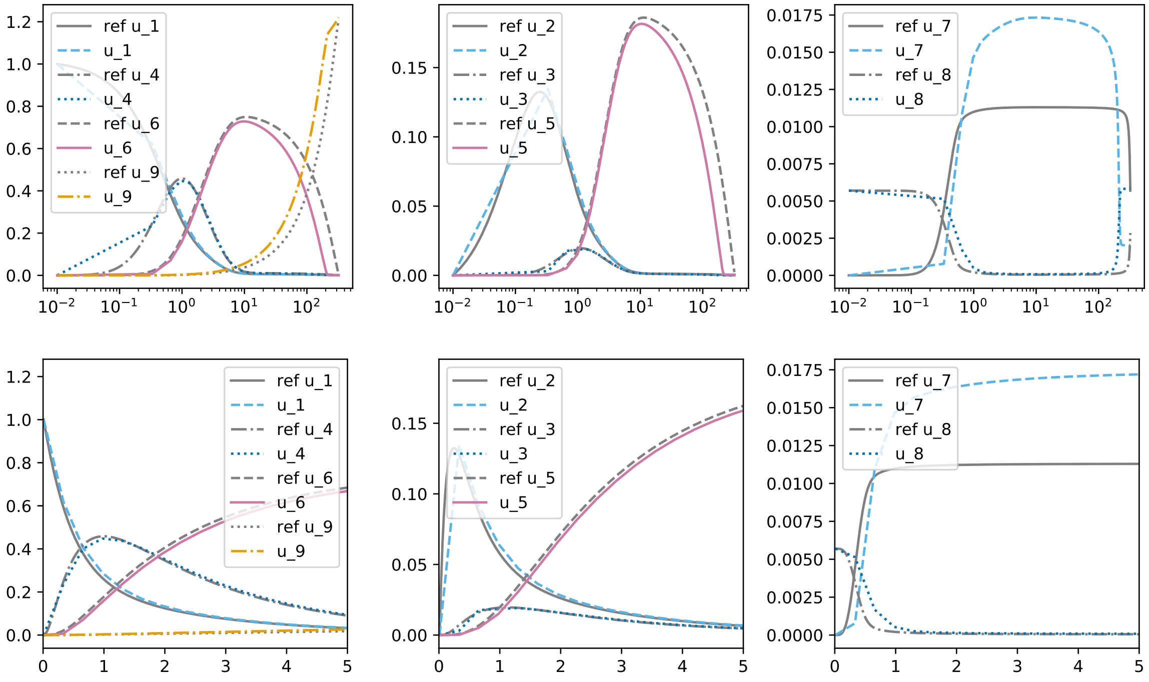


Figure 9: Simulations run with MPRKSO(4,3) with $\alpha = 0$ and $\beta = 8$ with $N = 1000$ timesteps, top logarithmic scale in time, bottom zoom on $t \in [0, 5]$

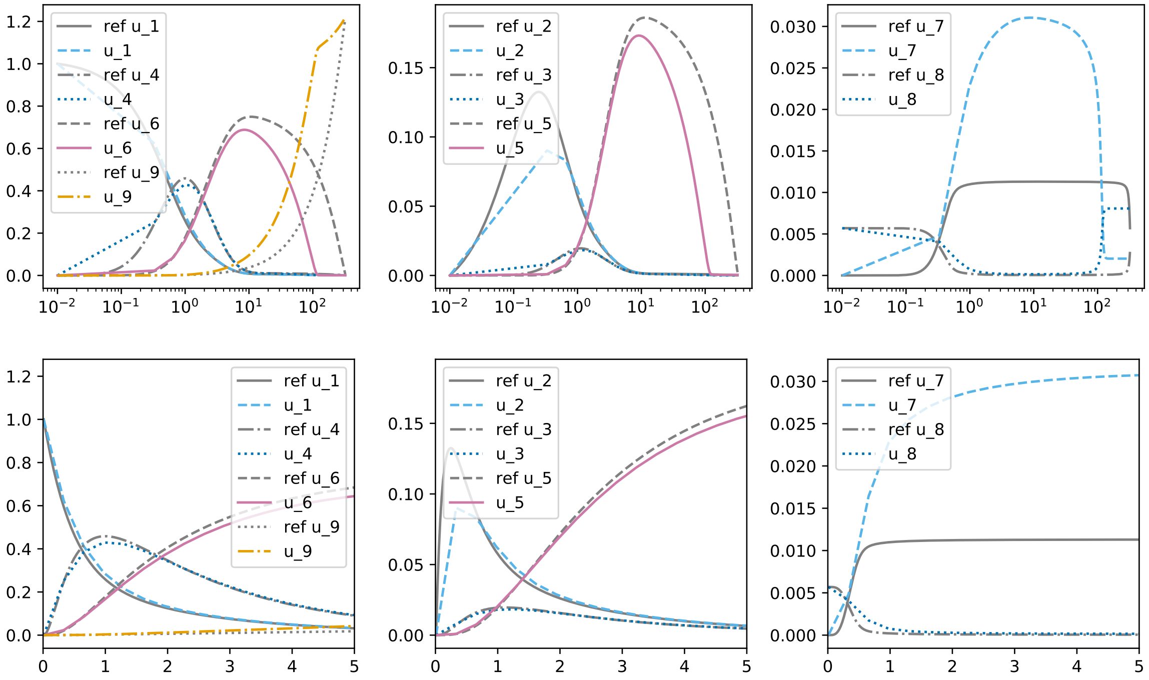


Figure 10: Simulations run with MPRK(3,2) with $N = 1000$ timesteps, top logarithmic scale in time, bottom zoom on $t \in [0, 5]$

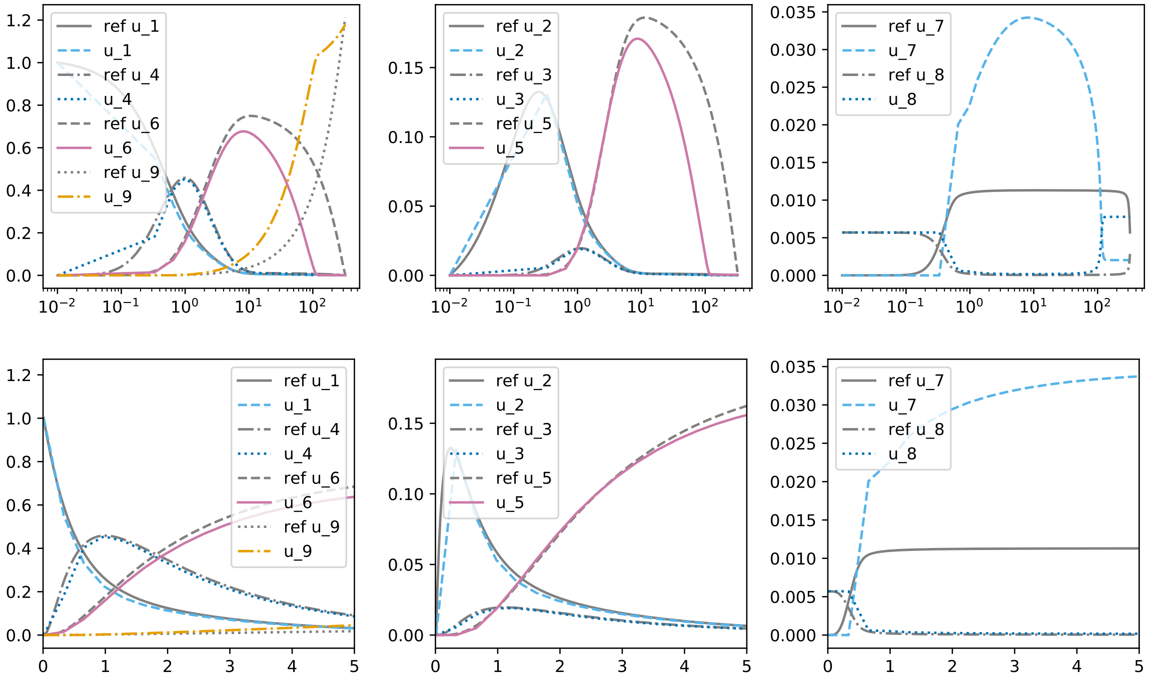


Figure 11: Simulations run with SIRK2 with $N = 1000$ timesteps, top logarithmic scale in time, bottom zoom on $t \in [0, 5]$

