Investigation of Exponential Time Differencing Schemes for Advection-Diffusion-Reaction Problems in the Presence of Significant Advection

Björn Gernot Müller

A Thesis Presented to the Graduate Faculty
in Partial Fulfillment of the Requirements for the Degree
Master of Science

University of Louisiana at Lafayette
Fall 2022

APPROVED:

Bruce A. Wade, Chair
Department of Mathematics

Xiang-Sheng Wang
Department of Mathematics

Kevin Zito
Department of Mathematics

Mary Farmer-Kaiser
Dean of the Graduate School

Abstract

Advection-diffusion-reaction equations are partial differential equations (PDEs) with various applications across the sciences. Exponential time differencing schemes are efficient methods of numerically solving PDEs of this type. We consider an exponential time differencing scheme called ETD-RDP-IF that approximates the arising matrix exponentials using a rational approximation with real distinct poles and employs a dimensional splitting technique to improve computational performance. The scheme has originally been derived for systems without advection. We show that the derivation still holds in the presence of advection and prove new results on the second-order temporal accuracy of the scheme. In numerical experiments, we investigate the real-world performance of the scheme depending on the strength of advection as quantified by the Péclet and Courant numbers. We confirm second-order convergence in space and time for linear problems with smooth initial condition and observe order reduction for non-smooth initial conditions. We further find that upwind-biased discretizations of advection improve computational efficiency. A comparison with an ETD scheme that uses Krylov-subspace approximations of the matrix exponentials shows that the Krylov-subspace technique has a better computational performance in low-advection regimes. Outside of these regimes, ETD-RDP-IF is more robust and therefore more widely applicable.

Acknowledgments

I am highly grateful to my advisor, Dr. Bruce Wade, whose ideas allowed me to explore and whose insight and guidance gave me the direction to finish this work. With his ever-lasting optimism and encouragement, he helped me overcome various kinds of hurdles along the way.

I want to thank my committee, Dr. Xiang-Sheng Wang and Dr. Kevin Zito, for taking the time to review multiple drafts of my thesis and their helpful suggestions and encouragement.

I also wish to express my deep gratitude to Dr. Andreas Kleefeld, especially for his constant faith in me, guidance, and support throughout this journey. Not only has he introduced me to the topic and taught me what I needed to know, he also encouraged and enabled my stay at the University of Louisiana at Lafayette.

I want to thank Dr. Gerhard Dikta for initiating the cooperation between Faculty 9 (Medical Engineering and Technomathematics) at FH Aachen University of Applied Sciences and the Department of Mathematics at the University of Louisiana at Lafayette and his great help in organizing my stay and studies at the University of Louisiana at Lafayette.

I am very grateful to Dr. Emmanuel Asante-Asamani, whose PhD dissertation this manuscript builds upon and who helped significantly shape the direction of this work and provided detailed ideas and invaluable suggestions.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

ADR	Advection-Diffusion-Reaction
CFL	Courant-Friedrichs-Lewy
CN	Crank-Nicolson
EETD	Extrapolated Exponential Time Differencing
ETD	Exponential Time Differencing
ETD-RDP-IF	Exponential Time Differencing Scheme with Real Distinct Poles Approximation and Integrating Factor Dimensional Splitting Technique
IBVP	Initial Boundary Value Problem
IF	Integrating Factor
IMEX	Implicit-Explicit
IVP	Initial Value Problem
ODE	Ordinary Differential Equation
PDE	Partial Differential Equation
RDP	Real Distinct Poles
RK	Runge-Kutta

1 Introduction

1.1 Applications of Advection-Diffusion-Reaction Equations

Advection-diffusion-reaction (ADR) equations originally describe how the concentration of chemical substances (species) in a flowing fluid medium evolve over time (Hundsdorfer and Verwer 2003; LeVeque 2007). This makes them applicable to the modeling of a wide range of phenomena. These include air pollution (Verwer et al. 2002; Lanser and Verwer 1999) and water pollution (e. g., van Herwaarden 1994; James 2002). Combustion processes are described by non-linear ADR systems (Berestycki 2002; Mickens 2005). The models we implement in this work, the Schnakenberg (Schnakenberg 1979; Madzvamuse 2006; Fernandes and Fairweather 2012; Bhatt et al. 2018) and Brusselator (Tyson 1976; Kang and Pesin 2005; Bhatt et al. 2018) models consider chemical reactions in an abstract sense, having originally been derived from stoichiometric equations and then generalized into ADR systems.

Beyond this, ADR systems have various applications in biological modeling, e. g., enzyme kinetics with Michaelis-Menten type equations (Johnson and Goody 2011; Chapwanya et al. 2013; Bhatt and Khaliq 2015; Asante-Asamani et al. 2016) and pattern formation in biological systems (Turing 1952; Tyson et al. 1999).

1.2 Problem Statement

In this work, we consider systems of time-dependent, semi-linear advection-diffusion-reaction (ADR) equations of the form

$$\frac{\partial u_i}{\partial t} = \mathbf{a}_i \cdot \nabla u_i + \nabla \cdot \mathbf{D}_i \nabla u_i + f_i(\mathbf{u}), \quad i = 1, \ldots, s, \tag{1}$$

where $\cdot$ denotes the (Euclidean) dot product of two vectors. In this context, $\mathbf{u} = \mathbf{u}(\mathbf{x}, t) = (u_1(\mathbf{x}, t), \ldots, u_s(\mathbf{x}, t))^\top$, $\mathbf{x} \in \Omega, t \in (0, T)$, can be understood as the concentration function of s species in the domain Ω. $\Omega = \times_{i=1}^{d}(\alpha_i, \alpha_i + \omega) \subset \mathbb{R}^d$ is a (hyper-)cube domain, where ω is the length of the sides. The reaction function $\mathbf{F}(\mathbf{u}) = (f_1(\mathbf{u}), \ldots, f_s(\mathbf{u}))^\top$ is a possibly non-linear function that we assume to be sufficiently smooth and bounded on $\overline{\Omega}$. $\mathbf{D}_i = \mathrm{diag}(d_{i1}, \ldots, d_{id}) \in \mathbb{R}^{d \times d}, i = 1, \ldots, s$ are diagonal matrices containing diffusion coefficients and are assumed to be constant in space and time. We require the entries of $\mathbf{D}_i, i = 1, \ldots, s$ to be non-negative in order to obtain a well-posed problem (cf. Hundsdorfer and Verwer 2003 p.13). This allows for a certain kind of anisotropic diffusion where the diffusion coefficients along the different coordinate axes differ. $\mathbf{a}_i = (a_{i1}, \ldots, a_{id})^\top \in \mathbb{R}^d$ is the advection vector for the i-th species.

We consider four different types of boundary conditions for such systems:

1. Homogeneous Dirichlet:

$$\mathbf{u}(\mathbf{x}, t) = 0 \quad \forall \mathbf{x} \in \partial\Omega$$

2. Vanishing normal derivative:

$$\frac{\partial}{\partial \nu}\mathbf{u}(x, t) = 0 \quad \forall \mathbf{x} \in \partial\Omega,$$

where ν denotes the exterior normal of Ω.

3. Homogeneous Neumann:

$$^\circ \cdot (\mathbf{D}_i \nabla u_i(\mathbf{x}, t) + \mathbf{a}_i u_i) = 0 \quad \forall \mathbf{x} \in \partial\Omega\,,$$

where ν denotes the exterior normal of Ω.

4. Periodic:

$$\mathbf{u}(\mathbf{x} + n\omega \mathbf{e}_i, t) = \mathbf{u}(\mathbf{x}, t) \;\; \forall \mathbf{x} \in \overline{\Omega},\; i = 1, \ldots, d,\; n \in \mathbb{Z}$$

Remark. The "vanishing normal derivative" boundary condition has been considered in the literature about advection-diffusion-reaction systems. It has sometimes been called (e. g., Bhatt et al. 2018) Neumann boundary condition. In the absence of advection, it is the same as what we refer to as "Neumann" boundary condition. However, in the presence of advection, the flux across the boundary is not equal to the normal derivative, so a vanishing normal derivative is not equivalent to zero flux across the boundary. To the best of our knowledge, Neumann conditions refer to the flux. Therefore, we chose the naming as described above.

An initial condition which is assumed to be compatible with the boundary condition, is given by $\mathbf{u}(\mathbf{x}, 0) = \mathbf{u}_0(\mathbf{x})$ for $\mathbf{x} \in \Omega$. Note that, numerically, we will consider initial conditions that are incompatible with the boundary conditions as well. For theoretical considerations, we limit ourselves to compatible initial and boundary conditions.

1.3 Significant Advection

The strength of advection in ADR systems can be quantified by the so-called Péclet number. Consider a one-dimensional problem with a single species

$$\frac{\partial u}{\partial t} = a u_x + d u_{xx} + f(u) \tag{2}$$

with suitable boundary conditions on a domain $\Omega = (\omega, \omega + L)$. The Péclet number is defined as

$$P = \frac{aL}{d} \, .$$

(see Gommes and Tharakan (2020) and references therein). L is the characteristic length of the problem. As explained in Gommes and Tharakan (2020), the Péclet number describes the relation between the orders of magnitude of the advection and diffusion term. The relevance of L is that it incorporates the scale dependence of this ratio.

Note that the definition given here can be extended to higher-dimensional problems and multiple species in various ways. In this work, we will only consider component-wise Péclet numbers.

Two regimes are usually distinguished. High Péclet numbers ($P \gg 1$) mean that advection is dominant relative to the diffusion whereas low Péclet numbers ($P \ll 1$) mean that diffusion is dominant.

In this work, we aim to consider problems with significant advection, i. e., where diffusion is at most of the same order of magnitude as advection ($P \gtrsim 1$).

1.4 ETD Methods

A common approach to solving time-dependent PDEs is the method of lines. This approach transforms the (system of) PDE(s) into a system of ordinary differential equations (ODEs) by first performing a discretization in space, i. e., in particular, of the spatial derivatives. The initial boundary value problem (IBVP) becomes an initial value problem (IVP).

The discretized differential operators make the ODE system stiff (Hundsdorfer and Verwer 2003). Explicit schemes suffer from stability issues when applied to stiff ODEs, requiring short time steps that make computation infeasible. To solve this issue, various implicit and semi-implicit schemes have been developed. Fully-implicit schemes like Runge-Kutta and BDF schemes are suitable for solving stiff ODEs. However, they require solving systems of non-linear equations in each time step when applied to non-linear ODEs, which reduces performance significantly. Various schemes have been suggested to resolve these performance penalties, e. g., Chen and Shen (1998) and Gear and Kevrekidis (2003).

In the semi-linear problem we are considering, the resulting ODEs are of the form

$$\mathbf{u}_t + \mathbf{M}\mathbf{u} = \mathbf{f}(\mathbf{u})$$

where $\mathbf{M}$ is a linear operator and $\mathbf{f}$ is a possibly non-linear function of $\mathbf{u}$.

For this case, various methods have been developed that use this structure to avoid having to solve non-linear systems of equations. Since the stiffness is contained inside the linear part of the ODE, many schemes solve the linear part

implicitly and treat the non-linear part explicitly (Hundsdorfer and Verwer 2003). Examples of these so-called linearly implicit or implicit-explicit (IMEX) schemes include Ascher et al. (1995), Akrivis et al. (1999), and Voss and Khaliq (1999).See Hundsdorfer and Verwer (2003) for a survey of IMEX methods.

Another class of schemes that rely on the semi-linear structure are called exponential time differencing schemes. These solve the linear part exactly in terms of exponentials, using Duhamel's principle. A seminal paper on ETD schemes is Cox and Matthews (2002) who used a Runge-Kutta (RK) type approach for constructing higher-order ETD schemes, called ETD-RK. Kassam and Trefethen (2005) and Hochbruck and Ostermann (2005) performed further analysis and introduced various improvements to the original method.

Cox and Matthews (2002) and Hochbruck and Ostermann (2005) did not consider the evaluation of the (matrix) exponentials that arise in ETD schemes, instead using standard implementations for numerical computations. Kassam and Trefethen (2005) suggested a contour integral technique to evaluate the exponentials. Since then, various ways of approximating the matrix exponentials have been proposed. Recently, Bhatt et al. (2018) used a Krylov-subspace approximation.

Khaliq et al. (2009) approximated the exponentials arising in ETD-RK schemes with low-order Padé approximations. This approach was also followed by Kleefeld et al. (2012) who used a Padé [1,1] approximation to obtain an ETD-Crank-Nicolson (ETD-CN) scheme. Yousuf et al. (2012) applied a Padé [0,2] approximation instead, making the scheme L-stable, and hence better at damping out spurious oscillations.

As Asante-Asamani (2016) notes, the Padé [0,2] approximation has complex poles, which makes a partial fraction decomposition computationally expensive. Instead, he proposes to approximate the exponential with a rational function that has real and distinct poles, following the approach of Voss and Khaliq (1996). Furthermore, Asante-Asamani (2016) introduced an operator splitting technique similar to the one used by Bhatt and Khaliq (2015). This scheme, named ETD-RDP-IF or short ETD-RDP, was presented again in Asante-Asamani et al. (2020) with more details on the potential for highly efficient implementation.

In this work, we further consider the ETD-RDP scheme introduced by Asante-Asamani (2016). It was originally applied only to reaction-diffusion equations without an advective term. We, therefore, show the derivation of the scheme in the presence of advection (Chapter 2). While doing so, we show some new results on the accuracy of the scheme in situations where the linear part of the differential equation involves a non-invertible matrix. We further evaluate the performance with varying amounts of advection present for exemplary equations from the literature and give a comparison of ETD-RDP to the Krylov-EETD scheme described in Bhatt et al. (2018) (Chapter 3). Finally, we give a conclusion in Chapter 4.

2 Methods

2.1 Mathematical Background

2.1.1 Finite Difference Spatial Discretization of Advection-Diffusion-Reaction Systems

ETD schemes use a method-of-lines approach (LeVeque 2007). This means that the IBVP (1) is discretized in space first, and the resulting ODE and initial value problem (IVP) are solved afterwards. For the spatial discretization, we will employ finite difference schemes. In this section, we describe their derivation for a system of the form (1).

First, the spatial domain Ω is discretized. Recall that we are assuming a hyper-cube domain, i. e., equal extent in all spatial dimensions. Therefore, choosing an equidistant grid size h in all d dimensions we obtain a grid with p^d points $\mathbf{x}_i, i = 1, \ldots, p^d$, where p is the number of grid points for each dimension. $\mathbf{u}$ is discretized on these grid points, resulting in a vector $\mathbf{U} = (\mathbf{U}_1^\top, \ldots, \mathbf{U}_{p^d}^\top)^\top \in \mathbb{R}^m$, $m = s \cdot p^d$ with $\mathbf{U}_i \approx \mathbf{u}(\mathbf{x}_i) \in \mathbb{R}^s$. $\mathbf{F}(\mathbf{u})$ is discretized by applying it componentwise to $\mathbf{U}$, i. e., we define the discretization $\mathbf{f}(\mathbf{U}) \in \mathbb{R}^m$ as $\mathbf{f}(\mathbf{U}) = (\mathbf{F}(\mathbf{U}_1)^\top, \ldots, \mathbf{F}(\mathbf{U}_{p^d})^\top)^\top$.

The diffusion and advection operators, and thus also their sum, are linear operators. Therefore, the discretized operator acting on $\mathbf{U}$ can be represented as a matrix $\mathbf{M} \in \mathbb{R}^{m \times m}$. The boundary conditions we will consider are linear as well, therefore, they will be discretized in $\mathbf{M}$ as well.

Therefore, we obtain an IVP by discretizing Equation (1) in this way:

$$\frac{\partial \mathbf{U}}{\partial t} + \mathbf{M}\mathbf{U} = \mathbf{f}(\mathbf{U}), \quad \mathbf{U}(0) = \mathbf{U}_0. \tag{3}$$

Here, $\mathbf{U}_0 = (\mathbf{u}_0(\mathbf{x}_1)^\top, \ldots, \mathbf{u}_0(\mathbf{x}_{p^d})^\top)^\top \in \mathbb{R}^m$ is the initial condition $\mathbf{u}_0(\mathbf{x})$ evaluated at the grid points $\mathbf{x}_1, \ldots, \mathbf{x}_{p^d}$.

2.1.1.1 Derivation. In order to derive the spatial discretization matrix $\mathbf{M}$, we first assume a one-dimensional problem for a single species, i. e., we consider the following problem:

$$\begin{aligned} \frac{\partial u(x,t)}{\partial t} &= \tilde{a}\nabla u(x,t) + \tilde{d}\Delta u(x,t) + f(u(x,t)) \\ &= \tilde{a}u_x(x,t) + \tilde{d}u_{xx}(x,t) + f(u(x,t))\,, \\ & \qquad\qquad x \in \Omega = (x_a, x_b), \end{aligned} \tag{4}$$

where $\tilde{a}$ is the advection velocity and $\tilde{d}$ is the diffusion constant.

In order to be consistent and - regarding implementation - compatible with the scheme proposed in Asante-Asamani et al. (2020), we wish to consider a discretization with p grid points. We, therefore, consider an equidistant discretization of (x_a, x_b) with p grid points and grid size h. Note that we are not specifying here whether the endpoints x_a and x_b are included in our grid or not. This will be discussed later. For now, the only requirement is equidistant spacing with the grid size h.

Hence, we need to find a discretization for u_x and u_{xx} at the grid points $(x_1, \ldots, x_p)$. Let, as above, $U_i \approx u(x_i), i = 1, \ldots, p$, be the approximation of $u(x_i)$. We can now apply second-order central finite differences to approximate the first and second spatial derivatives of u. At a given point $x_i, 2 \leq i \leq p-1$, i. e., in the interior of Ω, we get the following approximations:

$$u_{xx}(x_i) \approx \frac{u(x_{i+1}) - 2u(x_i) + u(x_{i-1})}{h^2} \approx \frac{U_{i+1} - 2U_i + U_{i-1}}{h^2}\,. \tag{5}$$

$$u_x(x_i) \approx \frac{u(x_{i+1}) - u(x_{i-1})}{2h} \approx \frac{U_{i+1} - U_{i-1}}{2h}\,. \tag{6}$$

We can express this in matrix form, analogously denoting the approximated derivative $U_{xx,i} \approx u_{xx}(x_i)$:

$$-\mathbf{U}_{xx} = -\begin{pmatrix} U_{xx,1} \\ U_{xx,2} \\ \vdots \\ U_{xx,p-1} \\ U_{xx,p} \end{pmatrix} \approx -\frac{1}{h^2}\begin{pmatrix} * & * & \dots & * & * \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ * & * & \dots & * & * \end{pmatrix}\begin{pmatrix} U_1 \\ U_2 \\ \vdots \\ U_{p-1} \\ U_p \end{pmatrix} =: \tilde{\mathbf{B}}\mathbf{U}$$

$$-\mathbf{U}_{x} = -\begin{pmatrix} U_{x,1} \\ U_{x,2} \\ \vdots \\ U_{x,p-1} \\ U_{x,p} \end{pmatrix} \approx -\frac{1}{2h}\begin{pmatrix} * & * & \dots & * & * \\ -1 & 0 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 0 & 1 \\ * & * & \dots & * & * \end{pmatrix}\begin{pmatrix} U_1 \\ U_2 \\ \vdots \\ U_{p-1} \\ U_p \end{pmatrix} =: \tilde{\mathbf{C}}\mathbf{U}$$

Note that we flipped the signs of the derivatives so that we can use this to discretize our reduced problem (4) to the form (3) with the discretization matrix

$$\tilde{\mathbf{M}} = \tilde{d}\tilde{\mathbf{B}} + \tilde{a}\tilde{\mathbf{C}} \tag{7}$$

For $2 \leq i \leq p-1$, this works without any restrictions as the differential operator in Equation (1) is the sum of the diffusion and advection operators. However, the first

and last rows of $\tilde{\mathrm{M}}$ are not yet well-defined, because the above approximation would yield

$$u_{xx}(x_1) \approx \frac{u(x_2) - 2u(x_1) + u(x_0)}{h^2} \approx \frac{U_2 - 2U_1 + U_0}{h^2}.$$

However U_0 is not considered in our discretization. Analogously, U_{p+1} would be required for approximating $u_{xx}(x_p)$.

2.1.1.2 Boundary Conditions. We can deal with this issue by treating the boundary points according to the respective boundary condition. Note that, above, we considered the discretization of advection (first-order derivative) and diffusion (second-order derivative) separately. Some boundary conditions introduce coupling between the advection and diffusion term, so we can only consider M as a whole.

We start by assuming the following discretization of Ω: $\mathbf{X} = (x_0, \ldots, x_n)^\top$ where $n = \frac{x_b - x_a}{h}$, $x_0 = x_a$ and $x_n = x_b$. Here, it is well determined where the n grid points lie.

Homogeneous Dirichlet. In the homogeneous Dirichlet case, we know $u(x_a) = u(x_b) = 0$. Thus, we set $U_0 = U_n = 0$ in the discretization. Therefore,

$$u_{xx}(x_1) \approx \frac{U_2 - 2U_1 + U_0}{h^2} = \frac{U_2 - 2U_1}{h^2}$$

$$u_x(x_1) \approx \frac{U_2 - U_0}{2h} = \frac{U_2}{2h}.$$

The same holds analogously for $u_x(x_{n-1})$. As a consequence, we need not consider U_0 and U_n further, leaving $n - 1$ points. We therefore consider the grid points $(x_1, \ldots, x_{n-1})$, excluding the boundary points. Therefore, the number of points, p, in our resulting discretization is $p = n - 1$, and we can express $h = \frac{x_b - x_a}{p+1}$. Clearly, we

applied this to the discretization of diffusion and advection independently, therefore we can continue to represent $\tilde{\mathbf{M}}$ as a sum of $\tilde{\mathbf{B}}$ and $\tilde{\mathbf{C}}$, namely $\tilde{\mathbf{M}} = \tilde{d}\tilde{\mathbf{B}} + \tilde{a}\tilde{\mathbf{C}}$ where

$$\tilde{\mathbf{B}} = -\frac{1}{h^2}\begin{pmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ & & & 1 & -2 \end{pmatrix} \in \mathbb{R}^{p\times p}$$

$$\tilde{\mathbf{C}} = -\frac{1}{2h}\begin{pmatrix} 0 & 1 & & & \\ -1 & 0 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 0 & 1 \\ & & & -1 & 0 \end{pmatrix} \in \mathbb{R}^{p\times p}.$$

Periodic. Periodic boundary conditions prescribe a periodic continuation of the solution outside of Ω. In particular, this means, for a given grid size h, $u(x_a) = u(x_b)$, $u(x_a + h) = u(x_b + h)$, and $u(x_a - h) = u(x_b - h)$. Thus, setting $U_0 = U_n$, $U_1 = U_{n+1}$ and $U_{-1} = U_{n-1}$, we need to only consider $U_0, \ldots, U_{n-1}$, and obtain on the boundaries:

$$u_{xx}(x_0) \approx \frac{U_1 - 2U_0 + U_{-1}}{h^2} = \frac{U_1 - 2U_0 + U_{n-1}}{h^2}$$

$$u_{xx}(x_{n-1}) \approx \frac{U_n - 2U_{n-1} + U_{n-2}}{h^2} = \frac{U_0 - 2U_{n-1} + U_{n-2}}{h^2}$$

$$u_x(x_1) \approx \frac{U_2 - U_0}{2h} = \frac{U_2 - U_n}{2h}$$

$$u_x(x_n) \approx \frac{U_{n+1} - U_{n-1}}{2h} = \frac{U_1 - U_{n-1}}{2h}$$

We therefore consider n grid points, including one boundary point. So, we set $p = n$, which yields $h = \frac{x_b - x_a}{p}$. This, too, can be applied independently to $\tilde{\mathbf{B}}$ and $\tilde{\mathbf{C}}$ and we get

$$\tilde{\mathbf{B}} = -\frac{1}{h^2}\begin{pmatrix} -2 & 1 & & & 1 \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ 1 & & & 1 & -2 \end{pmatrix} \in \mathbb{R}^{p\times p}$$

$$\tilde{\mathbf{C}} = -\frac{1}{2h}\begin{pmatrix} 0 & 1 & & & -1 \\ -1 & 0 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 0 & 1 \\ 1 & & & -1 & 0 \end{pmatrix} \in \mathbb{R}^{p\times p},$$

where $\tilde{\mathbf{M}} = \tilde{d}\tilde{\mathbf{B}} + \tilde{a}\tilde{\mathbf{C}}$.

Vanishing Normal Derivative. For a boundary with vanishing normal derivative, we use a ghost point approach to obtain a second-order discretization, Since we are assuming a one-dimensional domain, the exterior normal at $x = x_0$ is $\nu = -1$. Therefore, the boundary condition can be discretized to

$$-u_x(x_0) \approx -\frac{U_1 - U_{-1}}{2h} \stackrel{!}{=} 0\,. \tag{8}$$

U_{-1} is outside of our domain, hence it is called a ghost point. The discretization of the advection and diffusion terms in x_0 gives

$$\tilde{d}u_{xx}(x_0) + \tilde{a}u_x(x_0) \approx \tilde{d}\frac{-U_{-1} + 2U_0 - U_1}{h^2} + \tilde{a}\frac{-U_{-1} + U_1}{2h}\,. \tag{9}$$

Combining Equations (8) and (9) yields

$$\tilde{d}u_{xx}(x_0) + \tilde{a}u_x(x_0) \approx \tilde{d}\frac{2U_0 - 2U_1}{h^2}\,.$$

Clearly, the advection term vanishes on the boundary. Thus, we set the first and last rows of $\tilde{\mathbf{C}}$ to 0. The remaining diffusion term is expressed in terms of points within our domain only. For this construction, we need U_0 and U_n, i. e., $n+1$ grid points. We therefore set $p = n+1$ and $h = \frac{x_b - x_a}{p-1}$. This results in the matrices

$$\tilde{\mathbf{B}} = -\frac{1}{h^2}\begin{pmatrix} -2 & 2 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ & & & 2 & -2 \end{pmatrix} \in \mathbb{R}^{p\times p}$$

$$\tilde{\mathbf{C}} = -\frac{1}{2h}\begin{pmatrix} 0 & 0 & & & \\ -1 & 0 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 0 & 1 \\ & & & 0 & 0 \end{pmatrix} \in \mathbb{R}^{p\times p}\,,$$

and $\tilde{\mathbf{M}} = \tilde{d}\tilde{\mathbf{B}} + \tilde{a}\tilde{\mathbf{C}}$.

Homogeneous Neumann. Again employing a ghost point approach, we can derive the combined discretization matrix $\tilde{\mathbf{M}}$. The homogeneous Neumann boundary condition for a single species in one dimension can be discretized to

$$-\tilde{d}\frac{U_1 - U_{-1}}{2h} - \tilde{a}U_0 \stackrel{!}{=} 0\,.$$

Combining this with Equation (9) yields

$$\tilde{d}u_{xx}(x_0) + \tilde{a}u_x(x_0) \approx \left(\frac{2\tilde{d}}{h^2} - \frac{2\tilde{a}}{h} - \frac{\tilde{a}^2}{\tilde{d}}\right) U_0 - \frac{2\tilde{d}}{h^2} U_1 \,.$$

The discretization at x_p can be derived analogously. The result is the following discretization matrix:

$$\tilde{\mathbf{M}} = \begin{pmatrix} \frac{2\tilde{d}}{h^2} - \frac{2\tilde{a}}{h} - \frac{\tilde{a}^2}{\tilde{d}} & -\frac{2\tilde{d}}{h^2} & & & & \\ -\frac{\tilde{d}}{h^2} - \frac{\tilde{a}}{2h} & \frac{2\tilde{d}}{h^2} & -\frac{\tilde{d}}{h^2} + \frac{\tilde{a}}{2h} & & & \\ & \ddots & \ddots & \ddots & & \\ & & -\frac{\tilde{d}}{h^2} - \frac{\tilde{a}}{2h} & \frac{2\tilde{d}}{h^2} & -\frac{\tilde{d}}{h^2} + \frac{\tilde{a}}{2h} & \\ & & & -\frac{2\tilde{d}}{h^2} & \frac{2\tilde{d}}{h^2} + \frac{2\tilde{a}}{h} - \frac{\tilde{a}^2}{\tilde{d}} \end{pmatrix} \in \mathbb{R}^{p \times p}, \quad h = \frac{1}{p+1} \,.$$

For the purpose of legibility, we rewrite $\tilde{\mathbf{M}}$ as a sum of matrices:

$$\tilde{\mathbf{M}} = \frac{\tilde{d}}{h^2} \begin{pmatrix} 2 & -2 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -2 & 2 \end{pmatrix} + \frac{\tilde{a}}{2h} \begin{pmatrix} -4 & 0 & & & \\ -1 & 0 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 0 & 1 \\ & & & 0 & 4 \end{pmatrix}$$
$$+ \frac{\tilde{a}^2}{\tilde{d}} \begin{pmatrix} -1 & & & & \\ & 0 & & & \\ & & \ddots & & \\ & & & 0 & \\ & & & & -1 \end{pmatrix}$$

2.1.1.3 Upwind-Biased Schemes. Besides the second-order central finite difference scheme, we also applied so-called upwind-biased finite difference schemes. These are used only to discretize the advection term.

Consider the one-dimensional PDE in Equation (4)

$$\frac{\partial u(x,t)}{\partial t} = \tilde{a}u_x(x,t) + \tilde{d}u_{xx}(x,t) + f(u(x,t))\,,$$
$$x \in \Omega = (x_a, x_b),$$

The advection term $\tilde{a}u_x(x,t)$ causes the solution to be transported through the domain Ω. If $\tilde{a} < 0$, the solution moves to the right, i. e., in positive x direction. The upwind direction refers to where the "wind" (or, more generally, the current) that transports the solution is coming from. Here, this would be to the left, i. e., negative x direction. If $\tilde{a} > 0$, the solution moves to the left, and upwind is to the right.

In these cases, as described by Versteeg and Malalasekera (2007), the concentration of a species at a certain point is more strongly influenced by the concentrations that are upwind (upstream) of that point. Intuitively, this can be understood by considering that "material" is transported from upwind to downwind, and whatever is downwind of a point moves away from that point. In a pure advection problem, the concentrations downwind of a point have no influence on that point at all.

In order to reflect that in a finite difference scheme, it should not be symmetrical like the central difference scheme. Instead, the stencil is extended in the upwind direction, and higher weights are given to values upwind. This reduces the influence that downwind points have in the numerical scheme, thereby modeling the physical reality.

We considered two different upwind-type schemes, specifically, upwind-biased schemes. These are the second-order accurate Fromm scheme and a third-order upwind-biased scheme. Both are described in Hundsdorfer and Verwer (2003).

Fromm Scheme. The Fromm scheme was originally described by Fromm (1968). If $\tilde{a} < 0$, it discretizes u_x in the following way:

$$u_x(x_i) \approx \frac{1}{4}U_{i-2} - \frac{5}{4}U_{i-1} + \frac{3}{4}U_i + \frac{1}{4}U_{i+1}$$

If $\tilde{a} > 0$, the scheme needs to be reflected around x_i:

$$u_x(x_i) \approx -\frac{1}{4}U_{i-1} - \frac{3}{4}U_i + \frac{5}{4}U_{i+1} - \frac{1}{4}U_{i+2}$$

In both cases, the coefficients are greater in magnitude in the upwind direction. Also, the stencils extend further in the upwind direction than in the downwind direction. This is important to note since the Fromm scheme uses a larger stencil than the second-order central differencing scheme. Expressing this scheme in matrix form, therefore, produces a matrix with more bands. As we will see, this reduces performance in solving systems with the corresponding matrix.

Third-Order Upwind-Biased Scheme. However, on the same stencil, a third-order upwind-biased scheme can be constructed. Therefore, we consider this as well. It approximates u_x as

$$u_x(x_i) \approx \frac{1}{6}U_{i-2} - U_{i-1} + \frac{1}{2}U_i + \frac{1}{3}U_{i+1}$$

if $\tilde{a} < 0$ and

$$u_x(x_i) \approx -\frac{1}{3}U_{i-1} - \frac{1}{2}U_i + U_{i+1} - \frac{1}{6}U_{i+2}$$

if $\tilde{a} > 0$.

Matrix Form. Both of these schemes can be expressed in matrix form analogous to Section 2.1.1.1. Boundary conditions are applied in similar ways as in Section 2.1.1.2. Due to the larger stencils, further complications arise for some boundary conditions. Therefore, we consider the upwind-biased schemes only for periodic boundary conditions where a derivation analogous to that in Section 2.1.1.2 is immediately possible.

2.1.1.4 Generalization to Arbitrary Dimensions and Species. To compute the discretization matrix $\mathbf{M}$ for the case with d dimensions and s species, Asante-Asamani et al. (2020) employ a Kronecker-product formalism that we shall use here as well. In order to derive this, we construct the formalism in steps.

One Dimension, Two Species. First consider a one-dimensional problem with two species.

$$\frac{\partial u_i(x,t)}{\partial t} = \tilde{a}_i \frac{\partial}{\partial x} u_i(x,t) + \tilde{d}_i \frac{\partial^2}{\partial x^2} u_i(x,t) + f_i(u(x,t)), \ i = 1,2$$
$$x \in \Omega = (x_a, x_b)$$

Discretizing this at the grid points $x_1, \ldots, x_p$, we obtain $\mathbf{U} = (\mathbf{U}_1^\top, \ldots, \mathbf{U}_p^\top)$ where

$$\mathbf{U}_j = (U_{j1}, U_{j2})^\top \qquad \forall j = 1, \ldots, p.$$

Let $\mathbf{U}^{(i)} = (U_{1i}, \ldots, U_{pi}), i = 1, 2$. Note that our problem contains two equations that are only coupled by the nonlinear reaction term f. We discretize both equations separately to obtain matrices $\mathbf{M}^{(1)}$ and $\mathbf{M}^{(2)}$, and the following systems of ODEs:

$$\frac{\partial \mathbf{U}^{(i)}}{\partial t} + \mathbf{M}^{(i)} \mathbf{U}^{(i)} = \mathbf{f}^{(i)}(\mathbf{U}), \quad \mathbf{U}^{(i)}(0) = \mathbf{U}_0^{(i)}, i = 1,2$$

In order to combine this into a single system of ODEs, we observe that for any vector $\mathbf{V} = (V_{11}, V_{12}, \ldots, V_{p1}, V_{p2})$, i.e., with the same structure as $\mathbf{U}$,

$$\mathbf{M}^{(i)}\mathbf{U}^{(i)} = \mathbf{V}^{(i)},\ i = 1, 2$$

if and only if

$$\left(\mathbf{M}^{(1)} \otimes \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} + \mathbf{M}^{(2)} \otimes \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \right) \mathbf{U} = \mathbf{V}$$

where $\otimes$ denotes the Kronecker product (see Van Loan (2000) for details).

