Universidad de Concepción
Dirección de Postgrado
Facultad de Ciencias Físicas y Matemáticas
Programa de Doctorado en Ciencias Aplicadas
con Mención en Ingeniería Matemática

IMPLICIT-EXPLICIT METHODS FOR NONLINEAR AND NONLOCAL CONVECTION-DIFFUSION-REACTION PROBLEMS

(MÉTODOS IMPLÍCITOS-EXPLÍCITOS PARA PROBLEMAS DE CONVECCIÓN-DIFUSIÓN-REACCIÓN NO LINEALES Y NO LOCALES)

Tesis para optar al grado de Doctor en Ciencias
Aplicadas con mención en Ingeniería Matemática

Daniel Eduardo Inzunza Herrera

concepción-chile
2019

Profesor Guía: Raimund Bürger
CI^{2}MA y Departamento de Ingeniería Matemática
Universidad de Concepción, Chile

Cotutor: Pep Mulet Mestre
Departament de Matemàtiques
Universitat de València, España

Cotutor: Luis Miguel Villada Osorio
GIMNAP–Departamento de Matemática y CI^{2}MA
Universidad del Bío-Bío y Universidad de Concepción, Chile

Implicit-explicit methods for nonlinear and nonlocal convection-diffusion-reaction problems

Daniel Eduardo Inzunza Herrera

Directores de Tesis: Raimund Bürger, Universidad de Concepción, Chile.
Pep Mulet, Universitat de Valencia.
Luis Miguel Villada, Universidad del Bío-Bío, Chile.

Director de Programa: Rodolfo Rodriguez, Universidad de Concepción, Chile.

Comisión evaluadora

Prof. Paola Goatin, Inria Sophia Antipolis - Méditerranée, France.

Prof. Dante Kalise, University of Nottingham, United Kingdom.

Prof. Jorge E. Macías-Díaz, Universidad Autónoma de Aguascalientes, México.

Prof. Carlos E. Mejia, Universidad Nacional de Colombia, Colombia.

Comisión examinadora

Firma: _____
Prof. Raimund Bürger, Universidad de Concepción, Chile.

Firma: _____
Prof. Mauricio Sepulveda, Universidad de Concepción, Chile.

Firma: _____
Prof. Héctor Torres, Universidad de la Serena, Chile.

Firma: _____
Prof. Luis Miguel Villada, Universidad del Bío-Bío, Chile.

Calificación: _____

Concepción, 16 de Diciembre de 2019

Abstract

In this thesis, high-order numerical methods are used to approximate the solution of nonlinear and nonlocal equations with gradient-type flow structure. Specifically, numerical schemes are proposed for aggregation models and for convection-diffusion problems. The thesis has the following objectives.

The first objective of this thesis is to propose a high-order scheme for a non-linear and non-local equation with gradient flow, analyzing its properties and applications for both the one-dimensional and the multidimensional case.

The second objective of this thesis is to show that the implicit-explicit Runge-Kutta (IMEX-RK) schemes allow to obtain an efficient numerical solution of both the generated error and also the CPU time for convection-diffusion problems with nonlocal and nolinear terms. These schemes consist in handling the convective part by treating Runge-Kutta schemes, and the diffusive part by implicit schemes. For the latter, by discretizing the resulting implicit scheme, a system of nonlinear equations is obtained, which solved by the Newton-Raphson method with descent algorithm. The obtained scheme allows a less restrictive CFL condition compared with an explicit scheme.

The third objective of this thesis is to show an application of high-order schemes to population dynamics and pedestrian movement models. It turns out that for coarse discretizations of the computational mesh the numerical solutions obtained are more sharply resolved than those obtained with first-order schemes.

Resumen

En este trabajo de tesis se desarrollan métodos numéricos de alto orden para aproximar la solución de ecuaciones no lineales y no locales con estructura de flujo de tipo gradiente. Especificamente se plantean esquemas numéricos para modelos de agregación y para problemas de convección-difusión. La tesis tiene los siguientes objetivos.

El primer objetivo de esta tesis es plantear un esquema de alto orden para un ecuación no lineal y no local con flujo de tipo gradiente, analizando sus propiedaes y aplicaciones tanto para el caso unidimensional como para el vaso multi-dimensional.

El segundo objetivo de esta tesis es mostrar que los esquemas Implícitos-Explícitos Runge-Kutta (IMEX-RK) permiten obtener una solución numérica eficiente tanto del error generado como también del tiempo de cálculo computacional para los problemas de convección-difusión con términos no locales y no lineales. Estos esquemas consisten el trabajar la parte convectiva mediante tratamiento de esquemas Runge-Kutta, y la parte difusiva mediante esquemas implícitos. Para esta última, al discretizar el esquema implícito resultante, se obtiene un sistema de ecuaciones no lineal, el cual se resuelve mediante el método de Newton-Raphson con algoritmo de descenso. El esquema resultante obtiene una condición CFL menos restructiva en comparación con un esquema explícto.

El tercer objetivo de esta tesis es mostrar una aplicación de los esquemas de alto orden a los modelos de dinamica de poblaciones y movimiento de peatones, mostrando que para discretizaciones gruesas de la malla computacional las soluciones numéricas obtenidas tienen mejor resolución comparadas con las que se obtienen con esquemas de primer orden.

Agradecimientos

Quiero agredecer a mi director de tesis, profesor Raimund Bürger por su constante apoyo y preocupación tanto en instancias académicas como también personales. Su disposición, buena voluntad y trabajo son dignas de imitar.

A mi co-tutor, Pep Mulet por aceptarme para realizar esta tesis bajo su co-dirección, su continuo apoyo, multiples consejos, generosidad y excelente disposición en todo ámbito. Muchos de los avances más importantes de esta tesis se obtuvieron trabajando a su lado.

A mi co-tutor Luis Miguel Villada por su gran paciencia y por estar siempre ayudándome a la "vuelta de un llamado o mensaje" cuando lo necesité.

Al Centro de Investigación en Ingeniería Matemática (CI^2MA) de la Universidad de Concepción, por brindarme el espacio y las instalaciones para poder trabajar comodamente. Al personal administrativo tanto pasado como actual, principalmente a la Sra. Lorena por recibirme siempre con una sonrisa en las mañanas.

A mis compañeros del programa doctorado por todas esas conversaciones, sobremesas y cafés que compartimos.

A mis padres Daniel y Gloria, a mis hermanos Loreto y Edison; por estar siempre a mi lado entregando su amor y cariño incondicional. A mi sobrina Josefa (la "Jo") por enseñarme un tipo de amor que desconocía.

Agradezco también a todas aquellas personas que de alguna u otra forma me ayudaron en este camino: Edgardo Olate, Luis Sánchez (q.e.p.d.), Joaquín Fernandez, Gonzalo Rivera, Ramiro Rebolledo, Mari Carmen Martí, Daniela Mena, Maribel Cuevas y a los que se me puedan quedar en el tintero.

Agradecer en especial a Elvis Gavilán por su constante apoyo (y también insistencia) para que yo lograra ingresar al doctorado. Sin lugar a dudas, eres el causante de que hoy yo esté escribiendo esto.

Agradecer además a todas las instituciones que permitieron realizar este trabajo: a la Comisión Nacional de Ciencia y Tecnología (CONICYT) a través de su programa de becas CONICYT-PCHA/Doctorado Nacional/2014-21140362 y su programa CONICYT/PAI/Concurso Apoyo a Centros Científicos y Tecnológicos de Excelencia con Financiamiento Basal AFB-170001 del Centro de Modelamiento Matemático (CMM) de la Universidad de Chile; a Red

Doctoral Ciencia, Tecnología y Medio Ambiente (REDOCT.CTA); a INRIA Associate Team "Efficient numerical schemes for non-local transport phenomena" (NOLOCO; 2018-2020); al Centro de Recursos Hídrico para la Agricultura y Minería (CRHIAM).