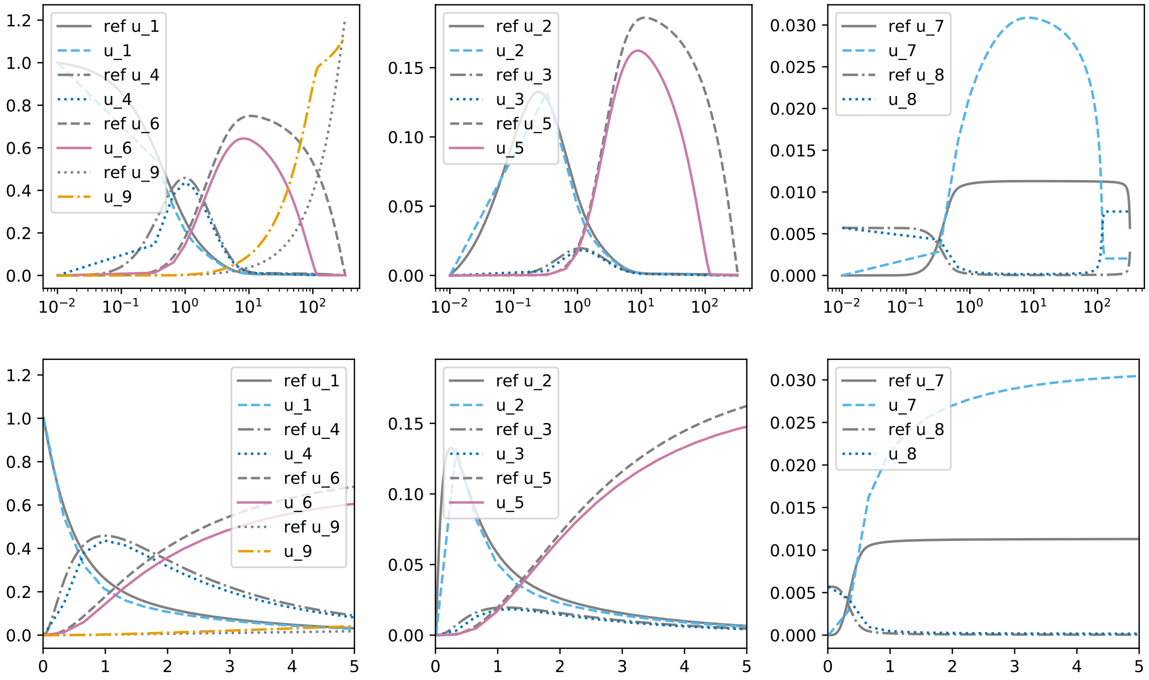


Figure 12: Simulations run with SIRK3 with $N = 1000$ timesteps, top logarithmic scale in time, bottom zoom on $t \in [0, 5]$

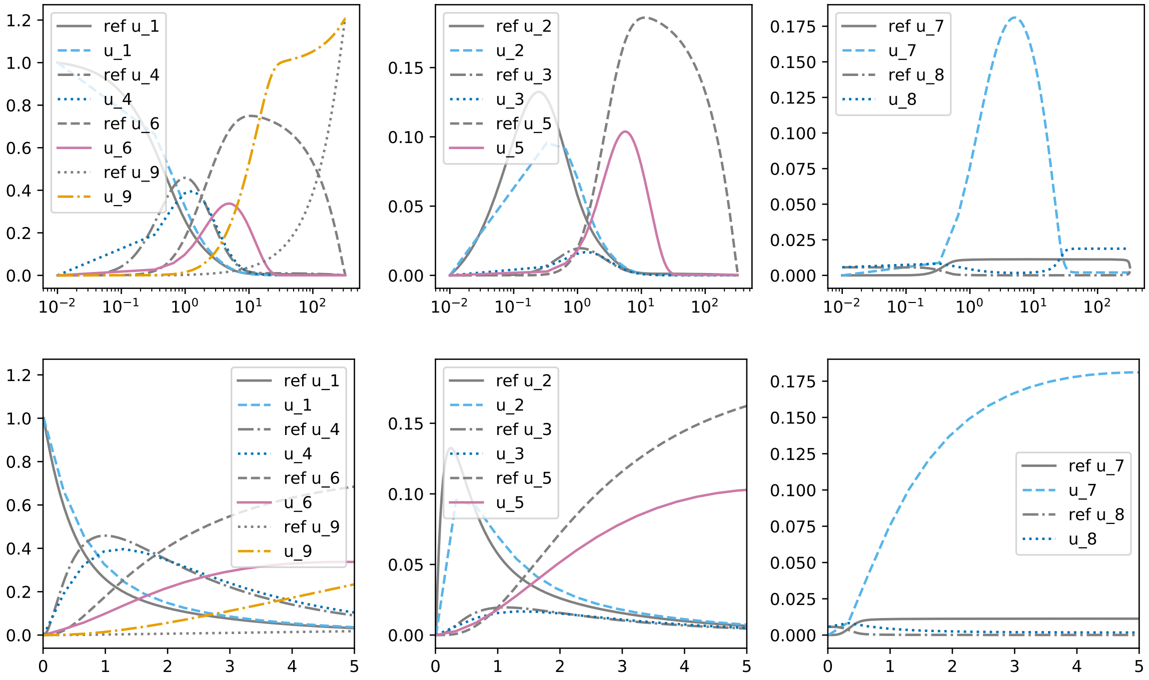


Figure 13: Simulations run with mPDeC1 with Gauss–Lobatto points with $N = 1000$ timesteps, top logarithmic scale in time, bottom zoom on $t \in [0, 5]$

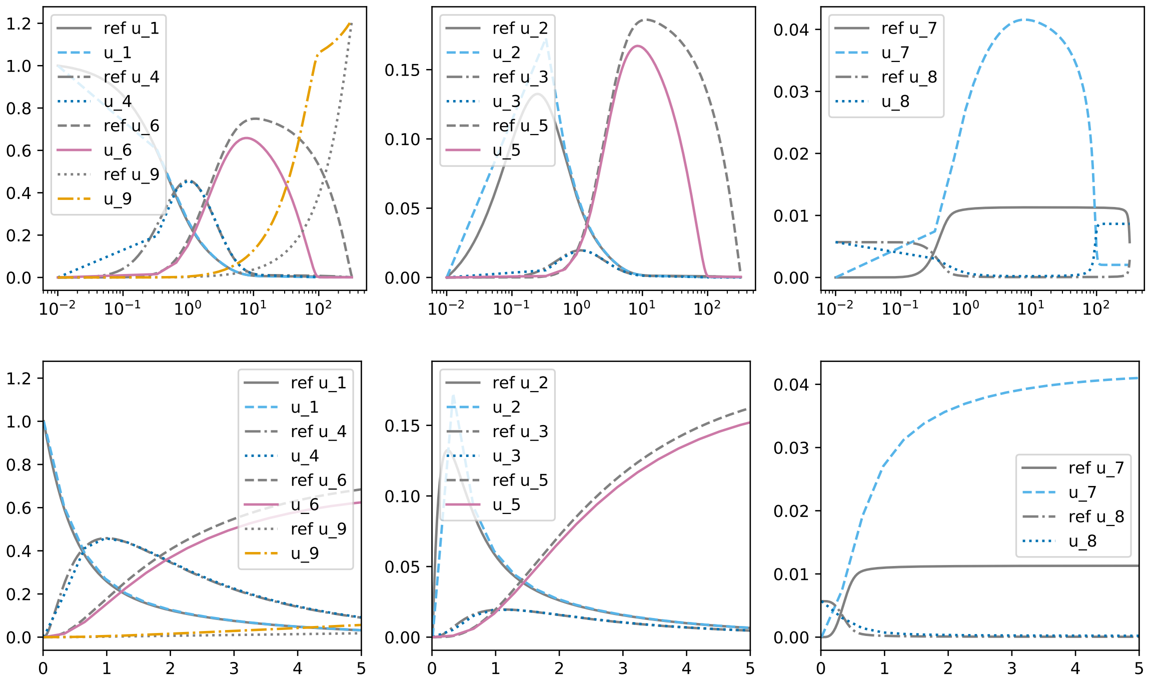


Figure 14: Simulations run with mPDeC2 with Gauss–Lobatto points with $N = 1000$ timesteps, top logarithmic scale in time, bottom zoom on $t \in [0,5]$

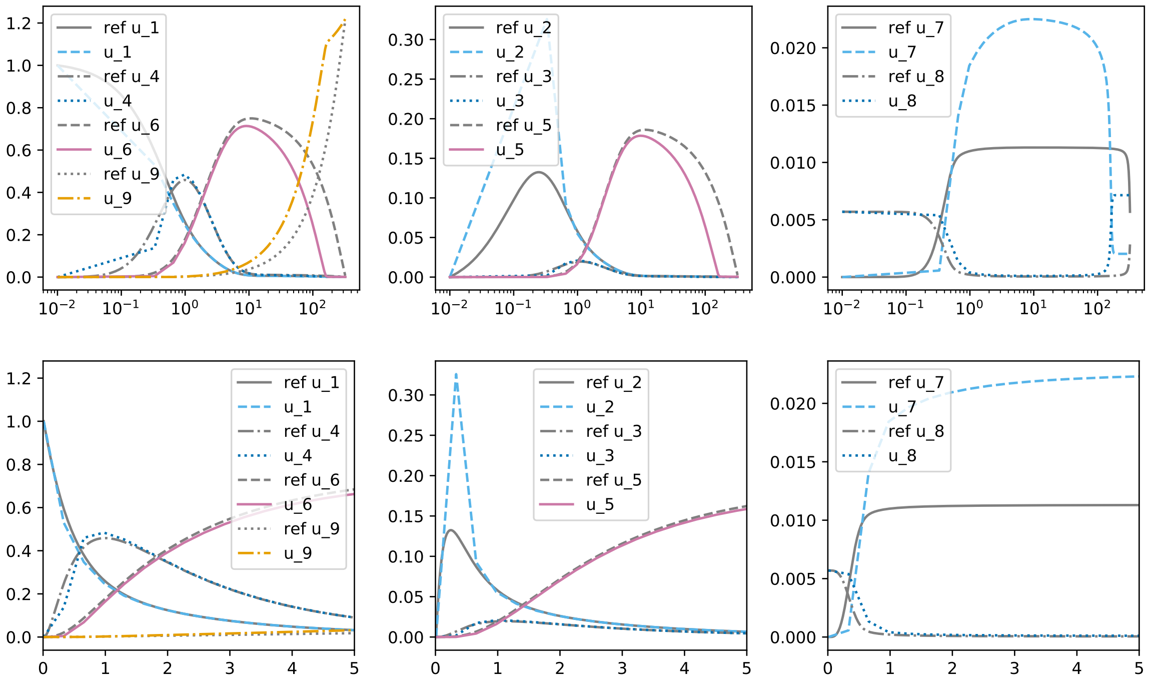


Figure 15: Simulations run with mPDeC3 with Gauss–Lobatto points with $N = 1000$ timesteps, top logarithmic scale in time, bottom zoom on $t \in [0,5]$