This gives us the full system of ODEs

$$\frac{\partial \mathbf{U}}{\partial t} + \mathbf{M}\mathbf{U} = \mathbf{f}(\mathbf{U}), \quad \mathbf{U}(0) = \mathbf{U}_0$$

with the matrix

$$\mathbf{M} = \mathbf{M}^{(1)} \otimes \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} + \mathbf{M}^{(2)} \otimes \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}$$

One Dimension, Multiple Species. This can be generalized for arbitrary s as follows:

$$\mathbf{M} = \sum_{i=1}^{s} \mathbf{M}^{(i)} \otimes \mathbf{E}^{(i)} \tag{10}$$

where

$$\mathbf{E}^{(i)} = \operatorname{diag}(\underbrace{0, \ldots, 0}_{i-1 \text{ times}}, 1, \underbrace{0, \ldots, 0}_{s-i \text{ times}}) = \begin{pmatrix} 0 & & & & \\ & \ddots & & & \\ & & 1 & & \\ & & & \ddots & \\ & & & & 0 \end{pmatrix} \in \mathbb{R}^{s \times s}$$

where the 1 is the i-th diagonal entry of $\mathbf{E}^{(i)}$. In this case,

$$\mathbf{U} = (\mathbf{U}_1^\top, \ldots, \mathbf{U}_p^\top)$$

where

$$\mathbf{U}_j = (U_{j1}, \ldots, U_{js})^\top \qquad \forall j = 1, \ldots, p.$$

Two Dimensions, One Species. In order to discretize problems in an arbitrary number of spatial dimensions, we first consider a problem in two dimensions with a single species:

$$\begin{aligned} \frac{\partial u(x,y,t)}{\partial t} = \tilde{\mathbf{a}} \cdot \nabla u(x,y,t) + \nabla \cdot \tilde{\mathbf{D}} \nabla u(x,y,t) + f(u(x,y,t)) \\ (x,y) \in \Omega = (x_a, x_b) \times (y_a, y_b), \end{aligned} \tag{11}$$

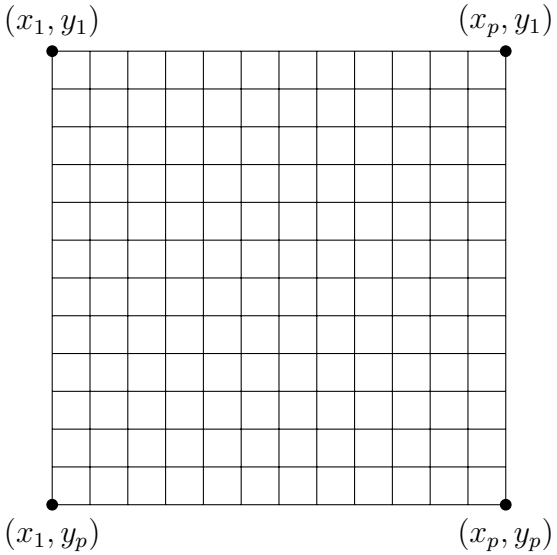
where $\tilde{\mathbf{a}} = (a_1, a_2)$ and

$$\tilde{\mathbf{D}} = \begin{pmatrix} d_1 & 0 \\ 0 & d_2 \end{pmatrix}.$$

We can rewrite (11) to

$$\frac{\partial u(x,y,t)}{\partial t} = a_1 u_x(x,y,t) + a_2 u_y(x,y,t) + d_1 u_{xx}(x,y,t) + d_2 u_{yy}(x,y,t) + f(u(x,y,t))$$

In order to discretize Ω, we start by first discretizing (x_a, x_b) to $x_1, \ldots, x_p$ and (y_a, y_b) to $y_1, \ldots, y_p$. Our discretization grid is then $(x_i, y_j), i, j = 1, \ldots, p$:



We approximate $u(x_i, y_j)$ by U_{ij} in our discretization of u. By ordering the points U_{ij} in row-major order, i.e, we obtain the vector

$\mathbf{U} = (U_{11}, \ldots, U_{p1}, \ldots\ldots, U_{1p}, \ldots, U_{pp})$.

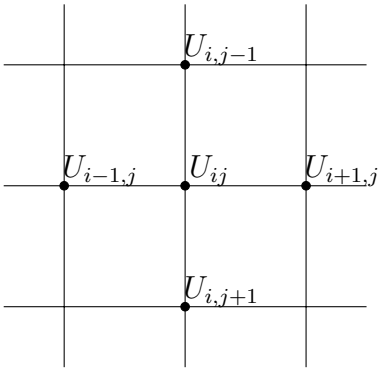


Figure 1. 5-point stencil about U_{ij}

Recall the discretization of u_{xx} and u_x in Equations (5) and (6), respectively. In the 2-dimensional case, we obtain similar discretizations

$$u_{xx}(x_i, y_j) \approx \frac{U_{i+1,j} - 2U_{ij} + U_{i-1,j}}{h^2}$$
$$u_x(x_i, y_j) \approx \frac{U_{i+1,j} - U_{i-1,j}}{2h}$$

and analogously

$$u_{yy}(x_i, y_j) \approx \frac{U_{i,j+1} - 2U_{ij} + U_{i,j-1}}{h^2}$$
$$u_y(x_i, y_j) \approx \frac{U_{i,j+1} - U_{i,j-1}}{2h}\,.$$

in the interior of Ω. Clearly, we can perform this discretization separately on each row (and column, respectively). This is effectively a reduction to the 1-dimensional case. Also considering the boundary conditions, we obtain a discretization matrix $\tilde{\mathbf{M}}_1$ per row, i.e.,

$$-d_1(\mathbf{U}_{i\cdot})_{xx} - a_1(\mathbf{U}_{i\cdot})_x \approx \tilde{\mathbf{M}}_1 \mathbf{U}_{i\cdot}, i = 1, \ldots, p$$

where $\tilde{\mathbf{M}}_1$ is constructed according to the boundary condition as in Section 2.1.1.

Combining these discretizations across all rows, we obtain

$$-d_1\mathbf{U}_{xx} - a_1\mathbf{U}_x = -d_1\begin{pmatrix} (\mathbf{U}_{1\cdot})_{xx} \\ \vdots \\ (\mathbf{U}_{p\cdot})_{xx} \end{pmatrix} - a_1\begin{pmatrix} (\mathbf{U}_{1\cdot})_{x} \\ \vdots \\ (\mathbf{U}_{p\cdot})_{x} \end{pmatrix} = \begin{pmatrix} \tilde{\mathbf{M}}_1 & & \\ & \ddots & \\ & & \tilde{\mathbf{M}}_1 \end{pmatrix}\begin{pmatrix} \mathbf{U}_{1\cdot} \\ \vdots \\ \mathbf{U}_{p\cdot} \end{pmatrix}$$

Note that we can express the block matrix on the right hand side as a Kronecker product

$$\begin{pmatrix} \tilde{\mathbf{M}}_1 & & \\ & \ddots & \\ & & \tilde{\mathbf{M}}_1 \end{pmatrix} = \mathbf{I}_p \otimes \tilde{\mathbf{M}}_1 \in \mathbb{R}^{p^2 \times p^2}$$

where $\mathbf{I}_p \in \mathbb{R}^{p\times p}$ is an identity matrix.

Similarly, considering the columns, it is true that

$$-d_2(\mathbf{U}_{\cdot j})_{yy} - a_2(\mathbf{U}_{\cdot j})_y \approx \tilde{\mathbf{M}}_2\mathbf{U}_{\cdot j}, \quad j = 1, \ldots, p$$

where $\tilde{\mathbf{M}}_2$ is constructed according to the boundary condition as in Section 2.1.1.

Due to the ordering of points in our vector $\mathbf{U}$, however, U_{ij} and U_{ij+1} are exactly p places apart. Hence, to obtain $-d_2\mathbf{U}_{yy} - a_2\mathbf{U}_y$, we need to correctly space the entries of $\tilde{\mathbf{M}}_2$. We can again achieve this using a Kronecker product:

$$
\begin{aligned}
-d_2\mathbf{U}_{yy} - a_2\mathbf{U}_y = &\begin{pmatrix}
m_{11} & \dots & m_{12} & & & & \\
\vdots & *\,*\,* & \vdots & *\,*\,* & & & \\
m_{21} & \dots & m_{22} & \dots & m_{23} & & \\
 & *\,*\,* & \vdots & *\,*\,* & \vdots & *\,*\,* & \\
 & & m_{p-1,p-2} & \dots & m_{p-1,p-1} & \dots & m_{p-1,p} \\
 & & & *\,*\,* & \vdots & *\,*\,* & \vdots \\
 & & & & m_{p,p-1} & \dots & m_{pp}
\end{pmatrix} \\
= &(\tilde{\mathbf{M}}_2 \otimes \mathbf{I}_p)\mathbf{U}.
\end{aligned}
$$

Here, $\vdots$ and $\cdots$ each represent a spacing of $p-1$ entries of 0. $*\,*\,*$ represent non-zero entries. Note that $(\tilde{\mathbf{M}}_2 \otimes \mathbf{I}_p) \in \mathbb{R}^{p^2 \times p^2}$ as well.

So far, we have discretized the derivatives along each spatial axis separately. Note that in equation (12) we add all the derivatives. Hence, we obtain the full discretization matrix

$$\mathbf{M} = \mathbf{I}_p \otimes \tilde{\mathbf{M}}_1 + \tilde{\mathbf{M}}_2 \otimes \mathbf{I}_p.$$

Multiple Dimensions, One Species. This process can be generalized for higher dimensions as well, yielding for three dimensions

$$\mathbf{M} = \mathbf{I}_p \otimes \mathbf{I}_p \otimes \tilde{\mathbf{M}}_1 + \mathbf{I}_p \otimes \tilde{\mathbf{M}}_2 \otimes \mathbf{I}_p + \tilde{\mathbf{M}}_3 \otimes \mathbf{I}_p \otimes \mathbf{I}_p.$$

For d dimensions, we obtain

$$\mathbf{M}_k = (\bigotimes_{j=1}^{d-k} \mathbf{I}_p) \otimes \tilde{\mathbf{M}}_k \otimes (\bigotimes_{j=1}^{k-1} \mathbf{I}_p) \quad \in \mathbb{R}^{p^d \times p^d} \tag{12}$$

and

$$\mathbf{M} = \sum_{k=1}^{d} \mathbf{M}_k. \tag{13}$$

Multiple Dimensions, Multiple Species. Finally, we can consider a problem with s species in d dimensions. Above, we discretized the problem separately for each species and combined the resulting matrices. I. e., we solve s problems in d dimensions and obtain sd matrices $\mathbf{M}_k^{(i)}, k = 1, \ldots, d, i = 1, \ldots, s$ as in Equation (12). Applying Equation (13), we obtain s matrices $\mathbf{M}^{(i)} = \sum_{k=1}^{d} \mathbf{M}_k^{(i)}$. Now we can combine these using Equation (10):

$$\begin{aligned}\mathbf{M} &= \sum_{i=1}^{s} \mathbf{M}^{(i)} \otimes \mathbf{E}^{(i)} \\ &= \sum_{i=1}^{s} \left(\sum_{k=1}^{d} \mathbf{M}_k^{(i)} \right) \otimes \mathbf{E}^{(i)}\end{aligned}$$

Due to the linearity of the Kronecker product, we can rewrite this as

$$\mathbf{M} = \sum_{k=1}^{d} \left(\sum_{i=1}^{s} \mathbf{M}_k^{(i)} \otimes \mathbf{E}^{(i)} \right) = \sum_{k=1}^{d} \mathbf{M}_k$$

where we define $\mathbf{M}_k = \sum_{i=1}^{s} \mathbf{M}_k^{(i)} \otimes \mathbf{E}^{(i)}$. We will use this final representation in the derivation of the scheme below.

2.1.2 Properties of the Matrix Exponential

We define the matrix exponential for an arbitrary square matrix $\mathbf{A} \in \mathbb{C}^{n \times n}$ as

$$\mathrm{e}^{\mathbf{A}} = \sum_{i=0}^{\infty} \frac{\mathbf{A}^i}{i!},$$

in analogy to the scalar exponential function.

The matrix exponential is used at the core of the derivation of the ETD-RDP-IF scheme we are considering in this work. We will be using some properties of the matrix exponential that we shall give here as lemmata. Suppose that $\mathbf{A}, \mathbf{B} \in \mathbb{C}^{n \times n}$.

Lemma 1. If $\mathbf{AB} = \mathbf{BA}$, i. e., $\mathbf{A}$ and $\mathbf{B}$ commute, then

$$\mathbf{A}\mathrm{e}^{\mathbf{B}} = \mathrm{e}^{\mathbf{B}}\mathbf{A},$$

i. e., $\mathbf{A}$ and $\mathrm{e}^{\mathbf{B}}$ commute.

Proof.

$$\mathbf{A}\mathrm{e}^{\mathbf{B}} = \mathbf{A}\sum_{i=0}^{\infty}\frac{\mathbf{B}^i}{i!} = \sum_{i=0}^{\infty}\mathbf{A}\frac{\mathbf{B}^i}{i!} = \sum_{i=0}^{\infty}\frac{\mathbf{B}^i}{i!}\mathbf{A} = \left(\sum_{i=0}^{\infty}\frac{\mathbf{B}^i}{i!}\right)\mathbf{A} = \mathrm{e}^{\mathbf{B}}\mathbf{A}$$

The proof is analogous to Higham (2008, Thm. 10.2). □

Lemma 2. , If $\mathbf{AB} = \mathbf{BA}$, i. e., $\mathbf{A}$ and $\mathbf{B}$ commute, then

$$\mathrm{e}^{\mathbf{A}}\mathrm{e}^{\mathbf{B}} = \mathrm{e}^{\mathbf{A}+\mathbf{B}} = \mathrm{e}^{\mathbf{B}}\mathrm{e}^{\mathbf{A}}.$$

Proof. See Higham (2008, chap. 10). □

Lemma 3. The exponential of a matrix is invertible and

$$(\mathrm{e}^{\mathbf{B}})^{-1} = \mathrm{e}^{-\mathbf{B}}$$

Proof. See Higham (2008, chap. 10). □

Lemma 4. The matrix exponential function $f(t) = \mathrm{e}^{\mathbf{A}t}$ is differentiable and

$$\frac{d}{dt}\mathrm{e}^{\mathbf{A}t} = \mathbf{A}\mathrm{e}^{\mathbf{A}t} = \mathrm{e}^{\mathbf{A}t}\mathbf{A}$$

Proof. See Higham (2008, chap. 10). □

Lemma 5. Suppose $\mathbf{A}$ has the Jordan decomposition (Jordan normal form) $\mathbf{A} = \mathbf{P}\mathbf{J}\mathbf{P}^{-1}$. Then,

$$\mathrm{e}^{\mathbf{A}} = \mathbf{P}\mathrm{e}^{\mathbf{J}}\mathbf{P}^{-1}.$$

Proof. See Horn and Johnson (1991, chap. 6). □

2.2 ETD-RDP-IF

Asante-Asamani (2016) and Asante-Asamani et al. (2020) developed an exponential time differencing (ETD) method using a rational (non-Padé) approximation of the matrix exponential with real and simple distinct poles (RDP) and an integrating factor (IF) dimensional splitting technique, called ETD-RDP-IF. Their ETD-RDP-IF scheme was developed for diffusion-reaction systems. After a description of the scheme, they suggested a possible way of implementing it, using parallelization on three cores. The primary goal of this work is to extend their scheme for the solution of advection-diffusion-reaction systems in order to be able to use the same implementation techniques. In the following, we recount their derivation of the scheme and adapt it to the more general context including advection that we are considering here. Unless otherwise noted, this section is based on Asante-Asamani et al. (2020).

2.2.1 Dimensional Splitting

After a finite difference discretization of the problem (1) with s species in d dimensions, we obtain an ODE as in (3), i. e.,

$$\frac{\partial \mathbf{U}}{\partial t} + \mathbf{M}\mathbf{U} = \mathbf{f}(\mathbf{U}), \quad \mathbf{U}(0) = \mathbf{U}_0 \in \mathbb{R}^m$$

where $m = s \cdot p^d$, and $\mathbf{M} = \sum_{k=1}^{d} \mathbf{M}_k$.

Each $\mathbf{M}_k$ represents the discretization considering one spatial dimension. The initial value problem (3) can be simplified by considering these separately. This is called dimensional splitting. Asante-Asamani et al. (2020) use an integrating factor technique to achieve dimensional splitting and simplify the problem. In this section, we describe their approach and prove that it is still applicable in the more general context we are considering here in order to accommodate for advection.

For the purpose of demonstration, we will assume a 3-dimensional problem, i.e., $\mathbf{M}$ can be represented as $\mathbf{M} = \mathbf{M}_1 + \mathbf{M}_2 + \mathbf{M}_3$.

As suggested before, an integrating factor is used. In this case, it is $\mathrm{e}^{\mathbf{M}_1 t}$. $\mathbf{U}$ is multiplied with this function to obtain a new time-dependent vector-valued function $\mathbf{V} = \mathrm{e}^{\mathbf{M}_1 t}\mathbf{U}$. Note here that $\mathbf{U}$ is a function of t as well. We will later see how this reduces the problem and thus simplifies it.

Taking the derivative of $\mathbf{V}$ with respect to time, we get

$$\mathbf{V}_t = \mathrm{e}^{\mathbf{M}_1 t}\mathbf{U}_t + \mathbf{M}_1 \mathrm{e}^{\mathbf{M}_1 t}\mathbf{U}$$

Using Equation (3), we can rewrite this to

$$\begin{aligned}\mathbf{V}_t &= \mathrm{e}^{\mathbf{M}_1 t}(\mathbf{f}(\mathbf{U}) - \mathbf{M}\mathbf{U}) + \mathbf{M}_1 \mathrm{e}^{\mathbf{M}_1 t}\mathbf{U} \\ &= \mathrm{e}^{\mathbf{M}_1 t}\mathbf{f}(\mathbf{U}) - \mathrm{e}^{\mathbf{M}_1 t}\mathbf{M}\mathbf{U} + \mathbf{M}_1 \mathrm{e}^{\mathbf{M}_1 t}\mathbf{U}\end{aligned}$$

By Lemma 1,

$$\mathrm{e}^{\mathbf{M}_1 t}\mathbf{M}\mathbf{U} = \mathbf{M}\mathrm{e}^{\mathbf{M}_1 t}\mathbf{U}$$

provided that $\mathbf{M}_1$ and $\mathbf{M}$ commute. We will prove this below. So,

$$\begin{aligned}\mathbf{V}_t &= \mathrm{e}^{\mathbf{M}_1 t}\mathbf{f}(\mathbf{U}) - \mathbf{M}\mathrm{e}^{\mathbf{M}_1 t}\mathbf{U} + \mathbf{M}_1\mathrm{e}^{\mathbf{M}_1 t}\mathbf{U} \\ &= \mathrm{e}^{\mathbf{M}_1 t}\mathbf{f}(\mathbf{U}) - (\mathbf{M} - \mathbf{M}_1)\mathrm{e}^{\mathbf{M}_1 t}\mathbf{U} \\ &= \mathrm{e}^{\mathbf{M}_1 t}\mathbf{f}(\mathbf{U}) - (\mathbf{M}_2 + \mathbf{M}_3)\mathrm{e}^{\mathbf{M}_1 t}\mathbf{U}\end{aligned}$$

where the last equality comes from $\mathbf{M} = \mathbf{M}_1 + \mathbf{M}_2 + \mathbf{M}_3$.

Since $\mathbf{V} = \mathrm{e}^{\mathbf{M}_1 t}\mathbf{U}$, and hence, by Lemma 3, $\mathbf{U} = \mathrm{e}^{-\mathbf{M}_1 t}\mathbf{V}$,

$$\begin{aligned}\mathbf{V}_t &= \mathrm{e}^{\mathbf{M}_1 t}\mathbf{f}(\mathrm{e}^{-\mathbf{M}_1 t}\mathbf{V}) - (\mathbf{M}_2 + \mathbf{M}_3)\mathbf{V} \\ &= \mathbf{g}(\mathbf{V}) - (\mathbf{M}_2 + \mathbf{M}_3)\mathbf{V}\end{aligned}$$

where we define $\mathbf{g}(\mathbf{V}) = \mathrm{e}^{\mathbf{M}_1 t}\mathbf{f}(\mathrm{e}^{-\mathbf{M}_1 t}\mathbf{V})$.

This is a transformed differential equation. We adapt the initial conditions $\mathbf{V}(0) = \mathrm{e}^{\mathbf{M}_1 \cdot 0}\mathbf{U}(0) = \mathbf{U}(0) = \mathbf{U}_0$. Hence, we obtain the initial value problem

$$\mathbf{V}_t = \mathbf{g}(\mathbf{V}) - (\mathbf{M}_2 + \mathbf{M}_3)\mathbf{V}, \quad \mathbf{V}(0) = \mathbf{U}_0$$

Analogously, we can define $\mathbf{W} = \mathrm{e}^{\mathbf{M}_2 t}\mathbf{V}$. We obtain

$$\begin{aligned}\mathbf{W}_t &= \mathrm{e}^{\mathbf{M}_2 t}\mathbf{V}_t + \mathbf{M}_2\mathrm{e}^{\mathbf{M}_2 t}\mathbf{V} \\ &= \mathrm{e}^{\mathbf{M}_2 t}(\mathbf{g}(\mathbf{V}) - (\mathbf{M}_2 + \mathbf{M}_3)\mathbf{V}) + \mathbf{M}_2\mathrm{e}^{\mathbf{M}_2 t}\mathbf{V} \\ &= \mathrm{e}^{\mathbf{M}_2 t}\mathbf{g}(\mathbf{V}) - (\mathbf{M}_2 + \mathbf{M}_3)\mathrm{e}^{\mathbf{M}_2 t}\mathbf{V} + \mathbf{M}_2\mathrm{e}^{\mathbf{M}_2 t}\mathbf{V} \\ &= \mathrm{e}^{\mathbf{M}_2 t}\mathbf{g}(\mathbf{V}) - \mathbf{M}_3\mathrm{e}^{\mathbf{M}_2 t}\mathbf{V} \\ &= \mathrm{e}^{\mathbf{M}_2 t}\mathbf{g}(\mathrm{e}^{-\mathbf{M}_2 t}\mathbf{W}) - \mathbf{M}_3\mathrm{e}^{\mathbf{M}_2 t}\mathbf{V}\end{aligned}$$

$$= \mathbf{h}(\mathbf{W}) - \mathbf{M}_3\mathbf{W}$$

with $\mathbf{h}(\mathbf{W}) = \mathrm{e}^{\mathbf{M}_2 t}\mathbf{g}(\mathrm{e}^{-\mathbf{M}_2 t}\mathbf{W})$.

By again transforming the initial conditions, we obtain the following IVP

$$\mathbf{W}_t = \mathbf{h}(\mathbf{W}) - \mathbf{M}_3\mathbf{W}, \quad \mathbf{W}(0) = \mathbf{U}_0$$

This system is what we will set out to solve. Once we have the solution $\mathbf{W}$, it is easy to compute

$$\mathbf{V} = \mathrm{e}^{-\mathbf{M}_2 t}\mathbf{W}$$

$$\mathbf{U} = \mathrm{e}^{-\mathbf{M}_1 t}\mathbf{V}$$

Notice that we can express $\mathbf{h}(\mathbf{W})$ as

$$\mathbf{h}(\mathbf{W}) = \mathrm{e}^{\mathbf{M}_2 t}\mathbf{g}(\mathrm{e}^{-\mathbf{M}_2 t}\mathbf{W}) = \mathrm{e}^{\mathbf{M}_2 t}\mathrm{e}^{\mathbf{M}_1 t}\mathbf{f}(\mathrm{e}^{-\mathbf{M}_1 t}\mathrm{e}^{-\mathbf{M}_2 t}\mathbf{W})$$

Considering this, we can generalize the splitting technique we derived above to d dimensions:

$$\mathbf{Z}_t = \mathbf{h}(\mathbf{Z}) - \mathbf{M}_d\mathbf{Z}, \quad \mathbf{Z}(0) = \mathbf{U}_0 \tag{14}$$

where

$$\mathbf{h}(\mathbf{Z}) = \mathrm{e}^{\sum_{k=1}^{d-1}\mathbf{M}_k t}\mathbf{f}(\mathrm{e}^{-\sum_{k=1}^{d-1}\mathbf{M}_k t}).$$

Note that, by Lemma 2,

$$\mathrm{e}^{\sum_{k=1}^{d-1}\mathbf{M}_k} = \prod_{k=1}^{d-1}\mathrm{e}^{\mathbf{M}_k},$$

provided $\mathbf{M}_k, k = 1, \ldots, d-1$ commute pairwise.

Having solved Equation (14), we can compute

$$\mathbf{U} = \mathrm{e}^{-\sum_{k=1}^{d-1}\mathbf{M}_k t}\mathbf{Z}.$$

Therefore, the scheme we construct below will be applied to Equation (14) and then solved for $\mathbf{U}$.

In order to justify this construction, what is left to prove is that $\mathbf{M}$ and $\mathbf{M}_k$ commute for all $k = 1, \ldots, d$ and that $\mathbf{M}_k, k = 1, \ldots, d-1$ commute pairwise. Asante-Asamani et al. (2020) proved similar statements. However, we are considering more general matrices here and therefore repeat the proof in this more general version, adapted to the precise formalism we described in Section 2.1.1.4. Since $\mathbf{M} = \sum_{k=1}^{d} \mathbf{M}_k$, it suffices to show the following

Lemma 6. $\mathbf{M}_k, k = 1, \ldots, d$ are pairwise commutative, i. e.,

$$\mathbf{M}_k \cdot \mathbf{M}_\ell = \mathbf{M}_\ell \cdot \mathbf{M}_k \quad \forall k, \ell = 1, \ldots, d\,.$$

Proof. Let k, ℓ be fixed. Recall that $\mathbf{M}_k = \sum_{i=1}^{s} \mathbf{M}_k^{(i)} \otimes \mathbf{E}^{(i)}$ and $\mathbf{M}_\ell = \sum_{j=1}^{s} \mathbf{M}_\ell^{(j)} \otimes \mathbf{E}^{(j)}$ in the notation of Section 2.1.1.4. Assuming that $\mathbf{M}_k^{(i)}$ and $\mathbf{M}_\ell^{(j)}$ commute for all $i, j = 1, \ldots, s$, we show the statement using the mixed product property of the Kronecker product and matrix multiplication (Van Loan 2000).

$$\begin{aligned}
\mathbf{M}_k \cdot \mathbf{M}_\ell &= \sum_{i=1}^{s} \mathbf{M}_k^{(i)} \otimes \mathbf{E}^{(i)} \cdot \sum_{j=1}^{s} \mathbf{M}_\ell^{(j)} \otimes \mathbf{E}^{(j)} \\
&= \sum_{i,j=1}^{s} (\mathbf{M}_k^{(i)} \otimes \mathbf{E}^{(i)}) \cdot (\mathbf{M}_\ell^{(j)} \otimes \mathbf{E}^{(j)}) \\
&= \sum_{i,j=1}^{s} (\mathbf{M}_k^{(i)} \cdot \mathbf{M}_\ell^{(j)}) \otimes (\mathbf{E}^{(i)} \cdot \mathbf{E}^{(j)}) \\
&= \sum_{i,j=1}^{s} (\mathbf{M}_\ell^{(j)} \cdot \mathbf{M}_k^{(i)}) \otimes (\mathbf{E}^{(j)} \cdot \mathbf{E}^{(i)}) \\
&= \mathbf{M}_\ell \cdot \mathbf{M}_k
\end{aligned}$$

as $\mathbf{E}^{(i)} \cdot \mathbf{E}^{(j)} = \delta_{ij}\mathbf{E}^{(i)} = \delta_{ij}\mathbf{E}^{(j)}$, where δ_{ij} denotes the Kronecker delta.

What is left to prove is that $\mathbf{M}_k^{(i)}$ and $\mathbf{M}_\ell^{(j)}$ commute (see also Asante-Asamani et al. (2020)). Suppose, without loss of generality, $k < \ell$ and let i, j be fixed.

$$\begin{aligned}\mathbf{M}_k^{(i)} \cdot \mathbf{M}_\ell^{(j)} &= \left(\left(\bigotimes_{m=1}^{d-k} \mathbf{I}_p \right) \otimes \tilde{\mathbf{M}}_k^{(i)} \otimes \left(\bigotimes_{m=1}^{k-1} \mathbf{I}_p \right) \right) \\ &\cdot \left(\left(\bigotimes_{m=1}^{d-\ell} \mathbf{I}_p \right) \otimes \tilde{\mathbf{M}}_\ell^{(j)} \otimes \left(\bigotimes_{m=1}^{\ell-1} \mathbf{I}_p \right) \right) \\ &= \left(\bigotimes_{m=1}^{d-\ell} \mathbf{I}_p \right) \otimes \tilde{\mathbf{M}}_\ell^{(j)} \otimes \left(\bigotimes_{m=1}^{\ell-k-1} \mathbf{I}_p \right) \otimes \tilde{\mathbf{M}}_k^{(i)} \otimes \left(\bigotimes_{m=1}^{k-1} \mathbf{I}_p \right)\end{aligned}$$

Analogously,

$$\begin{aligned}\mathbf{M}_\ell^{(j)} \cdot \mathbf{M}_k^{(i)} &= \left(\left(\bigotimes_{m=1}^{d-\ell} \mathbf{I}_p \right) \otimes \tilde{\mathbf{M}}_\ell^{(j)} \otimes \left(\bigotimes_{m=1}^{\ell-1} \mathbf{I}_p \right) \right) \\ &\cdot \left(\left(\bigotimes_{m=1}^{d-k} \mathbf{I}_p \right) \otimes \tilde{\mathbf{M}}_k^{(i)} \otimes \left(\bigotimes_{m=1}^{k-1} \mathbf{I}_p \right) \right) \\ &= \left(\bigotimes_{m=1}^{d-\ell} \mathbf{I}_p \right) \otimes \tilde{\mathbf{M}}_\ell^{(j)} \otimes \left(\bigotimes_{m=1}^{\ell-k-1} \mathbf{I}_p \right) \otimes \tilde{\mathbf{M}}_k^{(i)} \otimes \left(\bigotimes_{m=1}^{k-1} \mathbf{I}_p \right)\end{aligned}$$

Therefore, we conclude

$$\mathbf{M}_k^{(i)} \cdot \mathbf{M}_\ell^{(j)} = \mathbf{M}_\ell^{(j)} \cdot \mathbf{M}_k^{(i)} \ .$$

□

Thus, the required commutativity is given, even in the presence of advection, and the above construction is justified.

2.2.2 Discretization in Time

In the previous sections, we have obtained a (spatial) semi-discretization of our initial problem and reduced it via dimensional splitting to (14). The next step is to discretize the equation in time.

For the purpose of constructing the scheme, we consider an ODE of the form (14) where we assume $\mathbf{h}$ to be sufficiently smooth.

Duhamel's Principle. From Duhamel's principle (Weissler 1979; Zheng 2004), we know that a solution $\mathbf{Z}$ of (14) satisfies the following equation:

$$\mathbf{Z}(t) = \mathrm{e}^{-\mathbf{M}_d t}\mathbf{Z}(0) + \int_0^t \mathrm{e}^{-\mathbf{M}_d(t-s)}\mathbf{h}(\mathbf{Z}(s))\,\mathrm{d}s \tag{15}$$

Based on this, we can derive an exact recurrence relation that is also satisfied by the exact solution $\mathbf{Z}$:

$$\begin{aligned}
\mathbf{Z}(t_{n+1}) &= \mathrm{e}^{-\mathbf{M}_d t_{n+1}}\mathbf{Z}(0) + \int_0^{t_{n+1}} \mathrm{e}^{-\mathbf{M}_d(t_{n+1}-s)}\mathbf{h}(\mathbf{Z}(s))\,\mathrm{d}s \\
&= \mathrm{e}^{-\mathbf{M}_d k}\mathbf{Z}(0)\mathrm{e}^{-\mathbf{M}_d t_n}\mathbf{Z}(0) \\
&\quad + \int_0^{t_n} \mathrm{e}^{-\mathbf{M}_d(t_n+k-s)}\mathbf{h}(\mathbf{Z}(s))\,\mathrm{d}s + \int_{t_n}^{t_{n+1}} \mathrm{e}^{-\mathbf{M}_d(t_n+k-s)}\mathbf{h}(\mathbf{Z}(s))\,\mathrm{d}s \\
&= \mathrm{e}^{-\mathbf{M}_d k}\mathbf{Z}(t_n) + \int_{t_n}^{t_{n+1}} \mathrm{e}^{-\mathbf{M}_d(t_{n+1}-s)}\mathbf{h}(\mathbf{Z}(s))\,\mathrm{d}s
\end{aligned}$$

where k is the time step and $t_n = nk \quad \forall n$.