Finalmente, quiero agradecer de todo corazón a mi esposa Yasna. Tu dedicación, compromiso y amor son los motivos por los que me levanto en las mañanas. Te amo.

Daniel Eduardo Inzunza Herrera

Contents

List of Tables

List of Figures

Introduction

A class of nonlinear nonlocal gradient flow equations

A nonlinear nonlocal equations with a gradient flow structure is a nonlinear convection–diffusion equations with nonlocal flux and possibly degenerate diffusion of the type

$$u_t + \nabla \cdot \big(u\nabla(V(\boldsymbol{x}) + W * u)\big) = \nabla \cdot \big(u\nabla(H'(u))\big), \quad \boldsymbol{x} \in \mathbb{R}^d, \quad t > 0, \tag{0.1}$$

$$u(\boldsymbol{x}, 0) = u_0(\boldsymbol{x}), \quad \boldsymbol{x} \in \mathbb{R}^d, \tag{0.2}$$

where $u(\boldsymbol{x}, t) \geq 0$ is an unknown probability distribution function or population density, $W(\boldsymbol{x})$ is an interaction potential, which is assumed to be symmetric, and $H(u)$ is a density of internal energy and where

$$\big(W * u(\cdot, t)\big)(\boldsymbol{x}) = \int_{\mathbb{R}^d} W(\boldsymbol{y})u(\boldsymbol{x} - \boldsymbol{y}, t)\,\mathrm{d}\boldsymbol{y} = \int_{\mathbb{R}^d} W(\boldsymbol{x} - \boldsymbol{y})u(\boldsymbol{y}, t)\,\mathrm{d}\boldsymbol{y}. \tag{0.3}$$

Equations such as (0.1) appear in various contexts including interacting gases, porous media flows and collective behavior in biology (see Section 1.1.2 and [21] for references). Clearly, if $W = 0$ and $H(u) = u \log u - u$ or $H(u) = u^m$, the classical heat equation and porous medium/fast diffusion equation are recovered, respectively [58]. The function W is related to the interaction energy (see below), and may be as singular as the Newtonian potential in the chemotaxis system [38] or as smooth as $W(\boldsymbol{x}) = |\boldsymbol{x}|^\alpha$ with $\alpha > 2$ in granular flow [5]. Their numerical solution by an explicit finite difference method is costly due to the necessity of discretizing a local spatial convolution for each evaluation of the convective numerical flux, and due to the disadvantageous Courant–Friedrichs–Lewy (CFL) condition incurred by the diffusion term [21].

One of the purposes in this thesis is to discuss techniques for solving numerically (0.1) that perform better in terms of resolution accuracy and efficiency. Based on explicit schemes for such models devised in the study of Carrillo et al., we propose an implicit-explicit Runge-Kutta method (IMEX-RK method) that treats the diffusive term implicitly and the convective term explicitly. This method avoids the restrictive time step limitation of explicit schemes since the diffusion term is handled implicitly, but entails the necessity to solve nonlinear algebraic

systems in every time step. To explain the main idea, we consider the problem

$$\frac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}t} = \boldsymbol{\mathcal{C}}(\boldsymbol{u}) + \boldsymbol{\mathcal{D}}(\boldsymbol{u}), \tag{0.4}$$

which is assumed to represent a method-of-lines semi-discretization of (0.1), where $\boldsymbol{u} = \boldsymbol{u}(t)$ is a spatial discretization of the solution and $\boldsymbol{\mathcal{C}}(\boldsymbol{u})$ and $\boldsymbol{\mathcal{D}}(\boldsymbol{u})$ are discretizations of the convective and diffusive terms, respectively. Implicit treatment of both $\boldsymbol{\mathcal{C}}(\boldsymbol{u})$ and $\boldsymbol{\mathcal{D}}(\boldsymbol{u})$ would remove any stability restriction on Δt. However, the upwind nonlinear discretization of the convective terms contained in $\boldsymbol{\mathcal{C}}(\boldsymbol{u})$ that is needed for stability, makes its implicit treatment extremely involved. In fact, numerical integrators that deal implicitly with $\boldsymbol{\mathcal{D}}(\boldsymbol{u})$ and explicitly with $\boldsymbol{\mathcal{C}}(\boldsymbol{u})$ can be used with a time step restriction dictated by the convective term alone. These schemes, apart from having been profusely used in convection-diffusion problems and convection problems with stiff reaction term [3,30], have been recently used to deal with stiff terms in hyperbolic systems with relaxation [11–14,51].

For this thesis we use the integrators given by the pair of Butcher arrays:

- $$\begin{array}{c|c} \boldsymbol{c} & \boldsymbol{A} \\ \hline & \boldsymbol{b}^{\mathrm{T}} \end{array} = \begin{array}{c|cc} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 1/2 \\ \hline & 1/2 & 1/2 \end{array}, \qquad \begin{array}{c|c} \tilde{\boldsymbol{c}} & \tilde{\boldsymbol{A}} \\ \hline & \tilde{\boldsymbol{b}}^{\mathrm{T}} \end{array} = \begin{array}{c|cc} 0 & 0 & 0 \\ 1 & 1 & 0 \\ \hline & 1/2 & 1/2 \end{array}, \tag{0.5}$$

  denoted by H-CN(2,2,2) in [9] since it is a natural choice when dealing with convection-diffusion problems, since Heun's method is an SSP explicit RK one [32], and the Crank-Nicolson method is A-stable and widely used for diffusion problems.

- The classical second-order IMEX-RK method

$$\begin{array}{c|c} \boldsymbol{c} & \boldsymbol{A} \\ \hline & \boldsymbol{b}^{\mathrm{T}} \end{array} = \begin{array}{c|ccc} 1/4 & 1/4 & 0 & 0 \\ 1/4 & 0 & 1/4 & 0 \\ 1 & 1/3 & 1/2 & 1/3 \\ \hline & 1/3 & 1/3 & 1/3 \end{array}, \qquad \begin{array}{c|c} \tilde{\boldsymbol{c}} & \tilde{\boldsymbol{A}} \\ \hline & \tilde{\boldsymbol{b}}^{\mathrm{T}} \end{array} = \begin{array}{c|ccc} 0 & 0 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 \\ 1 & 1/2 & 1/2 & 0 \\ \hline & 1/3 & 1/3 & 1/3 \end{array} \tag{0.6}$$

  due to Pareschi and Russo [51], denoted by IMEX-SSP2(3,3,2) as in [9].

- The third-order scheme

$$\begin{array}{c|c} \boldsymbol{c} & \boldsymbol{A} \\ \hline & \boldsymbol{b}^{\mathrm{T}} \end{array} = \begin{array}{c|ccccc} \alpha & \alpha & 0 & 0 & 0 \\ 0 & -\alpha & \alpha & 0 & 0 \\ 1 & 0 & 1-\alpha & \alpha & 0 \\ 1/2 & \beta & \eta & 1/2-\beta-\eta-\alpha & \alpha \\ \hline & 0 & 1/6 & 1/6 & 2/3 \end{array}, \qquad \begin{array}{c|c} \tilde{\boldsymbol{c}} & \tilde{\boldsymbol{A}} \\ \hline & \tilde{\boldsymbol{b}}^{\mathrm{T}} \end{array} = \begin{array}{c|cccc} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1/2 & 0 & 1/4 & 1/4 & 0 \\ \hline & 0 & 1/6 & 1/6 & 2/3, \end{array} \tag{0.7}$$

  where $\alpha = 0.24169426078821, \quad \beta = \alpha/4, \quad \eta = 0.12915286960590,$

  also introduced in [51], and which is here denoted by IMEX-SSP3(4,3,3) following [10].