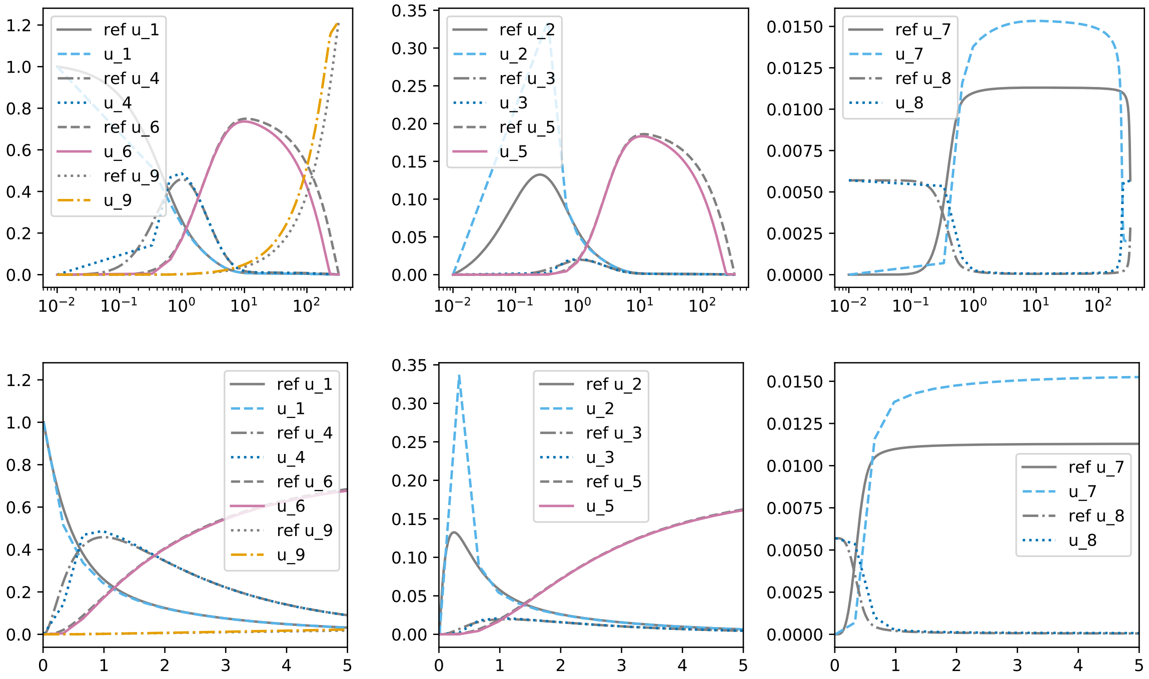


Figure 16: Simulations run with mPDeC4 with Gauss–Lobatto points with $N = 1000$ timesteps, top logarithmic scale in time, bottom zoom on $t \in [0,5]$

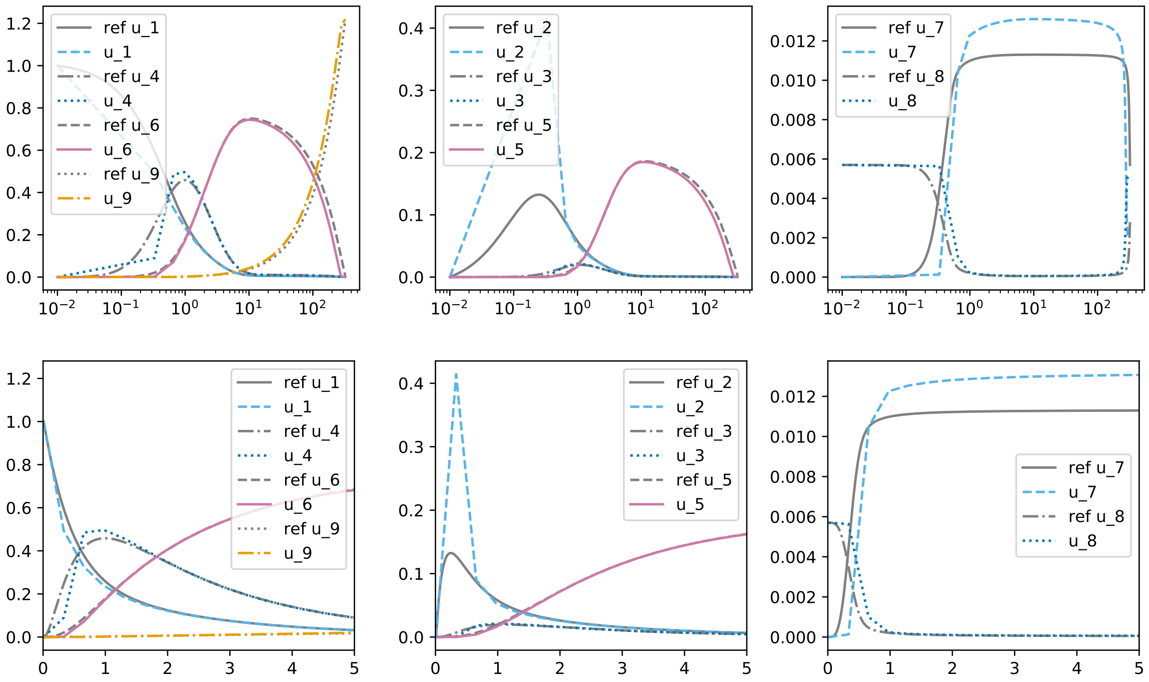


Figure 17: Simulations run with mPDeC5 with Gauss–Lobatto points with $N = 1000$ timesteps, top logarithmic scale in time, bottom zoom on $t \in [0,5]$

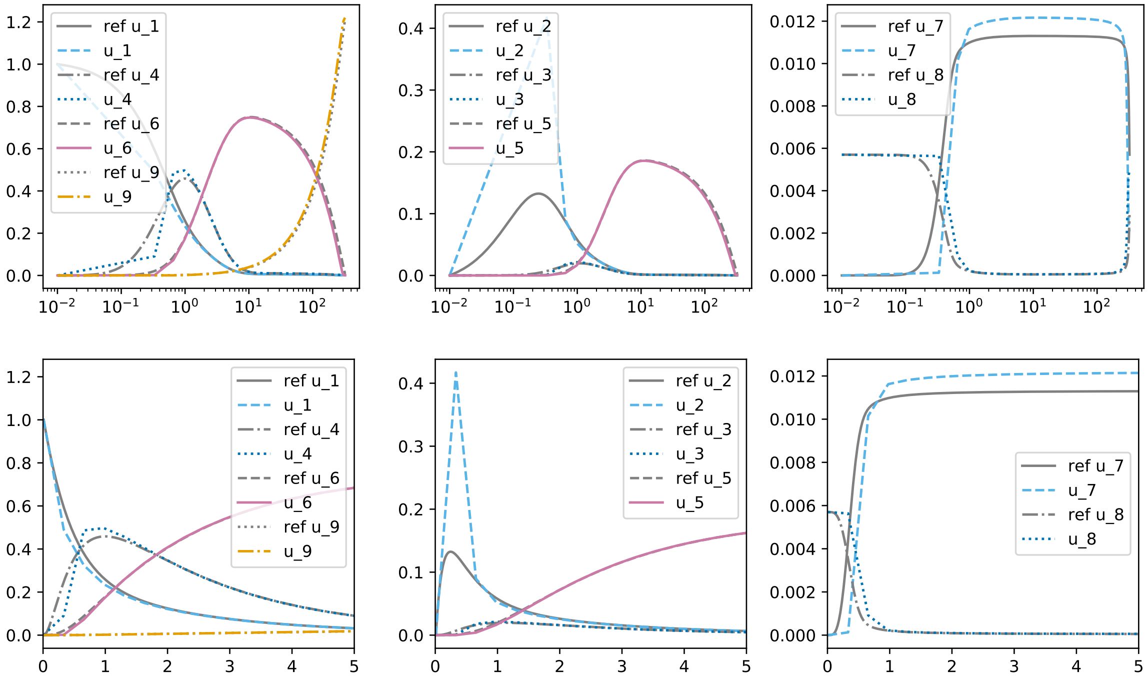


Figure 18: Simulations run with mPDeC6 with Gauss–Lobatto points with $N = 1000$ timesteps, top logarithmic scale in time, bottom zoom on $t \in [0, 5]$