All ETD schemes are based on this recurrence relation. The difficulty arises here in evaluating the integral and approximating the matrix exponential (Cox and Matthews 2002; Asante-Asamani 2016). We describe techniques for both below, again following Asante-Asamani et al. (2020).

Before we continue, we perform a change of variables $\tau = (s - t_n)/k$ to simplify the recurrence relation.

$$\mathbf{Z}(t_{n+1}) = \mathrm{e}^{-\mathbf{M}_d k}\mathbf{Z}(t_n) + k\int_0^1 \mathrm{e}^{-k\mathbf{M}_d(1-\tau)}\mathbf{h}(\mathbf{Z}(t_n + k\tau))\,\mathrm{d}\tau \tag{16}$$

Approximation of the Integral. Note that the recurrence relation contains an integration of $\mathbf{h}(\mathbf{Z}(t))$ over the interval $[t_n;\ t_{n+1}]$, and hence requires knowledge of

$\mathbf{Z}(t)$ on the whole interval. Since this is not given, we interpolate $\mathbf{h}$ by a linear function between $\mathbf{Z}(t_{n+1})$ and $\mathbf{Z}(t_n)$:

$$\mathbf{h}(\mathbf{Z}(t_n + k\tau)) \approx \mathbf{h}(\mathbf{Z}(t_n)) + (\mathbf{h}(\mathbf{Z}(t_n + k)) - \mathbf{h}(\mathbf{Z}(t_n)))\tau$$

This yields the following approximation for the integral:

$$\begin{aligned}\mathbf{Z}(t_{n+1}) &\approx \mathrm{e}^{-\mathbf{M}_d k}\mathbf{Z}(t_n) \\ &\quad + k\int_0^1 \mathrm{e}^{-k\mathbf{M}_d(1-\tau)}\left[\mathbf{h}(\mathbf{Z}(t_n)) + (\mathbf{h}(\mathbf{Z}(t_n + k)) - \mathbf{h}(\mathbf{Z}(t_n)))\tau\right]\mathrm{d}\tau \\ &= \mathrm{e}^{-\mathbf{M}_d k}\mathbf{Z}(t_n) + k\int_0^1 \mathrm{e}^{-k\mathbf{M}_d(1-\tau)}\,\mathrm{d}\tau \cdot \mathbf{h}(\mathbf{Z}(t_n)) \\ &\quad + k\int_0^1 \mathrm{e}^{-k\mathbf{M}_d(1-\tau)}\tau\,\mathrm{d}\tau \cdot (\mathbf{h}(\mathbf{Z}(t_{n+1})) - \mathbf{h}(\mathbf{Z}(t_n)))\end{aligned} \tag{17}$$

This approximation has a truncation error of order 3, i. e., the truncation error is $\mathcal{O}(k^3)$.

Notice now that we can formally evaluate the above integrals exactly. However, the resulting equations are fully implicit since they depend on $\mathbf{h}(\mathbf{Z}(t_{n+1}))$. To make them explicit, we also obtain a locally second-order (i. e., the local truncation error is $\mathcal{O}(k^2)$) approximation (cf. also Cox and Matthews 2002, Asante-Asamani 2016) by approximating $\mathbf{h}$ with a constant $\mathbf{h}(\mathbf{Z}(t_n))$:

$$\tilde{\mathbf{Z}}(t_{n+1}) = \mathrm{e}^{-\mathbf{M}_d k}\mathbf{Z}(t_n) + k\int_0^1 \mathrm{e}^{-k\mathbf{M}_d(1-\tau)}\mathbf{h}(\mathbf{Z}(t_n))\,\mathrm{d}\tau \tag{18}$$

We use this to approximate $\mathbf{h}(\mathbf{Z}(t_{n+1}))$ in the above equation (17) and obtain

$$\begin{aligned}\mathbf{Z}(t_{n+1}) &\approx \mathrm{e}^{-\mathbf{M}_d k}\mathbf{Z}(t_n) + k\int_0^1 \mathrm{e}^{-k\mathbf{M}_d(1-\tau)}\,\mathrm{d}\tau \cdot \mathbf{h}(\mathbf{Z}(t_n)) \\ &\quad + k\int_0^1 \mathrm{e}^{-k\mathbf{M}_d(1-\tau)}\tau\,\mathrm{d}\tau \cdot (\mathbf{h}(\tilde{\mathbf{Z}}(t_{n+1})) - \mathbf{h}(\mathbf{Z}(t_n)))\end{aligned} \tag{19}$$

Note that $\tilde{\mathbf{Z}}(t_{n+1})$ in Equation (18) can be interpreted as a predictor step, so we end up with a predictor-corrector type scheme. Cox and Matthews (2002) derive the

same scheme (18) and (19) by extending Runge-Kutta methods to ETD. They therefore call this scheme ETD2RK. They show that in the case of a scalar ODE (imagine $\mathbf{M}_d$ and $\mathbf{Z}$ are scalars), this is an approximation of locally third order in k (i. e., the local truncation error is $O(k^3)$). It is worth noting here that, according to their analysis, the error constant depends only on $\mathbf{h}$. This suggests that it is likely independent of $\mathbf{M}_d$ in our case as well. However, we are not aware of a proof for this.

Evaluation of the Remaining Integrals. The further derivation of the scheme in Asante-Asamani et al. (2020) is based on the assumption that $\mathbf{M}_d$ is a non-singular matrix. However, in many practical cases, this is not given. For example, in this work, we consider periodic boundary conditions. To show that the linear operator (consisting of advection and diffusion with boundary conditions) is not invertible, consider the steady-state system

$$0 = \mathbf{a}_i \cdot \nabla u_i + \nabla \cdot \mathbf{D}_i \nabla u_i \,, \quad i = 1, \ldots, s$$

with periodic boundary conditions. Note that we set f identically 0.

Suppose $\mathbf{u}$ is a solution of the steady-state system and consider $\mathbf{v} = \mathbf{u} + \mathbf{c}$ where $\mathbf{c}$ is a constant vector in $\mathbb{R}^s$. Now let $1 \leq i \leq s$ be given. Then,

$$\begin{aligned}
\mathbf{a}_i \cdot \nabla v_i + \nabla \cdot \mathbf{D}_i \nabla v_i &= \mathbf{a}_i \cdot \nabla (u_i + c_i) + \nabla \cdot \mathbf{D}_i \nabla (u_i + c_i) \\
&= \mathbf{a}_i \cdot (\nabla u_i + \nabla c_i) + \nabla \cdot \mathbf{D}_i (\nabla u_i + \nabla c_i) \\
&= \mathbf{a}_i \cdot \nabla u_i + \nabla \cdot \mathbf{D}_i \nabla u_i \\
&= 0
\end{aligned}$$

Therefore, $\mathbf{v}$ is also a solution, and the operator is not invertible.

Therefore, we also do not expect the discretized operator, the discretization matrix $\mathbf{M}_d$ to be invertible. And indeed we observed for small discretization matrices with periodic or vanishing normal derivative boundary conditions that these matrices have some zero eigenvalues.

However, after a method-of-lines discretization of (1), the resulting IVP (3) — with a regular or singular matrix — is uniquely solvable for reaction terms that fulfill the conditions of the Picard-Lindelöf theorem (Hartman 2002). In the following, we also perform the derivation assuming non-singular matrices to justify why the same scheme works for singular matrices as well. Note that Shampine (1994) shows that under the assumptions we pose, in particular, positive diffusion constants, the eigenvalues of the discretization matrix have non-negative real parts for Dirichlet boundary conditions and boundary conditions involving normal derivatives. It is, therefore, a reasonable assumption that all eigenvalues of $\mathbf{M}_d$ have non-negative real parts.

We can now consider the integrals in (19) and note that these are integrals of exponentials of linear functions that can be formally evaluated by standard techniques. Asante-Asamani (2016) and Asante-Asamani et al. (2020) give the following lemma:

Lemma 7. Let $\mathbf{A} \in \mathbb{C}^{n\times n}$ be a non-singular matrix. Then

$$k\int_0^1 \mathrm{e}^{-k\mathbf{A}(1-\tau)}\mathrm{d}\tau = \mathbf{A}^{-1}(\mathbf{I}-\mathrm{e}^{-k\mathbf{A}})$$

$$k\int_0^1 \mathrm{e}^{-k\mathbf{A}(1-\tau)}\tau\mathrm{d}\tau = k^{-1}\mathbf{A}^{-2}(k\mathbf{A}-\mathbf{I}+\mathrm{e}^{-k\mathbf{A}})$$

Here, $\mathbf{I} \in \mathbb{C}^{n\times n}$ is the identity matrix.

Proof. See Asante-Asamani (2016) and Asante-Asamani et al. (2020). □

However, the matrix exponential is well-defined for any matrix $\mathbf{A} \in \mathbb{C}^{n\times n}$. By Lemma 4, the given exponential functions in Lemma 7 are, in fact, differentiable, for any $\mathbf{A}$. Therefore, the integrals are well-defined for any matrix $\mathbf{A}$. Hence, we know that the integrals can be computed even for singular matrices.

We prove a generalization of Lemma 7.

Lemma 8. Let $\mathbf{A} \in \mathbb{C}^{n\times n}$ be a arbitrary square matrix. Then

$$k\mathbf{A}\int_0^1 \mathrm{e}^{-k\mathbf{A}(1-\tau)}\mathrm{d}\tau = \mathbf{I} - \mathrm{e}^{-k\mathbf{A}}$$
$$k\mathbf{A}^2\int_0^1 \mathrm{e}^{-k\mathbf{A}(1-\tau)}\tau\mathrm{d}\tau = k^{-1}(k\mathbf{A} - \mathbf{I} + \mathrm{e}^{-k\mathbf{A}})$$

Proof. We prove this analogously to Asante-Asamani et al. (2020).

We know that

$$\frac{\mathrm{d}}{\mathrm{d}\tau}\mathrm{e}^{-k\mathbf{A}(1-\tau)} = k\mathbf{A}\mathrm{e}^{-k\mathbf{A}(1-\tau)}.$$

Therefore,

$$k\mathbf{A}\int_0^1 \mathrm{e}^{-k\mathbf{A}(1-\tau)}\mathrm{d}\tau = \int_0^1 k\mathbf{A}\mathrm{e}^{-k\mathbf{A}(1-\tau)}\mathrm{d}\tau = \int_0^1 \frac{\mathrm{d}}{\mathrm{d}\tau}\mathrm{e}^{-k\mathbf{A}(1-\tau)}\mathrm{d}\tau = \mathbf{I} - \mathrm{e}^{-k\mathbf{A}}$$

In order to obtain the second equality, we first employ a change of variables $\sigma = (1-\tau)k$:

$$\begin{aligned} k\mathbf{A}^2\int_0^1 \mathrm{e}^{-k\mathbf{A}(1-\tau)}\tau\mathrm{d}\tau &= -\mathbf{A}^2\int_0^k \mathrm{e}^{-\sigma\mathbf{A}}(1-k^{-1}\sigma)\mathrm{d}\sigma \\ &= \mathbf{A}^2\int_0^k \mathrm{e}^{-\sigma\mathbf{A}}\mathrm{d}\sigma - k^{-1}\mathbf{A}^2\int_0^k \mathrm{e}^{-\sigma\mathbf{A}}\sigma\mathrm{d}\sigma \end{aligned}$$

Note that

$$\frac{\mathrm{d}}{\mathrm{d}\sigma}\mathrm{e}^{-\sigma\mathbf{A}} = -\mathbf{A}\mathrm{e}^{-\sigma\mathbf{A}},$$

so we can evaluate the first integral on the right hand side to

$$\mathbf{A}^2 \int_0^k \mathrm{e}^{-\sigma\mathbf{A}} \mathrm{d}\sigma = -\mathbf{A} \int_0^k \frac{\mathrm{d}}{\mathrm{d}\sigma} \mathrm{e}^{-\sigma\mathbf{A}} \mathrm{d}\sigma = -\mathbf{A}(\mathrm{e}^{-k\mathbf{A}} - \mathbf{I})$$

Also,

$$\frac{\mathrm{d}}{\mathrm{d}\sigma} \mathbf{A}\mathrm{e}^{-\sigma\mathbf{A}}\sigma + \mathrm{e}^{-\sigma\mathbf{A}} = -\mathbf{A}^2\mathrm{e}^{-\sigma\mathbf{A}},$$

so

$$-k^{-1}\mathbf{A}^2 \int_0^k \mathrm{e}^{-\sigma\mathbf{A}}\sigma \mathrm{d}\sigma = k^{-1} \int_0^k \frac{\mathrm{d}}{\mathrm{d}\sigma} \mathbf{A}\mathrm{e}^{-\sigma\mathbf{A}}\sigma + \mathrm{e}^{-\sigma\mathbf{A}} \mathrm{d}\sigma = \mathbf{A}\mathrm{e}^{-k\mathbf{A}} + k^{-1}\mathrm{e}^{-k\mathbf{A}} - k^{-1}\mathbf{I}$$

Adding these, we obtain

$$\begin{aligned} k\mathbf{A}^2 \int_0^1 \mathrm{e}^{-k\mathbf{A}(1-\tau)}\tau \mathrm{d}\tau &= -\mathbf{A}\mathrm{e}^{-k\mathbf{A}} + \mathbf{A} + \mathbf{A}\mathrm{e}^{-k\mathbf{A}} + k^{-1}\mathrm{e}^{-k\mathbf{A}} - k^{-1}\mathbf{I} \\ &= \mathbf{A} + k^{-1}(\mathrm{e}^{-k\mathbf{A}} - \mathbf{I}) \\ &= k^{-1}(k\mathbf{A} - \mathbf{I} + \mathrm{e}^{-k\mathbf{A}}), \end{aligned}$$

the desired result. □

Note that we did not directly evaluate the integrals from Lemma 7. In fact, we do not need this. The next step in the derivation of Asante-Asamani et al. (2020) is to approximate the matrix exponentials, so only approximations are necessary. We first consider the approximation of the matrix exponential before we show that the relevant approximations of the original integrals hold also in the case of singular matrices. After that, we summarize all in a proof of second-order accuracy for the resulting scheme.

2.2.3 Approximating the Matrix Exponential

Asante-Asamani (2016) and Asante-Asamani et al. (2020) apply the following RDP approximation of the matrix exponential to the results of Lemma 7.

$$\mathbf{R}_{RDP}(k\mathbf{A}) = \left(\mathbf{I} - \frac{5k}{12}\mathbf{A}\right)\left(\mathbf{I} + \frac{k}{4}\mathbf{A}\right)^{-1}\left(\mathbf{I} + \frac{k}{3}\mathbf{A}\right)^{-1} = \mathrm{e}^{-k\mathbf{A}} + \mathcal{O}(k^3), \tag{20}$$

They obtain

$$\begin{aligned}
&k\int_0^1 \mathrm{e}^{-k\mathbf{A}(1-\tau)}\mathrm{d}\tau \\
&= \mathbf{A}^{-1}\left(\mathbf{I} - \mathrm{e}^{-k\mathbf{A}}\right) \\
&= \mathbf{A}^{-1}\left(\mathbf{I} - \mathbf{R}_{RDP}(k\mathbf{A}) + \mathcal{O}(k^3)\right) \\
&= \mathbf{A}^{-1}\left(\mathbf{I} - \mathbf{R}_{RDP}(k\mathbf{A})\right) + \mathcal{O}(k^3) \\
&= \mathbf{A}^{-1}\left(\mathbf{I} - \left(\mathbf{I} - \frac{5k}{12}\mathbf{A}\right)\left(\mathbf{I} + \frac{k}{4}\mathbf{A}\right)^{-1}\left(\mathbf{I} + \frac{k}{3}\mathbf{A}\right)^{-1}\right) + \mathcal{O}(k^3) \\
&= \mathbf{A}^{-1}\left(\left(\mathbf{I} + \frac{k}{3}\mathbf{A}\right)\left(\mathbf{I} + \frac{k}{4}\mathbf{A}\right) - \left(\mathbf{I} - \frac{5k}{12}\mathbf{A}\right)\right)\left(\mathbf{I} + \frac{k}{4}\mathbf{A}\right)^{-1}\left(\mathbf{I} + \frac{k}{3}\mathbf{A}\right)^{-1} + \mathcal{O}(k^3) \\
&= \mathbf{A}^{-1}\left(\mathbf{I} + \frac{7k}{12}\mathbf{A} + \frac{k^2}{12}\mathbf{A}^2 - \left(\mathbf{I} - \frac{5k}{12}\mathbf{A}\right)\right)\left(\mathbf{I} + \frac{k}{4}\mathbf{A}\right)^{-1}\left(\mathbf{I} + \frac{k}{3}\mathbf{A}\right)^{-1} + \mathcal{O}(k^3) \\
&= \mathbf{A}^{-1}\left(k\mathbf{A} + \frac{k^2}{12}\mathbf{A}^2\right)\left(\mathbf{I} + \frac{k}{4}\mathbf{A}\right)^{-1}\left(\mathbf{I} + \frac{k}{3}\mathbf{A}\right)^{-1} + \mathcal{O}(k^3) \\
&= \left(k\mathbf{I} + \frac{k^2}{12}\mathbf{A}\right)\left(\mathbf{I} + \frac{k}{4}\mathbf{A}\right)^{-1}\left(\mathbf{I} + \frac{k}{3}\mathbf{A}\right)^{-1} + \mathcal{O}(k^3) \\
&= k\left(\mathbf{I} + \frac{k}{12}\mathbf{A}\right)\left(\mathbf{I} + \frac{k}{4}\mathbf{A}\right)^{-1}\left(\mathbf{I} + \frac{k}{3}\mathbf{A}\right)^{-1} + \mathcal{O}(k^3)
\end{aligned}$$

Analogously,

$$\begin{aligned}
&k\int_0^1 \mathrm{e}^{-k\mathbf{A}(1-\tau)}\tau\mathrm{d}\tau \\
&= k^{-1}\mathbf{A}^{-2}\left(k\mathbf{A} - \mathbf{I} + \mathrm{e}^{-k\mathbf{A}}\right)
\end{aligned}$$

$$
\begin{aligned}
&= k^{-1}\mathbf{A}^{-2}\left(k\mathbf{A} - \mathbf{I} + \mathbf{R}_{RDP}(k\mathbf{A}) + \mathcal{O}(k^3)\right) \\
&= k^{-1}\mathbf{A}^{-2}\left(k\mathbf{A} - \mathbf{I} + \mathbf{R}_{RDP}(k\mathbf{A})\right) + \mathcal{O}(k^2) \\
&= k^{-1}\mathbf{A}^{-2}\left(k\mathbf{A} - \mathbf{I} + \left(\mathbf{I} - \frac{5k}{12}\mathbf{A}\right)\left(\mathbf{I} + \frac{k}{4}\mathbf{A}\right)^{-1}\left(\mathbf{I} + \frac{k}{3}\mathbf{A}\right)^{-1}\right) + \mathcal{O}(k^2) \\
&= k^{-1}\mathbf{A}^{-2}\left(k\mathbf{A}\left(\mathbf{I} + \frac{k}{3}\mathbf{A}\right)\left(\mathbf{I} + \frac{k}{4}\mathbf{A}\right) - \left(\mathbf{I} + \frac{k}{3}\mathbf{A}\right)\left(\mathbf{I} + \frac{k}{4}\mathbf{A}\right) + \left(\mathbf{I} - \frac{5k}{12}\mathbf{A}\right)\right) \\
&\quad \cdot \left(\mathbf{I} + \frac{k}{4}\mathbf{A}\right)^{-1}\left(\mathbf{I} + \frac{k}{3}\mathbf{A}\right)^{-1} + \mathcal{O}(k^2) \\
&= k^{-1}\mathbf{A}^{-2}\left(k\mathbf{A}\left(\mathbf{I} + \frac{k}{3}\mathbf{A}\right)\left(\mathbf{I} + \frac{k}{4}\mathbf{A}\right) - \left(\mathbf{I} + \frac{7k}{12}\mathbf{A} + \frac{k^2}{12}\mathbf{A}^2\right) + \left(\mathbf{I} - \frac{5k}{12}\mathbf{A}\right)\right) \\
&\quad \cdot \left(\mathbf{I} + \frac{k}{4}\mathbf{A}\right)^{-1}\left(\mathbf{I} + \frac{k}{3}\mathbf{A}\right)^{-1} + \mathcal{O}(k^2) \\
&= k^{-1}\mathbf{A}^{-2}\left(k\mathbf{A}\left(\mathbf{I} + \frac{k}{3}\mathbf{A}\right)\left(\mathbf{I} + \frac{k}{4}\mathbf{A}\right) - k\mathbf{A} - \frac{k^2}{12}\mathbf{A}^2\right)\left(\mathbf{I} + \frac{k}{4}\mathbf{A}\right)^{-1}\left(\mathbf{I} + \frac{k}{3}\mathbf{A}\right)^{-1} \\
&\quad + \mathcal{O}(k^2) \\
&= \mathbf{A}^{-1}\left(\left(\mathbf{I} + \frac{k}{3}\mathbf{A}\right)\left(\mathbf{I} + \frac{k}{4}\mathbf{A}\right) - \mathbf{I} - \frac{k}{12}\mathbf{A}\right)\left(\mathbf{I} + \frac{k}{4}\mathbf{A}\right)^{-1}\left(\mathbf{I} + \frac{k}{3}\mathbf{A}\right)^{-1} + \mathcal{O}(k^2) \\
&= \mathbf{A}^{-1}\left(\left(\mathbf{I} + \frac{7k}{12}\mathbf{A} + \frac{k^2}{12}\mathbf{A}^2\right) - \mathbf{I} - \frac{k}{12}\mathbf{A}\right)\left(\mathbf{I} + \frac{k}{4}\mathbf{A}\right)^{-1}\left(\mathbf{I} + \frac{k}{3}\mathbf{A}\right)^{-1} + \mathcal{O}(k^2) \\
&= \mathbf{A}^{-1}\left(\frac{k}{2}\mathbf{A} + \frac{k^2}{12}\mathbf{A}^2\right)\left(\mathbf{I} + \frac{k}{4}\mathbf{A}\right)^{-1}\left(\mathbf{I} + \frac{k}{3}\mathbf{A}\right)^{-1} + \mathcal{O}(k^2) \\
&= \frac{k}{2}\left(\mathbf{I} + \frac{k}{6}\mathbf{A}\right)\left(\mathbf{I} + \frac{k}{4}\mathbf{A}\right)^{-1}\left(\mathbf{I} + \frac{k}{3}\mathbf{A}\right)^{-1} + \mathcal{O}(k^2)
\end{aligned}
$$

Since we wish to also allow for singular matrices $\mathbf{A}$, we need to rely on Lemma 8. The following Lemma 9 shows that the same results hold, i. e., that we obtain the same orders of approximations with the formulas we just derived.

Lemma 9. Let $\mathbf{A} \in \mathbb{R}^{n\times n}$ be a matrix such that all eigenvalues of $\mathbf{A}$ have non-negative real parts. Then, the RDP approximation is well-defined and

$$
k\int_0^1 \mathrm{e}^{-k\mathbf{A}(1-\tau)}\mathrm{d}\tau = k\left(\mathbf{I} + \frac{k}{12}\mathbf{A}\right)\left(\mathbf{I} + \frac{k}{4}\mathbf{A}\right)^{-1}\left(\mathbf{I} + \frac{k}{3}\mathbf{A}\right)^{-1} + \mathcal{O}(k^3)
$$

Proof. Note first that, by a standard result from linear algebra, each eigenvalue μ of $\mathbf{I} + \mathbf{B}$ is equal to $\lambda + 1$ for some eigenvalue λ of $\mathbf{B}$. Hence, the real parts of the eigenvalues of

$$\mathbf{I} + \frac{k}{12}\mathbf{A}, \ \mathbf{I} + \frac{k}{4}\mathbf{A}, \text{ and } \mathbf{I} + \frac{k}{3}\mathbf{A}$$

are greater than 1. Hence, the matrices are invertible and the approximation is well-defined.

We know that $\mathbf{A}$ has a, possibly complex, Jordan normal form $\mathbf{A} = \mathbf{P}\mathbf{J}\mathbf{P}^{-1}$, where

$$\mathbf{J} = \begin{pmatrix} \mathbf{J}_1 & & & \\ & \mathbf{J}_2 & & \\ & & \ddots & \\ & & & \mathbf{J}_p \end{pmatrix} \in \mathbb{C}^{n \times n}$$

with the Jordan blocks

$$\mathbf{J}_i = \begin{pmatrix} \lambda_i & 1 & & \\ & \lambda_i & \ddots & \\ & & \ddots & 1 \\ & & & \lambda_i \end{pmatrix} \in \mathbb{C}^{n_i \times n_i}, \qquad i = 1, \ldots, p$$

Depending on the algebraic and geometric multiplicities of the eigenvalues of $\mathbf{A}$, there may exist $i \neq j$ such that $\lambda_i = \lambda_j$

We first consider the matrix $\mathbf{J}$ and prove that the approximation holds for all Jordan blocks $\mathbf{J}_i$. As $\mathbf{J}$ is a block-diagonal matrix, the exponential and the integral can be applied blockwise (Horn and Johnson 1991, chap. 6). Therefore, proving the

approximation for all Jordan blocks $\mathbf{J}_i$, we can conclude that it also holds for $\mathbf{J}$ as a whole.

Knowing that for all non-zero eigenvalues λ_i, $\mathbf{J}_i$ is invertible, this follows immediately from the previous derivation. What is left is to prove the approximation for all $\mathbf{J}_i$ with $\lambda_i = 0$. In this case, $\mathbf{J}_i$ has the following structure:

$$\mathbf{J}_i = \begin{pmatrix} 0 & 1 & & \\ & 0 & \ddots & \\ & & \ddots & 1 \\ & & & 0 \end{pmatrix} \in \mathbb{R}^{n_i \times n_i}$$

We can now apply the matrix exponential explicitly and get

$$\mathrm{e}^{-k\mathbf{J}_i(1-\tau)} = \begin{pmatrix} 1 & \frac{(-k(1-\tau))^1}{1!} & \dots & \frac{(-k(1-\tau))^{n_i-1}}{(n_i-1)!} \\ & 1 & \ddots & \vdots \\ & & \ddots & \frac{(-k(1-\tau))^1}{1!} \\ & & & 1 \end{pmatrix} \in \mathbb{R}^{n_i \times n_i}.$$

Applying the integral elementwise yields for the left hand side

$$k\int_0^1 \mathrm{e}^{-k\mathbf{J}_i(1-\tau)}\mathrm{d}\tau = \begin{pmatrix} \frac{k^1}{1!} & -\frac{k^2}{2!} & \dots & (-1)^{n_i-1}\frac{k^{n_i}}{n_i!} \\ & \frac{k^1}{1!} & \ddots & \vdots \\ & & \ddots & -\frac{k^2}{2!} \\ & & & \frac{k^1}{1!} \end{pmatrix} = \begin{pmatrix} k & -\frac{k^2}{2} & & \\ & k & \ddots & \\ & & \ddots & -\frac{k^2}{2} \\ & & & k \end{pmatrix} + \mathcal{O}(k^3).$$

Note that the above approximation is valid with the same error constant for arbitrary dimension of this $\mathbf{J}_i$, i. e., the error constant is independent of n_i. This is due to the

fact that the row-sum norm of the matrix

$$\begin{pmatrix} \frac{k^1}{1!} & -\frac{k^2}{2!} & \dots & (-1)^{n_i-1}\frac{k^{n_i}}{n_i!} \\ & \frac{k^1}{1!} & \ddots & \vdots \\ & & \ddots & -\frac{k^2}{2!} \\ & & & \frac{k^1}{1!} \end{pmatrix} - \begin{pmatrix} k & -\frac{k^2}{2} & & \\ & k & \ddots & \\ & & \ddots & -\frac{k^2}{2} \\ & & & k \end{pmatrix}$$

$$= \begin{pmatrix} 0 & 0 & \frac{k^3}{3!} & \dots & (-1)^{n_i-1}\frac{k^{n_i}}{n_i!} \\ & 0 & 0 & \ddots & \vdots \\ & & & \ddots & \frac{k^3}{3!} \\ & & & \ddots & 0 \\ & & & & 0 \end{pmatrix}$$

is bounded above by the infinite sequence

$$\sum_{j=3}^{\infty} \frac{k^j}{(j)!} = k^3 \sum_{j=0}^{\infty} \frac{k^j}{(j+3)!} \leq k^3 \sum_{j=0}^{\infty} \frac{k^j}{(j)!} = k^3 \mathrm{e}^k.$$

This depends only on k, not on the dimensions of the matrix. Furthermore, since we care about the limit as $k \to 0$, we may assume that $k <= 1$, and so $\mathrm{e}^k < \mathrm{e}$.

As $\mathbf{J}_i$ is nilpotent with $\mathbf{J}_i^{n_i} = 0$, we can compute the right hand side using the following Neumann series approach for $c \in \{3, 4\}$:

$$\left(\mathbf{I} + \frac{k}{c}\mathbf{J}_i\right)^{-1} = \left(\mathbf{I} - \left(-\frac{k}{c}\mathbf{J}_i\right)\right)^{-1} = \sum_{j=0}^{\infty} \left(-\frac{k}{c}\mathbf{J}_i\right)^j = \sum_{j=0}^{n_i} \left(-\frac{k}{c}\mathbf{J}_i\right)^j$$

$$= \begin{pmatrix} 1 & (-\frac{k}{c})^1 & \dots & (-\frac{k}{c})^{n_i-1} \\ & 1 & \ddots & \vdots \\ & & \ddots & (-\frac{k}{c})^1 \\ & & & 1 \end{pmatrix} \tag{21}$$

Clearly, this can be approximated as

$$\sum_{j=0}^{2}\left(-\frac{k}{c}\mathbf{J}_i\right)^j+\mathcal{O}(k^3)$$

The error constant is, again, independent of n_i. This can be seen by the following argument:

$$\sum_{j=0}^{n_i}\left(-\frac{k}{c}\mathbf{J}_i\right)^j-\sum_{j=0}^{2}\left(-\frac{k}{c}\mathbf{J}_i\right)^j=\sum_{j=3}^{n_i}\left(-\frac{k}{c}\mathbf{J}_i\right)^j=k^3\sum_{j=0}^{n_i}(-1)^{j+1}\frac{k^j}{c^{j+3}}\mathbf{J}_i^{j+3}$$

Note that the row-sum norm of $\mathbf{J}_i^j$ is $||\mathbf{J}_i^j||_\infty=1$ if $j\leq n_i$, and $||\mathbf{J}_i^j||_\infty=0$ otherwise. Hence, the row-sum norm of

$$k^3\sum_{j=0}^{n_i}(-1)^{j+1}\frac{k^j}{c^{j+3}}\mathbf{J}_i^{j+3}$$

is bounded above by the infinite sequence

$$k^3\sum_{j=3}^{\infty}\frac{k^j}{c^{j+3}}\leq k^3\sum_{j=0}^{\infty}\left(\frac{k}{c}\right)^j=k^3\frac{c}{c-k}$$

assuming, like above, that $k<1$. Assuming $k<0.5$, we can furthermore bound $\frac{c}{c-k}$ by 2 given that $c\in\{3,4\}$. These bounds are independent of n_i, as desired.

From the Neumann series, we obtain

$$k\left(\mathbf{I}+\frac{k}{12}\mathbf{J}_i\right)\left(\mathbf{I}+\frac{k}{4}\mathbf{J}_i\right)^{-1}\left(\mathbf{I}+\frac{k}{3}\mathbf{J}_i\right)^{-1}$$

$$=k\begin{pmatrix}1&\frac{k}{12}&\dots&\\&1&\ddots&\vdots\\&&\ddots&\frac{k}{12}\\&&&1\end{pmatrix}\begin{pmatrix}1&(-\frac{k}{4})^1&\dots&(-\frac{k}{4})^{n_i-1}\\&1&\ddots&\vdots\\&&\ddots&(-\frac{k}{4})^1\\&&&1\end{pmatrix}\begin{pmatrix}1&(-\frac{k}{3})^1&\dots&(-\frac{k}{3})^{n_i-1}\\&1&\ddots&\vdots\\&&\ddots&(-\frac{k}{3})^1\\&&&1\end{pmatrix}$$

which after multiplication and applying the binomial theorem and the above approximations equals

$$
\begin{aligned}
& k\left(\mathbf{I}+\frac{k}{12}\mathbf{J}_i\right)\sum_{j=0}^{n_i}\sum_{\ell=0}^{n_i}\left(-\frac{k}{4}\mathbf{J}_i\right)^j\left(-\frac{k}{3}\mathbf{J}_i\right)^\ell \\
=& k\left(\mathbf{I}+\frac{k}{12}\mathbf{J}_i\right)\sum_{j=0}^{n_i}\sum_{\ell=0}^{n_i}(-1)^{j+\ell}k^{j+\ell}\left(\frac{1}{4}\right)^j\left(\frac{1}{3}\right)^\ell\mathbf{J}_i^{j+\ell} \\
=& \sum_{j=0}^{n_i}\sum_{\ell=0}^{n_i}(-1)^{j+\ell}k^{j+\ell+1}\left(\frac{1}{4}\right)^j\left(\frac{1}{3}\right)^\ell\mathbf{J}_i^{j+\ell}+\frac{1}{12}\sum_{j=0}^{n_i}\sum_{\ell=0}^{n_i}(-1)^{j+\ell}k^{j+\ell+2}\left(\frac{1}{4}\right)^j\left(\frac{1}{3}\right)^\ell\mathbf{J}_i^{j+\ell+1} \\
=& \sum_{j=0}^{1}\sum_{\ell=0}^{1-j}(-1)^{j+\ell}k^{j+\ell+1}\left(\frac{1}{4}\right)^j\left(\frac{1}{3}\right)^\ell\mathbf{J}_i^{j+\ell} \\
& +\frac{1}{12}\sum_{j=0}^{0}\sum_{\ell=0}^{0}(-1)^{j+\ell}k^{j+\ell+2}\left(\frac{1}{4}\right)^j\left(\frac{1}{3}\right)^\ell\mathbf{J}_i^{j+\ell+1}+\mathcal{O}(k^3) \\
=& k\mathbf{I}-\frac{k^2}{3}\mathbf{J}_i-\frac{k^2}{4}\mathbf{J}_i+\frac{k^2}{12}\mathbf{J}_i+\mathcal{O}(k^3) \\
=& \begin{pmatrix} k & & & \\ & k & & \\ & & \ddots & \\ & & & k \end{pmatrix}+\begin{pmatrix} 0 & -\frac{k^2}{3} & & \\ & 0 & \ddots & \\ & & \ddots & -\frac{k^2}{3} \\ & & & 0 \end{pmatrix}+\begin{pmatrix} 0 & -\frac{k^2}{4} & & \\ & 0 & \ddots & \\ & & \ddots & -\frac{k^2}{4} \\ & & & 0 \end{pmatrix} \\
& +\begin{pmatrix} 0 & \frac{k^2}{12} & & \\ & 0 & \ddots & \\ & & \ddots & \frac{k^2}{12} \\ & & & 0 \end{pmatrix}+\mathcal{O}(k^3) \\
=& \begin{pmatrix} k & -\frac{k^2}{2} & & \\ & k & \ddots & \\ & & \ddots & -\frac{k^2}{2} \\ & & & k \end{pmatrix}+\mathcal{O}(k^3)
\end{aligned}
$$

We conclude from this representation that the approximation holds up to third order for $\mathbf{J}_i$ where $\mathbf{J}_i$ is a Jordan block for the eigenvalue $\lambda_i = 0$. As stated above, this approximation can be applied independently for each Jordan block as the full Jordan matrix $\mathbf{J}$ is block diagonal and the block structure is identical for all involved matrices $\mathbf{J}$ and $\mathbf{I}$. We can, thus, conclude that the approximation holds for any Jordan matrix $\mathbf{J}$.