Crowd dynamics and pedestrians movement

A moving crowd is described by its density $\rho = \rho(\boldsymbol{x}, t)$. In standard situations, the total number of individuals is constant, so that conservation laws of the type

$$\partial_t \rho + \operatorname{div}_{\boldsymbol{x}} (\rho \boldsymbol{V}) = 0 \tag{0.8}$$

are the natural tool for the description of the crowd dynamics. The choice of the velocity $\boldsymbol{V}$ is key to describe the target of the pedestrian and their speed, in addition to adapt their path choice to the crowd density they estimate to find along this path. The vector $\boldsymbol{V}$ is, in general, a function of the space coordinate $\boldsymbol{x}$ in a domain Ω and of the density ρ.

The existence and stability of the solutions of (0.8) are given in [1]. Here the authors consider a Lax-Friedrichs type algorithm that yields a sequence of approximate solutions to (0.8) that, up to a subsequence, converges to a weak entropy solution (see [1, Theorem 2.2]). In addition they obtain bounds for the stability of the solution.

Equation (0.8) is used to represent physical phenomena in biology, sedimentation, vehicular traffic, supply chains, granular materials, vortex dynamics and crowd dynamics (see [25]). We mention that, in some cases, due to its nonlocal nature, this equation is particularly suitable for describing the behavior of crowds, where each member moves according to her/his evaluation of the crowd density and its variations within her/his view horizon.

Our purpose is to propose a high-order WENO scheme that approximates the solution of (0.8), and to show through examples the advantages of using high-order scheme both to capture the numerical solutions as well as the evacuation times.

Organization of this thesis

The present thesis is organized as follows:

In **Chapter 1**, for the case $d = 1$ and $V \equiv 0$, we propose IMEX methods that numerically solves (0.1) and we prove that these methods are well defined. Numerical experiments illustrate that, for fine discretizations, it is more efficient in terms of reduction of error versus CPU time than the original explicit method. One of the test cases is given by a strongly degenerate parabolic, nonlocal equation modeling aggregation studied in [7]. This model can be transformed to a local partial differential equation that can be solved numerically easily to generate a reference solution for the IMEX-RK method, but it is limited to one space dimension.

The contents of Chapter 1 correspond to the article [18]:

- R. Bürger, D. Inzunza, P. Mulet, L.M. Villada, Implicit-explicit schemes for nonlinear nonlocal equations with a gradient flow structure in one space dimension, *Numer. Methods Partial Differential Equations* 35 (2019) 1008–1034.

In **Chapter 2**, we focus in the multi-dimensional case. The resulting IMEX-RK methods require solving nonlinear algebraic systems at every time step. It is proven, for a general number of space dimensions, that this method is well defined. Numerical experiments for spatially two-dimensional problems motivated by models of collective behaviour are conducted with several alternative choices of the pair of Runge-Kutta schemes defining an IMEX-RK method. For fine discretizations, IMEX-RK methods turn out to be more efficient in terms of reduction of error versus CPU time than the original explicit method.

The contents of Chapter 2 correspond to the article [17]:

- R. Bürger, D. Inzunza, P. Mulet and L. M. Villada, Implicit-explicit methods for a class of nonlinear nonlocal gradient flow equations modelling collective behavior, *Applied Numerical Mathematics* 144 (2019) 234–252.

In **Chapter 3**, for the case $d = 2$ we propose a finite difference WENO schemes for (0.8). Numerical experiments show that for coarse discretizations the finite difference WENO scheme better captures both the numerical solution and evacuation time and also the formation of lanes.

The contents of Chapter 3 correspond to research:

- R. Bürger, D. Inzunza and L. M. Villada, High-order finite-difference WENO schemes for models of crowd dynamics *(in preparation)*.

Introducción

Una clase de ecuación no lineal y no local con flujo gradiente

Una ecuación no lineal y no local con una estructura de tipo gradiente es una ecuación de convección-difusión no lineal con flujo no local y posiblemente con difusión degenerada del tipo

$$u_t + \nabla \cdot \big(u\nabla(V(\boldsymbol{x}) + W * u)\big) = \nabla \cdot \big(u\nabla(H'(u))\big), \quad \boldsymbol{x} \in \mathbb{R}^d, \quad t > 0, \tag{0.9}$$

$$u(\boldsymbol{x}, 0) = u_0(\boldsymbol{x}), \quad \boldsymbol{x} \in \mathbb{R}^d, \tag{0.10}$$

donde $u(\boldsymbol{x}, t) \geq 0$ es una función de distribución de probabilidad o densidad de población no conocida, $W(\boldsymbol{x})$ es un potencial de interacción, el cual se asume simétrico, y $H(u)$ es un densidad de energía interna y donde

$$\big(W * u(\cdot, t)\big)(\boldsymbol{x}) = \int_{\mathbb{R}^d} W(\boldsymbol{y})u(\boldsymbol{x} - \boldsymbol{y}, t)\,\mathrm{d}\boldsymbol{y} = \int_{\mathbb{R}^d} W(\boldsymbol{x} - \boldsymbol{y})u(\boldsymbol{y}, t)\,\mathrm{d}\boldsymbol{y}. \tag{0.11}$$

Ecuaciones como (0.1) aparecen en varios contextos incluyendo interacción de gases, flujos de medios porosos y comportamiento colectivo en biología (ver Sección 1.1.2 y [21] para referencias). Claramente, si $W = 0$ y $H(u) = u \log u - u$ o $H(u) = u^m$, se obtienen la ecuación del calor clásica y la ecuación de difusión porosa medio/rápida, respecticamente [58]. La función W está relacionada a la interacción de energía (ver más abajo), y puede ser tan singular como el potencial de Newton en sistemas de quimiotaxis [38] o tan suave como $W(\boldsymbol{x}) = |\boldsymbol{x}|^\alpha$ con $\alpha > 2$ en flujos granulares [5]. Su solución numérica mediante un método de diferencias finitas explícito es costosa debido a la necesidad de discretizar una convolución espacial local para cada evaluación del flujo numérico convectivo, y debido a la condición de Courant – Friedrichs – Lewy (CFL) desventajosa en que se incurre por el término de difusión [21].

Uno de los propósitos de esta tesis es discutir técnicas que resuelvan numeéricamente (0.1) que resulten mejor en términos de precisión de resolución y eficiencia. Basados en esquemas explícitos para tales modelos planteados en los estudios de Carrillo y compañia proponemos un método implícito-explícito de Runge-Kutta (método IMEX-RK) que trata el término difusivo implícitamente y el término convectivo de forma explícita. Este método evita la limitación restrictiva del paso de tiempo de los esquemas explícitos ya que el término de difusión se maneja implícitamente, pero conlleva la necesidad de resolver sistemas algebraicos no lineales

en cada paso de tiempo. Para explicar la idea principal, consideramos el problema

$$\frac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}t} = \boldsymbol{\mathcal{C}}(\boldsymbol{u}) + \boldsymbol{\mathcal{D}}(\boldsymbol{u}), \tag{0.12}$$

que se asume que representa una semi-discretización del método de lineas de (0.1), donde $\boldsymbol{u} = \boldsymbol{u}(t)$ es la discretización espacial del la solución y $\boldsymbol{\mathcal{C}}(\boldsymbol{u})$ y $\boldsymbol{\mathcal{D}}(\boldsymbol{u})$ son las discretizaciones de los términos convectivo y difusivo respectivamente. Supongamos, por simplicidad, que el ancho de la malla espacial es $\Delta x > 0$. Entonces la restricción de estabilidad sobre el tiempo Δt que el esquema explícito impone cuando se aplica (0.4) es muy severa (Δt debe ser proporcional al cuadrado de Δx^2 sobre el espacio de malla), debido a la presencia de $\boldsymbol{\mathcal{D}}(\boldsymbol{u})$. El tratamiento implícito de ambos $\boldsymbol{\mathcal{C}}(\boldsymbol{u})$ y $\boldsymbol{\mathcal{D}}(\boldsymbol{u})$ removerá toda restricción de estabilidad sobre Δt. Sin embargo, la discretización no lineal upwind del término convectivo contenido en $\boldsymbol{\mathcal{C}}(\boldsymbol{u})$ que es necesario para la estabilidad hace que el tratamiento implícito sea extremadamente complicado. De hecho, después del pionero trabajo de Crouzeix [27], integradores numércios que tratan implícitamente con $\boldsymbol{\mathcal{D}}(\boldsymbol{u})$ y explícitamente con $\boldsymbol{\mathcal{C}}(\boldsymbol{u})$ se pueden usar con una restricción en el paso del tiempo dictada solo por el término convectivo. Estos esquemas, además de ser profundamente usados en problemas de convección-difusión con término de reacción rígido [3, 30], se han utilizado recientemente para tratar términos rígidos en sistemas hiperbólicos con relajación [11–14, 51].