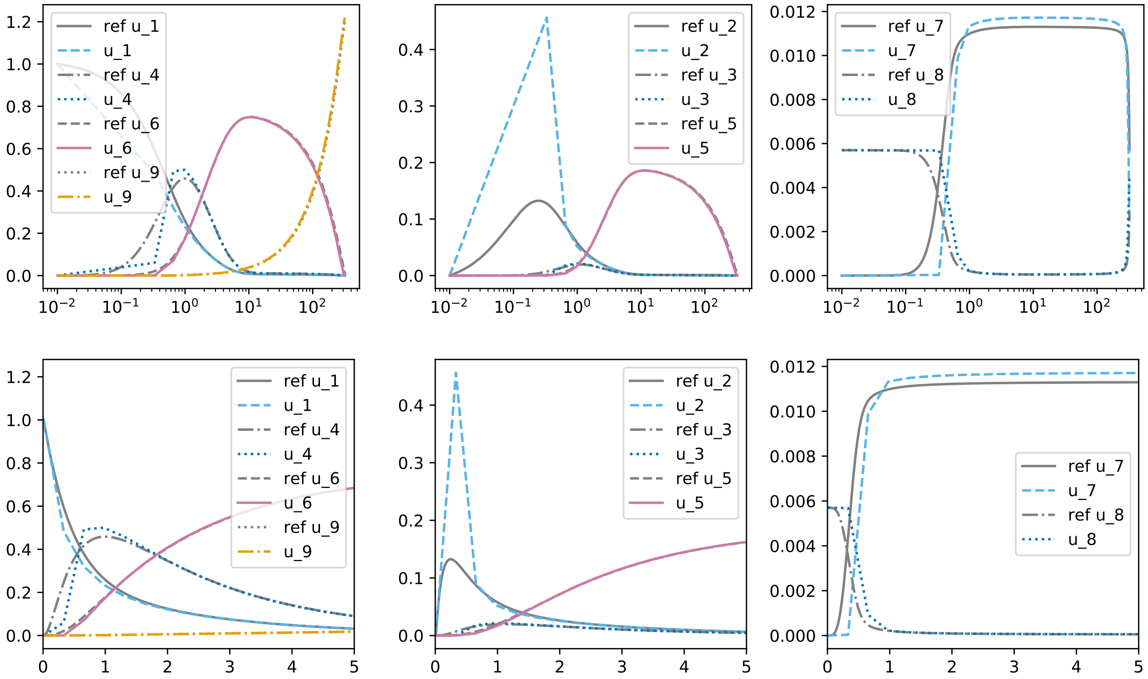


Figure 19: Simulations run with mPDeC7 with Gauss–Lobatto points with $N = 1000$ timesteps, top logarithmic scale in time, bottom zoom on $t \in [0, 5]$

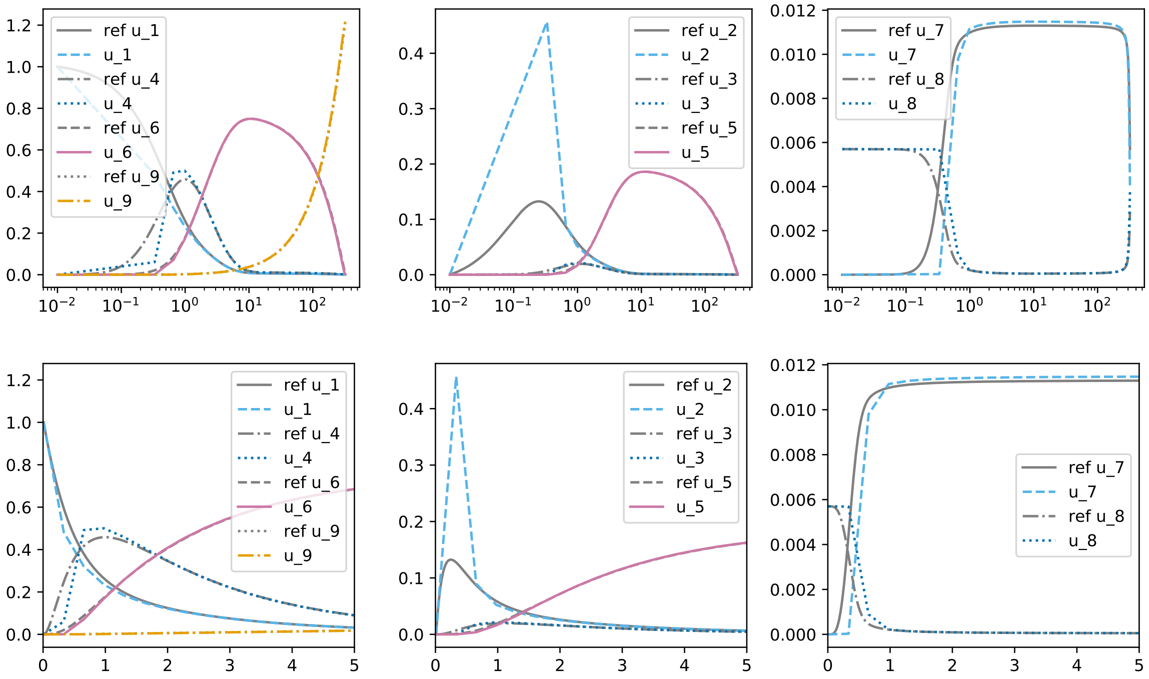


Figure 20: Simulations run with mPDeC8 with Gauss–Lobatto points with $N = 1000$ timesteps, top logarithmic scale in time, bottom zoom on $t \in [0, 5]$

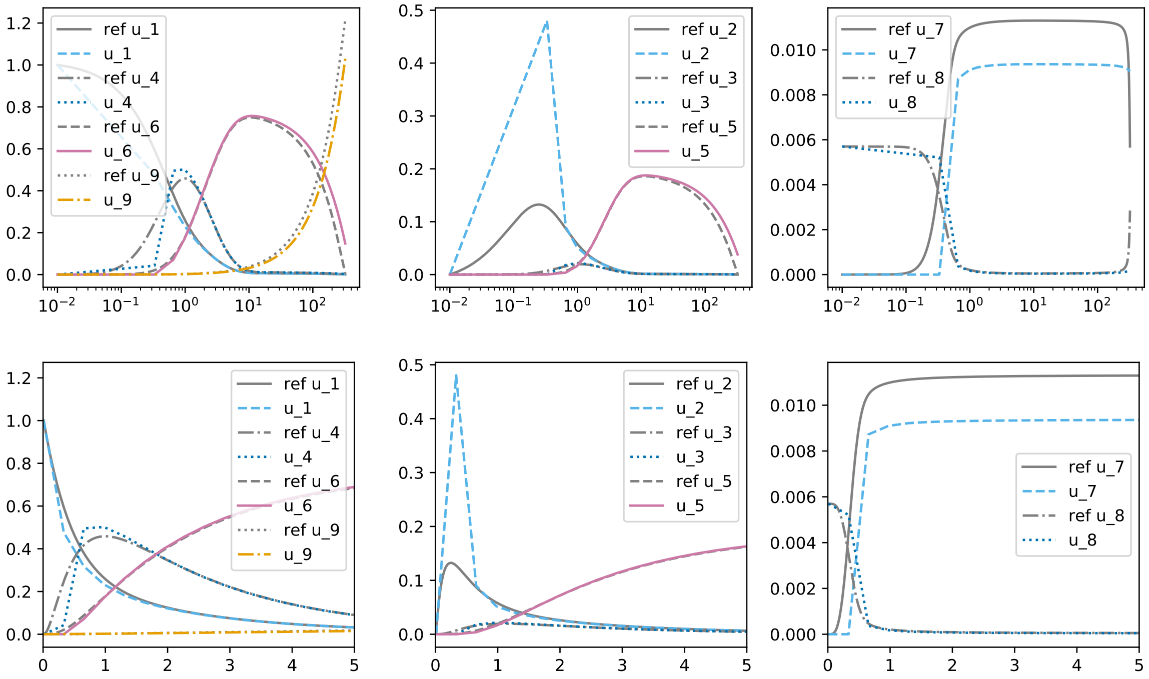


Figure 21: Simulations run with mPDeC9 with Gauss–Lobatto points with $N = 1000$ timesteps, top logarithmic scale in time, bottom zoom on $t \in [0, 5]$

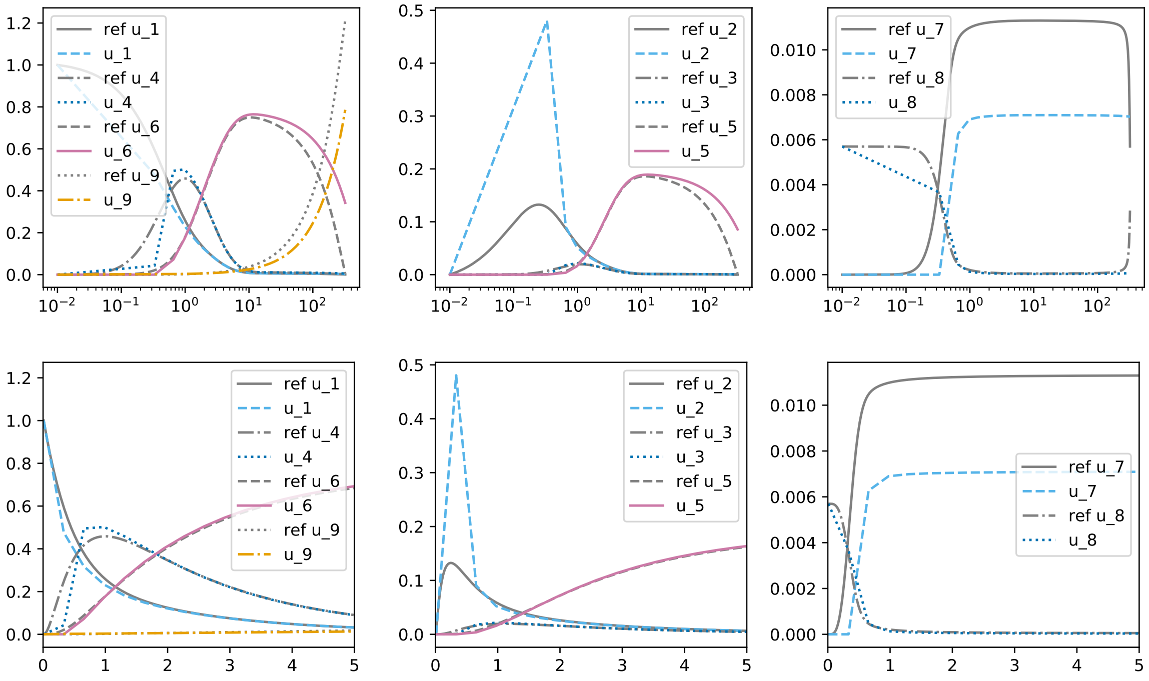


Figure 22: Simulations run with mPDeC10 with Gauss–Lobatto points with $N = 1000$ timesteps, top logarithmic scale in time, bottom zoom on $t \in [0, 5]$