From this, we can now derive that it holds for any matrix $\mathbf{A}$ by representing $\mathbf{A}$ as $\mathbf{A} = \mathbf{P}\mathbf{J}\mathbf{P}^{-1}$ where $\mathbf{A}, \mathbf{J}, \mathbf{P}$ are independent of k. Using Lemma 5, we obtain

$$
\begin{aligned}
& k \int_0^1 \mathrm{e}^{-k\mathbf{A}(1-\tau)} \mathrm{d}\tau \\
=& k \int_0^1 \mathrm{e}^{-k\mathbf{P}\mathbf{J}\mathbf{P}^{-1}(1-\tau)} \mathrm{d}\tau \\
=& k \int_0^1 \mathbf{P}\mathrm{e}^{-k\mathbf{J}(1-\tau)}\mathbf{P}^{-1} \mathrm{d}\tau \\
=& \mathbf{P} k \int_0^1 \mathrm{e}^{-k\mathbf{J}(1-\tau)} \mathrm{d}\tau \mathbf{P}^{-1} \\
=& \mathbf{P} \left(k \left(\mathbf{I} + \frac{k}{12}\mathbf{J} \right) \left(\mathbf{I} + \frac{k}{4}\mathbf{J} \right)^{-1} \left(\mathbf{I} + \frac{k}{3}\mathbf{J} \right)^{-1} + \mathcal{O}(k^3) \right) \mathbf{P}^{-1} \\
=& k\mathbf{P} \left(\mathbf{I} + \frac{k}{12}\mathbf{J} \right) \mathbf{P}^{-1}\mathbf{P} \left(\mathbf{I} + \frac{k}{4}\mathbf{J} \right)^{-1} \mathbf{P}^{-1}\mathbf{P} \left(\mathbf{I} + \frac{k}{3}\mathbf{J} \right)^{-1} \mathbf{P}^{-1} + \mathcal{O}(k^3) \\
=& k \left(\mathbf{P}\mathbf{P}^{-1} + \frac{k}{12}\mathbf{P}\mathbf{J}\mathbf{P}^{-1} \right) \left(\mathbf{P}\mathbf{P}^{-1} + \frac{k}{4}\mathbf{P}\mathbf{J}\mathbf{P}^{-1} \right)^{-1} \left(\mathbf{P}\mathbf{P}^{-1} + \frac{k}{3}\mathbf{P}\mathbf{J}\mathbf{P}^{-1} \right)^{-1} + \mathcal{O}(k^3) \\
=& k \left(\mathbf{I} + \frac{k}{12}\mathbf{A} \right) \left(\mathbf{I} + \frac{k}{4}\mathbf{A} \right)^{-1} \left(\mathbf{I} + \frac{k}{3}\mathbf{A} \right)^{-1} + \mathcal{O}(k^3)
\end{aligned}
$$

Note that the error constants still do not depend on the size of the matrix $\mathbf{A}$, thus are independent of n.

This concludes the proof. □

In the same way, the above approximation for the second integral holds as well with the same order.

Lemma 10. Let $\mathbf{A} \in \mathbb{R}^{n\times n}$ be a matrix such that all eigenvalues of $\mathbf{A}$ have non-negative real parts. Then,

$$k\int_0^1 \mathrm{e}^{-k\mathbf{A}(1-\tau)}\tau \mathrm{d}\tau = \frac{k}{2}\left(\mathbf{I}+\frac{k}{6}\mathbf{A}\right)\left(\mathbf{I}+\frac{k}{4}\mathbf{A}\right)^{-1}\left(\mathbf{I}+\frac{k}{3}\mathbf{A}\right)^{-1}+\mathcal{O}(k^2)$$

Proof. We perform this proof like the previous proof of Lemma 9 and consider the Jordan normal form $\mathbf{A} = \mathbf{PJP}^{-1}$, where

$$\mathbf{J} = \begin{pmatrix} \mathbf{J}_1 & & & \\ & \mathbf{J}_2 & & \\ & & \ddots & \\ & & & \mathbf{J}_p \end{pmatrix} \in \mathbb{C}^{n\times n}$$

with the Jordan blocks

$$\mathbf{J}_i = \begin{pmatrix} \lambda_i & 1 & & \\ & \lambda_i & \ddots & \\ & & \ddots & 1 \\ & & & \lambda_i \end{pmatrix} \in \mathbb{C}^{n_i\times n_i}, \qquad i = 1,\dots,p$$

We need only consider the Jordan block $\mathbf{J}_i$ for $\lambda_i = 0$. Explicitly computing the integral with the matrix exponential

$$\mathrm{e}^{-k\mathbf{J}_i(1-\tau)}\tau = \begin{pmatrix} \tau & \frac{(-k(1-\tau))^1\tau}{1!} & \cdots & \frac{(-k(1-\tau))^{n_i-1}\tau}{(n_i-1)!} \\ & \tau & \ddots & \vdots \\ & & \ddots & \frac{(-k(1-\tau))^1\tau}{1!} \\ & & & \tau \end{pmatrix} \in \mathbb{R}^{n_i\times n_i}$$

gives

$$k\int_0^1 \mathrm{e}^{-k\mathbf{J_i}(1-\tau)}\tau \mathrm{d}\tau = \begin{pmatrix} \frac{k}{2} & \frac{-k^2}{3!} & \cdots & \frac{(-1)^{n_i-1}k^n}{(n_i+1)!} \\ & \frac{k}{2} & \ddots & \vdots \\ & & \ddots & \frac{-k^2}{3!} \\ & & & \frac{k}{2} \end{pmatrix}$$

$$= \begin{pmatrix} \frac{k}{2} & \frac{-k^2}{3!} & & \\ & \frac{k}{2} & \ddots & \\ & & \ddots & \frac{-k^2}{3!} \\ & & & \frac{k}{2} \end{pmatrix} + \mathcal{O}(k^3)$$

$$= \begin{pmatrix} \frac{k}{2} & & & \\ & \frac{k}{2} & & \\ & & \ddots & \\ & & & \frac{k}{2} \end{pmatrix} + \mathcal{O}(k^2)$$

For the right hand side, we can use the same Neumann series approach as in (21) to obtain

$$\left(\mathbf{I}+\frac{k}{c}\mathbf{J}_i\right)^{-1} = \sum_{j=0}^{n_i}\left(-\frac{k}{c}\mathbf{J}_i\right)^j = \begin{pmatrix} 1 & (-\frac{k}{c})^1 & \cdots & (-\frac{k}{c})^{n_i-1} \\ & 1 & \ddots & \vdots \\ & & \ddots & (-\frac{k}{c})^1 \\ & & & 1 \end{pmatrix}$$

Thus,

$$\begin{aligned} &\frac{k}{2}\left(\mathbf{I}+\frac{k}{6}\mathbf{J}_i\right)\left(\mathbf{I}+\frac{k}{4}\mathbf{J}_i\right)^{-1}\left(\mathbf{I}+\frac{k}{3}\mathbf{J}_i\right)^{-1} \\ =&\frac{k}{2}\left(\mathbf{I}+\frac{k}{6}\mathbf{J}_i\right)\sum_{j=0}^{n_i}\sum_{\ell=0}^{n_i}\left(-\frac{k}{4}\mathbf{J}_i\right)^j\left(-\frac{k}{3}\mathbf{J}_i\right)^\ell \end{aligned}$$

$$
\begin{aligned}
=&\frac{k}{2}\left(\mathbf{I}+\frac{k}{6}\mathbf{J}_i\right)\sum_{j=0}^{n_i}\sum_{\ell=0}^{n_i}(-1)^{j+\ell}k^{j+\ell}\left(\frac{1}{4}\right)^j\left(\frac{1}{3}\right)^\ell\mathbf{J}_i^{j+\ell}\\
=&\frac{k}{2}\sum_{j=0}^{n_i}\sum_{\ell=0}^{n_i}(-1)^{j+\ell}k^{j+\ell}\left(\frac{1}{4}\right)^j\left(\frac{1}{3}\right)^\ell\mathbf{J}_i^{j+\ell}+\frac{k^2}{12}\sum_{j=0}^{n_i}\sum_{\ell=0}^{n_i}(-1)^{j+\ell}k^{j+\ell}\left(\frac{1}{4}\right)^j\left(\frac{1}{3}\right)^\ell\mathbf{J}_i^{j+\ell+1}\\
=&\frac{k}{2}\sum_{j=0}^{1}\sum_{\ell=0}^{1-j}(-1)^{j+\ell}k^{j+\ell}\left(\frac{1}{4}\right)^j\left(\frac{1}{3}\right)^\ell\mathbf{J}_i^{j+\ell}\\
&+\frac{k^2}{12}\sum_{j=0}^{0}\sum_{\ell=0}^{0}(-1)^{j+\ell}k^{j+\ell}\left(\frac{1}{4}\right)^j\left(\frac{1}{3}\right)^\ell\mathbf{J}_i^{j+\ell+1}+\mathcal{O}(k^3)\\
=&\frac{k}{2}\mathbf{I}-\frac{k^2}{6}\mathbf{J}_i-\frac{k^2}{8}\mathbf{J}_i+\frac{k^2}{12}\mathbf{J}_i+\mathcal{O}(k^3)\\
=&\begin{pmatrix}\frac{k}{2}&&&\\&\frac{k}{2}&&\\&&\ddots&\\&&&\frac{k}{2}\end{pmatrix}+\begin{pmatrix}0&-\frac{k^2}{6}&&\\&0&\ddots&\\&&\ddots&-\frac{k^2}{6}\\&&&0\end{pmatrix}+\begin{pmatrix}0&-\frac{k^2}{8}&&\\&0&\ddots&\\&&\ddots&-\frac{k^2}{8}\\&&&0\end{pmatrix}\\
&+\begin{pmatrix}0&\frac{k^2}{12}&&\\&0&\ddots&\\&&\ddots&\frac{k^2}{12}\\&&&0\end{pmatrix}+\mathcal{O}(k^3)\\
=&\begin{pmatrix}\frac{k}{2}&-\frac{5k^2}{24}&&\\&\frac{k}{2}&\ddots&\\&&\ddots&-\frac{5k^2}{24}\\&&&\frac{k}{2}\end{pmatrix}+\mathcal{O}(k^3)=\begin{pmatrix}\frac{k}{2}&&&\\&\frac{k}{2}&&\\&&\ddots&\\&&&\frac{k}{2}\end{pmatrix}+\mathcal{O}(k^2)
\end{aligned}
$$

Therefore, the desired approximation holds for Jordan block $\mathbf{J}_i$ corresponding to the eigenvalue $\lambda_i = 0$ of $\mathbf{A}$. As seen in the derivations for regular matrices above, it

also holds for all other possible Jordan blocks since they are by construction regular. Just like in the proof of Lemma 9, we can conclude that it holds for $\mathbf{J}$.

Since $\mathbf{B}\tau = \tau\mathbf{B}$, we do not show again that the approximation then holds for $\mathbf{A}$ as well. Instead, we refer to the corresponding section of the proof for Lemma 9. Also analogously to this proof, the error constants do not depend on the size of the matrix $\mathbf{A}$, thus are independent of n.

This concludes the proof. □

Note that similar expansions as we did here can be performed for regular matrices as well. These yield the same results. We do not perform these explicitly since we have shown the abstract derivation above (before Lemma 9).

We can now apply the approximations to the scheme of Equation (19) and obtain the following fully discrete scheme.

$$\begin{aligned}\mathbf{Z}_{n+1} = &\left(\mathbf{I} - \frac{5k}{12}\mathbf{M}_d\right)\left(\mathbf{I} + \frac{k}{4}\mathbf{M}_d\right)^{-1}\left(\mathbf{I} + \frac{k}{3}\mathbf{M}_d\right)^{-1}\mathbf{Z}_n \\ &+ k\left(\mathbf{I} + \frac{k}{12}\mathbf{M}_d\right)\left(\mathbf{I} + \frac{k}{4}\mathbf{M}_d\right)^{-1}\left(\mathbf{I} + \frac{k}{3}\mathbf{M}_d\right)^{-1}\mathbf{h}(\mathbf{Z}_n) \\ &+ \frac{k}{2}\left(\mathbf{I} + \frac{k}{6}\mathbf{M}_d\right)\left(\mathbf{I} + \frac{k}{4}\mathbf{M}_d\right)^{-1}\left(\mathbf{I} + \frac{k}{3}\mathbf{M}_d\right)^{-1}(\mathbf{h}(\tilde{\mathbf{Z}}_{n+1}) - \mathbf{h}(\mathbf{Z}_n))\end{aligned} \tag{22}$$

This can be simplified to

$$\begin{aligned}\mathbf{Z}_{n+1} = &\left(\mathbf{I} - \frac{5k}{12}\mathbf{M}_d\right)\left(\mathbf{I} + \frac{k}{4}\mathbf{M}_d\right)^{-1}\left(\mathbf{I} + \frac{k}{3}\mathbf{M}_d\right)^{-1}\mathbf{Z}_n \\ &+ \frac{k}{2}\left(\mathbf{I} + \frac{k}{4}\mathbf{M}_d\right)^{-1}\left(\mathbf{I} + \frac{k}{3}\mathbf{M}_d\right)^{-1}\mathbf{h}(\mathbf{Z}_n) \\ &+ \frac{k}{2}\left(\mathbf{I} + \frac{k}{6}\mathbf{M}_d\right)\left(\mathbf{I} + \frac{k}{4}\mathbf{M}_d\right)^{-1}\left(\mathbf{I} + \frac{k}{3}\mathbf{M}_d\right)^{-1}\mathbf{h}(\tilde{\mathbf{Z}}_{n+1})\end{aligned} \tag{23}$$

The predictor (Equation (18)) has a truncation error of $\mathcal{O}(k^2)$, so we can apply a second-order approximation of the exponential without loss of precision. We use

the [0/1] Padé approximation

$$\mathbf{R}_{01}(k\mathbf{A}) = (\mathbf{I} + k\mathbf{A})^{-1} = \mathrm{e}^{-k\mathbf{A}} + \mathcal{O}(k^2)$$

We then obtain in the case of a regular matrix $\mathbf{A}$

$$\begin{aligned}
&k \int_0^1 \mathrm{e}^{-k\mathbf{A}(1-\tau)} \mathrm{d}\tau \\
&= \mathbf{A}^{-1} \left(\mathbf{I} - \mathrm{e}^{-k\mathbf{A}}\right) \\
&= \mathbf{A}^{-1} \left(\mathbf{I} - \mathbf{R}_{01}(k\mathbf{A}) + \mathcal{O}(k^2)\right) \\
&= \mathbf{A}^{-1} \left(\mathbf{I} - \mathbf{R}_{01}(k\mathbf{A})\right) + \mathcal{O}(k^2) \\
&= \mathbf{A}^{-1} \left(\mathbf{I} - (\mathbf{I} + k\mathbf{A})^{-1}\right) + \mathcal{O}(k^2) \\
&= \mathbf{A}^{-1} \left((\mathbf{I} + k\mathbf{A})(\mathbf{I} + k\mathbf{A})^{-1} - (\mathbf{I} + k\mathbf{A})^{-1}\right) + \mathcal{O}(k^2) \\
&= \mathbf{A}^{-1} \left((\mathbf{I} + k\mathbf{A})^{-1} + k\mathbf{A}(\mathbf{I} + k\mathbf{A})^{-1} - (\mathbf{I} + k\mathbf{A})^{-1}\right) + \mathcal{O}(k^2) \\
&= \mathbf{A}^{-1} \left(k\mathbf{A}(\mathbf{I} + k\mathbf{A})^{-1}\right) + \mathcal{O}(k^2) \\
&= k(\mathbf{I} + k\mathbf{A})^{-1} + \mathcal{O}(k^2)
\end{aligned}$$

Lemma 11. Let $\mathbf{A} \in \mathbb{R}^{n \times n}$ be a matrix such that all eigenvalues of $\mathbf{A}$ have non-negative real parts. Then,

$$k \int_0^1 \mathrm{e}^{-k\mathbf{A}(1-\tau)} \mathrm{d}\tau = k(\mathbf{I} + k\mathbf{A})^{-1} + \mathcal{O}(k^2)$$

Proof. We consider, again, the Jordan normal form $\mathbf{A} = \mathbf{P}\mathbf{J}\mathbf{P}^{-1}$ with Jordan blocks $\mathbf{J}_i$. As previously, it is sufficient to consider the Jordan block $\mathbf{J}_i$ for the eigenvalue λ_i.

From the proof of Lemma 9, we know that for $\mathbf{J}_i$, the following holds:

$$k\int_0^1 \mathrm{e}^{-k\mathbf{J}_i(1-\tau)}\mathrm{d}\tau = \begin{pmatrix} k & -\frac{k^2}{2} & & \\ & k & \ddots & \\ & & \ddots & -\frac{k^2}{2} \\ & & & k \end{pmatrix} + \mathcal{O}(k^3).$$

and

$$k\left(\mathbf{I}+k\mathbf{J}_i\right)^{-1} = \begin{pmatrix} k & (-k)^2 & \dots & (-k)^{n_i} \\ & k & \ddots & \vdots \\ & & \ddots & (-k)^2 \\ & & & k \end{pmatrix}$$

So, clearly,

$$k\int_0^1 \mathrm{e}^{-k\mathbf{J}_i(1-\tau)}\mathrm{d}\tau = k\left(\mathbf{I}+k\mathbf{J}_i\right)^{-1} + \mathcal{O}(k^2)$$

We refer back to the proof of Lemma 9 for the conclusion that

$$k\int_0^1 \mathrm{e}^{-k\mathbf{A}(1-\tau)}\mathrm{d}\tau = k\left(\mathbf{I}+k\mathbf{A}\right)^{-1} + \mathcal{O}(k^2)$$

Note again that the error constants do not depend on the size of $\mathbf{A}$, thus are independent of n. □

The predictor $\tilde{\mathbf{Z}}_{n+1}$ is, therefore, computed as

$$\begin{aligned} \tilde{\mathbf{Z}}_{n+1} &= \left(\mathbf{I}+k\mathbf{M}_d\right)^{-1}\mathbf{Z}_n \\ &\quad + k\left(\mathbf{I}+k\mathbf{M}_d\right)^{-1}\mathbf{h}(\mathbf{Z}_n) \\ &= \left(\mathbf{I}+k\mathbf{M}_d\right)^{-1}\left(\mathbf{Z}_n + k\mathbf{h}(\mathbf{Z}_n)\right) \end{aligned} \tag{24}$$

2.2.4 Accuracy

Asante-Asamani (2016) proves that the scheme derived above is second-order accurate and stable. He is able to show this in the setting of strongly continuous semigroups, where $\mathbf{M}_d$ and other matrices can be more general operators. This allows him to consider the limit as $h \to 0$ while we are only able to consider fixed, finite h in order to obtain finite matrices. However, the proof requires the operator to be invertible, just like the derivation in Asante-Asamani et al. (2020).

Our derivation so far does not require the matrices to be invertible and we have found all relevant approximations to hold with the same order. To complete this, we, therefore, prove second-order accuracy of the time-stepping scheme for finite, but not necessarily invertible matrices, i. e., for a fixed, finite spatial grid. This serves as a first step in a potential attempt for a full proof.

In the following, assume that $\mathbf{Z}(t)$ is the exact solution of the initial value problem 14.

Lemma 12. The truncation error for the predictor $\tilde{\mathbf{Z}}_{n+1}$ is of second order in time, i. e., $\tilde{\mathbf{Z}}_{n+1} - \mathbf{Z}(t_{n+1}) = \mathcal{O}(k^2)$.

Proof. Let, as in Equation (24),

$$\tilde{\mathbf{Z}}_{n+1} = (\mathbf{I} + k\mathbf{M}_d)^{-1} (\mathbf{Z}_n + k\mathbf{h}(\mathbf{Z}_n))$$

We want to show that $\tilde{\mathbf{Z}}_{n+1} - \mathbf{Z}(t_{n+1}) = \mathcal{O}(k^2)$. Note that $\mathbf{Z}(t)$ satisfies Duhamel's principle, i. e., (15).

$$\mathbf{Z}(t) = \mathrm{e}^{-\mathbf{M}_d t}\mathbf{Z}(0) + \int_0^t \mathrm{e}^{-\mathbf{M}_d(t-s)}\mathbf{h}(\mathbf{Z}(s))ds\,.$$

From (18) and the approximations in Section 2.2.3, we know that

$$\begin{aligned}\mathbf{Z}(t_{n+1}) &= \mathrm{e}^{-\mathbf{M}_d k}\mathbf{Z}(t_n) + k\int_0^1 \mathrm{e}^{-k\mathbf{M}_d(1-\tau)}\mathbf{h}(\mathbf{Z}(t_n))\,\mathrm{d}\tau + \mathcal{O}(k^2)\\ &= (I + k\mathbf{M}_d)^{-1}\,\mathbf{Z}(t_n) + \mathcal{O}(k^2) + k\,(I + k\mathbf{M}_d)^{-1}\,\mathbf{h}(\mathbf{Z}(t_n)) + \mathcal{O}(k^2)\\ &= (\mathbf{I} + k\mathbf{M}_d)^{-1}\,(\mathbf{Z}_n + k\mathbf{h}(\mathbf{Z}_n)) + \mathcal{O}(k^2)\\ &= \tilde{\mathbf{Z}}_{n+1} + \mathcal{O}(k^2)\end{aligned}$$

This finishes the proof. □

Theorem 1. Given the predictor of (24), the method in (23) is second-order accurate in time, i. e., has a local truncation error of $\mathcal{O}(k^3)$. In other words,

$\mathbf{Z}_{n+1} - \mathbf{Z}(t_{n+1}) = \mathcal{O}(k^3)$

Proof. Let

$$\tilde{\mathbf{Z}}_{n+1} = (\mathbf{I} + k\mathbf{M}_d)^{-1}\,(\mathbf{Z}_n + k\mathbf{h}(\mathbf{Z}(t_n)))$$

and

$$\begin{aligned}\mathbf{Z}_{n+1} = &\left(\mathbf{I} - \frac{5k}{12}\mathbf{M}_d\right)\left(\mathbf{I} + \frac{k}{4}\mathbf{M}_d\right)^{-1}\left(\mathbf{I} + \frac{k}{3}\mathbf{M}_d\right)^{-1}\mathbf{Z}(t_n)\\ &+ \frac{k}{2}\left(\mathbf{I} + \frac{k}{4}\mathbf{M}_d\right)^{-1}\left(\mathbf{I} + \frac{k}{3}\mathbf{M}_d\right)^{-1}\mathbf{h}(\mathbf{Z}(t_n))\\ &+ \frac{k}{2}\left(\mathbf{I} + \frac{k}{6}\mathbf{M}_d\right)\left(\mathbf{I} + \frac{k}{4}\mathbf{M}_d\right)^{-1}\left(\mathbf{I} + \frac{k}{3}\mathbf{M}_d\right)^{-1}\mathbf{h}(\tilde{\mathbf{Z}}_{n+1})\end{aligned}$$

as in Equations (24) and (23)

We want to show that $\mathbf{Z}_{n+1} - \mathbf{Z}(t_{n+1}) = \mathcal{O}(k^3)$. Note that $\mathbf{Z}(t)$ satisfies Duhamel's principle, i. e., (15).

$$\mathbf{Z}(t) = \mathrm{e}^{-\mathbf{M}_d t}\mathbf{Z}(0) + \int_0^t \mathrm{e}^{-\mathbf{M}_d(t-s)}\mathbf{h}(\mathbf{Z}(s))\,\mathrm{d}s$$

By Equation (19) and the approximations from Section 2.2.3,

$$
\begin{aligned}
\mathbf{Z}(t_{n+1}) &= \mathrm{e}^{-\mathbf{M}_d k}\mathbf{Z}(t_n) + k\mathbf{h}(\mathbf{Z}(t_n)) \int_0^1 \mathrm{e}^{-k\mathbf{M}_d(1-\tau)}\,\mathrm{d}\tau \\
&\quad + k(\mathbf{h}(\tilde{\mathbf{Z}}_{n+1}) - \mathbf{h}(\mathbf{Z}(t_n))) \int_0^1 \mathrm{e}^{-k\mathbf{M}_d(1-\tau)}\tau\,\mathrm{d}\tau + \mathcal{O}(k^3) \\
&= \left(\mathbf{I} - \frac{5k}{12}\mathbf{M}_d\right)\left(\mathbf{I} + \frac{k}{4}\mathbf{M}_d\right)^{-1}\left(\mathbf{I} + \frac{k}{3}\mathbf{M}_d\right)^{-1}\mathbf{Z}(t_n) + \mathcal{O}(k^3) \\
&\quad + k\left(\mathbf{I} + \frac{k}{12}\mathbf{M}_d\right)\left(\mathbf{I} + \frac{k}{4}\mathbf{M}_d\right)^{-1}\left(\mathbf{I} + \frac{k}{3}\mathbf{M}_d\right)^{-1}\mathbf{h}(\mathbf{Z}(t_n)) + \mathcal{O}(k^3) \\
&\quad + \left(\frac{k}{2}\left(\mathbf{I} + \frac{k}{6}\mathbf{M}_d\right)\left(\mathbf{I} + \frac{k}{4}\mathbf{M}_d\right)^{-1}\left(\mathbf{I} + \frac{k}{3}\mathbf{M}_d\right)^{-1} + \mathcal{O}(k^2)\right)\left(\mathbf{h}(\tilde{\mathbf{Z}}_{n+1}) - \mathbf{h}(\mathbf{Z}(t_n))\right) \\
&= \left(\mathbf{I} - \frac{5k}{12}\mathbf{M}_d\right)\left(\mathbf{I} + \frac{k}{4}\mathbf{M}_d\right)^{-1}\left(\mathbf{I} + \frac{k}{3}\mathbf{M}_d\right)^{-1}\mathbf{Z}(t_n) \\
&\quad + k\left(\mathbf{I} + \frac{k}{12}\mathbf{M}_d\right)\left(\mathbf{I} + \frac{k}{4}\mathbf{M}_d\right)^{-1}\left(\mathbf{I} + \frac{k}{3}\mathbf{M}_d\right)^{-1}\mathbf{h}(\mathbf{Z}(t_n)) \\
&\quad + \frac{k}{2}\left(\mathbf{I} + \frac{k}{6}\mathbf{M}_d\right)\left(\mathbf{I} + \frac{k}{4}\mathbf{M}_d\right)^{-1}\left(\mathbf{I} + \frac{k}{3}\mathbf{M}_d\right)^{-1}\left(\mathbf{h}(\tilde{\mathbf{Z}}_{n+1}) - \mathbf{h}(\mathbf{Z}(t_n))\right) + \mathcal{O}(k^3) \\
&= \mathbf{Z}_{n+1} + \mathcal{O}(k^3)
\end{aligned}
$$

Note that the penultimate equality holds provided

$$\mathbf{h}(\tilde{\mathbf{Z}}_{n+1}) - \mathbf{h}(\mathbf{Z}(t_n)) = \mathcal{O}(k)$$

This is a kind of Lipschitz condition on $\mathbf{h}$ that we assume to be true since we are assuming $\mathbf{h}$ to be sufficiently smooth. □

Using Theorem 1, we can continue following the derivation of the scheme from Asante-Asamani et al. (2020) with the same order of accuracy.

2.2.5 Partial Fraction Decomposition

Like Asante-Asamani (2016) and Asante-Asamani et al. (2020), we can simplify Equation (23) by a partial fraction decomposition. Considering the scalar

variable z, and the rational function

$$\frac{1-\frac{5z}{12}}{(1+\frac{z}{4})(1+\frac{z}{3})},$$

we obtain the partial fraction decomposition

$$\frac{1-\frac{5z}{12}}{(1+\frac{z}{4})(1+\frac{z}{3})} = \frac{9}{1+\frac{z}{3}} - \frac{8}{1+\frac{z}{4}}.$$

Similarly,

$$\frac{1}{(1+\frac{z}{4})(1+\frac{z}{3})} = \frac{4}{1+\frac{z}{3}} - \frac{3}{1+\frac{z}{4}}$$

and

$$\frac{1+\frac{z}{6}}{(1+\frac{z}{4})(1+\frac{z}{3})} = \frac{2}{1+\frac{z}{3}} - \frac{1}{1+\frac{z}{4}}.$$

We can apply this analogously to the terms of the corrector. That this is possible follows, e. g., from Horn and Johnson (1991, chap. 6), Corollary 6.2.10 (e).

$$\begin{aligned}
\left(\mathbf{I}-\frac{5k}{12}\mathbf{M}_d\right)\left(\mathbf{I}+\frac{k}{4}\mathbf{M}_d\right)^{-1}\left(\mathbf{I}+\frac{k}{3}\mathbf{M}_d\right)^{-1} &= 9\left(\mathbf{I}+\frac{k}{3}\mathbf{M}_d\right)^{-1} - 8\left(\mathbf{I}+\frac{k}{4}\mathbf{M}_d\right)^{-1} \\
\left(\mathbf{I}+\frac{k}{4}\mathbf{M}_d\right)^{-1}\left(\mathbf{I}+\frac{k}{3}\mathbf{M}_d\right)^{-1} &= 4\left(\mathbf{I}+\frac{k}{3}\mathbf{M}_d\right)^{-1} - 3\left(\mathbf{I}+\frac{k}{4}\mathbf{M}_d\right)^{-1} \\
\left(\mathbf{I}+\frac{k}{6}\mathbf{M}_d\right)\left(\mathbf{I}+\frac{k}{4}\mathbf{M}_d\right)^{-1}\left(\mathbf{I}+\frac{k}{3}\mathbf{M}_d\right)^{-1} &= 2\left(\mathbf{I}+\frac{k}{3}\mathbf{M}_d\right)^{-1} - 1\left(\mathbf{I}+\frac{k}{4}\mathbf{M}_d\right)^{-1}
\end{aligned}$$

Applying this to Equation (23) yields

$$\begin{aligned}
\mathbf{Z}_{n+1} = &\left(9\left(\mathbf{I}+\frac{k}{3}\mathbf{M}_d\right)^{-1} - 8\left(\mathbf{I}+\frac{k}{4}\mathbf{M}_d\right)^{-1}\right)\mathbf{Z}_n \\
&+ \frac{k}{2}\left(4\left(\mathbf{I}+\frac{k}{3}\mathbf{M}_d\right)^{-1} - 3\left(\mathbf{I}+\frac{k}{4}\mathbf{M}_d\right)^{-1}\right)\mathbf{h}(\mathbf{Z}_n) \\
&+ \frac{k}{2}\left(2\left(\mathbf{I}+\frac{k}{3}\mathbf{M}_d\right)^{-1} - 1\left(\mathbf{I}+\frac{k}{4}\mathbf{M}_d\right)^{-1}\right)\mathbf{h}(\tilde{\mathbf{Z}}_{n+1}) \\
&= \left(\mathbf{I}+\frac{k}{3}\mathbf{M}_d\right)^{-1}\left(9\mathbf{Z}_n + 2k\mathbf{h}(\mathbf{Z}_n) + k\mathbf{h}(\tilde{\mathbf{Z}}_{n+1})\right)
\end{aligned}$$

$$- \left(\mathbf{I} + \frac{k}{4}\mathbf{M}_d\right)^{-1} \left(8\mathbf{Z}_n + \frac{3k}{2}\mathbf{h}(\mathbf{Z}_n) + \frac{k}{2}\mathbf{h}(\tilde{\mathbf{Z}}_{n+1})\right)$$

The full scheme is, therefore,

$$\begin{aligned}
\mathbf{Z}_{n+1} &= \left(\mathbf{I} + \frac{k}{3}\mathbf{M}_d\right)^{-1} \left(9\mathbf{Z}_n + 2k\mathbf{h}(\mathbf{Z}_n) + k\mathbf{h}(\tilde{\mathbf{Z}}_{n+1})\right) \\
&\quad - \left(\mathbf{I} + \frac{k}{4}\mathbf{M}_d\right)^{-1} \left(8\mathbf{Z}_n + \frac{3k}{2}\mathbf{h}(\mathbf{Z}_n) + \frac{k}{2}\mathbf{h}(\tilde{\mathbf{Z}}_{n+1})\right) \qquad (25) \\
\tilde{\mathbf{Z}}_{n+1} &= \left(\mathbf{I} + k\mathbf{M}_d\right)^{-1} \left(\mathbf{Z}_n + k\mathbf{h}(\mathbf{Z}_n)\right)
\end{aligned}$$

The advantage of this representation of the scheme is that it allows for parallelization (Asante-Asamani 2016; Asante-Asamani et al. 2020). Specifically, the systems

$$\left(\mathbf{I} + \frac{k}{3}\mathbf{M}_d\right)^{-1}$$

and

$$\left(\mathbf{I} + \frac{k}{4}\mathbf{M}_d\right)^{-1}$$

can be solved simultaneously instead of sequentially.