Para esta tesis, usamos los integradores numéricos definidos por los siguientes arreglos de Butcher:

- $$\begin{array}{c|c} \boldsymbol{c} & \boldsymbol{A} \\ \hline & \boldsymbol{b}^{\mathrm{T}} \end{array} = \begin{array}{c|cc} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 1/2 \\ \hline & 1/2 & 1/2 \end{array}, \qquad \begin{array}{c|c} \tilde{\boldsymbol{c}} & \tilde{\boldsymbol{A}} \\ \hline & \tilde{\boldsymbol{b}}^{\mathrm{T}} \end{array} = \begin{array}{c|cc} 0 & 0 & 0 \\ 1 & 1 & 0 \\ \hline & 1/2 & 1/2 \end{array}, \tag{0.13}$$

  denotado por H-CN(2,2,2) en [9] ya que es la elección natural cuando se trabaja con problemas de convección-difusión, pues el método de Heun es RK-SSP explícito [32], y el método de Crank-Nicolson es A-estable y es ampliamente usado para problemas de difusión.

- El clásico metodo IMEX-RK de segundo orden

$$\begin{array}{c|c} \boldsymbol{c} & \boldsymbol{A} \\ \hline & \boldsymbol{b}^{\mathrm{T}} \end{array} = \begin{array}{c|ccc} 1/4 & 1/4 & 0 & 0 \\ 1/4 & 0 & 1/4 & 0 \\ 1 & 1/3 & 1/2 & 1/3 \\ \hline & 1/3 & 1/3 & 1/3 \end{array}, \qquad \begin{array}{c|c} \tilde{\boldsymbol{c}} & \tilde{\boldsymbol{A}} \\ \hline & \tilde{\boldsymbol{b}}^{\mathrm{T}} \end{array} = \begin{array}{c|ccc} 0 & 0 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 \\ 1 & 1/2 & 1/2 & 0 \\ \hline & 1/3 & 1/3 & 1/3 \end{array} \tag{0.14}$$

  debido a Pareschi y Russo [51], denotado por IMEX-SSP2(3,3,2) como en [9].

- El esquema de tercer orden

$$\begin{array}{c|c} \boldsymbol{c} & \boldsymbol{A} \\ \hline & \boldsymbol{b}^{\mathrm{T}} \end{array} = \begin{array}{c|cccc} \alpha & \alpha & 0 & 0 & 0 \\ 0 & -\alpha & \alpha & 0 & 0 \\ 1 & 0 & 1-\alpha & \alpha & 0 \\ 1/2 & \beta & \eta & 1/2-\beta-\eta-\alpha & \alpha \\ \hline & 0 & 1/6 & 1/6 & 2/3 \end{array}, \quad \begin{array}{c|c} \tilde{\boldsymbol{c}} & \tilde{\boldsymbol{A}} \\ \hline & \tilde{\boldsymbol{b}}^{\mathrm{T}} \end{array} = \begin{array}{c|cccc} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1/2 & 0 & 1/4 & 1/4 & 0 \\ \hline & 0 & 1/6 & 1/6 & 2/3, \end{array} \tag{0.15}$$

  donde $\alpha = 0.24169426078821$, $\beta = \alpha/4$, $\eta = 0.12915286960590$, es también introducido en [51], y que se denota por IMEX-SSP3(4,3,3) después de [10].

Dinámica de poblaciones y movimiento de peatones

El movimiento de poblaciones es descrito por su densidad $\rho = \rho(\boldsymbol{x}, t)$. En situaciones normales, el número total de individuos es constante, de modo que la ley de conservación del tipo

$$\partial_t \rho + \operatorname{div}_{\boldsymbol{x}} (\rho \boldsymbol{V}) = 0 \tag{0.16}$$

es la herramienta natural para la descripción de la dinámica de la población. La elección de la velocidad $\boldsymbol{V}$ es clave para describir el objetivo del peatón y su velocidad, además de adaptar su elección del camino a la estimación de la densidad de la población que encuentra a lo largo de esta ruta. El vector $\boldsymbol{V}$ es, en general, una función de la coordenada $\boldsymbol{x}$ en un dominio Ω, y de densidad ρ.

La existencia y la estabilidad de las solución de (0.16) se encuentra en [1]. Aquí los autores consideran un algoritmo de tipo Lax-Friedrichs que produce una sucesión de soluciones aproximaciones de (0.16) que, a través de una subsucesión, converge a su solución de débil de entropía (ver [1, Teorema 2.2]), además obtienen cotas para la estabilidad de la solución.

La ecuación (0.16) es usada para representar fenḿenos físicos en biologá, sedimentación, tráfico vehicular, cadena de suministros, materiales granulados, dinámica de vórtices y dinámica de poblaciones. Sobre esto último mencionamos que, en algunos casos, debido a la naturaleza no local de esta ecuación es particularmente adecuada para describir el comportamiento de problaciones.

Nuestro propósito es mostrar un esquema WENO de alto orden que se aproxime a la solución de (0.16), y mostrar a través de ejemplos las ventajas de usar un esquema de alto orden tanto para capturar las soluciones numéricas como los tiempos de evacuación.

Organización de esta tesis

La presente tesis se organiza como sigue:

En el **capítulo 1**, para el caso $d = 1$ y $V \equiv 0$, proponemos un método implícito-explícito que resuelve numéricamente la ecuación (0.1) y analizamos sus propiedades obteniendo condiciones de positividad tanto para el esquema explícito como también para el esquema IMEX-RK. Expermientos numéricos ilustran que para discretizaciones finas es más eficiente en términos de reducción del error versus el tiempo de CPU que el método explícito original. Uno de los ejemplos, estudiado en [7], es dado por una ecuación parabólica no local, fuertemente degenerada que modela agregación. Este modelo se puede transformar en una ecuación diferencial parcial no local que se puede facilmente resolver numéricamente para generar una solución de referencia para el método IMEX-RK, pero se limita solo al caso de dimensión uno.

Los contenidos del **Capítulo 1** corresponden al artículo [18]:

- R. Bürger, D. Inzunza, P. Mulet, L.M. Villada, Implicit-explicit schemes for nonlinear nonlocal equations with a gradient flow structure in one space dimension, *Numer. Methods Partial Differential Equations* 35 (2019) 1008–1034.

En el **Capítulo 2** nos enfocamos en el caso multi-dimensional con $V \neq 0$. El método IMEX-RK resultante requiere resolver un sistema algebraico no lineal en cada paso de tiempo. Se demuestra, para un número general de dimensiones espaciales, este método es bien definido. Los experimentos numéricos para problemas de dimensión espacial dos, motivados por modelos de comportamiento colectivo, se realizan con varias opciones alternativas del par de esquemas Runge-Kutta que definen un método IMEX-RK. Para discretizaciones finas, los métodos IMEX-RK resultan más eficientes en términos de reducción de error versus tiempo de CPU que el método explícito original.