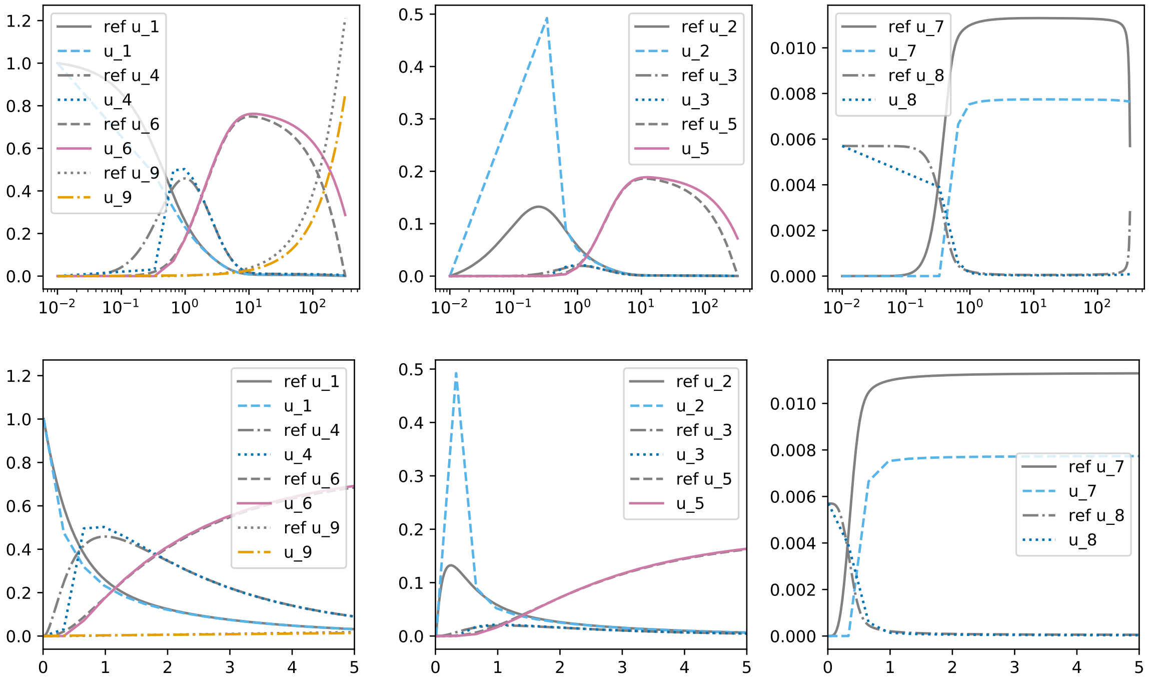


Figure 23: Simulations run with mPDeC11 with Gauss–Lobatto points with $N = 1000$ timesteps, top logarithmic scale in time, bottom zoom on $t \in [0, 5]$

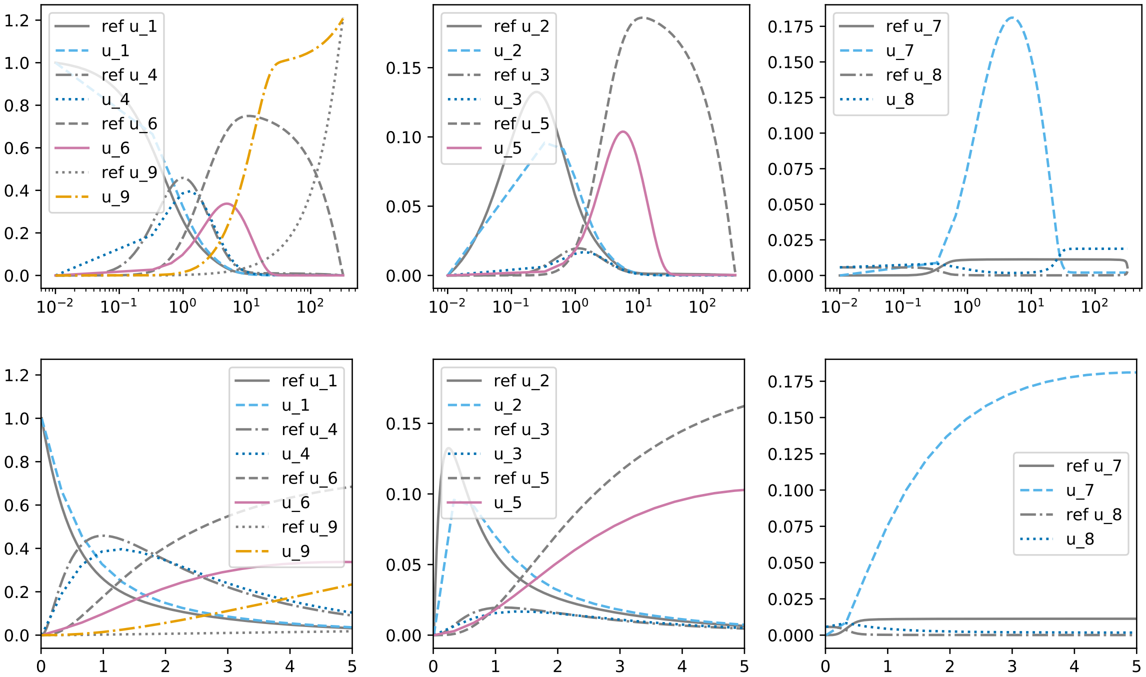


Figure 24: Simulations run with mPDeC1 with equispaced points with $N = 1000$ timesteps, top logarithmic scale in time, bottom zoom on $t \in [0, 5]$

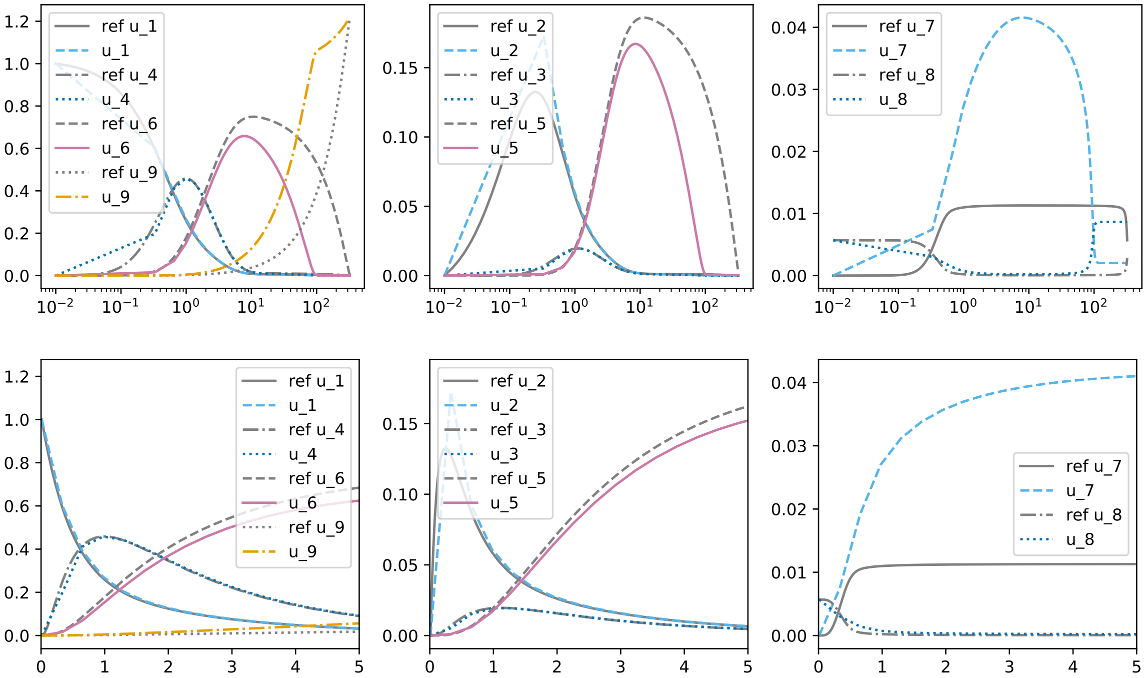


Figure 25: Simulations run with mPDeC2 with equispaced points with $N = 1000$ timesteps, top logarithmic scale in time, bottom zoom on $t \in [0, 5]$