2.2.6 Unwinding the Dimensional Splitting

Notice now that

$$\mathbf{h}(\mathbf{Z}_n) = \mathrm{e}^{\sum_{i=1}^{d-1} \mathbf{M}_i t_n} \mathbf{f}(\mathrm{e}^{-\sum_{i=1}^{d-1} \mathbf{M}_i t_n} \mathbf{Z}_n)$$

depends on matrix exponentials as well. These were introduced by the dimensional splitting. We, therefore, unwind the dimensional splitting by reducing the scheme back to $\mathbf{U}$ and $\mathbf{f}$, the functions involved in the problem before the splitting.

Recall that, by construction,

$$\mathbf{Z}(t_n) = \mathrm{e}^{\sum_{i=1}^{d-1} \mathbf{M}_i t_n} \mathbf{U}(t_n).$$

So we set

$$\mathbf{Z}_n = \mathrm{e}^{\sum_{i=1}^{d-1} \mathbf{M}_i t_n} \mathbf{U}_n$$

and, analogously,

$$\mathbf{Z}_{n+1} = \mathrm{e}^{\sum_{i=1}^{d-1} \mathbf{M}_i t_{n+1}} \mathbf{U}_{n+1}$$

and

$$\tilde{\mathbf{Z}}_{n+1} = \mathrm{e}^{\sum_{i=1}^{d-1} \mathbf{M}_i t_{n+1}} \tilde{\mathbf{U}}_{n+1}$$

Therefore,

$$\mathbf{h}(\mathbf{Z}_n) = \mathrm{e}^{\sum_{i=1}^{d-1} \mathbf{M}_i t_n} \mathbf{f}(\mathbf{U}_n).$$

Inserting this into our full scheme (25) yields

$$\begin{aligned}
\mathrm{e}^{\sum_{i=1}^{d-1} \mathbf{M}_i t_{n+1}} \mathbf{U}_{n+1} &= \left(\mathbf{I} + \frac{k}{3}\mathbf{M}_d\right)^{-1} \mathrm{e}^{\sum_{i=1}^{d-1} \mathbf{M}_i t_n} \left(9\mathbf{U}_n + 2k\mathbf{f}(\mathbf{U}_n) + k\mathbf{f}(\tilde{\mathbf{U}}_{n+1})\right) \\
&\quad - \left(\mathbf{I} + \frac{k}{4}\mathbf{M}_d\right)^{-1} \mathrm{e}^{\sum_{i=1}^{d-1} \mathbf{M}_i t_n} \left(8\mathbf{U}_n + \frac{3k}{2}\mathbf{f}(\mathbf{U}_n) + \frac{k}{2}\mathbf{f}(\tilde{\mathbf{U}}_{n+1})\right) \\
\mathrm{e}^{\sum_{i=1}^{d-1} \mathbf{M}_i t_{n+1}} \tilde{\mathbf{U}}_{n+1} &= (\mathbf{I} + k\mathbf{M}_d)^{-1} \mathrm{e}^{\sum_{i=1}^{d-1} \mathbf{M}_i t_n} (\mathbf{U}_n + k\mathbf{f}(\mathbf{U}_n))
\end{aligned}$$

We can rewrite

$$\mathrm{e}^{\sum_{i=1}^{d-1} \mathbf{M}_i t_n} = \mathrm{e}^{\sum_{i=1}^{d-1} \mathbf{M}_i nk}.$$

Thus, by Lemma 2,

$$\mathrm{e}^{\sum_{i=1}^{d-1} \mathbf{M}_i t_{n+1}} = \mathrm{e}^{\sum_{i=1}^{d-1} \mathbf{M}_i (nk+k)} = \mathrm{e}^{\sum_{i=1}^{d-1} \mathbf{M}_i nk} \cdot \mathrm{e}^{\sum_{i=1}^{d-1} \mathbf{M}_i k}.$$

Then, we multiply

$$\mathrm{e}^{-\sum_{i=1}^{d-1} \mathbf{M}_i (nk+k)}$$

on both sides to obtain, by Lemmas 3 and 1,

$$\begin{aligned}\mathbf{U}_{n+1} &= \left(\mathbf{I}+\frac{k}{3}\mathbf{M}_d\right)^{-1} \mathrm{e}^{-\sum_{i=1}^{d-1}\mathbf{M}_i k}\left(9\mathbf{U}_n + 2k\mathbf{f}(\mathbf{U}_n) + k\mathbf{f}(\tilde{\mathbf{U}}_{n+1})\right)\\ &\quad - \left(\mathbf{I}+\frac{k}{4}\mathbf{M}_d\right)^{-1} \mathrm{e}^{-\sum_{i=1}^{d-1}\mathbf{M}_i k}\left(8\mathbf{U}_n + \frac{3k}{2}\mathbf{f}(\mathbf{U}_n) + \frac{k}{2}\mathbf{f}(\tilde{\mathbf{U}}_{n+1})\right)\\ \tilde{\mathbf{U}}_{n+1} &= (\mathbf{I}+k\mathbf{M}_d)^{-1}\,\mathrm{e}^{-\sum_{i=1}^{d-1}\mathbf{M}_i k}(\mathbf{U}_n + k\mathbf{f}(\mathbf{U}_n))\end{aligned}$$

Note that we also used the fact that if $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n\times n}$ and $\mathbf{A}$ and $\mathbf{B}$ commute, then also $\mathbf{A}^{-1}$ and $\mathbf{B}$ commute.

The final step is now to approximate the matrix exponential again and simplify. In the predictor, we use the $\mathbf{R}_{01}$ approximation and in the corrector, we use $\mathbf{R}_{RDP}$. This yields, due to the commutativity of the $\mathbf{M}_i, i = 1, \ldots, d$,

$$\begin{aligned}\mathbf{U}_{n+1} &= \left(\mathbf{I}+\frac{k}{3}\mathbf{M}_d\right)^{-1} \mathrm{e}^{-\mathbf{M}_1 k}\cdot\ldots\cdot\mathrm{e}^{-\mathbf{M}_{d-1}k}\left(9\mathbf{U}_n + 2k\mathbf{f}(\mathbf{U}_n) + k\mathbf{f}(\tilde{\mathbf{U}}_{n+1})\right)\\ &\quad - \left(\mathbf{I}+\frac{k}{4}\mathbf{M}_d\right)^{-1} \mathrm{e}^{-\mathbf{M}_1 k}\cdot\ldots\cdot\mathrm{e}^{-\mathbf{M}_{d-1}k}\left(8\mathbf{U}_n + \frac{3k}{2}\mathbf{f}(\mathbf{U}_n) + \frac{k}{2}\mathbf{f}(\tilde{\mathbf{U}}_{n+1})\right)\\ &= \left(\mathbf{I}+\frac{k}{3}\mathbf{M}_d\right)^{-1}\left(9\left(\mathbf{I}+\frac{k}{3}\mathbf{M}_1\right)^{-1} - 8\left(\mathbf{I}+\frac{k}{4}\mathbf{M}_1\right)^{-1}\right)\cdot\ldots\\ &\quad\cdot\left(9\left(\mathbf{I}+\frac{k}{3}\mathbf{M}_{d-1}\right)^{-1} - 8\left(\mathbf{I}+\frac{k}{4}\mathbf{M}_{d-1}\right)^{-1}\right)\left(9\mathbf{U}_n + 2k\mathbf{f}(\mathbf{U}_n) + k\mathbf{f}(\tilde{\mathbf{U}}_{n+1})\right)\\ &\quad - \left(\mathbf{I}+\frac{k}{4}\mathbf{M}_d\right)^{-1}\left(9\left(\mathbf{I}+\frac{k}{3}\mathbf{M}_1\right)^{-1} - 8\left(\mathbf{I}+\frac{k}{4}\mathbf{M}_1\right)^{-1}\right)\cdot\ldots\\ &\quad\cdot\left(9\left(\mathbf{I}+\frac{k}{3}\mathbf{M}_{d-1}\right)^{-1} - 8\left(\mathbf{I}+\frac{k}{4}\mathbf{M}_{d-1}\right)^{-1}\right)\left(8\mathbf{U}_n + \frac{3k}{2}\mathbf{f}(\mathbf{U}_n) + \frac{k}{2}\mathbf{f}(\tilde{\mathbf{U}}_{n+1})\right)\end{aligned}$$

and

$$\begin{aligned}\tilde{\mathbf{U}}_{n+1} &= (\mathbf{I}+k\mathbf{M}_d)^{-1}\,\mathrm{e}^{-\mathbf{M}_1 k}\cdot\ldots\cdot\mathrm{e}^{-\mathbf{M}_{d-1}k}(\mathbf{U}_n + k\mathbf{f}(\mathbf{U}_n))\\ &= (\mathbf{I}+k\mathbf{M}_d)^{-1}(\mathbf{I}+k\mathbf{M}_1)^{-1}\cdot\ldots\cdot(\mathbf{I}+k\mathbf{M}_{d-1})^{-1}(\mathbf{U}_n + k\mathbf{f}(\mathbf{U}_n))\\ &= (\mathbf{I}+k\mathbf{M}_d)^{-1}\cdot\ldots\cdot(\mathbf{I}+k\mathbf{M}_1)^{-1}(\mathbf{U}_n + k\mathbf{f}(\mathbf{U}_n))\end{aligned}$$

This concludes the derivation of the final scheme.

2.2.7 Accuracy of the Final Scheme

What is left now is to prove accuracy for the final scheme.

Lemma 13. The truncation error for $\tilde{\mathbf{U}}_{n+1}$ is of second order, i. e., $\tilde{\mathbf{U}}_{n+1} - \mathbf{U}(t_{n+1}) = \mathcal{O}(k^2)$.

Proof. By Lemma 12, the truncation error of $\tilde{\mathbf{Z}}_{n+1}$ is of second order.

We assume that $\mathbf{U}(t)$ and $\mathbf{Z}(t)$ are the exact solutions of the respective problems. Since all transformations are exact, and

$$\mathrm{e}^{-\sum_{i=1}^{d-1} \mathbf{M}_i (nk+k)}$$

is bounded in k (recall that the eigenvalues of M_i are non-negative for all $i = 1, \ldots, d$), we know immediately that

$$\tilde{\mathbf{U}}_{n+1} = (\mathbf{I} + k\mathbf{M}_d)^{-1} \, \mathrm{e}^{-\sum_{i=1}^{d-1} \mathbf{M}_i k} (\mathbf{U}_n + k\mathbf{f}(\mathbf{U}_n)) + \mathcal{O}(k^2)$$

Knowing that

$$\mathrm{e}^{-\mathbf{M}_i k} = (\mathbf{I} + k\mathbf{M}_i)^{-1} + \mathcal{O}(k^2) \qquad \forall i = 1, \ldots, d,$$

we obtain inductively

$$\begin{aligned}
&\mathrm{e}^{-\mathbf{M}_{d-1}k} \cdot \mathrm{e}^{-\mathbf{M}_{d-2}k} \cdot \ldots \cdot \mathrm{e}^{-\mathbf{M}_1 k} \\
&= \left((\mathbf{I} + k\mathbf{M}_{d-1})^{-1} + \mathcal{O}(k^2)\right) \left((\mathbf{I} + k\mathbf{M}_{d-2})^{-1} + \mathcal{O}(k^2)\right) \cdot \mathrm{e}^{-\mathbf{M}_{d-3}k} \cdot \ldots \cdot \mathrm{e}^{-\mathbf{M}_1 k} \\
&= \left((\mathbf{I} + k\mathbf{M}_{d-1})^{-1} (\mathbf{I} + k\mathbf{M}_{d-2})^{-1} + \mathcal{O}(k^2)\right) \cdot \mathrm{e}^{-\mathbf{M}_{d-3}k} \cdot \ldots \cdot \mathrm{e}^{-\mathbf{M}_1 k} \\
&= (\mathbf{I} + k\mathbf{M}_{d-1})^{-1} (\mathbf{I} + k\mathbf{M}_{d-2})^{-1} \cdot \mathrm{e}^{-\mathbf{M}_{d-3}k} \cdot \ldots \cdot \mathrm{e}^{-\mathbf{M}_1 k} + \mathcal{O}(k^2) \\
&= \ldots
\end{aligned}$$

$$= (\mathbf{I} + k\mathbf{M}_{d-1})^{-1} \cdot \ldots \cdot (\mathbf{I} + k\mathbf{M}_1)^{-1} + \mathcal{O}(k^2)$$

So, finally,

$$\begin{aligned}
\tilde{\mathbf{U}}_{n+1} &= (\mathbf{I} + k\mathbf{M}_d)^{-1} \, \mathrm{e}^{-\sum_{i=1}^{d-1} \mathbf{M}_i k} (\mathbf{U}_n + k\mathbf{f}(\mathbf{U}_n)) + \mathcal{O}(k^2) \\
&= (\mathbf{I} + k\mathbf{M}_d)^{-1} \left((\mathbf{I} + k\mathbf{M}_{d-1})^{-1} \cdot \ldots \cdot (\mathbf{I} + k\mathbf{M}_1)^{-1} + \mathcal{O}(k^2) \right) (\mathbf{U}_n + k\mathbf{f}(\mathbf{U}_n)) + \mathcal{O}(k^2) \\
&= (\mathbf{I} + k\mathbf{M}_d)^{-1} \cdot \ldots \cdot (\mathbf{I} + k\mathbf{M}_1)^{-1} (\mathbf{U}_n + k\mathbf{f}(\mathbf{U}_n)) + \mathcal{O}(k^2)
\end{aligned}$$

□

Finally, we can prove the following theorem:

Theorem 2. The full scheme is second-order accurate in time, i. e., if $\mathbf{U}(t)$ is the exact solution of (3), then

$$\mathbf{U}_{n+1} - \mathbf{U}(t_{n+1}) = \mathcal{O}(k^3)$$

Proof. Since all transformations are exact, and

$$\mathrm{e}^{-\sum_{i=1}^{d-1} \mathbf{M}_i (nk+k)}$$

is bounded in k, we know immediately that

$$\begin{aligned}
\mathbf{U}_{n+1} = &\left(\mathbf{I} + \frac{k}{3}\mathbf{M}_d \right)^{-1} \mathrm{e}^{-\mathbf{M}_1 k} \cdot \ldots \cdot \mathrm{e}^{-\mathbf{M}_{d-1} k} \left(9\mathbf{U}(t_n) + 2k\mathbf{f}(\mathbf{U}(t_n)) + k\mathbf{f}(\tilde{\mathbf{U}}_{n+1}) \right) \\
&- \left(\mathbf{I} + \frac{k}{4}\mathbf{M}_d \right)^{-1} \mathrm{e}^{-\mathbf{M}_1 k} \cdot \ldots \cdot \mathrm{e}^{-\mathbf{M}_{d-1} k} \left(8\mathbf{U}(t_n) + \frac{3k}{2}\mathbf{f}(\mathbf{U}(t_n)) + \frac{k}{2}\mathbf{f}(\tilde{\mathbf{U}}_{n+1}) \right) + \mathcal{O}(k^3)
\end{aligned}$$

Since

$$\mathrm{e}^{-\mathbf{M}_i k} = \left(9 \left(\mathbf{I} + \frac{k}{3}\mathbf{M}_i \right)^{-1} - 8 \left(\mathbf{I} + \frac{k}{4}\mathbf{M}_i \right)^{-1} \right) + \mathcal{O}(k^3) \qquad \forall i = 1, \ldots, d,$$

we can conclude analogously to the proof of Lemma 13 that

$$\begin{aligned}
\mathbf{U}(t_{n+1}) = &\left(\mathbf{I}+\frac{k}{3}\mathbf{M}_d\right)^{-1}\left(9\left(\mathbf{I}+\frac{k}{3}\mathbf{M}_1\right)^{-1}-8\left(\mathbf{I}+\frac{k}{4}\mathbf{M}_1\right)^{-1}\right)\cdot\ldots\\
&\cdot\left(9\left(\mathbf{I}+\frac{k}{3}\mathbf{M}_{d-1}\right)^{-1}-8\left(\mathbf{I}+\frac{k}{4}\mathbf{M}_{d-1}\right)^{-1}\right)\left(9\mathbf{U}(t_n)+2k\mathbf{f}(\mathbf{U}(t_n))+k\mathbf{f}(\tilde{\mathbf{U}}_{n+1})\right)\\
&-\left(\mathbf{I}+\frac{k}{4}\mathbf{M}_d\right)^{-1}\left(9\left(\mathbf{I}+\frac{k}{3}\mathbf{M}_1\right)^{-1}-8\left(\mathbf{I}+\frac{k}{4}\mathbf{M}_1\right)^{-1}\right)\cdot\ldots\\
&\cdot\left(9\left(\mathbf{I}+\frac{k}{3}\mathbf{M}_{d-1}\right)^{-1}-8\left(\mathbf{I}+\frac{k}{4}\mathbf{M}_{d-1}\right)^{-1}\right)\left(8\mathbf{U}(t_n)+\frac{3k}{2}\mathbf{f}(\mathbf{U}(t_n))+\frac{k}{2}\mathbf{f}(\tilde{\mathbf{U}}_{n+1})\right)\\
&+\mathcal{O}(k^3)
\end{aligned}$$

□

The last theorem shows that the method is consistent and second-order accurate in time for a fixed spatial grid.

2.2.8 Numerical Implementation

For details on the implementation, we refer to Asante-Asamani et al. (2020). They first describe the parallelizations that are possible by introducing the partial fraction decomposition, giving detailed flow charts for parallelized implementations in the two- and three-dimensional cases. Besides that, they describe optimizations that use the specific band structure of the finite difference matrices. Namely, the matrices $\mathbf{M}_i, i = 1, \ldots, d$ only have three non-zero bands, one of which is the main diagonal. This suggests that variants of the Thomas algorithm can be used. Since the two outer bands are not necessarily adjacent to the main diagonal, it needs to be adapted. As Asante-Asamani et al. (2020) show, this is possible in the case of the vanishing normal derivative and Dirichlet boundary conditions.

For periodic boundary conditions, the structure is different making this approach impossible.

In this work, in order to allow for the most generality, we do not use any optimizations and instead call a standard LU decomposition algorithm that is not specifically optimized.

The source code can be found at `github.com/muellerbjoern/ETDRDPIF`.

2.3 Krylov-EETD

Another ETD scheme is introduced in Bhatt et al. (2018). It uses an extrapolation technique based on a first-order ETD scheme. Then, the arising matrix exponentials are approximated using a Krylov-subspace technique. We performed a comparison of the ETD-RDP-IF scheme to this Krylov-EETD scheme below. Therefore, we describe the derivation of this scheme here. Note that we do not focus on theoretical results such as error bounds or stability.

The starting point in Bhatt et al. (2018) is an ADR system like (1). Since Krylov-EETD also uses a method-of-lines approach, the first step is to discretize the problem to the form (3). This is done by a central finite difference scheme, analogous to Section 2.1.1.

First-Order ETD Scheme. Applying Duhamel's principle, they obtain a recurrence relation analogous to (16)

$$\mathbf{U}(t_{n+1}) = \mathrm{e}^{-\mathbf{M}k}\mathbf{U}(t_n) + k\int_0^1 \mathrm{e}^{-k\mathbf{M}(1-\tau)}\mathbf{f}(\mathbf{U}(t_n + k\tau))\,\mathrm{d}\tau$$

A first-order approximation of the integral, assuming

$\mathbf{f}(\mathbf{U}(t)) = \mathbf{f}(\mathbf{U}(t_n))\ \forall t \in [t_n; t_{n+1}]$, gives

$$\mathbf{U}(t_{n+1}) \approx \mathrm{e}^{-\mathbf{M}k}\mathbf{U}(t_n) + k\int_0^1 \mathrm{e}^{-k\mathbf{M}(1-\tau)}\mathbf{f}(\mathbf{U}(t_n))\,\mathrm{d}\tau$$

Then, Bhatt et al. (2018) evaluate the integral, assuming that $\mathbf{M}$ is invertible, i. e., as in Lemma 7. Denoting the approximation of $\mathbf{U}(t_n)$ by $\mathbf{U}_n$, they obtain the first-order scheme

$$\mathbf{U}_{n+1} = \mathrm{e}^{-k\mathbf{M}}\mathbf{U}_n + \mathbf{M}^{-1}\left(\mathbf{I} - \mathrm{e}^{-k\mathbf{M}}\right)\mathbf{f}(\mathbf{U}_n)$$

This is a first-order accurate ETD scheme.

Since the scheme is first-order accurate, the local truncation error is $\mathcal{O}(k^2)$. Hence, introducing another error of $\mathcal{O}(k^2)$ does not change the asymptotics. Therefore, Bhatt et al. (2018) rewrite the scheme as

$$\mathbf{U}_{n+1} = \mathrm{e}^{-k\mathbf{M}}\left(\mathbf{U}_n + k\mathbf{f}(\mathbf{U}_n)\right) + \mathcal{O}(k^2)$$

Extrapolation. In order to obtain a second-order scheme, the authors perform an extrapolation. Taking a single time step of $2k$, an approximation $\mathbf{U}^{(2)}_{n+2}$ is obtained as

$$\mathbf{U}^{(2)}_{n+2} = \mathrm{e}^{-2k\mathbf{M}}\left(\mathbf{U}_n + 2k\mathbf{f}(\mathbf{U}_n)\right) + \mathcal{O}(k^2).$$

Instead taking 2 steps of k, they obtain

$$\mathbf{U}^{(1)}_{n+2} = \mathrm{e}^{-k\mathbf{M}}\left(\mathbf{U}_{n+1} + k\mathbf{f}(\mathbf{U}_{n+1})\right) + \mathcal{O}(k^2),$$

where $\mathbf{U}_{n+1}$ is computed as in the original scheme.

Having performed a Taylor expansion of both schemes up to k^2, Bhatt et al. (2018) report that the extrapolation

$$\mathbf{U}_{n+2} = 2\mathbf{U}^{(1))}_{n+2} - \mathbf{U}^{(2))}_{n+2}$$

yields a second-order accurate scheme which they call EETD (short for extrapolated ETD).

Krylov-Subspace Approximation. Computing the matrix exponentials in the above scheme is a challenge. The matrix $\mathbf{M}$ is large and sparse. However, the exponential will generally be a dense matrix. Is is, therefore, desirable to not compute the full matrix exponential (Moler and Van Loan 2003). We can see from the scheme above that we only need to compute the action of the matrix exponential on some vectors.

Suppose, now, we intend to compute $\mathrm{e}^{-k\mathbf{A}}\mathbf{v}$ for some square matrix $\mathbf{A}$ and vector $\mathbf{v}$. We follow the description of the Krylov subspace approximation given in Bhatt et al. (2018). A Krylov-subspace approximation considers the M-dimensional subspace

$$K = \mathrm{span}(\{\mathbf{v}, \mathbf{A}\mathbf{v}, \mathbf{A}^2\mathbf{v}, \ldots, \mathbf{A}^{M-1}\mathbf{v}\})$$

The so-called Arnoldi algorithm (Saad 2011, chap. 6) computes an orthonormal basis of this subspace, say $\{\mathbf{v}_1, \ldots, \mathbf{v}_M\}$, and a projection matrix $\mathbf{H}$ of $\mathbf{A}$ onto K. $\mathbf{H}$ is an $M \times M$ Hessenberg matrix.

Then, $\mathrm{e}^{-k\mathbf{A}}\mathbf{v}$ can be approximated by

$$\mathrm{e}^{-k\mathbf{A}}\mathbf{v} \approx \gamma\mathbf{V}\mathrm{e}^{-k\mathbf{H}}\mathrm{e}_1$$

where $\gamma = ||\mathbf{v}||_2$ and $\mathbf{V} = (\mathbf{v}_1, \ldots, \mathbf{v}_M)$. e_1 is the first canonical unit vector in $\mathbb{C}^M$.

Due to the low dimensionality, the computation of $\mathrm{e}^{-k\mathbf{H}}$ can be done using standard methods for dense matrices. Bhatt et al. (2018) suggest a scaling-and-squaring approach which is also used by the `expm` function in Matlab and

`scipy.linalg.expm` for Python. For details on this, see Moler and Van Loan (2003), Higham (2005), and Al-Mohy and Higham (2010). For the numerical experiments in Chapter 3, we chose a Krylov-subspace dimension of $M = 10$.

Expokit Implementation. The function `expv` in the Matlab package Expokit (Sidje 1998) provides an implementation of a Krylov-subspace approximation of $\mathrm{e}^{-k\mathbf{A}}\mathbf{v}$ for arbitrary matrices $\mathbf{A}$ and vectors $\mathbf{v}$, and small positive constants k. It uses an adaptive version of the Arnoldi algorithm and the procedure described above.

We discuss shortly the adaptive code employed by Expokit. The adaptivity is based on the fact that

$$\mathrm{e}^{-At} = \mathrm{e}^{-At_1} \cdot \mathrm{e}^{-A(t-t_1)} \qquad \forall 0 \leq t_1 \leq t.$$

This is the semi-group property of the exponential. Therefore, instead of computing $\mathbf{r} = \mathrm{e}^{-k\mathbf{A}}\mathbf{v}$ directly, it is possible to compute $\mathbf{w} = \mathrm{e}^{-t_1\mathbf{A}}\mathbf{v}$ and $\mathbf{r} = \mathrm{e}^{-(k-t_1)\mathbf{A}}\mathbf{w}$. Clearly, each of these steps can be broken down further.

Hence, the adaptive code in Expokit performs a time-stepping-type iteration to compute the final result. The time step is determined based on local error criteria. Algorithm 1 shows a pseudo-code representation of the adaptive code.

In our numerical experiments in Chapter 3, we compare ETD-RDP-IF to the Krylov-EETD scheme. We consider two different implementations of the Krylov-subspace approximation of the matrix exponential. One of these is the one described above, as implemented in Expokit. Besides that, we consider what we call a “non-adaptive” version. In this version, we removed the time-stepping from the Expokit code. Hence, only a single step is performed, and no error checking is

```
time = 0
determine initial time step t_step
while(time < k)
        Arnoldi process to compute H and V
        while (not rejected too often)
                // perform a step with t_step:
                v = expm(–H*t_step)v
                if (error >= tol)
                        reject the step
                        reduce t_step
        time = time + t_step
        adapt t_step
```

Algorithm 1. Pseudo-code representation of the Expokit implementation of `expv`

performed. This reduces the code to performing the Arnoldi process once and computing the final result based on this.

As we will show, we considered this non-adaptive implementation since the adaptivity caused excessively small time steps in the examples we tested, resulting in computation times becoming infeasible. Note that the scaling-and-squaring that is used internally to compute the exponential of $\mathbf{H}$ is still an adaptive algorithm. However, this did not cause any problems.

3 Numerical Experiments

In order to investigate the performance of the ETD-RDP-IF method for solving ADR problems, we performed some numerical experiments comparing the method to other ETD schemes. Most notably, we did a comparison with the Krylov-EETD scheme proposed in Bhatt et al. (2018).

3.1 Quantities Concerning Advection

The main focus of this work is to investigate the ETD-RDP-IF scheme in the presence of advection. There are two main quantifications of advection that need to be considered in the context of numerical schemes. We describe these below.

3.1.1 Cell Péclet Number

As described in Section 1.3, the Péclet number describes which of the physical phenomena advection or diffusion dominates on Ω. For the behavior of the numerical scheme, the grid size h plays an important role. We therefore consider the Péclet number on the scale of the grid size, choosing h as the characteristic length, and define the cell Péclet number

$$\mathbf{P}_c = \frac{|a|h}{d},$$

where a and d are the advection velocity and diffusion constant as in Equation (2) (Strikwerda 2004).

As described by Shampine (1994) and Voss and Khaliq (1996), the eigenvalues of the discretization matrix $\mathbf{M}$ depend the cell Péclet number. They give

explicit formulas for these eigenvalues in the case of second-order central differencing. In particular, they show that the eigenvalues are real for $\mathbf{P}_c \leq 2$ and complex for $\mathbf{P}_c > 2$ in the case of homogeneous Dirichlet boundary conditions. The angle (in the complex plane) of the eigenvalues then increases as $\mathbf{P}_c$ increases. Shampine (1994) derives similar statements for other boundary conditions as well.

Since $\mathbf{M}$ is the system matrix of an ODE system, imaginary eigenvalues introduce oscillatory solutions even if the ODE were solved exactly. According to Strikwerda (2004), these oscillations do not increase without bound (while they consider a specific scheme, the result is applicable more generally). However, the oscillations result from an insufficiently fine grid, i. e., too low resolution, and are therefore considered spurious.

In summary, there is a change in the behavior of the solution at a certain threshold value of $\mathbf{P}_c$. Versteeg and Malalasekera (2007) show that central difference schemes produce non-optimal results for large cell Péclet numbers. However, upwind differencing schemes, which we also consider, do not exhibit this. Hundsdorfer and Verwer (2003) give more extensive results on this as they consider the behavior of various spatial and temporal discretization schemes given different cell Péclet numbers.

In the following, we do not explicitly consider the cell Péclet number in many cases. However, by varying the advection constant a, we observe both low-Péclet and high-Péclet regimes. Note that often we change the spatial grid during an experiment, that makes it more difficult to interpret in terms of cell Péclet number.

However, this enables us to interpret our results in terms of properties of the original problem, namely the advection velocity a.

3.1.2 CFL Condition

In advection-heavy problems, e.g., in Computational Fluid Dynamics, an important consideration is the Courant-Friedrichs-Lewy (CFL) condition (Hundsdorfer and Verwer 2003).

The CFL condition was first described by Courant, Friedrichs and Lewy in their 1928 paper Courant et al. (1928) (see Courant et al. (1967) for a translation) where they primarily considered the wave equation in one spatial dimension. Using the method of characteristics, they first derived a domain of dependence of the solution at each point. Namely, the solution at (x^*, t^*) depends on the triangle spanned by the characteristic lines $x - vt = x^* - vt^*$ and $x + vt = x^* - vt^*$ for $t < t^*$, where v denotes the wave propagation speed. These characteristics pass through (x^*, t^*).

Given a specific fully discrete (explicit) scheme to solve the PDE, i. e., a difference equation that approximates the PDE, they obtained an analogous result describing the domain of dependence of each point. Specifically, they found that this domain of dependence includes the domain of dependence of the original PDE only under a certain constraint, namely when

$$C = \frac{vk}{h} \leq 1\,.$$

This is the CFL condition. We call C the Courant number or CFL number.

If it is not satisfied, the domain of dependence of the discrete scheme is smaller than that of the original PDE. This means that some changes of the initial condition influence the solution of the PDE at a specific point, but do not influence the solution of the numerical scheme (the difference equation) at the same point. Courant et al. then considered letting h and k approach 0 while keeping their ratio k/h, and hence the Courant number C, fixed. If the CFL condition is not satisfied, Courant et al. show that the solution of the scheme cannot generally converge to the solution of the PDE. Therefore, the CFL condition is a necessary condition for the convergence of an explicit scheme. Specifically, it is a necessary condition for stability (Hundsdorfer and Verwer 2003).

Since their original work, the condition has been refined and applied to various schemes. Different bounds on the Courant number have been described for different schemes (Hundsdorfer and Verwer 2003).

It can therefore be stated, adapted to advection-diffusion-reaction equations where a, the advection velocity, describes the propagation, in the following way:

$$C = \frac{ak}{h} \leq C_{max}$$

where C_{max} depends on the scheme. Note that this formulation holds for a one-dimensional problem, it may however be adapted for a multi-dimensional problem (see Courant et al. (1928)).

In implicit schemes, each time step involves solving a system of linear equations involving every point in space. Hence, the solution at each time step

depends on all points of the previous time step. I. e., the domain of dependence contains the full spatial domain for all $t < t^*$. Since this is independent of k and h, it suggests that the CFL condition is not necessary for implicit schemes (LeVeque 2007).

In this work, we have chosen to treat the advection term like the diffusion term, i. e., as a linear term to which we applied the exponential in the form of the RDP approximation. Applying the RDP approximation also involves solving a linear system of equations that includes the discretization matrix of the advection term. Hence, advection is in this sense treated implicitly. We may therefore expect that no CFL-type restrictions apply for our ETD-RDP-IF scheme. We investigate this statement below.

3.2 Pure Advection Equation

While the ETD-RDP scheme presented here is suitable for more general PDEs and a lot of considerations went into the treatment of possible nonlinearities, this work focuses on the treatment of the advection term. In order to judge the performance of the scheme in the presence of advection, we first consider a pure advection equation

$$u_t + a u_x = 0 \tag{26}$$

with initial condition $u_0(x)$. On an infinite domain, by the method of characteristics (Evans 2010), the solution to this is

$$u(x,t) = u_0(x - at).$$

This means that the initial condition is merely shifted. From a modeling perspective,

this is exactly what the advection term is used for — to describe how the solution is transported or “shifted”.