Los contenidos del **Capítulo 2** corresponden al artículo [17]:

- R. Bürger, D. Inzunza, P. Mulet and L. M. Villada, Implicit-explicit methods for a class of nonlinear nonlocal gradient flow equations modelling collective behavior, *Applied Numerical Mathematics* 144 (2019) 234–253.

En el **Capítulo 3** para el caso $d = 2$ proponemos un esquema WENO de diferencias finitas para 0.16. Experimentos numéricos muestran que para discretizaciones gruesas el esquema WENO de diferencias finitas captura de mejor forma las solución numérica, los tiempos de evacuación y la formación de lineas.

Los contenidos del Capítulo 3 corresponde a la investigación:

- R. Bürger, D. Inzunza and L. M. Villada, High-order finite-difference WENO schemes for models of crowd dynamics *(in preparation)*.

CHAPTER 1

Implicit-explicit schemes for nonlinear nonlocal equations with a gradient flow structure in one space dimension

1.1 Introduction

1.1.1 Scope

In the present chapter we are concerned with numerical methods for a nonlinear nonlocal equation with a gradient flow structure of the type

$$u_t + \nabla \cdot \big(u \nabla (W * u)\big) = \nabla \cdot \big(u \nabla (H'(u))\big), \quad \boldsymbol{x} \in \mathbb{R}^d, \quad t > 0, \tag{1.1}$$

$$u(\boldsymbol{x}, 0) = u_0(\boldsymbol{x}), \quad \boldsymbol{x} \in \mathbb{R}^d. \tag{1.2}$$

We limit the discussion to the case $d = 1$ (space dimension) for which (0.1), (0.10) reduce to

$$u_t = F[u]_x, \quad x \in \mathbb{R}, \quad t > 0; \quad F[u] = u(H'(u) - W * u)_x, \tag{1.3}$$

$$u(x, 0) = u_0(x), \quad x \in \mathbb{R}. \tag{1.4}$$

Here the notation $F[u] = F[u(\cdot, t)]$ means that the flux F depends on $u(\cdot, t)$ as a function of x as a whole, and we recall that

$$\big(W * u(\cdot, t)\big)(x) = \int_{\mathbb{R}} W(y) u(x - y, t) \, \mathrm{d}y = \int_{\mathbb{R}} W(x - y) u(y, t) \, \mathrm{d}y.$$

Although the available mathematical theory does not allow us to be conclusive about the existence, uniqueness and well-posedness of the solution of such convection-diffusion equations, it is plausible to perform simulations with appropriate numerical methods. Explicit schemes for hyperbolic first-order conservation laws are widely used in many applications nowadays. Although they can be rather slow for some steady-state computations due to CFL stability restrictions on the time step size, their use for unsteady computations is deemed practical in

many situations. When diffusion terms are present, one can resort to an implicit treatment of these terms to overcome the drastic step size stability restrictions imposed by their alternative explicit treatment. It is the purpose of the present work to demonstrate the benefits of using an implicit-explicit (IMEX) scheme for the efficient solution of (1.3), (1.4) under specific assumptions on the diffusive term. It is shown that the proposed scheme is more efficient, in terms of error reduction versus CPU time, than the explicit scheme of [21].

1.1.2 Related work

Equation (0.1), or some specific case of it, arises in many contexts including interacting gases [41], granular flows [56], flow in porous media [22, 50], and collective behavior in biology [55] (see these papers and [21] for further references). The one-dimensional model (1.3), (1.4) can also be understood as a model of the aggregation of populations by the following reasoning [7]. Assume that u is the density of the population (e.g., of animals) under study, and consider the equation

$$u_t + \left(-k \left[\int_{-\infty}^{x} u(y,t)\,\mathrm{d}y - \int_{x}^{\infty} u(y,t)\,\mathrm{d}y \right] u \right)_x = A(u)_{xx}, \quad x \in \mathbb{R}, \quad t > 0. \tag{1.5}$$

Here $k > 0$ is a constant, and the convective term models that an animal will move to the right (respectively, left) if the total population to its right is larger (respectively, smaller) than to its left. The aggregation mechanism is balanced by nonlinear diffusion described by the term $A(u)_{xx}$, known as density-dependent dispersal in mathematical ecology [7], where $A(u) = \int_0^u a(s)\,\mathrm{d}s$ and $a(u) \geq 0$ is a given diffusion function. Properties of (1.5) under various assumptions on the regularity of a were studied in [2, 7, 8, 43–46]. As in [7, 8], we allow that $a(u)$ may vanish on u-intervals of positive length, so (1.5) may be strongly degenerate. To see that (1.5) is an example of (1.3), we first rewrite (1.5) as

$$u_t + (u\tilde{W} * u)_x = A(u)_{xx} \tag{1.6}$$

with the odd kernel $\tilde{W}(x) = -k\,\mathrm{sgn}(x)$. Equation (1.6) becomes a one-dimensional example of (0.1) if we observe that

$$\tilde{W} * u = W' * u,$$

where W' denotes the derivative of W, if we choose the even kernel

$$W(x) = -k|x| + C,$$

where C is a constant and the function H is given by

$$H(u) = \int_0^u \int_0^r \frac{a(s)}{s}\,\mathrm{d}s\,\mathrm{d}r$$

(where possibly further restrictions on the function $u \mapsto a(u)$ need to be imposed so that H is well defined). The particular interest in this degenerate nonlocal aggregation equation arises from the numerical method for its solution constructed in [7], where the general equation

$$u_t + \left(\Phi' \left(\int_{-\infty}^{x} u(y,t)\,\mathrm{d}y \right) u \right)_x = A(u)_{xx} \tag{1.7}$$

is analyzed. If we assume that animals are conserved, i.e.,

$$C_0 := \int_{\mathbb{R}} u_0(y)\,\mathrm{d}y = \int_{\mathbb{R}} u(y,t)\,\mathrm{d}y \quad \text{for all } t > 0,$$

then the expression in squared brackets within (1.5) can be written as

$$\int_{-\infty}^{x} u(y,t)\,\mathrm{d}y - \int_{x}^{\infty} u(y,t)\,\mathrm{d}y = 2\int_{-\infty}^{x} u(y,t)\,\mathrm{d}y - C_0,$$

so if the function Φ is chosen such that

$$\Phi'(q) = -k(2q - C_0),$$

the equation (1.5) is obtained. The key observation here is that the primitive

$$q(x,t) := \int_{-\infty}^{x} u(y,t)\,\mathrm{d}y$$

is a solution of the following initial value problem for a *local* PDE:

$$q_t + \Phi(q)_x = A(q_x)_x, \quad x \in \mathbb{R}, \quad t > 0; \quad q(x,0) = q_0(x) := \int_{-\infty}^{x} u_0(\xi)\,\mathrm{d}\xi, \quad x \in \mathbb{R}. \tag{1.8}$$

Thus, by solving the local problem (1.8) numerically, and transforming back the numerical solution to u, we may conveniently generate a reference solution (for this particular aggregation model) that does not involve a discretization of the convolution but that can be used to assess the performance of the numerical scheme developed herein that solves (1.3) directly, and in particular does involve calculating the convolution at every time step. See Section 1.3.4 for further details.

Concerning numerical methods for (0.1), we mention that finite element approximations have been proposed in the literature, which are positivity preserving and entropy decreasing at the expense of constructing them by an implicit discretization in time but continuous in space [16]. We also mention that Carrillo, Chertock, and Huang [21] consider (1.3) adding the term $V(x)$ which represents a confinement potential, i.e.,

$$\rho_t(x) = \big(\rho(H'(\rho) + V(x) - W * \rho)_x\big)_x. \tag{1.9}$$

This a variant of (1.3) and has been extensively studied during the last fifteen years. In both cases of (1.3) and (1.9), the numerical methods studied for these equations are explicit schemes for convection-diffusion equations. In fact, Carrillo et al. [21] propose both one- and two-dimensional finite volume schemes for (0.1) and prove their positivity preserving and entropy dissipation properties along with error estimates and convergence results. These schemes follow a method of lines and are explicit by the choice of explicit SSP Runge-Kutta ODE integrators.