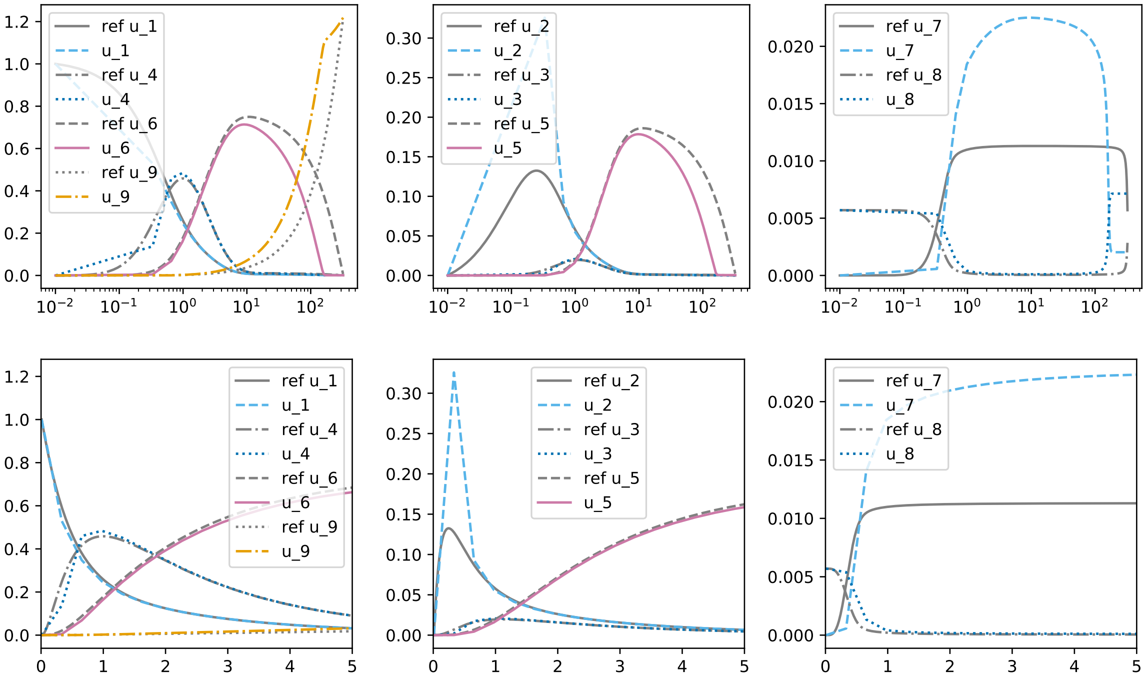


Figure 26: Simulations run with mPDeC3 with equispaced points with $N = 1000$ timesteps, top logarithmic scale in time, bottom zoom on $t \in [0, 5]$

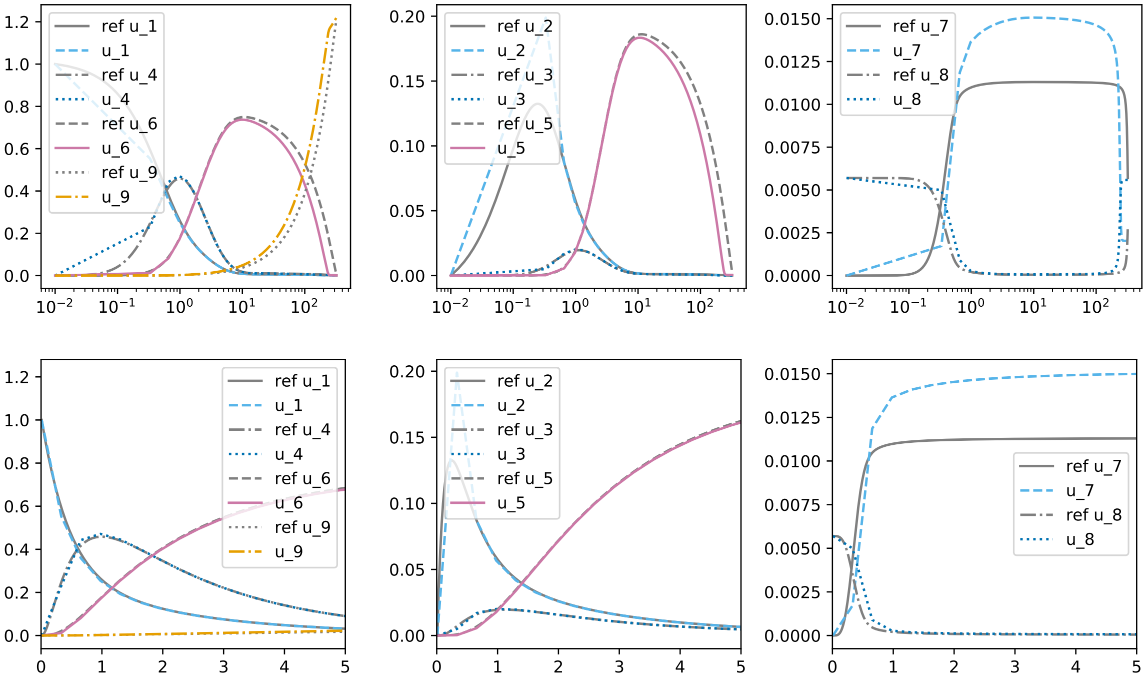


Figure 27: Simulations run with mPDeC4 with equispaced points with $N = 1000$ timesteps, top logarithmic scale in time, bottom zoom on $t \in [0, 5]$

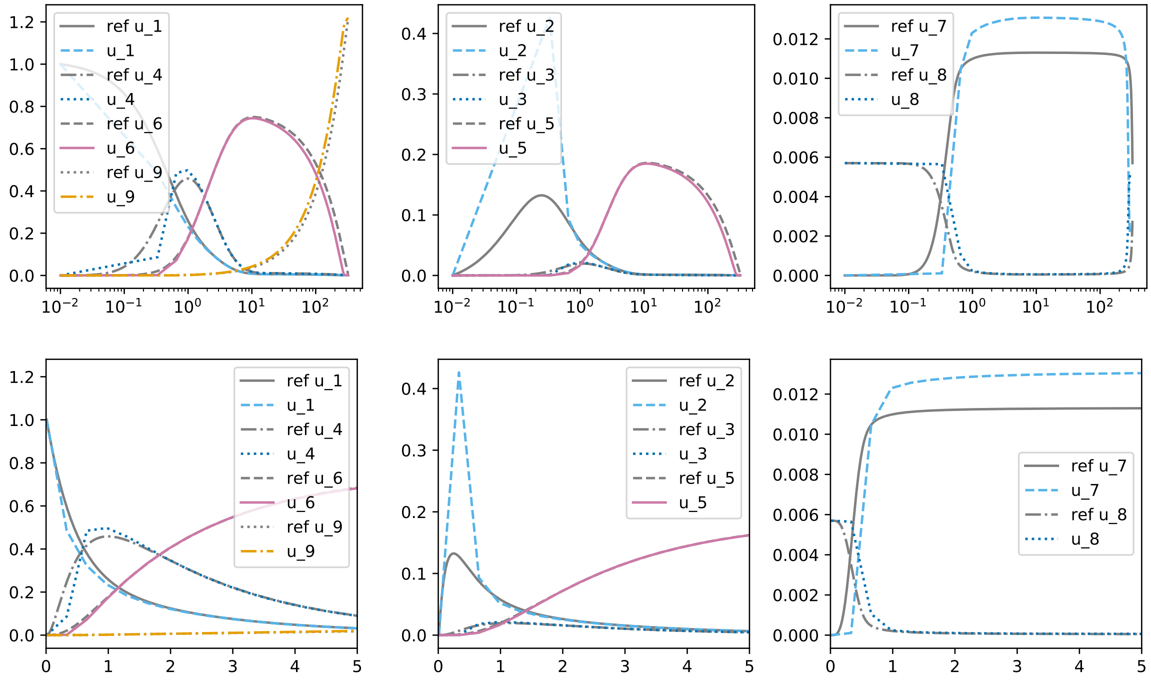


Figure 28: Simulations run with mPDeC5 with equispaced points with $N = 1000$ timesteps, top logarithmic scale in time, bottom zoom on $t \in [0, 5]$