We assume periodic boundary conditions for our initial analysis. This allows for an infinite shift of the initial condition since the part that is shifted out of the domain on one side, is shifted back in on the other side. Indeed, the solution given above also works for a periodic domain $(0; \omega)$ with the natural adaptation

$$u(x,t) = u_0((x - at) \bmod \omega) = u_0(x - at - \lfloor \frac{x - at}{\omega} \rfloor \omega).$$

Specifically, for a domain $(0; 1)$,

$$u(x,t) = u_0(x - at - \lfloor x - at \rfloor). \tag{27}$$

Below, we applied the ETD-RDP-IF scheme to this equation using different initial conditions u_0. We vary the strength of advection in order to investigate the behavior with respect to the quantities mentioned in Section 3.1. Since no diffusion is present, we only consider the CFL number.

3.2.1 Wave-Packet Initial Condition

To numerically test for the presence of any CFL-type condition, we consider the advection equation (26) first with a continuously differentiable initial function that lives on a limited domain, specifically, a wave-packet of the form

$$u_0(x) = \sin^2(2\pi x)\mathbb{I}_{[0;0.5]}(x) \tag{28}$$

where $\mathbb{I}$ denotes an indicator function. The full domain is $\Omega = (0; 1)$. The blue curve in Figure 3 shows this function. The exact solution of this problem can be obtained using Equation (27).

For the numerical simulation, we initially set $h = 0.1$ and $k = 0.005$. We then let h and k approach 0 in such a way that the ratio $k/h = 0.05$ remains fixed, namely, by successively decreasing h and k by a factor of 2. This allows us to analyze the convergence behavior analogous to how Courant et al. (Courant et al. 1928) considered it. Specifically, the Courant number is

$$C = \frac{ak}{h} = \frac{ak}{20k} = \frac{a}{20}.$$

In order to observe the behavior for various values of the Courant number, we then performed this experiment for multiple values of the advection velocity a. Since the exact solutions are available, we compute the error as

$$E(h, k) = ||\mathbf{U}(h, k) - \mathbf{U}_e||_\infty$$

where $\mathbf{U}(h, k)$ denotes the solution obtained by ETD-RDP-IF and $\mathbf{U}_e$ denotes the exact solution evaluated at the same grid points as $\mathbf{U}(h, k)$. Note that we consider the solutions at time $T = 1$.

Figure 2 shows the resulting errors obtained for certain values of a with a central difference discretization. While we have run the experiment for more different values of a, namely 0, 0.01, 0.1, 0.15, 0.2, 0.25, 0.3, 0.5, 1, 2, 3, 5, 10, 20, 100, 200, 500, and 1000, we only report values for a subset of these. Unless otherwise noted, the values we do not report did not give any new information. Note that h decreases with k as described above, so $h = 20k$ for each value of k.

We observe that the errors we obtain for a given grid size increase with a. This is plausible since, for the exact PDE, increasing a is equivalent to increasing the final time T. Therefore, this can be interpreted like going further in time. However, we

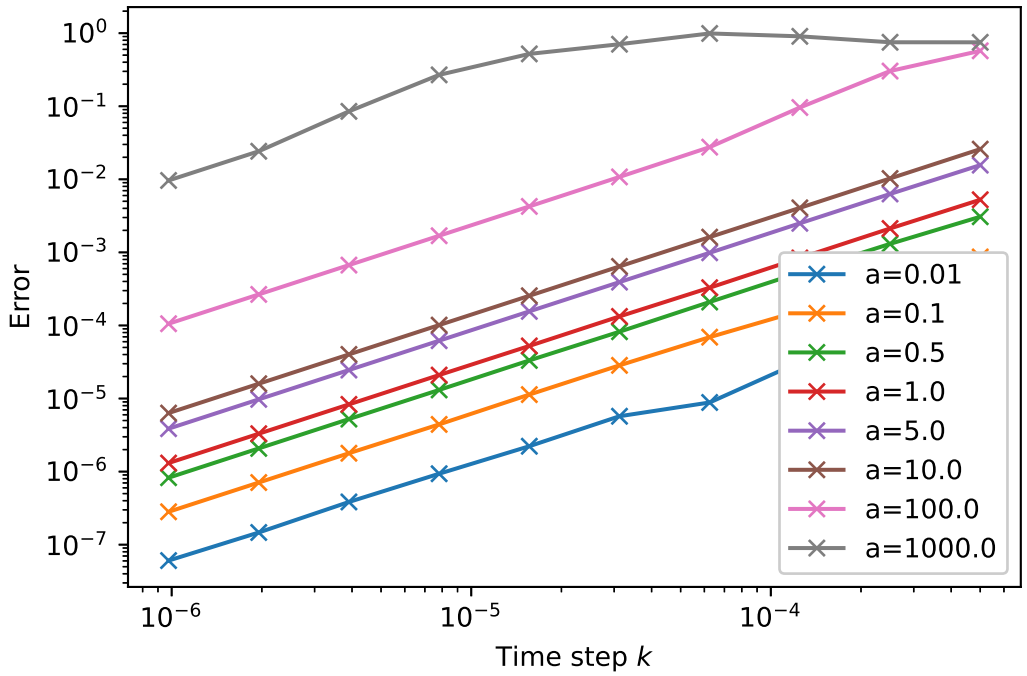


Figure 2. Errors of ETD-RDP-IF with central difference discretization for the advection equation (26) with wave-packet initial condition (28) and varying advection velocities a.

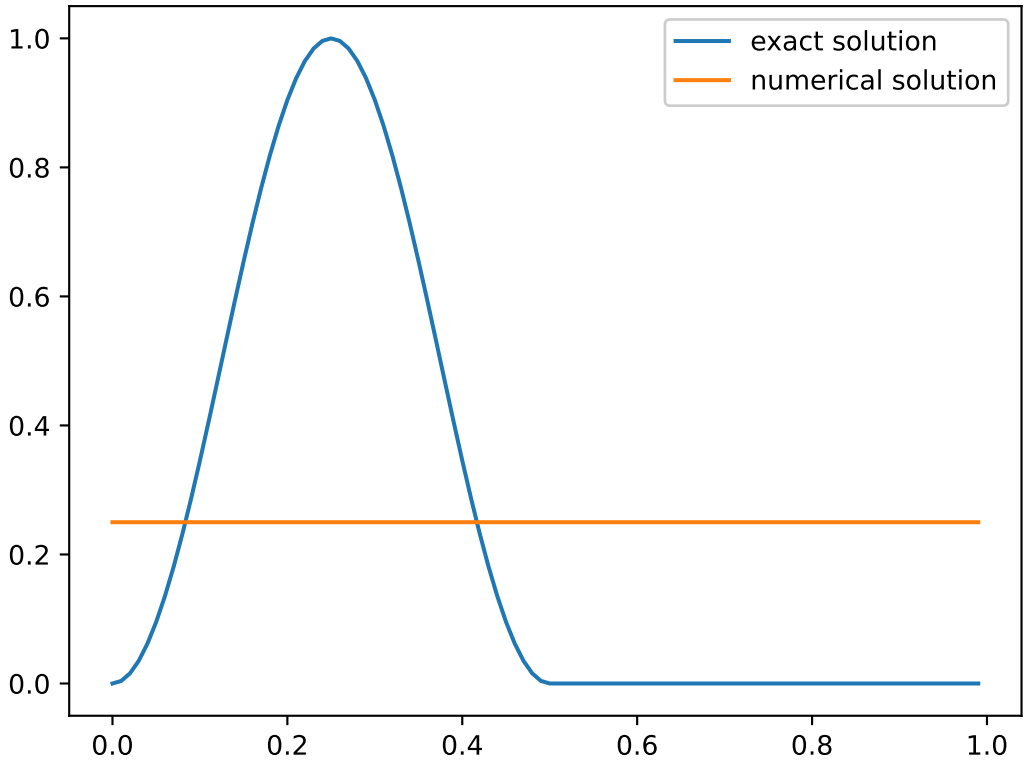


Figure 3. Solution of the advection equation (26) with wave-packet initial condition (28) for $a = 1000$ at $T = 1$. The numerical solution is produced by ETD-RDP-IF with central difference discretization and $h = 0.01$, $k = 0.0005$.

keep the same number of time steps. Thus, it is analogous to increasing the time step. Alternatively, one could argue that due to the increased advection velocity, the change per time step is greater, and therefore the errors introduced by the discretization are also greater.

It is important to observe, however, that the errors do not increase without bound. In fact, the solution for $a = 1000$, $h = 0.01$, $k = 0.0005$ (Figure 3) looks like all oscillations have been damped out, or, equivalently, that the solution has completely diffused, to a constant function. Since the second-order central finite difference

scheme we used to discretize the advection is non-dissipative (Hundsdorfer and Verwer 2003), this damping cannot be attributed to the spatial discretization. It must therefore be inherent in the ETD-RDP scheme. We thus speculate that this can be attributed to the L-stability (Asante-Asamani 2016) of the scheme which causes damping of oscillations, particularly for larger time steps. This is further supported by the fact that reducing the time step k while keeping h fixed yields non-constant solutions. On the other hand, reducing h while keeping k fixed does not change the solution profile. Therefore, we can clearly attribute the damping to the time-discretization scheme, not the spatial discretization.

On the other hand, we see that for all values of a the error decreases when the time step k is decreased. Indeed, the slope of all the curves in the log-log plot of Figure 2 is approximately the same. This suggests that the ETD-RDP-IF scheme converges to the true solution with approximately the same order of convergence for all a. This is confirmed by the plot in Figure 4 which shows the numerical order of convergence, i. e., the order by which the error is reduced when reducing h and k by a factor of 2. We define the order to be

$$O(h,k) = \log_2(E(2h,2k)/E(h,k))$$

This means that the error is reduced by a factor of $2^{O(h,k)}$ when h and k are reduced by a factor of 2. Since our scheme is a second-order scheme, we expect $O(h,k)$ to be approximately 2. In this case, we observe orders of only about 1.3. We speculate that this can be attributed to the fact that our initial condition u_0 is not a smooth function, it is continuously differentiable in x only once.

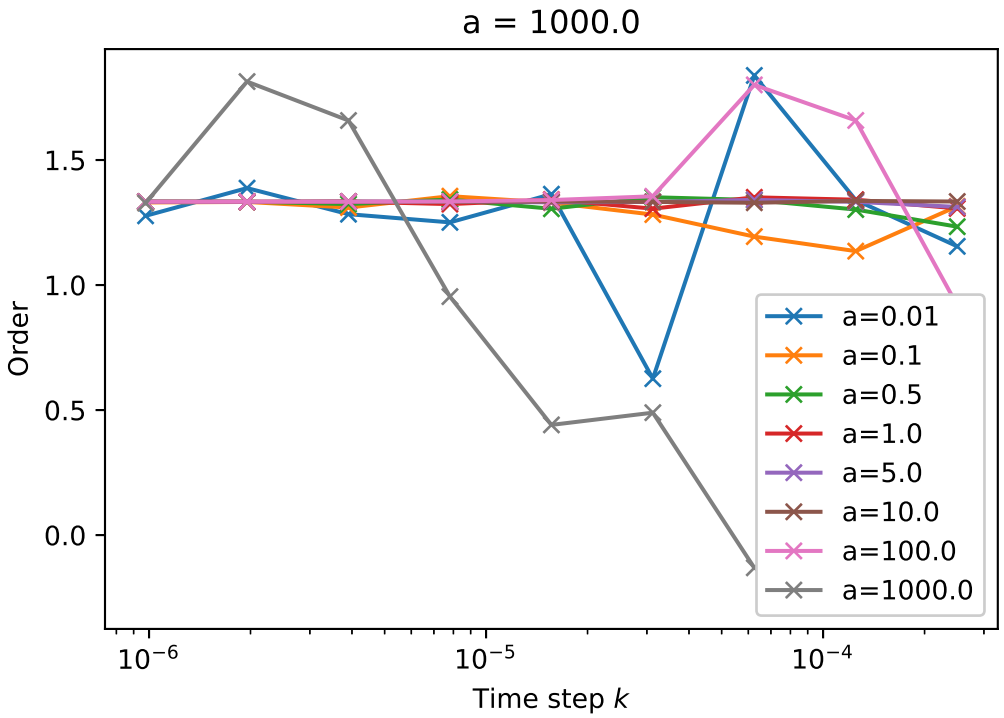


Figure 4. Numerical orders of convergence of ETD-RDP-IF with central difference discretization for the advection equation (26) with wave-packet initial condition (28) and varying advection velocities a.

From this observation, we conclude that we have not observed a CFL-type condition. An advection velocity of $a = 1000$ above corresponds to a Courant number $C = 50$. The fact that the numerical solution still converges to the exact solution at $C = 50$ means that if there is a CFL-type condition, C_{max} would need to be larger than 50. This is in line with our expectations that there is no CFL-type condition due to the implicit treatment of the advection (and diffusion) terms. Note that finer grids are necessary to numerically observe the convergence for large values of a. This is due to the fact that, as described above, the errors for grids are so large that the structure of the exact solution is no longer retained. Due to this and computational limitations, we were unable to obtain conclusive results for greater values of a.

The other discretizations we considered, namely second-order upwind-biased (Fromm) and third-order upwind-biased yielded very similar results. We therefore do not explicitly report them here.

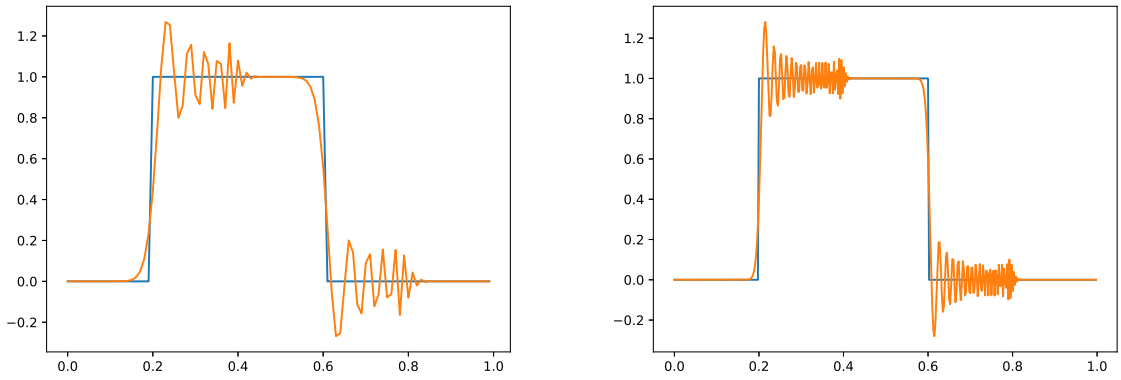
3.2.2 Boxcar Initial Condition

We performed the same experiment as described in the previous Section 3.2.1 with a discontinuous initial condition. For this, we chose a boxcar-type function

$$u_0(x) = \mathbb{I}_{[0.3;0.7]}(x) \tag{29}$$

on the domain $\Omega = (0; 1)$. We again assumed periodic boundary conditions.

Figure 5 shows the numerical and exact solutions of the advection equation (26) with the boxcar initial condition and $a = 1$ at $T = 0.1$ for differently fine grids using the second-order central finite difference scheme for advection. We observe that the



(a) $h = 0.01, k = 0.001/2$

(b) $h = 0.01/4, k = 0.001/8$

(c) $h = 0.01/16, k = 0.001/32$

(d) $h = 0.01/512, k = 0.001/1024$

Figure 5. Solution of the advection equation (26) with boxcar initial condition (29) for $a = 1$ at $T = 0.1$. Blue: exact solution, orange: numerical solution. The numerical solution is produced by ETD-RDP-IF with central difference discretization.

numerical solutions exhibit oscillations that are not present in the exact solution. These spurious oscillations can be attributed to numerical dispersion. As described, e. g., in Hundsdorfer and Verwer (2003) or Strikwerda (2004), numerical solutions to the advection equation cause different Fourier modes (corresponding to different frequencies in space) of the initial condition to be transported at different velocities. In particular, high-frequency Fourier modes are transported more slowly than lower-frequency Fourier modes. The initial condition we used here is not smooth, and therefore the coefficients of high-frequency modes are large. This causes a shift of the different Fourier modes relative to each other, causing the spurious oscillations.

We observe that most of the spurious oscillations are reduced as the grid becomes finer. However, there are "spikes" in the numerical solution at the discontinuities of the exact solution. These remain for the finest grids we were able to compute. Furthermore, they appear to neither increase nor decrease significantly in magnitude as the grid becomes finer. Instead, the spikes become thinner, their support becomes smaller. For much finer grids, a reduction in magnitude might occur but we were unable to observe this due to computational limitations.

Due to these spikes, the maximum norm we used before cannot capture the apparent improvement in the approximation. I. e., in our experiments we do not observe convergent behavior in the maximum norm, the error remains approximately constant. Instead, we use the Euclidean norm here. Note that we normalized it for each grid size with the Euclidean norm of the initial condition discretized with the corresponding grid size. Otherwise, the increase in "dimension" (i. e., number of grid points) alone would cause the norm of the error to increase as h decreases.

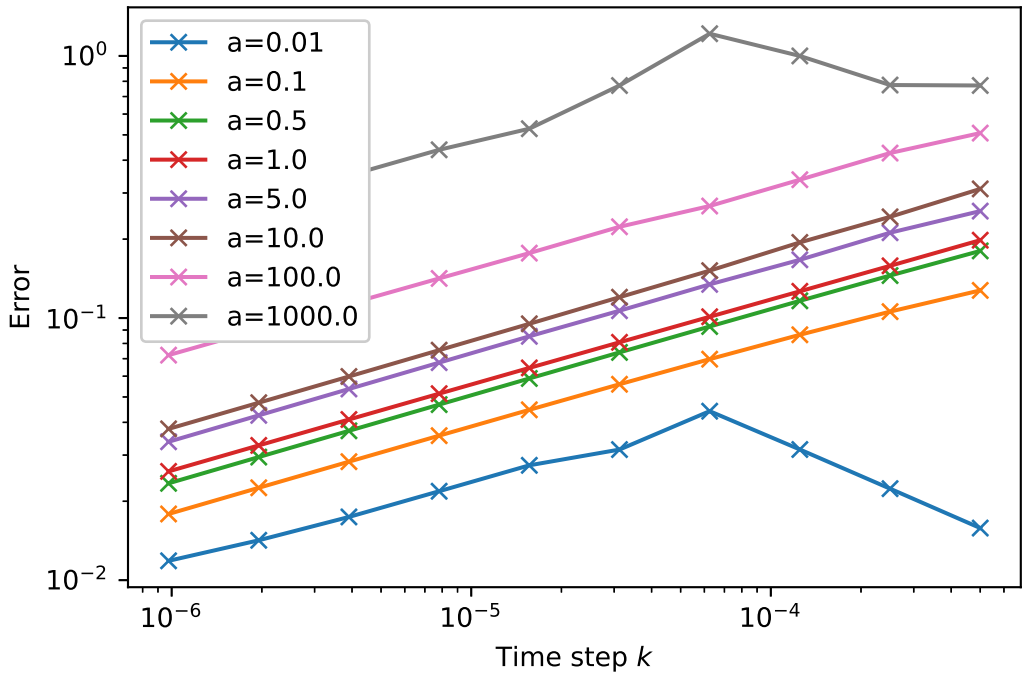
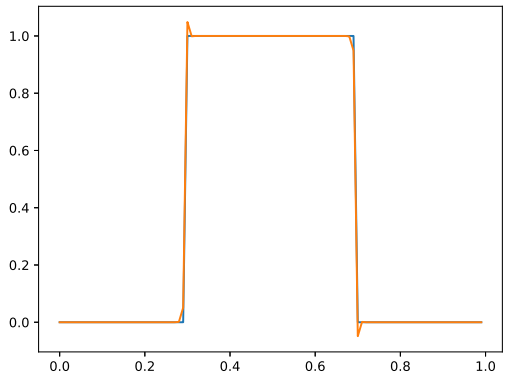


Figure 6. Errors of ETD-RDP-IF with central difference discretization for the advection equation (26) with boxcar initial condition (29) and varying advection velocities a.

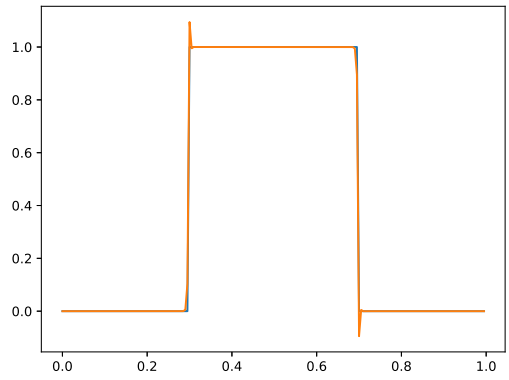
The errors, in the Euclidean norm, are shown in Figure 6. Like for the wave-packet, the errors increase as a increases, but convergence can be observed for all a. Therefore, we also do not observe a CFL-type condition for this initial condition. Just like before, larger a require finer grids before convergence can be observed since for large h and k the overall structure of the solution is lost. However, compared to the wave-packet, the order of convergence is smaller, about 0.3. We assume that this is due to the discontinuity of u_0, i. e., u_0 is highly non-smooth.

One anomaly can be seen in the plot in Figure 6. For $a = 0.01$, the error at first increases before it decreases. This is caused by the spikes we described above. Figure 7 shows the results we obtained for $a = 0.01$ for different grids. It can be seen that, initially, the spikes are small in magnitude while the slope at the discontinuities is noticeably lower than in the exact solution. As h and k decrease, the slope increases to more closely match the desired curve. However, the spike increases significantly in magnitude. This causes the error to increase. As the grid is further refined, the spike no longer grows, in fact it even decreases slightly. This is corresponding to the later decrease in error and convergent behavior.

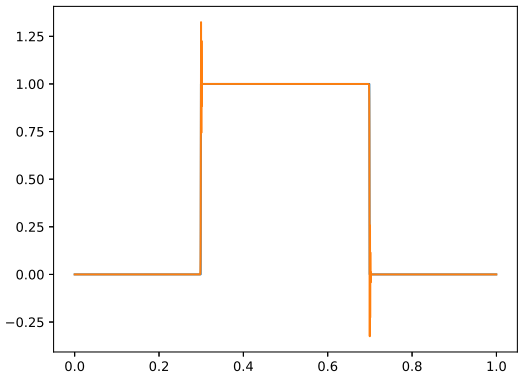
It is also noteworthy that the oscillations do not blow up for coarse grids. Instead, we observe a behavior similar to the wave-packet initial condition. For small a, the numerical solution is qualitatively very similar to the exact solution, even for coarse grids. For large a and large h and k, the numerical solution appears to completely diffuse and becomes a constant function. We attribute this, again, to the L-stability of the ETD-RDP-IF scheme which causes damping of oscillations. In fact, we know that the L^1-norm, i. e., the integral, of the initial condition is 0.4. This is also



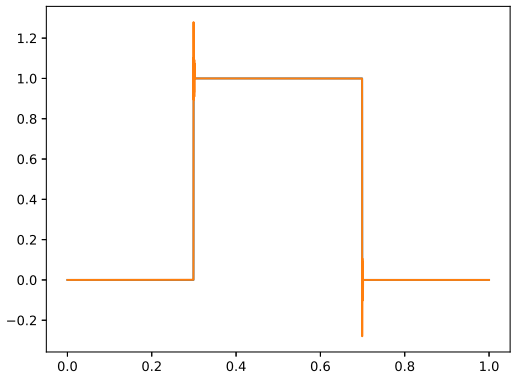
(a) $h = 0.01, k = 0.001/2$



(b) $h = 0.01/2, k = 0.001/4$

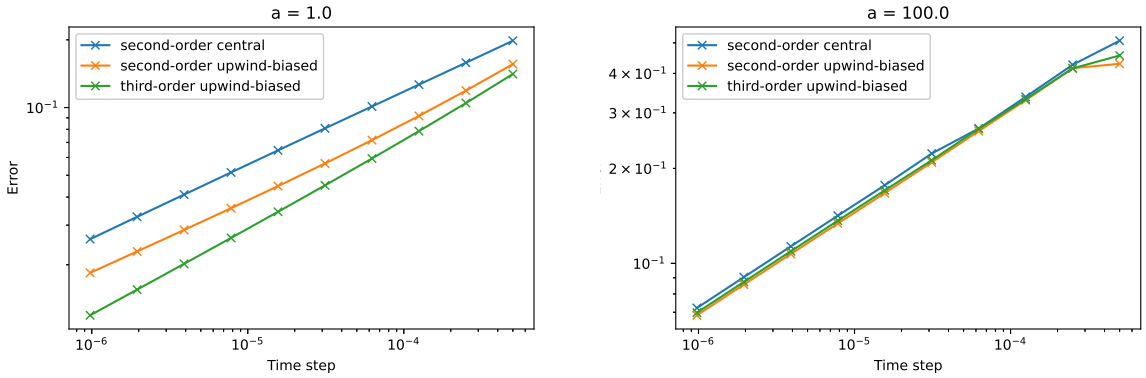


(c) $h = 0.01/32, k = 0.001/64$



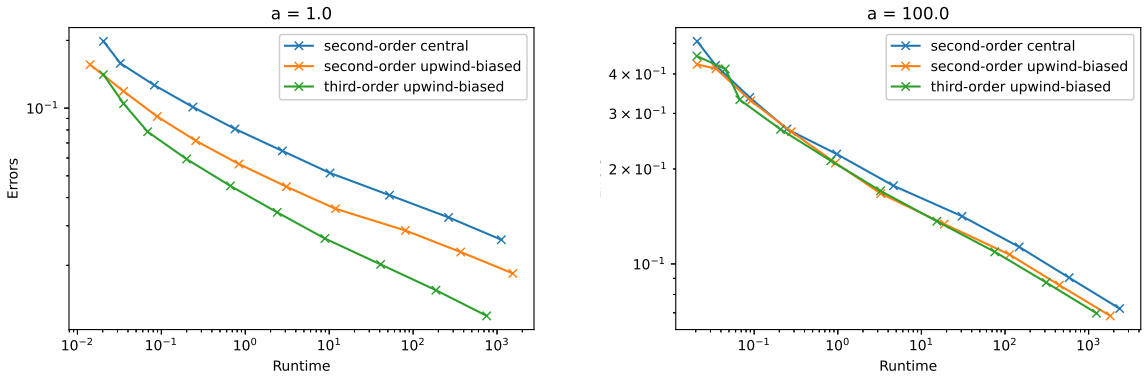
(d) $h = 0.01/512, k = 0.001/1024$

Figure 7. Solution of the advection equation (26) with boxcar initial condition (29) for $a = 0.01$ at $T = 0.1$. Blue: exact solution, orange: numerical solution. The numerical solution is produced by ETD-RDP-IF with central difference discretization.



(a) $a = 1$ **(b)** $a = 100$

Figure 8. Errors of ETD-RDP-IF with different spatial discretizations for the advection equation (26) with boxcar initial condition (29).



(a) $a = 1$ **(b)** $a = 100$

Figure 9. Efficiency of ETD-RDP-IF with different spatial discretizations for the advection equation (26) with boxcar initial condition (29).

the constant value our solution approaches in this case. This supports the assumption that the effect is due to a damping of oscillations in the scheme.

Besides that, we observe different results for the different discretizations of the advection term we implemented. Figure 8 shows the observed errors for selected values of a. Figure 9 shows the errors for the same values of a plotted against the computation time (single thread, Intel(R) Xeon(R) Gold 6248 CPU @ 2.50GHz) required. We show the latter plots since the different discretizations require different

stencils, and therefore change the band structure of our discretization matrix. This might have an impact on computation time. Note that this selection is representative insofar as we obtained similar results for other values of a.

We can see that for lower a, e. g., $a = 1$, the third-order upwind-biased scheme produces the lowest errors for the same time step. Also, it produces the lowest errors for a given runtime. This discretization, therefore, seems to be the most efficient choice. In fact, considering the slope of the lines in the aforementioned log-log plots, we can see that the error also decreases faster for this discretization. We observe a convergence order of about 0.32 for the second-order central finite difference scheme and an order of up to 0.38 for the third-order upwind-biased scheme. For larger values of a, the difference between the discretizations appears less pronounced.

3.3 Benchmark Example with Known Exact Solution

In order to evaluate the real-world performance of the ETD-RDP-IF scheme, we test it on some more complicated ADR problems. Bhatt et al. (2018) described multiple ADR problems that they used to evaluate their Krylov-EETD scheme. The results reported in that paper and the published code of the Krylov-EETD scheme allowed us to perform a comparison of both schemes in terms of accuracy and runtime.

Bhatt et al. (2018) first tested their Krylov-EETD scheme on a linear benchmark problem with periodic boundary conditions.

$$\begin{aligned} u_t &= \frac{d}{3}\Delta u - \frac{a}{3}(u_x + u_y + u_z) - bu + v \\ v_t &= \frac{d}{3}\Delta v - \frac{a}{3}(v_x + v_y + v_z) - cv \end{aligned} \tag{30}$$

on the three-dimensional domain $\Omega = (0; 2\pi)^3$ with real constants b and c. Given the initial condition

$$u_0(x, y, z) = 2 \cdot \cos(x + y + z)$$

$$v_0(x, y, z) = (b - c) \cdot \cos(x + y + z),$$

this problem has the exact solution

$$u(x, y, z, t) = (\mathrm{e}^{-(b+d)t} + \mathrm{e}^{-(c+d)t}) \cdot \cos(x + y + z - at)$$

$$v(x, y, z, t) = (b - c)\mathrm{e}^{-(c+d)t} \cdot \cos(x + y + z - at).$$

This closed-form analytical solution allows us to compute the errors precisely $E(h, k)$ as in Section 3.2.1 and compare them to the errors of Krylov-EETD. For a comparison, we performed the same numerical experiments as in Bhatt et al. (2018) (Table 1). The parameters were set to $a = 3$, $b = 100$, $c = 1$ and $d = 1$. We used $N = 10 \cdot 2^i$ spatial grid points per dimension, i. e., N^3 spatial grid points in total, corresponding to $h \approx 0.628/2^i$, and simulated up to $t_e = 1.0$ using a time step of $k = 0.005/2^i$. We were able to reproduce the same errors (up to at least 3 significant digits) in our tests, indicating that the precision of the ETD-RDP-IF scheme we used is approximately the same. Table 1 lists the precise results up to 3 significant digits. The errors are given in the maximum norm as described in Section 3.2.1. Like for Krylov-EETD, the order of convergence can consequently be concluded to be 2. We also notice that the computation time is lower for Krylov-EETD.

For a more extensive comparison, we performed the same experiment for different values of a while keeping all other parameters fixed. The values of a we used were 0, 0.01, 0.1, 0.15, 0.2, 0.25, 0.3, 0.5, 1, 2, 3, 5, 10, 20, 100, 200, 500, 1000. As

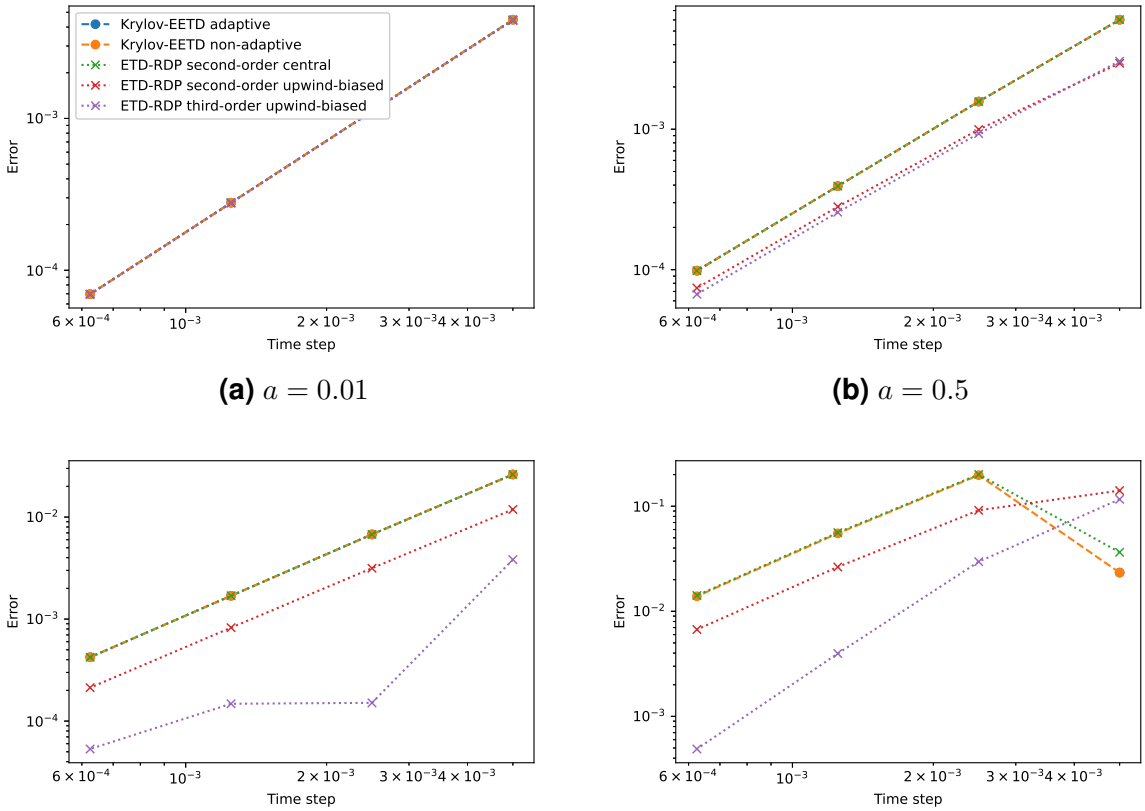
	h k	$2\pi/10$ 5E-03	$2\pi/20$ 5E-03/2	$2\pi/40$ 5E-03/4	$2\pi/80$ 5E-03/8
ETD-RDP-IF 2nd-order central	$E(h,k)$ CPU(s)	2.62E-02 0.38	6.76E-03 6.80	1.69E-03 78.8	4.23E-04 1088
ETD-RDP-IF 2nd-order upwind-biased	$E(h,k)$ CPU(s)	1.19E-02 0.44	3.15E-03 7.21	8.24E-04 84.0	2.13E-04 1177
ETD-RDP-IF 3rd-order upwind-biased	$E(h,k)$ CPU(s)	3.83E-02 0.44	1.51E-04 7.15	1.48E-04 84.2	5.32E-05 1170
Krylov-ETD adaptive	$E(h,k)$ CPU(s)	2.62E-02 0.46	6.76E-03 2.23	1.69E-03 27.1	4.23E-04 476
Krylov-ETD non-adaptive	$E(h,k)$ CPU(s)	2.62E-02 0.68	6.76E-03 4.49	1.69E-03 101	4.23E-04 2469

Table 1. Errors and computation time for the three-dimensional benchmark problem (30) with different types of spatial and temporal discretizations on different grids. CPU time is for a single core of an Intel(R) Xeon(R) Gold 6248 CPU @ 2.50GHz.

mentioned in Section 2.3, in certain situations the runtime of Krylov-EETD increased beyond feasibility due to the adaptive nature of the `expv` function. In this problem, we observed that more steps in `expv` were necessary to achieve the required precision for large values of a. We, therefore, chose a threshold for the number of steps that `expv` is allowed to take. If the time step becomes small enough that this threshold would be crossed, we abort the computation and state that no result could be obtained instead of attempting the computation. No values are reported in this case.