1.2 Numerical method

1.2.1 Spatial discretization

For the one-dimensional equation (1.3) we consider a spatial domain $\Omega = (-L, L)$ large enough so that the solution is compactly supported in it. We also consider a subdivision of Ω into M cells $C_j = [x_{j-1/2}, x_{j+1/2}]$ of a uniform size Δx with $x_j = -L + (j - 1/2)\Delta x$, $j \in \{1, \dots, M\}$, $x_{j\pm1/2} = x_j \pm \Delta x/2$, and denote by $u_j(t)$ an approximation to the solution cell average on C_j, i.e.,

$$u_j(t) \approx \frac{1}{\Delta x} \int_{x_{j-1/2}}^{x_{j+1/2}} u(x,t)\,\mathrm{d}x.$$

To obtain a semi-discrete finite volume scheme, (1.3) is first integrated over the j-th cell to give

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\int_{x_{j-1/2}}^{x_{j+1/2}} u(x,t)\,\mathrm{d}x\right) = F[u(\cdot,t)](x_{j+1/2}) - F[u(\cdot,t)](x_{j-1/2}).$$

We divide this equation by Δx and approximate the terms

$$\hat{F}_{j\pm1/2}(t) \approx F[u(\cdot,t)](x_{j\pm1/2})$$

by some numerical flux function $\hat{F}$ with arguments chosen among the variables u_l within some finite stencil around j,

$$\hat{F}_{j+1/2}(t) = \hat{F}\big(u_{j-p}(t), \dots, u_{j+q}(t)\big).$$

Then the final semidiscrete scheme takes the form of the following system of ODEs for u_j (notice that the signs are reversed with respect to the standard notation in conservation laws):

$$u_j'(t) = \frac{\hat{F}_{j+1/2}(t) - \hat{F}_{j-1/2}(t)}{\Delta x}. \tag{1.10}$$

For our separate treatment of convective and diffusive terms, we split $F[u]$ as follows:

$$F[u] = uH''(u)u_x - u(W * u)_x = K(u)_x - u(W * u)_x, \quad K'(u) = uH''(u), \tag{1.11}$$

where we assume the following properties of K:

$$\begin{aligned} &K \in \mathcal{C}^1([0,\infty)) \cap \mathcal{C}^2((0,\infty)),\\ &K(0) = K'(0) = 0,\\ &K''(u) \geq 0 \text{ for } u \in (0,\infty). \end{aligned} \tag{1.12}$$

In particular, $K(u), K'(u) \geq 0$ for $u \in [0,\infty)$.

Now we approximate both terms in (1.11) separately, omitting the dependence of quantities on $t \geq 0$ for sake of simplicity. The diffusive term is approximated by standard second-order three-point finite differences:

$$K(u)_{xx}(x_j) \approx \frac{K(u_{j+1}) - 2K(u_j) + K(u_{j-1})}{\Delta x^2} = \frac{1}{\Delta x}\big(\hat{F}^{\mathrm{d}}_{j+1/2} - \hat{F}^{\mathrm{d}}_{j-1/2}\big),$$
$$\hat{F}^{\mathrm{d}}_{j+1/2:} = \frac{K(u_{j+1}) - K(u_j)}{\Delta x}.$$

Regarding the convective term, the flux can be expressed as

$$uv[u], \quad v[u] = z[u]_x, \quad z[u] = W * u. \tag{1.13}$$

We use MUSCL reconstructions [57] and standard upwind techniques to obtain the convective numerical flux. One first constructs piecewise linear polynomials in each cell C_j, namely

$$\tilde{u}_j(x) = u_j + \sigma_j(x - x_j), \quad x \in C_j, \tag{1.14}$$

and computes right and left point values, u^{R}_j and u^{L}_j, at the respective cell interfaces $x_{j+1/2}$ and $x_{j-1/2}$ by

$$u^{\mathrm{R}}_j = \tilde{u}_j\big(x^-_{j+1/2}\big) = u_j + \frac{\Delta x}{2}\sigma_j, \quad u^{\mathrm{L}}_j = \tilde{u}_j\big(x^+_{j-1/2}\big) = u_j - \frac{\Delta x}{2}\sigma_j. \tag{1.15}$$

These values will be second-order accurate provided the slopes σ_j are at least first-order accurate approximations of the partial derivative $u_x(x,\cdot)$. To ensure that the point values (1.15) are both second-order accurate and non-negative, the slopes σ_j in (1.14) are calculated as follows. First, the centered difference approximation $\sigma_j = (u_{j+1} - u_{j-1})/(2\Delta x)$ is used for all j. Then, if a reconstructed point value at a cell boundary becomes negative (i.e., either $u^{\mathrm{R}}_j < 0$ or $u^{\mathrm{L}}_j < 0$), we correct the corresponding slope σ_j using a slope limiter, which guarantees that the reconstructed point values are non-negative as long as the cell averages u_j are non-negative. In the numerical experiments in [21] this is achieved by using a minmod limiter as follows:

$$\sigma_j = \begin{cases} \dfrac{u_{j+1} - u_{j-1}}{2\Delta x} & \text{if } u_j \geq \dfrac{|u_{j+1} - u_{j-1}|}{4}, \\ \vartheta \operatorname{minmod}\left(\dfrac{u_{j+1} - u_j}{\Delta x}, \dfrac{u_j - u_{j-1}}{\Delta x}\right) & \text{otherwise,} \end{cases}$$

where the standard minmod function is defined as

$$\operatorname{minmod}(z_1, z_2) := \begin{cases} \operatorname{sgn}(z_1)\min\{|z_1|, |z_2|\} & \text{if } \operatorname{sgn}(z_1) = \operatorname{sgn}(z_2), \\ 0 & \text{otherwise,} \end{cases}$$

and the parameter $\vartheta \in (0, 2]$ is used to control the numerical viscosity of the resulting scheme. The value $\vartheta = 2$ is used for the numerical examples in [21], and we adopt it in all our numerical examples.

To approximate $v[u](x_{j+1/2})$, we first use a second-order finite difference formula

$$z[u]_x(x_{j+1/2}) \approx \frac{z[u](x_{j+1}) - z[u](x_j)}{\Delta x}, \tag{1.16}$$

followed by discrete approximations of the convolutions $(W * u)(x_j)$, taking into account that u is compactly supported in $(-L, L)$, given by

$$(W * u)(x_j) = z[u](x_j) \approx \tilde{z}[u]_j = \tilde{z}[u^*_{j-s}, \dots, u^*_{j+s}] = \Delta x \sum_{l=-s}^{s} W_l u^*_{j-l}, \tag{1.17}$$

where we define

$$u^*_l := \begin{cases} u_l & \text{if } 1 \le l \le M, \\ 0 & \text{otherwise,} \end{cases}$$

where $W_l = W(l\Delta x)/\sum_{n=-s}^{s} W(n\Delta x)$, and the width of the convolution stencil $s = s_{\Delta x}$ is computed to retain second-order accuracy, by choosing s as the smallest natural such that

$$1 - \frac{\sum_{n=-s}^{s} W(n\Delta x)}{\sum_{n=-\infty}^{\infty} W(n\Delta x)} \le \mathcal{A}\Delta x^2,$$

where we have taken $\mathcal{A} = 10^{-8}$ in all our numerical examples and the term $\sum_{n=-\infty}^{\infty} W(n\Delta x)$ is approximated by $\sum_{n=-N}^{N} W(n\Delta x)$ for very large N.