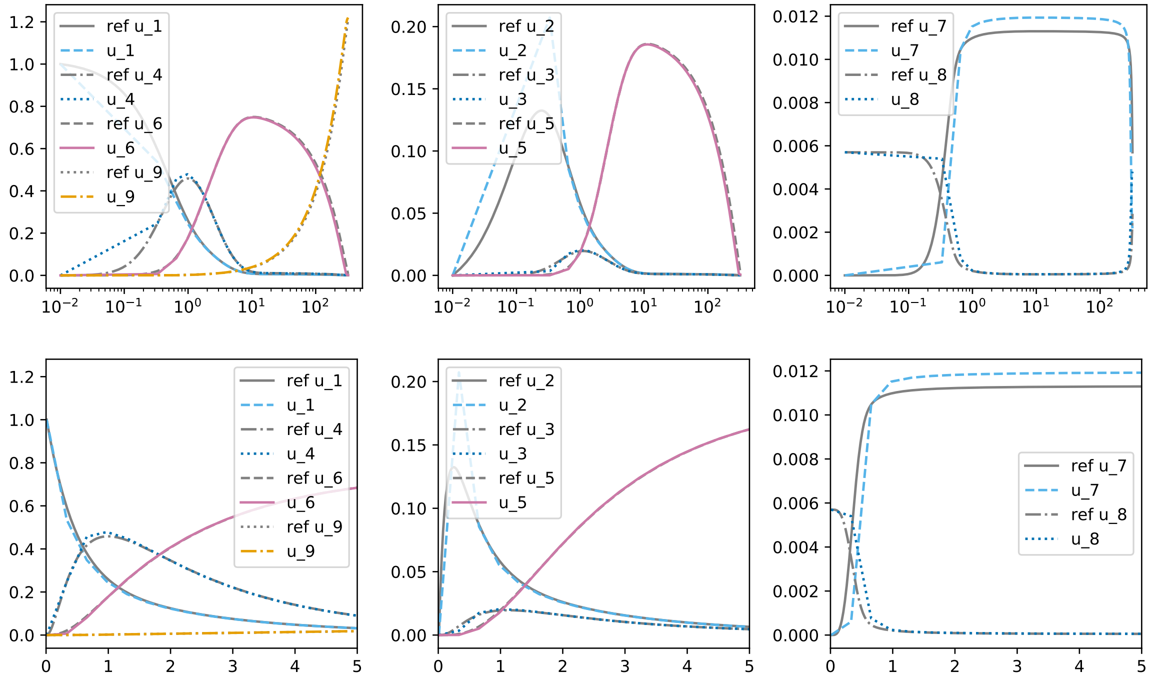


Figure 29: Simulations run with mPDeC6 with equispaced points with $N = 1000$ timesteps, top logarithmic scale in time, bottom zoom on $t \in [0, 5]$

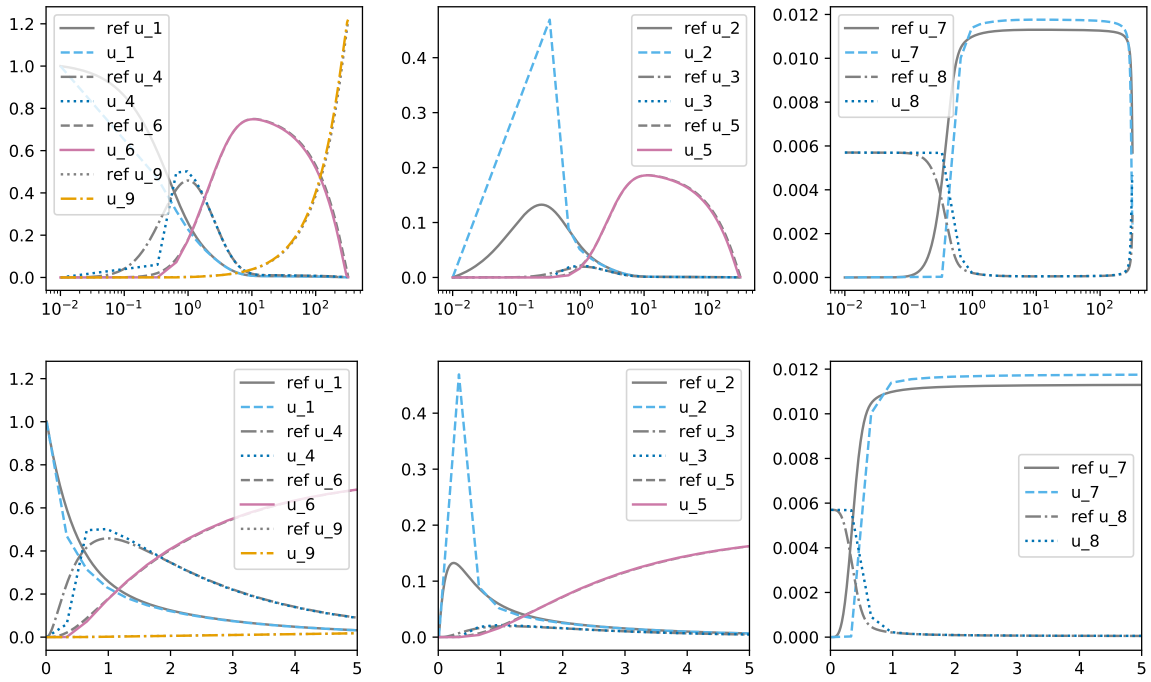


Figure 30: Simulations run with mPDeC7 with equispaced points with $N = 1000$ timesteps, top logarithmic scale in time, bottom zoom on $t \in [0, 5]$

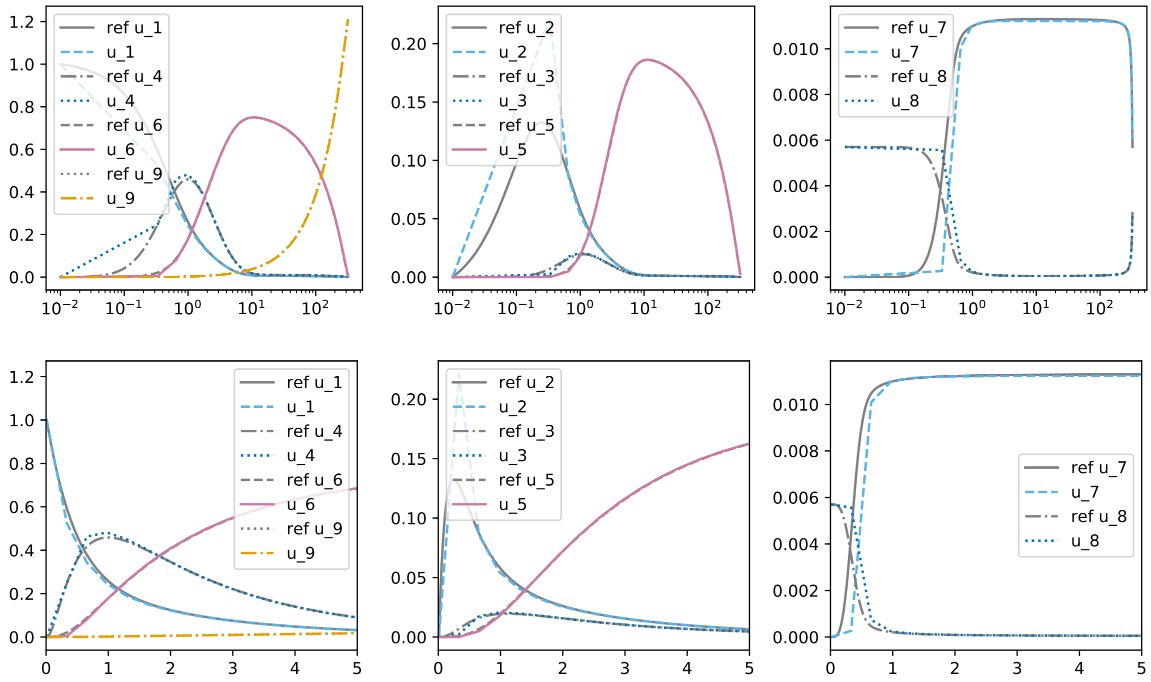


Figure 31: Simulations run with mPDeC8 with equispaced points with $N = 1000$ timesteps, top logarithmic scale in time, bottom zoom on $t \in [0, 5]$

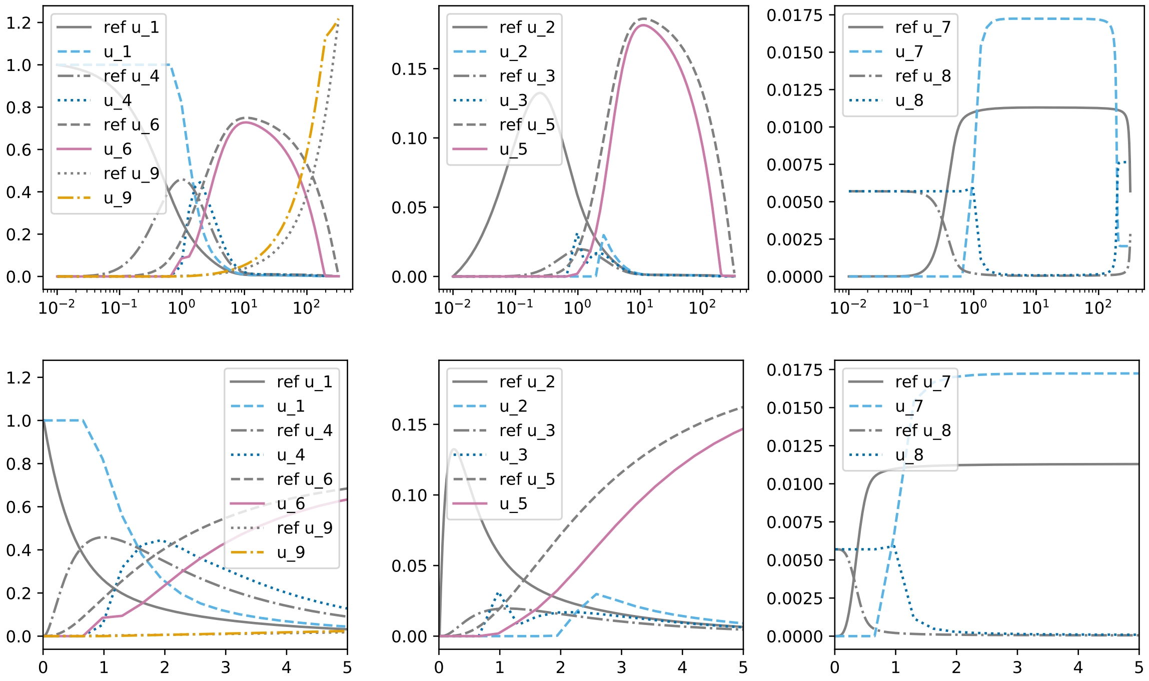


Figure 32: Simulations run with mPDeC9 with equispaced points with $N = 1000$ timesteps, top logarithmic scale in time, bottom zoom on $t \in [0, 5]$