For the non-adaptive version of Krylov-EETD, we noticed that instabilities arise for large values of a, i. e., the error increases beyond the order of magnitude of the true solution. In these cases, we note that we have clearly not obtained reasonable results and also do not report the values. We interpret this behavior as a kind of instability.

Besides this, we also tested the ETD-RDP-IF scheme with the different discretization schemes for advection we described in this work. Note that the



(a) $a = 0.01$ **(b)** $a = 0.5$

(c) $a = 3.0$ **(d)** $a = 100$

Figure 10. Error comparison of ETD-RDP-IF with different spatial discretizations and Krylov-EETD for the three-dimensional benchmark problem (30).

Krylov-EETD scheme uses a second-order central scheme for the discretization of advection. Therefore, we consider two main comparisons, the comparison of Krylov-EETD with ETD-RDP-IF given second-order central schemes, and the comparison of the central scheme with the upwind-biased schemes within ETD-RDP-IF. It is to be expected that Krylov-EETD would benefit in a similar way from upwind-biased schemes as ETD-RDP-IF does.

Errors. Figure 10 shows the errors we obtained for different values of a for the different schemes we tested. For a comparison of ETD-RDP-IF with the Krylov-EETD scheme, note first that for (a) – (c) the graphs of the errors of Krylov-EETD are not

clearly visible since they coincide with the values for ETD-RDP-IF with central finite difference discretization of advection. Therefore, both methods produce nearly identical errors. This holds for both implementations of the Krylov-EETD scheme with the adaptive and non-adaptive `expv` function. In fact, we also compared the respective solutions and noticed that they are nearly identical as well, i. e., the differences between the solutions are significantly lower (at least by a factor 10^3 lower for $a \leq 10$) than the errors of either scheme. Since the problem (30) is linear, this suggests the conclusion that both methods yield the same results for linear problems up to different rounding errors introduced by the different ways of computing the result. Note that the observed order of convergence for the Krylov-EETD scheme and the ETD-RDP-IF scheme with central finite differences is approximately 2.

We notice, however, that for larger values of a, Figure 10 (d), there are differences in the errors, and therefore also in the solutions. The Krylov-EETD scheme with the adaptive `expv` function exceeded feasible computation times here, so no values are reported. It is not immediately clear how the differences between the errors arise. One possible reason is that the above conclusion is incorrect, and both schemes do not produce identical results up to rounding for general linear problems. Instead, the solutions of the schemes depend differently on a. Since all errors are large compared to, e. g., Figure 10 (c), it is also possible that the differences can be attributed to a loss of precision in the non-adaptive `expv` function, i. e., the Krylov-subspace approximation of the matrix exponential. This is supported by the fact that the adaptive function would require a large number of steps that crossed our threshold. Further experiments are necessary to obtain conclusive results on this.

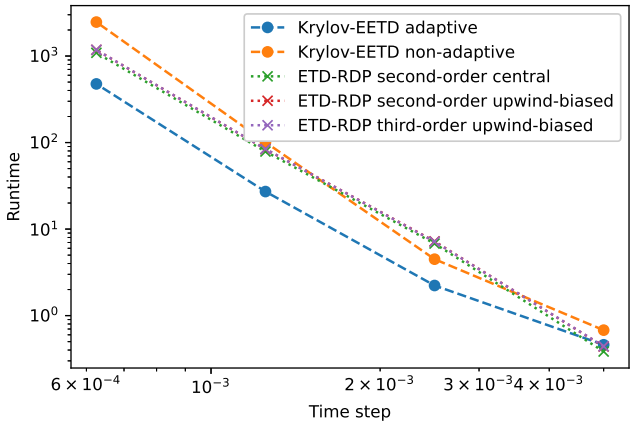


Figure 11. Runtime comparison of ETD-RDP-IF with different spatial discretizations and Krylov-EETD for the three-dimensional benchmark problem (30) with $a = 3$.

Considering the different discretization schemes in the ETD-RDP-IF implementation, the upwind-biased schemes produce lower errors, with the third-order scheme having lower errors than the Fromm scheme. This holds particularly for larger values of a. For small values of a and finer grids, the errors obtained by both upwind-biased schemes approach those of the central difference scheme. The observed order of convergence is therefore below 2. For larger values of a, the order of convergence fluctuates strongly. However, the errors are always below those produced by the central difference scheme. We therefore speculate that this fluctuation is reduced for finer grids. We were not able to test this due to computational limits.

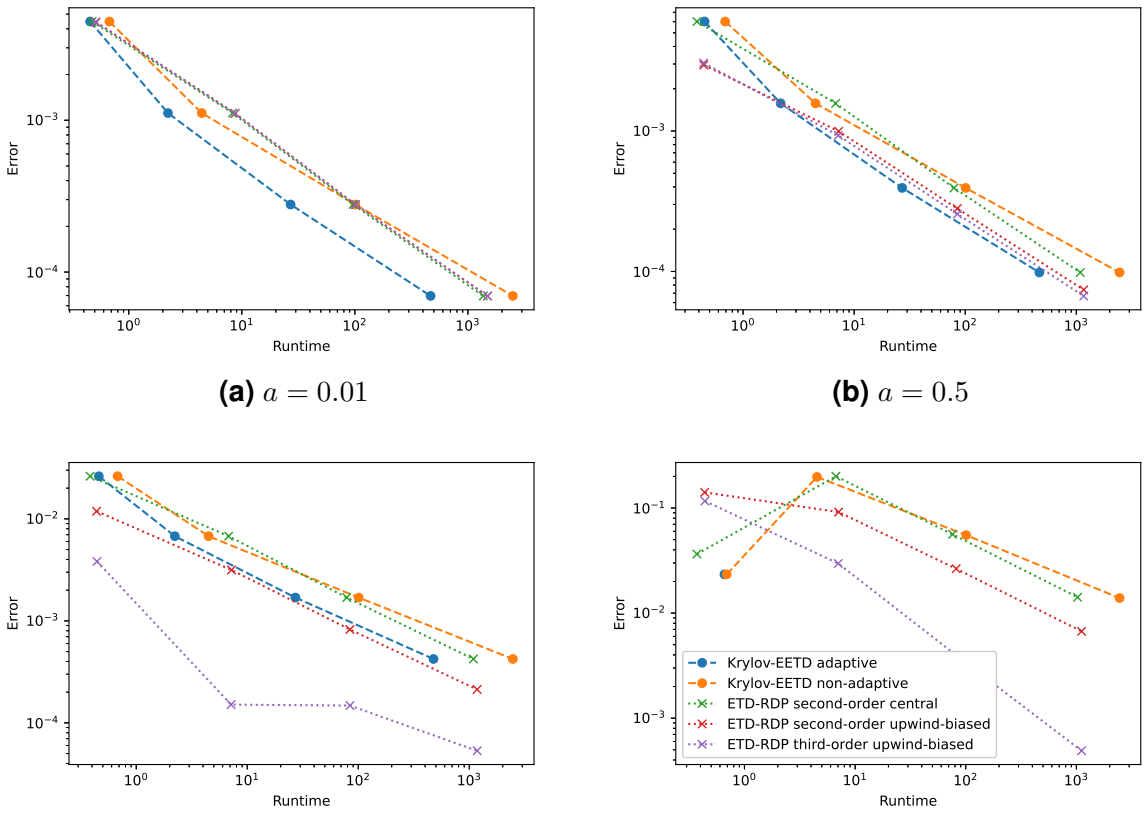
Runtimes. The adaptive Krylov-EETD scheme consistently exhibits the lowest runtimes when it is able to obtain a solution. As shown in Figure 11, it is 2 to 4 times faster than the ETD-RDP-IF implementations we tested, except for the coarsest grid we considered. This makes it the most efficient implementation for this problem. The

same pattern can be observed for all other values of a. However, in spite of the good performance for $a \leq 20$, for larger values of a, we were unable to complete the computation due to possibly excessive runtimes. This suggests the existence of a stability regime outside of which increasingly more computation time is required to maintain the desired precision in the matrix exponential. In particular, we find that difficulties arise in regimes with strong advections.

The non-adaptive implementation of the Krylov-EETD scheme, in contrast, exhibits the greatest computation times for all experiments. This can be attributed to the remaining adaptivity of the scaling-and-squaring algorithm in the `expm` function which compensates for the larger step in the Arnoldi algorithm. However, it is able to produce results for a wider range of advection velocities.

The different discretization schemes implemented with ETD-RDP-IF show similar runtimes. Due to the larger stencils used for both upwind-biased schemes, these have slightly longer runtimes. See Table 1 for detailed values. As shown in Figure 12, the efficiency is, however, better than for the central difference scheme for multiple values of a due to the lower errors.

Further Convergence Considerations for Finer Grids. In order to analyze the convergence behavior of ETD-RDP-IF in more detail, we performed an additional test run keeping the time step constant at its lowest tested value, $k = 0.005/8$, and testing different spatial accuracies for the central difference discretization. With $a = 3$, we found the same errors as seen in Table 1. Therefore, we conclude, that convergence is indeed $\mathcal{O}(h^2)$, as doubling the number of grid points leads to a reduction of the error by factor 4.



(a) $a = 0.01$ **(b)** $a = 0.5$

(c) $a = 3$ **(d)** $a = 100$

Figure 12. Efficiency comparison of ETD-RDP-IF with different spatial discretizations and Krylov-EETD for the three-dimensional benchmark problem (30).

In contrast, however, we found that keeping spatial accuracy constant (at $N = 80$, central differences) while iterating different time steps, no such convergence could be observed. We found rapidly decreasing errors that then remained approximately constant below a threshold of $k_0 \approx 0.005$. This kind of behavior can be explained by the limitations in spatial accuracy. I. e., decreasing k below k_0 does not improve the approximation quality since most of the error is introduced due to the spatial discretization.

In order to observe a convergence behavior in k, we reduced the problem to two dimensions:

$$\begin{aligned} u_t &= \frac{d}{2}\Delta u - \frac{a}{2}(u_x + u_y) - bu + v \\ v_t &= \frac{d}{2}\Delta v - \frac{a}{2}(v_x + v_y) - cv \end{aligned} \tag{31}$$

on the two-dimensional domain $\Omega = (0; 2\pi)^2$ where $a = 3$, $b = 100$, $c = 1$ and $d = 1$. Analogous to the three-dimensional problem above, given the initial condition

$$u_0(x, y) = 2 \cdot \cos(x + y)$$

$$v_0(x, y) = (b - c) \cdot \cos(x + y),$$

this problem has the exact solution

$$u(x, y, t) = (\mathrm{e}^{-(b+d)t} + \mathrm{e}^{-(c+d)t}) \cdot \cos(x + y - at)$$

$$v(x, y, t) = (b - c)\mathrm{e}^{-(c+d)t} \cdot \cos(x + y - at).$$

This allowed us to increase N by a factor of 4 while the computation remained feasible. With this increased spatial accuracy, we observed the same behavior of the errors with respect to k as above. There is a significant drop in errors for large k until

the limits imposed by spatial grid size are reached. Table 2 shows the observed errors.

h	$2\pi/320$	$2\pi/320$	$2\pi/320$	$2\pi/320$
k	4E-02	2E-02	1E-02	5E-03
$E(h,k)$	7.22E+16	2.37E-01	2.88E-05	2.44E-05
CPU(s)	1.18	2.35	5.24	9.97
h	$2\pi/320$	$2\pi/320$	$2\pi/320$	
k	5E-03/2	5E-03/4	5E-03/8	
$E(h,k)$	2.58E-05	2.63E-05	2.64E-05	
CPU(s)	21.6	38.5	80.1	

Table 2. Errors and computation time for the two-dimensional benchmark problem (31) with different types of spatial and temporal discretizations on different grids. CPU time is for a single core of an Intel(R) Core(TM) i7-10510U CPU @ 1.80GHz.

Therefore, we again conclude that the limitations in accuracy imposed by the grid size prevent the error from becoming small enough for smaller k in order to detect the asymptotic behavior as $k \to 0$. Restating the problem for a single dimension allowed us to increase N considerably and to consider smaller values of k. The problem reads:

$$\begin{aligned} u_t &= d\Delta u - a(u_x) - bu + v \\ v_t &= d\Delta v - a(v_x) - cv \end{aligned} \tag{32}$$

on the domain $\Omega = (0; 2\pi)$ with coefficients as above. The initial condition reduces to

$$u_0(x) = 2 \cdot \cos(x)$$

$$v_0(x) = (b - c) \cdot \cos(x),$$

and the exact solution becomes

$$u(x,t) = (\mathrm{e}^{-(b+d)t} + \mathrm{e}^{-(c+d)t}) \cdot \cos(x - at)$$

$$v(x,t) = (b - c)\mathrm{e}^{-(c+d)t} \cdot \cos(x - at).$$

Table 3 shows exemplary errors and the observed order of convergence. The lower order in the last column can be explained by the limits of spatial grid size as for smaller k the error does not decrease further.

h	$2\pi/10240$	$2\pi/10240$	$2\pi/10240$	$2\pi/10240$
k	1E-02	5E-03	5E-03/2	5E-03/4
$E(N,k)$	6.70E-05	1.68E-05	4.20E-06	1.04E-06
Order	-	2.00	2.00	2.01
h	$2\pi/10240$	$2\pi/10240$	$2\pi/10240$	
k	5E-03/8	5E-03/16	5E-03/32	
$E(N,k)$	2.55E-07	6.20E-08	2.55E-08	
Order	2.03	2.04	1.28	

Table 3. Errors and computation time for the one-dimensional benchmark problem (32) with different types of spatial and temporal discretizations on different grids. CPU time is for a single core of an Intel(R) Core(TM) i7-10510U CPU @ 1.80GHz.

In summary, since convergence considers the limit as $k \to 0$, we have observed a second-order convergence in k. However, this only becomes apparent for very fine spatial grids and small time steps k. Hence, we conclude that spatial accuracy, not temporal accuracy is the limiting factor for this problem.

3.4 Schnakenberg Problem

The Schnakenberg problem was originally described in Schnakenberg (1979). It models auto-catalytic chemical reactions with an oscillatory component (Bhatt and Khaliq 2015). It has been extended in Madzvamuse (2006), Fernandes and Fairweather (2012), and Bhatt et al. (2018) to two- or three-dimensional IBVPs, some of which include an advection term besides the diffusion and reaction terms.

Schnakenberg Reaction-Diffusion Model. The first problem we investigated is given as (Bhatt and Khaliq 2015)

$$\frac{\partial u_1}{\partial t} = d_1(\frac{\partial^2 u_1}{\partial x^2} + \frac{\partial^2 u_1}{\partial y^2}) + \gamma(a - u_1 + u_1^2 u_2)$$
$$\frac{\partial u_2}{\partial t} = d_2(\frac{\partial^2 u_2}{\partial x^2} + \frac{\partial^2 u_2}{\partial y^2}) + \gamma(b - u_1^2 u_2)$$
$$\Omega = (0;1) \times (0;1)\,, \qquad t > 0$$

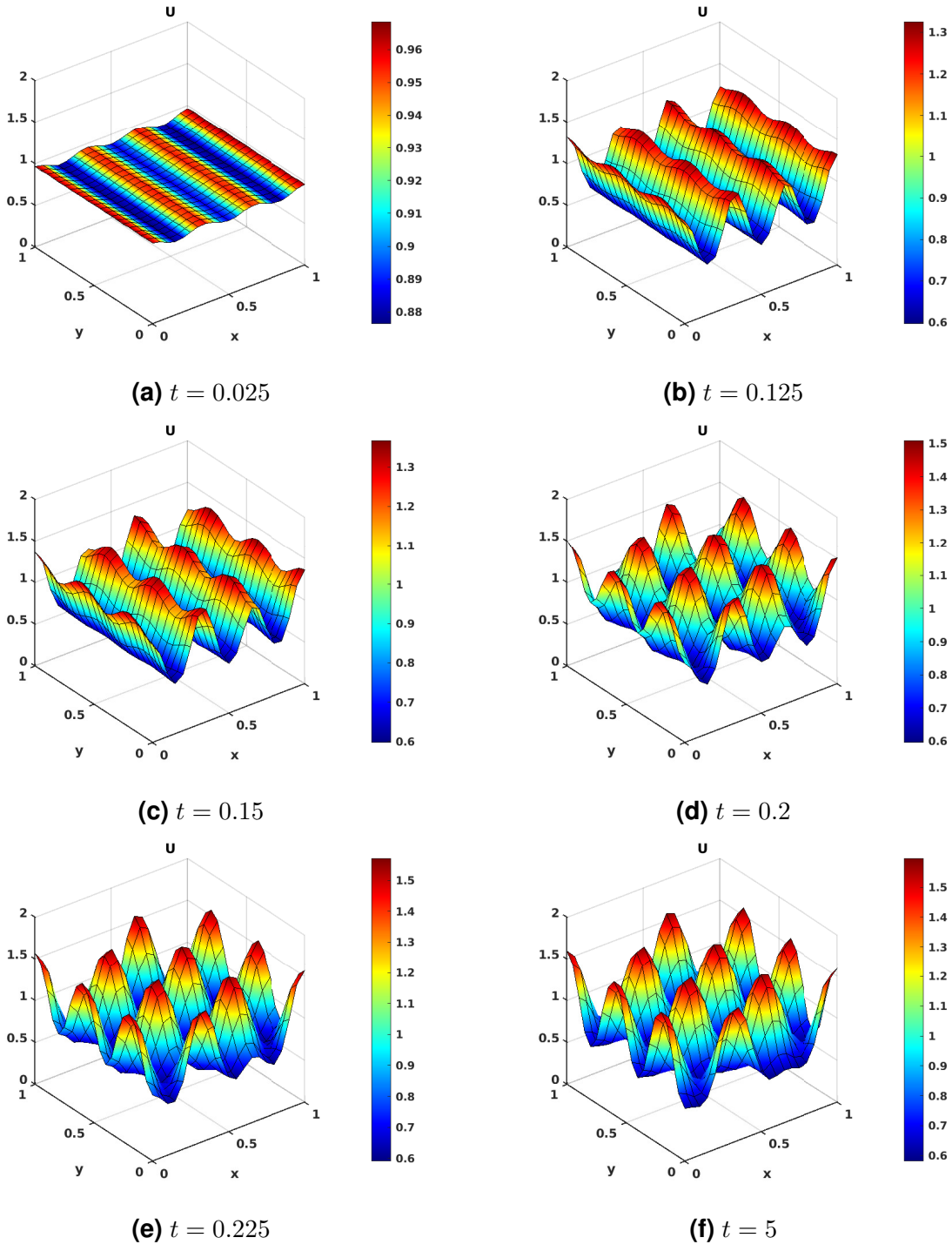
subject to Neumann boundary conditions. The initial conditions are:

$$u_1(x, y, 0) = 0.919145 + 0.0016 \cdot \cos(2\pi(x + y)) + 0.01 \cdot \sum_{j=1}^{8} \cos(2\pi j x)$$
$$u_2(x, y, 0) = 0.937903 + 0.0016 \cdot \cos(2\pi(x + y)) + 0.01 \cdot \sum_{j=1}^{8} \cos(2\pi j x)$$

We performed a simulation with coefficients given in Bhatt and Khaliq (2015) in order to reproduce the figures shown there. These are the following: $a = 0.126779$, $b = 0.792366$, $d_1 = 1.0$, $d_2 = 10.0$, and $\gamma = 1000$. Those coefficients have in turn been first used in Fernandes and Fairweather (2012). Therefore, we were able to compare our results to both papers. Only visual comparison of the resulting figures was possible as we do not have access to the code used for either paper.

Similarly to both papers, we focus on the concentration profile of u_1 in Ω. We simulate from $t_0 = 0$ to $t_e = 5$. Figures 13 and 14 show the results obtained at certain points in time.

At a glance, the images we produced look similar to those from both references. Close examination of the images shows that they visually match the images shown in Fernandes and Fairweather (2012) more closely than those shown in Bhatt and Khaliq (2015). For example, Figures 13 (b) and 14 (b) reveal that the

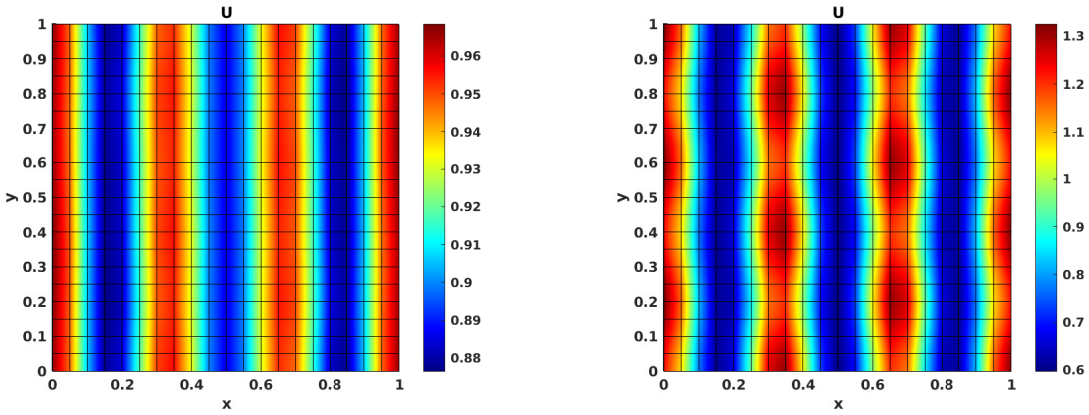


(a) $t = 0.025$ **(b)** $t = 0.125$

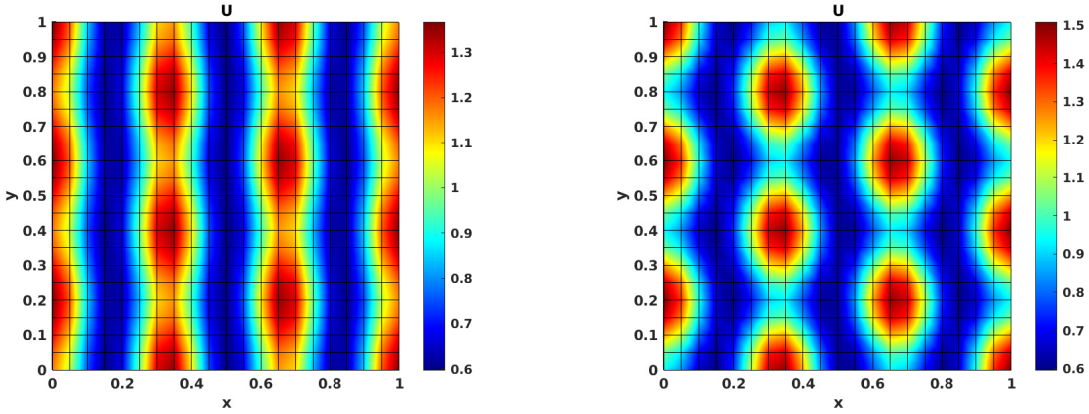
(c) $t = 0.15$ **(d)** $t = 0.2$

(e) $t = 0.225$ **(f)** $t = 5$

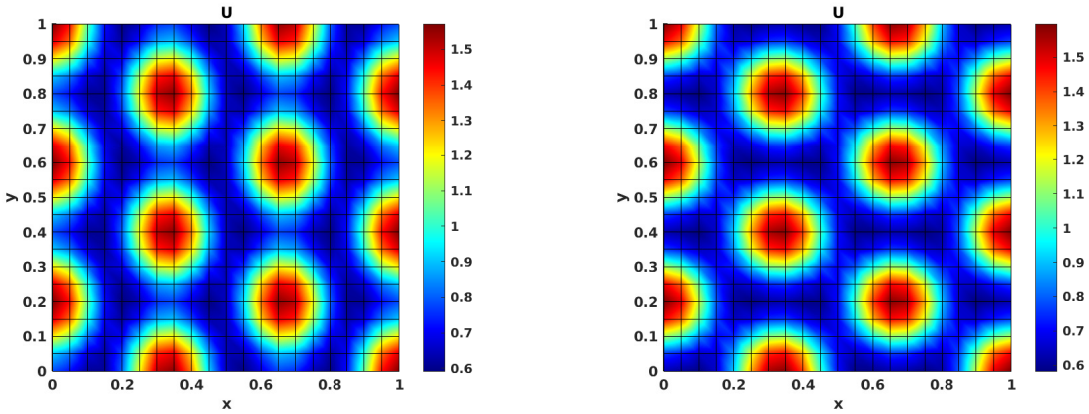
Figure 13. Numerical solution of the concentration profiles of species u_1 for the two-dimensional Schnakenberg reaction-diffusion model at different points in time. The numerical solutions were obtained with ETD-RDP-IF with central difference discretization.



(a) $t = 0.025$ **(b)** $t = 0.125$



(c) $t = 0.15$ **(d)** $t = 0.2$



(e) $t = 0.225$ **(f)** $t = 5$

Figure 14. Aerial views of Figure 13

saddle points that are located between the peaks (in direction y) seem darker in Figure 14 (b) than they do in Fernandes and Fairweather (2012), i. e., the values of u_1 at those points are relatively closer to the values at the peaks. In Bhatt and Khaliq (2015), the values are relatively lower than in Fernandes and Fairweather (2012).

Similar behavior can be observed in Figures 13 (c) – (e), and 14 (c) – (e), respectively. In 13 (f) and 14 (f) in contrast, the simulation has converged to its steady state well enough that any remaining differences are too minor to be observed visually.

Schnakenberg ADR Model. Furthermore, we performed simulations of a three-dimensional Schnakenberg ADR model as given in Bhatt et al. (2018)

$$\begin{aligned}
\frac{\partial u_1}{\partial t} + a_1(\frac{\partial u_1}{\partial x} + \frac{\partial u_1}{\partial y} + \frac{\partial u_1}{\partial z}) &= d_1(\frac{\partial^2 u_1}{\partial x^2} + \frac{\partial^2 u_1}{\partial y^2} + \frac{\partial^2 u_1}{\partial z^2}) + \gamma(\alpha - u_1 + u_1^2 u_2) \\
\frac{\partial u_2}{\partial t} + a_2(\frac{\partial u_2}{\partial x} + \frac{\partial u_2}{\partial y} + \frac{\partial u_2}{\partial z}) &= d_2(\frac{\partial^2 u_2}{\partial x^2} + \frac{\partial^2 u_2}{\partial y^2} + \frac{\partial^2 u_2}{\partial z^2}) + \gamma(\beta - u_1^2 u_2) \qquad (33) \\
\Omega = (0;1) \times (0;1) \times (0;1)\,, &\qquad t > 0,
\end{aligned}$$

subject to periodic boundary conditions. The initial conditions were

$$u_1(x,y,z,0) = 1 - \mathrm{e}^{-10((x-1/2)^2+(y-1/2)^2+(z-1/2)^2)}$$

$$u_2(x,y,z,0) = 0.9 - \mathrm{e}^{-10((x-1/2)^2+(y-1/2)^2+(z-1/2)^2)}$$

We considered the same parameters as Bhatt et al. (2018), $d_1 = 0.05$, $d_2 = 0.01$, $\alpha = 1.0$, $\beta = 0.9$, $\gamma = 1.0$. We set $a_1 = a_2 = a$ and varied a as described above. We simulated up to time $T = 1.0$ using a spatial grid with $h = 1/32$ and different values of k, specifically $k = 0.01/2^i, i = 0, \ldots, 5$.

Error Estimation. Since we do not know an exact analytical solution to this problem, we cannot directly compute the errors. Instead, we use an error estimate.

Like Bhatt et al. (2018), we define

$$E(h,k) = ||U(h,k) - U(h,2k)||_\infty$$

The assumption underlying this error estimate is that $U(h,k)$ is closer to the true solution than $U(h,2k)$. Assuming second-order convergence to the true solution, and assuming that h is small enough that it does not limit the precision, we would observe second-order convergence in k. Then,

$$||U(h,k) - U_e||_\infty = \frac{1}{4}||U(h,2k) - U_e||_\infty.$$

Therefore, $U(h,k)$ is sufficiently closer to the true solution for $E(h,k)$ to yield a good error estimate. We choose this definition instead of

$$\tilde{E}(h,k) = ||U(h,k) - U(h,k/2)||_\infty$$

to obtain a more conservative and less computationally expensive estimate.

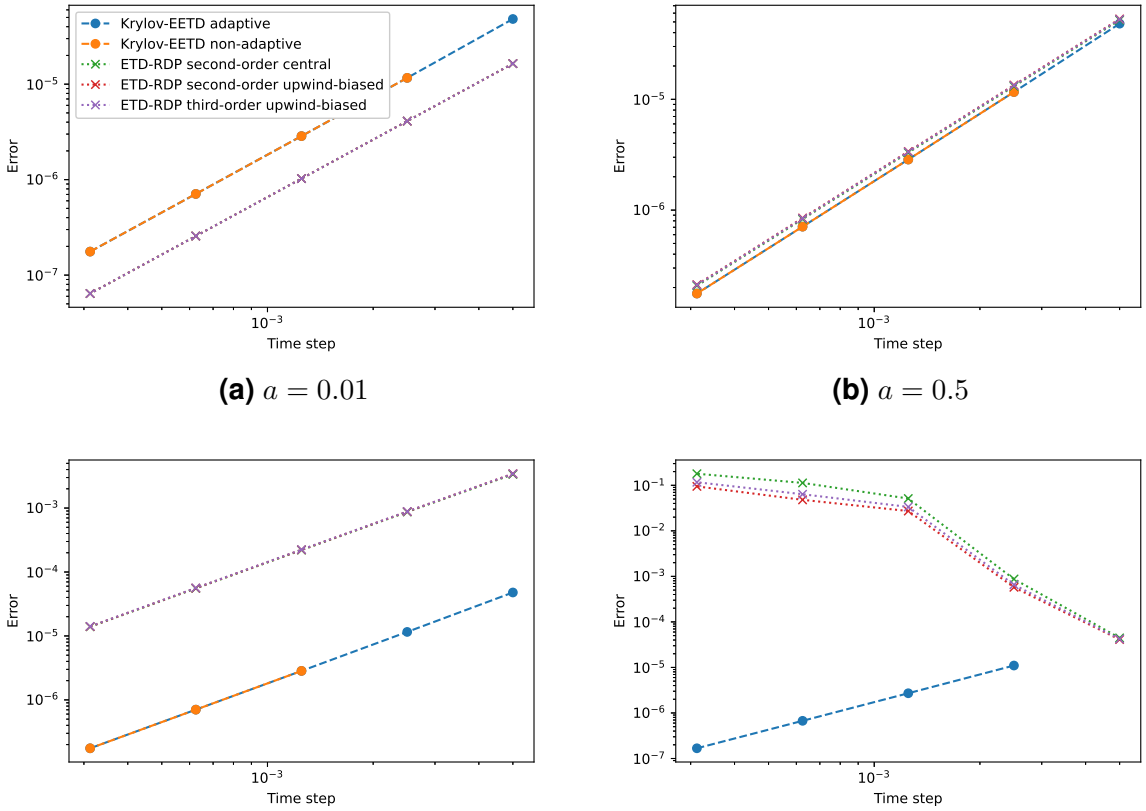
It is important to note that since the error estimate we used is independent of the true solution, we might not notice if our scheme instead converges to an incorrect solution. This might happen, for example, due to a bug in the code, like a sign error. We therefore first performed a sanity check and compared the solutions to each other. For the solutions obtained using the second-order central differencing scheme, we were able to show that the difference of the solutions is in the order of magnitude of the greatest error estimates. This means that any discrepancies between the solutions are explained by the errors. This strongly suggests that the solutions converge to the same limit. The solutions obtained using upwind-biased schemes, on the other hand, differ from the other solutions by more than the error estimates. Since

we are only letting k decrease, however, this can be attributed to the fact the differences in the spatial discretization remain constant over all the test runs we performed. Hence, we can conclude that we found no obvious discrepancies.

When computing the error estimates for the Krylov-EETD scheme, there were discrepancies from the errors reported in Bhatt et al. (2018). Our translation of the code to Python showed lower error estimates. Comparing the implementations showed that the original implementation by Bhatt et al. (2018) deviated from the description of the algorithm in the paper. Fixing this in the original Matlab implementation yielded the same error estimates as we had observed. Therefore, in the following, we report the error estimates we obtained after this fix in order to achieve a more meaningful comparison.

Results. Figure 15 shows the error estimates for select values of a. Note that we simulated for more different values of a and the plots depicted here show the trends we observed. For small values of a, the errors of ETD-RDP-IF are consistently lower than those of Krylov-EETD. Since the different discretization schemes we used for ETD-RDP-IF yield similar errors, we do not mention them separately. Note that where no values of the non-adaptive Krylov-EETD scheme are shown, the computation was not possible due to reaching the limits of floating point numbers, i. e., results were reported as infinite or `nan`. This suggests that the numerical solution blows up in these cases, indicating instability. This instability is reduced by the adaptive scheme.