Clearly, the computational bottleneck in this procedure is the discrete convolution in (1.17). This is a classical problem in scientific computing that is effectively evaluated using fast convolution algorithms, mainly based on Fast Fourier Transforms [59]. These techniques are applied here within the IMEX-RK version as well as within the explicit methods of [21].

To recap, the following approximation, obtained from (1.16) and (1.17),

$$v[u](x_{j+1/2}) = z[u]_x(x_{j+1/2}) \approx v_{j+1/2} = \tilde{v}[u]_{j+1/2} = \frac{\tilde{z}[u]_{j+1} - \tilde{z}[u]_j}{\Delta x}, \tag{1.18}$$

yields, in an upwind manner, the convective numerical flux associated with the cell interface $x_{j+1/2}$, namely

$$\hat{F}^{\mathrm{c}}_{j+1/2} = u_{j+1/2} v_{j+1/2}, \quad \text{where} \quad u_{j+1/2} = \begin{cases} u^{\mathrm{R}}_j & \text{if } v_{j+1/2} \ge 0, \\ u^{\mathrm{L}}_{j+1} & \text{if } v_{j+1/2} < 0, \end{cases} \tag{1.19}$$

which we can write in closed form as

$$\hat{F}^{\mathrm{c}}_{j+1/2} = u^{\mathrm{R}}_j v^-_{j+1/2} + u^{\mathrm{L}}_{j+1} v^+_{j+1/2}, \tag{1.20}$$

where $v^+_{j+1/2} := \max\{v_{j+1/2}, 0\}$ and $v^-_{j+1/2} := \min\{v_{j+1/2}, 0\}$. We set

$$\hat{F}_{j+1/2} = -\hat{F}^{\mathrm{c}}_{j+1/2} + \hat{F}^{\mathrm{d}}_{j+1/2}$$

compatibly with the splitting (1.11), so that (0.4) with

$$\boldsymbol{u} = (u_1, \dots, u_M)^{\mathrm{T}},\ \boldsymbol{\mathcal{C}}(\boldsymbol{u}) = (\mathcal{C}_1(\boldsymbol{u}), \dots, \mathcal{C}_M(\boldsymbol{u}))^{\mathrm{T}} \text{ and } \boldsymbol{\mathcal{D}}(\boldsymbol{u}) = (\mathcal{D}_1(\boldsymbol{u}), \dots, \mathcal{D}_M(\boldsymbol{u}))^{\mathrm{T}}$$

and (1.10) can be written as

$$u_j'(t) = \mathcal{C}_j(\boldsymbol{u}) + \mathcal{D}_j(\boldsymbol{u}),\ \mathcal{C}_j(\boldsymbol{u}) := -\frac{\hat{F}^{\mathrm{c}}_{j+1/2} - \hat{F}^{\mathrm{c}}_{j-1/2}}{\Delta x},\ \mathcal{D}_j(\boldsymbol{u}) := \frac{\hat{F}^{\mathrm{d}}_{j+1/2} - \hat{F}^{\mathrm{d}}_{j-1/2}}{\Delta x},\ \ j = 1, \dots, M. \tag{1.21}$$

It is worth pointing out that the original reference scheme in [21] is obtained through this procedure by taking $z[u] = H'(u) + W * u$ in (1.13) and $\hat{F}_{j+1/2} = \hat{F}^{\mathrm{c}}_{j+1/2}$.

1.2.2 A property of an explicit time discretization

The ODEs that form the semi-discrete scheme (1.10) need to be integrated numerically using a stable and accurate ODE solver. In all their numerical examples, Carrillo et al. [21] use the third-order strong stability preserving Runge-Kutta (SSP-RK) ODE solver [32]. The resulting scheme preserves positivity of the computed cell averages u_j, as stated in [21, Theorem 2.1]. Its proof is based on the forward Euler integration of (1.10), but it remains valid if the forward Euler method is replaced by a higher-order SSP-ODE solver [32], whose time step can be expressed as a convex combination of several forward Euler steps. For our scheme we can prove the following result, which is an analogue of [21, Theorem 2.1], following the lines stated therein.

Theorem 1.1. *If K' and K'' are non-negative on $(0, \infty)$, $u_j \geq 0$, for any j, and the CFL condition*

$$\Delta t \left(\frac{\max_j |v_{j+1/2}|}{\Delta x} + \frac{\max_j K'(u_j)}{\Delta x^2} \right) \leq \frac{1}{2} \tag{1.22}$$

is satisfied, then the quantity

$$\mathcal{E}(u)_j := u_j + \frac{\Delta t}{\Delta x} \big(\hat{F}_{j+1/2} - \hat{F}_{j-1/2} \big) \tag{1.23}$$

satisfies $\mathcal{E}(u)_j \geq 0$ for all j, i.e., the explicit Euler method applied to the semi-discrete scheme (1.21) yields a fully discrete positivity preserving scheme.

Proof. Since for $j = 1, \dots, M$ there holds $(u_j^{\mathrm{L}} + u_j^{\mathrm{R}})/2 = u_j$, $u_j^{\mathrm{L}}, u_j^{\mathrm{R}} \geq 0$ and there exists

$$\hat{u}_{j+1/2} = \vartheta_{j+1/2} u_j + (1 - \vartheta_{j+1/2}) u_{j+1}, \quad \vartheta_{j+1/2} \in (0, 1)$$

such that

$$K(u_{j+1}) - K(u_j) = K'(\hat{u}_{j+1/2})(u_{j+1} - u_j),$$

by (1.20) and (1.21) we may write $\mathcal{E}(u)_j$, which is defined in (1.23), as follows:

$$\begin{aligned}\mathcal{E}(u)_j &= \frac{u_j^{\mathrm{L}}+u_j^{\mathrm{R}}}{2}+\frac{\Delta t}{\Delta x}\big(-u_j^{\mathrm{R}}v_{j+1/2}^{+}-u_{j+1}^{\mathrm{L}}v_{j+1/2}^{-}+u_{j-1}^{\mathrm{R}}v_{j-1/2}^{+}+u_j^{\mathrm{L}}v_{j-1/2}^{-}\big)\\ &\quad+\frac{\Delta t}{\Delta x^2}\big(K(u_{j+1})-2K(u_j)+K(u_{j-1})\big)\\ &=\left(\frac{1}{2}+\frac{\Delta t}{\Delta x}v_{j-1/2}^{-}\right)u_j^{\mathrm{L}}+\left(\frac{1}{2}-\frac{\Delta t}{\Delta x}v_{j+1/2}^{+}\right)u_j^{\mathrm{R}}+\frac{\Delta t}{\Delta x}\big(-u_{j+1}^{\mathrm{L}}v_{j+1/2}^{-}+u_{j-1}^{\mathrm{R}}v_{j-1/2}^{+}\big)\\ &\quad+\frac{\Delta t}{\Delta x^2}\big(K'(\hat{u}_{j+1/2})(u_{j+1}-u_j)-K'(\hat{u}_{j-1/2})(u_j-u_{j-1})\big).\end{aligned}$$

Therefore, taking into account (1.22), $K'\geq 0$ and that

$$K'(\hat{u}_{l+1/2})\leq K'(\max\{u_l,u_{l+1}\})\leq \max_j K'(u_j)$$

for any l (due to $K''(u)\geq 0$ for any $u>0$), we obtain

$$\begin{aligned}\mathcal{E}(u)_j &\geq \left(\frac{1}{2}+\frac{\Delta t}{\Delta x}v_{j-1/2}^{-}-\frac{\Delta t}{2\Delta x^2}(K'(\hat{u}_{j+1/2})+K'(\hat{u}_{j-1/2}))\right)u_j^{\mathrm{L}}\\ &\quad+\left(\frac{1}{2}-\frac{\Delta t}{\Delta x}v_{j+1/2}^{+}-\frac{\Delta t}{2\Delta x^2}\big(K'(\hat{u}_{j+1/2})+K'(\hat{u}_{j-1/2})\big)\right)u_j^{\mathrm{R}}\\ &\quad+\frac{\Delta t}{\Delta x^2}\Big(K'(\hat{u}_{j+1/2})u_{j+1}+K'(\hat{u}_{j-1/2})u_{j-1}\Big)\\ &\geq\left(\frac{1}{2}-\frac{\Delta t}{\Delta x}\max_j|v_{j-1/2}|-\frac{\Delta t}{\Delta x^2}\max_j K'(u_j)\right)u_j^{\mathrm{L}}\\ &\quad+\left(\frac{1}{2}-\frac{\Delta t}{\Delta x}\max_j|v_{j-1/2}|-\frac{\Delta t}{\Delta x^2}\max_j K'(u_j)\right)u_j^{\mathrm{R}}\geq 0.\end{aligned}$$

This concludes the proof. □

The following result provides usable bounds for the velocities in (1.22).