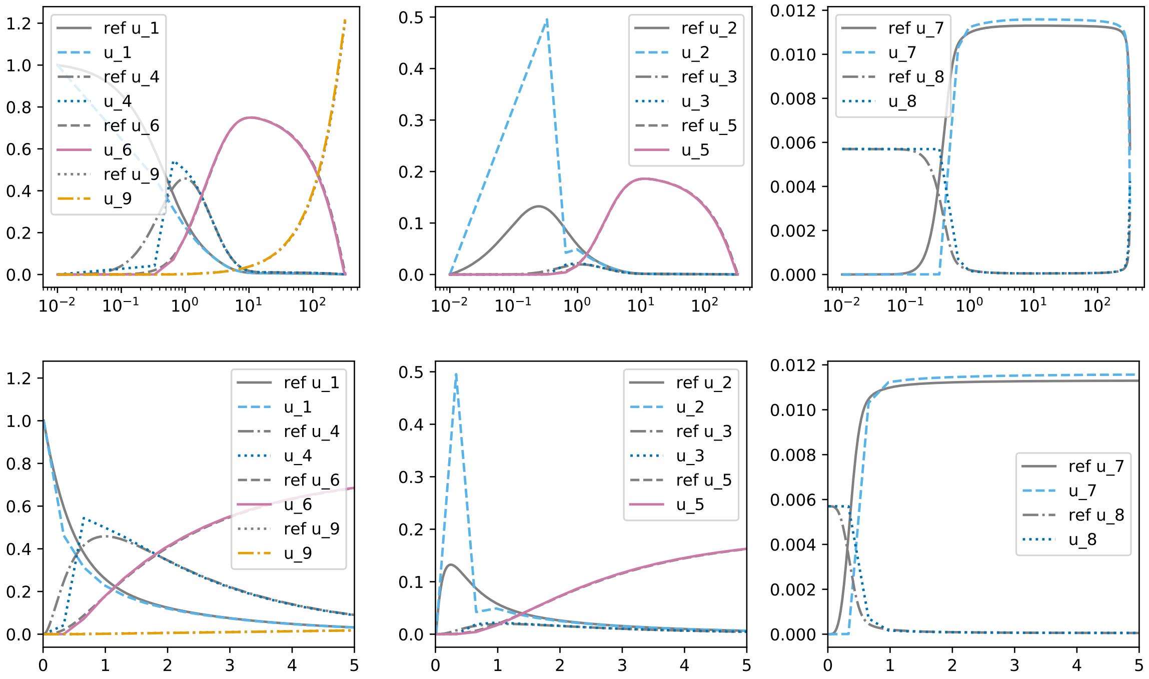


Figure 33: Simulations run with mPDeC10 with equispaced points with $N = 1000$ timesteps, top logarithmic scale in time, bottom zoom on $t \in [0, 5]$

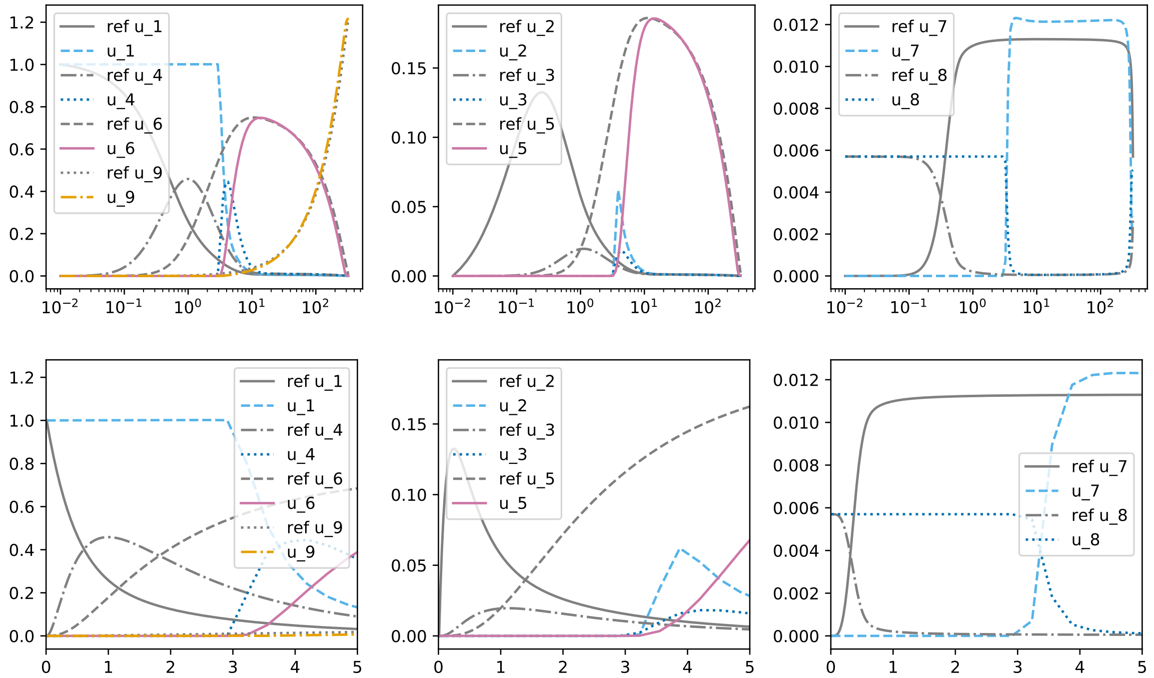


Figure 34: Simulations run with mPDeC11 with equispaced points with $N = 1000$ timesteps, top logarithmic scale in time, bottom zoom on $t \in [0, 5]$

We run the MPRK(2,2,α) with $\alpha \in [0.7, 1, 5]$. As for the linear case, only for $\alpha > 1$ we observe inconsistency as it is visible in Figure 3, where the evolution of some constituents is completely missed, e.g. u_2, u_3, u_5, u_9, while in Figure 2 we obtain consistent results.

We test MPRKSO(2,2,α,β) with $\alpha = 0.3$, $\beta = 2$ and $\alpha = 0$, $\beta = 8$ and, as expected, the second one shows the inconsistent spurious steady state. An oscillatory type behavior can be observed, though, in the first simulation, which is depicted in Figure 7. This is probably due to the CFL condition as, refining the time discretization, the oscillations disappear.

For MPRK(4,3,α, β) we test $\alpha = 0.9$, $\beta = 0.6$ and $\alpha = 5$, $\beta = 0.5$, observing inconsistency only in the second one, according with the linear tests. For MPRKSO(4,3), MPRK(3,2), SI-RK2 and SI-RK3 we do not observe inconsistencies, as in the linear test, nor other particular behaviors.

References

[1] H. Burchard, E. Deleersnijder, and A. Meister. “A high-order conservative Patankar-type discretisation for stiff systems of production–destruction equations.” In: *Applied Numerical Mathematics* 47.1 (2003), pp. 1–30. DOI: 10.1016/S0168-9274(03)00101-6.

[2] A. Chertock, S. Cui, A. Kurganov, and T. Wu. “Steady state and sign preserving semi-implicit Runge–Kutta methods for ODEs with stiff damping term.” In: *SIAM Journal on Numerical Analysis* 53.4 (2015), pp. 2008–2029. DOI: 10.1137/151005798.

[3] E. Hairer and G. Wanner. “Stiff differential equations solved by Radau methods.” In: *Journal of Computational and Applied Mathematics* 111.1-2 (1999), pp. 93–111. DOI: 10.1016/S0377-0427(99)00134-X.

[4] J. Huang and C.-W. Shu. “Positivity-Preserving Time Discretizations for Production–Destruction Equations with Applications to Non-equilibrium Flows.” In: *Journal of Scientific Computing* 78.3 (2019), pp. 1811–1839. DOI: 10.1007/s10915-018-0852-1.

[5] 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." In: *Journal of Scientific Computing* 79.2 (2019), pp. 1015–1056. DOI: 10.1007/s10915-018-0881-9.

[6] S. Kopecz and A. Meister. "Unconditionally positive and conservative third order modified Patankar–Runge–Kutta discretizations of production–destruction systems." In: *BIT Numerical Mathematics* 58.3 (2018), pp. 691–728. DOI: 10.1007/s10543-018-0705-1.

[7] F. Mazzia and C. Magherini. *Test Set for Initial Value Problem Solvers*. Technical Report Release 2.4. Italy: Department of Mathematics, University of Bari, Feb. 2008.

[8] P. Öffner and D. Torlo. "Arbitrary high-order, conservative and positivity preserving Patankar-type deferred correction schemes." In: *Applied Numerical Mathematics* 153 (2020), pp. 15–34.