As a increases, the errors of ETD-RDP-IF increase, similarly to what we observed in Section 3.2. At $a = 100$ and greater, we no longer observe convergent



(a) $a = 0.01$ **(b)** $a = 0.5$

(c) $a = 3.0$ **(d)** $a = 100$

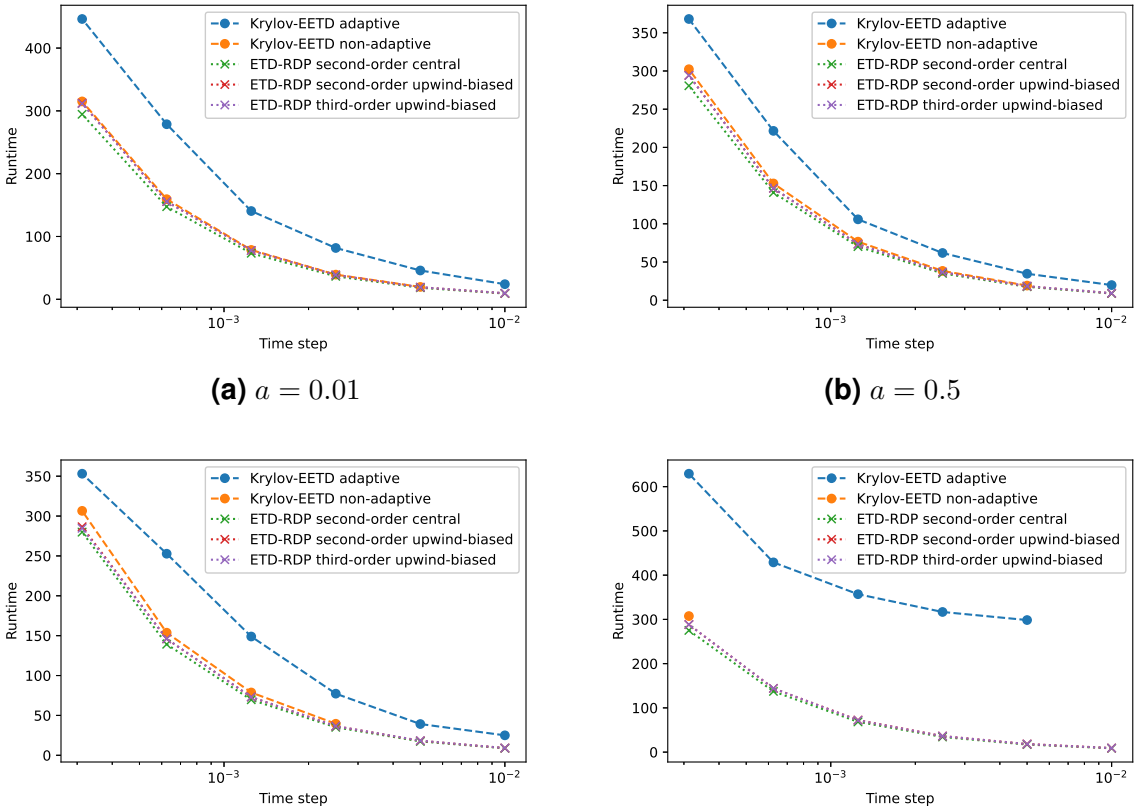
Figure 15. Error comparison of ETD-RDP-IF with different spatial discretizations and Krylov-EETD for the Schnakenberg ADR model (33). Note that where no values for Krylov-EETD are shown, computation time was infeasible or errors were exceedingly large.

behavior. Instead, errors increase as the temporal grid becomes finer. This is unexpected since the error for the coarse temporal grid is still low at about $4 \cdot 10^{-5}$. Further investigation is needed to explain this phenomenon.

The errors of the adaptive Krylov-EETD, in contrast, remain nearly constant for different a on the same grid, and we can still see convergent behavior at $a = 100$. The non-adaptive Krylov-EETD scheme produces the same errors wherever it is stable. However, it shows instabilities for coarse grids and for large values of a. We therefore conclude that the adaptive implementation of Krylov-EETD is the most reliable scheme for this problem.

Runtimes are shown in Figure 16. It can be seen that the adaptive Krylov-EETD has the longest runtimes while ETD-RDP-IF is the fastest. The non-adaptive implementation has runtimes that are slightly higher than those of ETD-RDP-IF. For ETD-RDP-IF and the non-adaptive Krylov-EETD, runtimes remain near constant across all values of a. The same is approximately true for $a < 20$ for the Krylov-EETD scheme. After $a = 1$, the errors of Krylov-EETD are sufficiently lower that it becomes more efficient than the ETD-RDP-IF scheme.

As a increases further, most notably for $a \geq 20$, the runtime of the adaptive implementation of Krylov-EETD increases. However, as noted above, in this regime, the adaptive Krylov-EETD is more reliable than all other implementations. This is due to the adaptive nature of the scheme. The reliability and low errors are obtained at the cost of runtime.



(a) $a = 0.01$ **(b)** $a = 0.5$

(c) $a = 3.0$ **(d)** $a = 100$

Figure 16. Runtime comparison of ETD-RDP-IF with different spatial discretizations and Krylov-EETD for the Schnakenberg ADR model (33). Note that where no values for Krylov-EETD are shown, computation time was infeasible or errors were exceedingly large.

3.5 Brusselator ADR Model

The last model we simulated is the Brusselator ADR model that can be interpreted as a generalization of the Schnakenberg model. According to Kang and Pesin (2005) the Brusselator model was first proposed by Prigogine and Lefever in 1968. The term Brusselator was first used by Tyson (Tyson 1976). Here, we consider an extended Brusselator model described in Bhatt et al. (2018) that also includes an advection term:

$$\begin{aligned}
&\frac{\partial u_1}{\partial t} + a_1(\frac{\partial u_1}{\partial x} + \frac{\partial u_1}{\partial y} + \frac{\partial u_1}{\partial z}) = d_1(\frac{\partial^2 u_1}{\partial x^2} + \frac{\partial^2 u_1}{\partial y^2} + \frac{\partial^2 u_1}{\partial z^2}) + u_1^2 u_2 - (\alpha+1)u_1 + \beta \\
&\frac{\partial u_2}{\partial t} + a_2(\frac{\partial u_2}{\partial x} + \frac{\partial u_2}{\partial y} + \frac{\partial u_2}{\partial z}) = d_2(\frac{\partial^2 u_2}{\partial x^2} + \frac{\partial^2 u_2}{\partial y^2} + \frac{\partial^2 u_2}{\partial z^2}) - u_1^2 u_2 + \alpha u_1 \qquad (34) \\
&\qquad\qquad \Omega = (0;1) \times (0;1) \times (0;1)\,, \qquad t > 0,
\end{aligned}$$

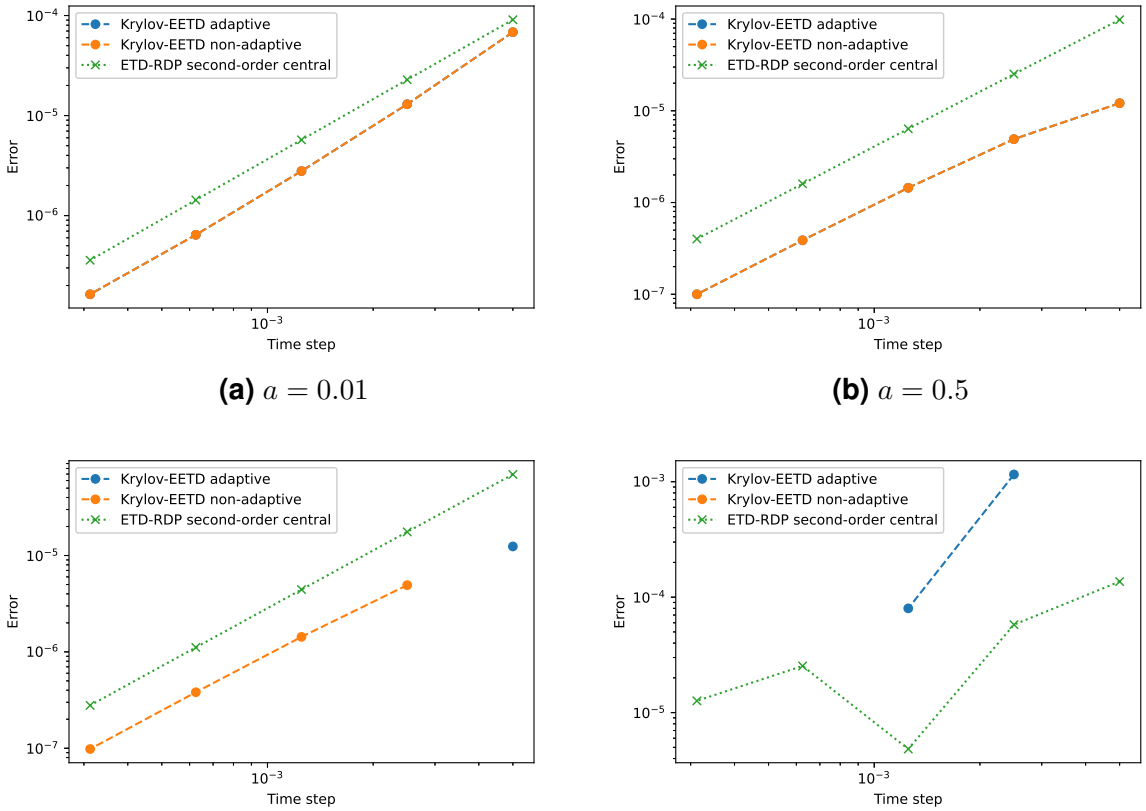
subject to the vanishing normal derivative boundary conditions and initial conditions

$$u_1(x, y, z, 0) = 1 + \sin(2\pi x) \cdot \sin(2\pi y) \cdot \sin(2\pi z)$$

$$u_2(x, y, z, 0) = 3$$

α and β can be interpreted rate constants of the biochemical reactions described by this system (Bhatt et al. 2018).

In order to compare ETD-RDP-IF and Krylov-ETD (Bhatt et al. 2018), we apply the same parameters as the authors do. These are $d_1 = 0.02$, $d_2 = 0.01$, $\alpha = 1.0$, $\beta = 2.0$. We also set $a_1 = a_2 = a$ and varied a to investigate the effect of advection. We simulate up to $t_e = 1.0$ using a fixed spatial grid of $32 \times 32 \times 32$ and varying temporal resolutions.



(a) $a = 0.01$ **(b)** $a = 0.5$

(c) $a = 3.0$ **(d)** $a = 100$

Figure 17. Error comparison of ETD-RDP-IF with central difference discretization and Krylov-EETD for the Brusselator ADR model (34). Note that where no values for Krylov-EETD are shown, computation time was infeasible or errors were exceedingly large.

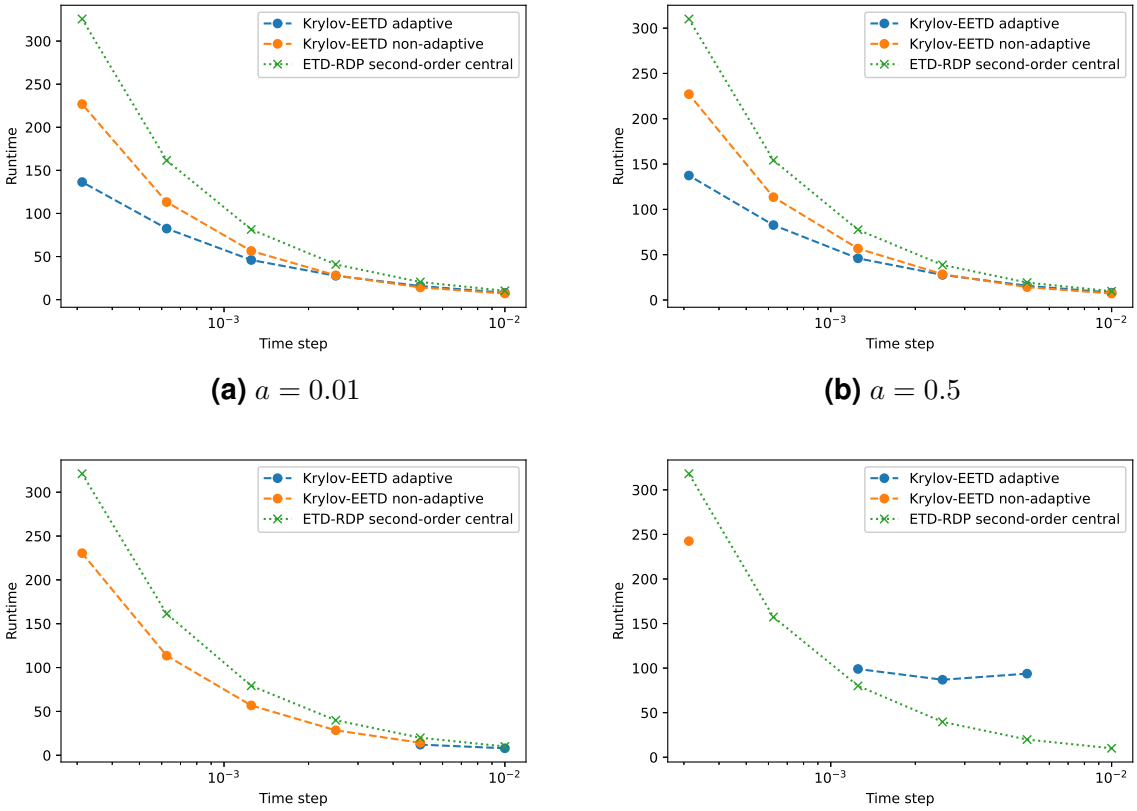
Since there is no closed-form solution to this problem, we estimate the error as described in Section 3.4. The errors for some values of a are shown in Figure 17. Note that we only show the central finite difference discretization of advection here since we have not implemented the upwind-biased discretizations for the vanishing normal derivative boundary condition. No values are reported, like previously, for the adaptive implementation of Krylov-EETD when computation times were suspected to be too high, and for the non-adaptive implementation when errors occurred due to reaching the limits of the floating point representation.

Considering ETD-RDP-IF, we observe errors that are consistently below 10^{-4}. For $a \leq 20$ decreasing with an order of about 2 as the grid is refined. As a increases further, errors remain below 10^{-4} but no clear convergent behavior can be seen. Instead, we see irregular patterns of errors increasing or decreasing as the temporal grid is refined. Since these fluctuations are small, this suggests that the spatial grid is not fine enough for the corresponding problem, so a finer temporal grid does not improve the error.

The errors of Krylov-EETD are below those of ETD-RDP-IF for $a \leq 20$ wherever we were able to obtain solutions. As can be seen, the orders of convergence fluctuate somewhat. However, we observed that they approach 2 as the grid is refined. For $a \geq 100$, the non-adaptive implementation was unable to produce any solutions, and the adaptive version was only able to produce three solutions for $a = 100$ and two for $a = 200$ - such that we obtained two and one error estimates, respectively. These 3 error estimates are greater than those observed with ETD-RDP-IF.

We therefore conclude that ETD-RDP-IF produces larger errors where Krylov-EETD works reliably. However, ETD-RDP-IF produces reliable over a wider range of parameters. This conclusion is subject to one caveat: When the error estimate no longer decreases, the fundamental assumption of convergence required for the error estimate is violated. The errors may therefore be reported incorrectly.

Figure 18 shows the runtimes needed by the different schemes. The runtimes of ETD-RDP-IF remain nearly constant for different values of a, since there is no adaptivity that could influence runtimes significantly. The same holds for the



(a) $a = 0.01$ **(b)** $a = 0.5$

(c) $a = 3.0$ **(d)** $a = 100$

Figure 18. Runtime comparison of ETD-RDP-IF with central difference discretization and Krylov-EETD for the Brusselator ADR model (34). Note that where no values for Krylov-EETD are shown, computation time was infeasible or errors were exceedingly large.

non-adaptive implementation of Krylov-EETD in spite of the remaining adaptivity in `expv`. The non-adaptive Krylov-EETD scheme is consistently faster than ETD-RDP-IF wherever solutions were obtained. Due to the lower errors, it is therefore more efficient for $a \leq 20$. However, it is unreliable for $a \geq 20$.

For $a \leq 1$, the adaptive implementation of Krylov-EETD exhibits the lowest runtimes and is therefore most efficient. For $a \geq 2$, in contrast, we canceled most of the computations due to suspected excessive runtimes. For $a = 100$, multiple computations with different grid sizes finished. However, notably, they showed long runtimes and large errors as shown above. Therefore, we consider this scheme to be unreliable for $a \geq 2$.

4 Conclusion and Future Work

We have described the derivation of the ETD-RDP-IF scheme originally developed in Asante-Asamani (2016) and Asante-Asamani et al. (2020), and extended the derivation and implementation to allow for an advection term. We discretized the advection term in space using a finite difference scheme in order to treat it linearly, like the diffusion, and showed that the derivation of the scheme still holds in this case. Considering the ODE that arises from such a spatial semi-discretization of the ADR system, we showed that the time stepping scheme of ETD-RDP-IF is second-order accurate also for the case of non-regular discretization matrices (system matrices of the ODE).

Numerically, we verified the second-order convergence in time for a linear benchmark problem with smooth initial conditions. We observed an order reduction for non-smooth initial conditions. We observed that the runtime increase introduced by upwind-biased schemes due to larger stencils is limited and the efficiency (error over runtime) is better for for upwind-biased schemes than for the central differencing scheme, in particular for strong advection.

A comparison to the Krylov-EETD scheme showed that, for low advection velocities, Krylov-EETD gives the same or smaller errors while exhibiting better computational performance. However, for large values of a, the computational performance and reliability of Krylov-EETD decrease, making ETD-RDP-IF the better choice. For ETD-RDP-IF, we did not observe any instabilities as a increased. This can most likely be attributed to the L-stability of the scheme (Asante-Asamani 2016).

We conclude from this that the choice of scheme depends on the parameters of the problem, especially the strength of advection. In a parameter regime where Krylov-EETD has been shown to work well, it is more efficient. Outside of these regimes, however, ETD-RDP-IF is more reliable in producing useful results.

There are numerous aspects that warrant further research. Considering accuracy for the time-stepping scheme alone as if it were applied to an ODE is not sufficient to show its convergence behavior for solving PDEs. For PDEs, simultaneous convergence in space and time needs to be considered. This can be done by considering the spatial discretization matrix as a more general linear operator. This includes the limit of the discretization matrix as $h \to 0$. The corresponding exponential can then be investigated in the framework of strongly continuous semigroups. Asante-Asamani (2016) proved second-order accuracy using this framework. However, he assumes that the operator is invertible, and all values in the spectrum have a strictly positive real part. In the case of, e. g., Neumann or periodic boundary conditions, this assumption is violated, the operators are not invertible. We, therefore, loosened the assumption, allowing for positive semi-definite matrices. However, we only allow for matrices of finite and fixed dimension. The next step is to investigate whether a similar proof as in Asante-Asamani (2016) can be achieved for semi-definite operators.

Furthermore, we considered the advection — after spatial semi-discretization — with the linear part of the arising ODE by adding the discretization matrix of the advection to that of the diffusion. Therefore, the advection is treated implicitly in our final scheme. As mentioned by Ascher et al. (1995), the advection term is usually

non-stiff or mildly stiff, so it is generally solved explicitly. In our case, this would mean adding the advection to the reaction function instead of the diffusion matrix. This warrants further investigation with respect to performance and accuracy. Furthermore, this would enable the consideration of non-linear advection terms, e. g., where the advection velocity depends on the concentration of the species.

If advection dominates significantly, as Shampine (1994) notes, the ODE becomes non-stiff. Then, treatment using a fully explicit scheme might be advantageous compared to an ETD scheme.

The runtime performance of the scheme is greatly influenced by the way advection is treated. Considering central differences, the optimizations given by Asante-Asamani et al. (2020) can all be applied since the band structure of the discretization matrix does not change. However, the upwind schemes exhibit a different band structure, therefore preventing some optimizations from being applied. In this work, we only considered non-optimized solvers that do not show these potential differences, in an effort to remain general and produce general implementations. It remains to be investigated how much performance can be gained by optimized solving of the linear systems, and if this outweighs the benefits introduced by upwind schemes.

Besides that, we only considered central differences for the Krylov-ETD scheme. Since the Krylov-subspace approximation does not rely on the band structure of the matrix, the performance penalty of upwind differencing schemes is expected to be small. The potentially higher accuracy might improve the reliability of Krylov-ETD as well. Such a comparison remains to be performed.

Lastly, increasing the Krylov-subspace dimension could yield improvements in reliability of Krylov-EETD.

Bibliography

Akrivis, G., Crouzeix, M., and Makridakis, C. (1999), "Implicit-explicit multistep methods for quasilinear parabolic equations," *Numerische Mathematik* 82.4, pp. 521–541. DOI: 10.1007/s002110050429.

Asante-Asamani, E. O. (2016), "An Exponential Time Differencing Scheme with a Real Distinct Poles Rational Function for Advection-Diffusion Reaction Equations," Doctoral Dissertation, Milwaukee, WI: University of Wisconsin - Milwaukee.

Asante-Asamani, E. O., Khaliq, A. Q. M., and Wade, B. A. (2016), "A real distinct poles Exponential Time Differencing scheme for reactiondiffusion systems," *Journal of Computational and Applied Mathematics* 299, pp. 24–34. DOI: 10.1016/j.cam.2015.09.017.

Asante-Asamani, E. O., Kleefeld, A., and Wade, B. A. (2020), "A second-order exponential time differencing scheme for non-linear reaction-diffusion systems with dimensional splitting," *Journal of Computational Physics* 415, p. 109490. DOI: 10.1016/j.jcp.2020.109490.

Ascher, U. M., Ruuth, S. J., and Wetton, B. T. R. (1995), "Implicit-Explicit Methods for Time-Dependent Partial Differential Equations," *SIAM Journal on Numerical Analysis* 32.3, pp. 797–823. DOI: 10.1137/0732037.

Berestycki, H. (2002), "The Influence of Advection on the Propagation of Fronts in Reaction-Diffusion Equations". *Nonlinear PDEs in Condensed Matter and Reactive Flows*. Ed. by H. Berestycki and Y. Pomeau. NATO Science Series. Dordrecht: Springer Netherlands, pp. 11–48. DOI: 10.1007/978-94-010-0307-0_2.

Bhatt, H. P. and Khaliq, A. Q. M. (2015), "The locally extrapolated exponential time differencing LOD scheme for multidimensional reactiondiffusion systems," *Journal of Computational and Applied Mathematics* 285, pp. 256–278. DOI: 10.1016/j.cam.2015.02.017.

Bhatt, H. P., Khaliq, A. Q. M., and Wade, B. A. (2018), "Efficient Krylov-based exponential time differencing method in application to 3D advection-diffusion-reaction systems," *Applied Mathematics and Computation* 338, pp. 260–273. DOI: 10.1016/j.amc.2018.06.025.

Chapwanya, M., Lubuma, J. M.-S., and Mickens, R. E. (2013), "Nonstandard finite difference schemes for MichaelisMenten type reaction-diffusion equations,"

Numerical Methods for Partial Differential Equations 29.1, pp. 337–360. DOI: 10.1002/num.21733.

Chen, L. Q. and Shen, J. (1998), "Applications of semi-implicit Fourier-spectral method to phase field equations," *Computer Physics Communications* 108.2, pp. 147–158. DOI: 10.1016/S0010-4655(97)00115-X.

Courant, R., Friedrichs, K., and Lewy, H. (1928), "Über die partiellen Differenzengleichungen der mathematischen Physik," *Mathematische Annalen* 100.1, pp. 32–74.

Courant, R., Friedrichs, K., and Lewy, H. (1967), "On the Partial Difference Equations of Mathematical Physics," *IBM Journal of Research and Development* 11.2, pp. 215–234. DOI: 10.1147/rd.112.0215.

Cox, S. M. and Matthews, P. C. (2002), "Exponential Time Differencing for Stiff Systems," *Journal of Computational Physics* 176.2, pp. 430–455. DOI: 10.1006/jcph.2002.6995.

Evans, L. (2010), *Partial Differential Equations (2nd ed.)* Vol. 19. Graduate Studies in Mathematics. ISSN: 1065-7339. Providence, RI: American Mathematical Society. DOI: 10.1090/gsm/019.

Fernandes, R. I. and Fairweather, G. (2012), "An ADI extrapolated Crank-Nicolson orthogonal spline collocation method for nonlinear reaction-diffusion systems," *Journal of Computational Physics* 231.19, pp. 6248–6267. DOI: 10.1016/j.jcp.2012.04.001.

Fromm, J. E. (1968), "A method for reducing dispersion in convective difference schemes," *Journal of Computational Physics* 3.2, pp. 176–189. DOI: 10.1016/0021-9991(68)90015-6.

Gear, C. W. and Kevrekidis, I. G. (2003), "Projective Methods for Stiff Differential Equations: Problems with Gaps in Their Eigenvalue Spectrum," *SIAM Journal on Scientific Computing* 24.4, pp. 1091–1106. DOI: 10.1137/S1064827501388157.

Gommes, C. J. and Tharakan, J. (2020), "The Péclet number of a casino: Diffusion and convection in a gambling context," *American Journal of Physics* 88.6, pp. 439–447. DOI: 10.1119/10.0000957.

Hartman, P. (2002), *Ordinary Differential Equations (2nd ed.)* Classics in Applied Mathematics. Philadelphia, PA: Society for Industrial and Applied Mathematics. DOI: 10.1137/1.9780898719222.

van Herwaarden, O. A. (1994), “Spread of Pollution by Dispersive Groundwater Flow,” *SIAM Journal on Applied Mathematics* 54.1, pp. 26–41. DOI: 10.1137/S0036139992227047.

Higham, N. J. (2005), “The Scaling and Squaring Method for the Matrix Exponential Revisited,” *SIAM Journal on Matrix Analysis and Applications* 26.4, pp. 1179–1193. DOI: 10.1137/04061101X.

Higham, N. J. (2008), *Functions of Matrices*, Other Titles in Applied Mathematics. Philadelphia, PA: Society for Industrial and Applied Mathematics. DOI: 10.1137/1.9780898717778.

Hochbruck, M. and Ostermann, A. (2005), “Explicit Exponential Runge–Kutta Methods for Semilinear Parabolic Problems,” *SIAM Journal on Numerical Analysis* 43.3, pp. 1069–1090. DOI: 10.1137/040611434.

Horn, R. A. and Johnson, C. R. (1991), *Topics in Matrix Analysis*, Cambridge: Cambridge University Press. DOI: 10.1017/CBO9780511840371.

Hundsdorfer, W. and Verwer, J. (2003), *Numerical Solution of Time-Dependent Advection-Diffusion-Reaction Equations*,

James, I. D. (2002), “Modelling pollution dispersion, the ecosystem and water quality in coastal waters: a review,” *Environmental Modelling & Software* 17.4, pp. 363–385. DOI: 10.1016/S1364-8152(01)00080-9.

Johnson, K. A. and Goody, R. S. (2011), “The Original Michaelis Constant: Translation of the 1913 Michaelis-Menten Paper,” *Biochemistry* 50.39, pp. 8264–8269. DOI: 10.1021/bi201284u.

Kang, H. and Pesin, Y. (2005), “Dynamics of a discrete Brusselator model: escape to infinity and Julia set,” *Milan Journal of Mathematics* 73.1, pp. 1–17. DOI: 10.1007/s00032-005-0036-y.

Kassam, A.-K. and Trefethen, L. N. (2005), “Fourth-Order Time-Stepping for Stiff PDEs,” *SIAM Journal on Scientific Computing* 26.4, pp. 1214–1233. DOI: 10.1137/S1064827502410633.

Khaliq, A. Q. M., Martín-Vaquero, J., Wade, B. A., and Yousuf, M. (2009), “Smoothing schemes for reaction-diffusion systems with nonsmooth data,” *Journal of Computational and Applied Mathematics* 223.1, pp. 374–386. DOI: 10.1016/j.cam.2008.01.017.

Kleefeld, B., Khaliq, A. Q. M., and Wade, B. A. (2012), “An ETD Crank-Nicolson method for reaction-diffusion systems,” *Numerical Methods for Partial*

Differential Equations 28.4, pp. 1309–1335.
DOI: https://doi.org/10.1002/num.20682.

Lanser, D. and Verwer, J. G. (1999), "Analysis of operator splitting for advection-diffusion-reaction problems from air pollution modelling," *Journal of Computational and Applied Mathematics* 111.1, pp. 201–216.
DOI: 10.1016/S0377-0427(99)00143-0.

LeVeque, R. (2007), *Finite difference methods for ordinary and partial differential equations - steady-state and time-dependent problems*, Other Titles in Applied Mathematics. Philadelphia, PA: Society for Industrial and Applied Mathematics.

Madzvamuse, A. (2006), "Time-stepping schemes for moving grid finite elements applied to reactiondiffusion systems on fixed and growing domains," *Journal of Computational Physics* 214.1, pp. 239–263.
DOI: 10.1016/j.jcp.2005.09.012.

Mickens, R. E. (2005), "A nonstandard finite difference scheme for a PDE modeling combustion with nonlinear advection and diffusion," *Mathematics and Computers in Simulation* 69.5, pp. 439–446.
DOI: 10.1016/j.matcom.2005.03.008.

Al-Mohy, A. H. and Higham, N. J. (2010), "A New Scaling and Squaring Algorithm for the Matrix Exponential," *SIAM Journal on Matrix Analysis and Applications* 31.3, pp. 970–989. DOI: 10.1137/09074721X.

Moler, C. and Van Loan, C. (2003), "Nineteen Dubious Ways to Compute the Exponential of a Matrix, Twenty-Five Years Later," *SIAM Review* 45.1, pp. 3–49. DOI: 10.1137/S00361445024180.

Saad, Y. (2011), *Numerical Methods for Large Eigenvalue Problems*, Classics in Applied Mathematics. Philadelphia, PA: Society for Industrial and Applied Mathematics. DOI: 10.1137/1.9781611970739.

Schnakenberg, J. (1979), "Simple chemical reaction systems with limit cycle behaviour," *Journal of Theoretical Biology* 81.3, pp. 389–400.
DOI: 10.1016/0022-5193(79)90042-0.

Shampine, L. F. (1994), "ODE solvers and the method of lines," *Numerical Methods for Partial Differential Equations* 10.6, pp. 739–755.
DOI: 10.1002/num.1690100608.

Sidje, R. B. (1998), "Expokit: a software package for computing matrix exponentials," *ACM Transactions on Mathematical Software* 24.1, pp. 130–156.
DOI: 10.1145/285861.285868.

Strikwerda, J. C. (2004), *Finite Difference Schemes and Partial Differential Equations (2nd ed.)* Other Titles in Applied Mathematics. Philadelphia, PA: Society for Industrial and Applied Mathematics.

Turing, A. M. (1952), "The chemical basis of morphogenesis," *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences* 237.641, pp. 37–72. DOI: 10.1098/rstb.1952.0012.

Tyson, J. J. (1976), *The Belousov-Zhabotinskii Reaction*, Lecture Notes in Biomathematics. Berlin, Heidelberg: Springer-Verlag.

Tyson, R., Lubkin, S. R., and Murray, J. D. (1999), "Model and analysis of chemotactic bacterial patterns in a liquid medium," *Journal of Mathematical Biology* 38.4, pp. 359–375. DOI: 10.1007/s002850050153.

Van Loan, C. F. (2000), "The ubiquitous Kronecker product," *Journal of Computational and Applied Mathematics* 123.1, pp. 85–100. DOI: 10.1016/S0377-0427(00)00393-9.

Versteeg, H. and Malalasekera, W. (2007), *An Introduction to Computational Fluid Dynamics: The Finite Volume Method (2nd ed.)* Harlow: Pearson Education Limited.

Verwer, J. G., Hundsdorfer, W. H., and Blom, J. G. (2002), "Numerical Time Integration for Air Pollution Models," *Surveys on Mathematics for Industry* 10, pp. 107–174.

Voss, D. A. and Khaliq, A. Q. M. (1996), "Time-stepping algorithms for semidiscretized linear parabolic PDEs based on rational approximants with distinct real poles," *Advances in Computational Mathematics* 6.1, pp. 353–363. DOI: 10.1007/BF02127713.

Voss, D. A. and Khaliq, A. Q. M. (1999), "A linearly implicit predictor-corrector method for reaction-diffusion equations," *Computers & Mathematics with Applications* 38.11, pp. 207–216. DOI: 10.1016/S0898-1221(99)00299-0.

Weissler, F. B. (1979), "Semilinear evolution equations in Banach spaces," *Journal of Functional Analysis* 32.3, pp. 277–296. DOI: 10.1016/0022-1236(79)90040-5.

Yousuf, M., Khaliq, A. Q., and Kleefeld, B. (2012), "The numerical approximation of nonlinear Black-Scholes model for exotic path-dependent American options with transaction cost," *International Journal of Computer Mathematics* 89.9, pp. 1239–1254. DOI: 10.1080/00207160.2012.688115.

Zheng, S. (2004), *Nonlinear Evolution Equations*, New York, NY: Chapman and Hall/CRC. DOI: 10.1201/9780203492222.

Biographical Sketch

Björn Müller was born in Würzburg on December 17, 1998. He graduated from FH Aachen University of Applied Sciences in 2020 with a Bachelor of Science degree in Scientific Programming. In 2020, he began his graduate studies in pursuit of a Master of Science degree in Applied Mathematics and Informatics at FH Aachen University of Applied Sciences before joining the University of Louisiana at Lafayette in the Fall of 2021. He graduated in the Fall of 2022 from the University of Louisiana at Lafayette with a Master of Science degree in Mathematics.