Lemma 1.1. *We have the following bounds:*

$$|v_{j+1/2}|\leq\|W\|_\infty\,\mathrm{TV}(u),\quad |v_{j+1/2}|\leq\|u\|_\infty\,\mathrm{TV}(W),$$

provided that the right-hand sides are finite.

Proof. These results follow directly from the definition of $v_{j+1/2}$ in (1.17) and (1.18). For instance,

$$v_{j+1/2}=\sum_{l=-s}^{s}W_l\big(u_{j+1-l}^{*}-u_{j-l}^{*}\big)$$

immediately yields the first bound. □

1.2.3 Stability

A rigorous study of the von Neumann stability of explicit ODE solvers applied to (1.21) or the original scheme in [21] is not possible since the linearization of (1.3) does not have a structure amenable to this analysis. Nevertheless, the closest scenario permitting a stability analysis would be the application of a linear scheme to the standard convection diffusion equation

$$u_t + \gamma u_x = \eta u_{xx}, \tag{1.24}$$

where $\gamma \approx (W * u)_x$ and $\eta \approx K'(u)$. For some simple explicit Runge-Kutta schemes, it can be readily seen that such schemes applied to (1.24) are stable provided

$$\Delta t \left(\frac{\gamma}{\Delta x} + \frac{\eta}{\Delta x^2} \right) \leq C_1$$

for some constant C_1. This result is coherent with (1.22) and will be illustrated in the numerics section. This bound can be severely restrictive for fine simulations and is therefore a clear motivation for the consideration of implicit-explicit Runge-Kutta schemes, that typically relax the latter bound to

$$\gamma \frac{\Delta t}{\Delta x} \leq C_2,$$

a restriction that is fine for accuracy requirements.

1.2.4 Implicit-explicit Runge-Kutta schemes

For the diffusive part $\boldsymbol{\mathcal{D}}(\boldsymbol{u})$ we utilize an implicit s-stage diagonally implicit (DIRK) scheme with coefficients $\boldsymbol{A} \in \mathbb{R}^{s\times s}$, $\boldsymbol{b}, \boldsymbol{c} \in \mathbb{R}^s$, in the common Butcher notation, where $\boldsymbol{A} = (a_{ij})$ with $a_{ij} = 0$ for $j > i$. For the convective term $\boldsymbol{\mathcal{C}}(\boldsymbol{u})$ we employ an s-stage explicit scheme with coefficients $\tilde{\boldsymbol{A}} \in \mathbb{R}^{s\times s}$, $\tilde{\boldsymbol{b}}, \tilde{\boldsymbol{c}} \in \mathbb{R}^s$ and $\tilde{\boldsymbol{A}} = (\tilde{a}_{ij})$ with $\tilde{a}_{ij} = 0$ for $j \geq i$. The corresponding Butcher arrays are

$$\begin{array}{c|c} \boldsymbol{c} & \boldsymbol{A} \\ \hline & \boldsymbol{b}^{\mathrm{T}} \end{array} \quad \text{and} \quad \begin{array}{c|c} \tilde{\boldsymbol{c}} & \tilde{\boldsymbol{A}} \\ \hline & \tilde{\boldsymbol{b}}^{\mathrm{T}} \end{array}. \tag{1.25}$$

If applied to (1.21), the IMEX-RK scheme gives rise to the following algorithm, where we recall that $\boldsymbol{u} = (u_1, \ldots, u_M)^{\mathrm{T}}$, etc.:

Algorithm 3.1: IMEX-RK scheme

Input: approximate solution vector $\boldsymbol{u}^n$ for $t = t_n$

do $i = 1, \ldots, s$

solve for $\boldsymbol{u}^{(i)}$ the nonlinear equation

$$\boldsymbol{u}^{(i)} = \boldsymbol{u}^n + \Delta t\left(\sum_{j=1}^{i-1} a_{ij}\boldsymbol{\kappa}_j + \sum_{j=1}^{i-1} \tilde{a}_{ij}\tilde{\boldsymbol{\kappa}}_j\right) + a_{ii}\Delta t\boldsymbol{\mathcal{D}}\big(\boldsymbol{u}^{(i)}\big)$$
$$\boldsymbol{\kappa}_i \leftarrow \boldsymbol{\mathcal{D}}\big(\boldsymbol{u}^{(i)}\big),\ \tilde{\boldsymbol{\kappa}}_i \leftarrow \boldsymbol{\mathcal{C}}\big(\boldsymbol{u}^{(i)}\big)$$

enddo

$$\boldsymbol{u}^{n+1} \leftarrow \boldsymbol{u}^n + \Delta t\sum_{j=1}^{s} b_j\,\boldsymbol{\kappa}_j + \Delta t\sum_{i=1}^{s} \tilde{b}_i\tilde{\boldsymbol{\kappa}}_i$$

Output: approximate solution vector $\boldsymbol{u}^{n+1}$ for $t = t^{n+1} = t^n + \Delta t$.

This algorithm requires solving for each $i \in \{1, \ldots, s\}$ the nonlinear system

$$\boldsymbol{F}(\boldsymbol{u}) := \boldsymbol{u} - a_{ii}\Delta t D(\boldsymbol{u}) - \boldsymbol{r} = \boldsymbol{0}, \quad i = 1, \ldots, s, \tag{1.26}$$

for the unknown vector $\boldsymbol{u} = \boldsymbol{u}^{(i)}$, where the vector $\boldsymbol{r}$ is given by

$$\boldsymbol{r} = \boldsymbol{u}^n + \Delta t\left(\sum_{j=1}^{i-1} a_{ij}\boldsymbol{\kappa}_j + \sum_{j=1}^{i-1} \tilde{a}_{ij}\tilde{\boldsymbol{\kappa}}_j\right). \tag{1.27}$$

The following results deal with the solution of (1.26).

Theorem 1.2. *Assume that K satisfies the conditions (1.12), $\mu > 0$, $\boldsymbol{c} \in \mathbb{R}^M$, and $\boldsymbol{c} \geq \boldsymbol{0}$, where such inequalities for vectors and matrices are understood in the component-wise sense. Then the equation*

$$\boldsymbol{z} - \mu\boldsymbol{\mathcal{D}}(\boldsymbol{z}) = \boldsymbol{c} \tag{1.28}$$

has a unique solution $\boldsymbol{z} \in \mathbb{R}^M$ satisfying $\boldsymbol{z} \geq \boldsymbol{0}$.

Proof. Equation (1.28) can be rewritten as

$$\boldsymbol{z} + \boldsymbol{G}\boldsymbol{K}(\boldsymbol{z}) = \boldsymbol{c}, \tag{1.29}$$

where the $M \times M$-matrix $\boldsymbol{G}$ and the M-vector $\boldsymbol{K}(\boldsymbol{z})$ are given by

$$\boldsymbol{G} = \begin{bmatrix} 2 & -1 & 0 & \cdots & 0 \\ -1 & 2 & -1 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & -1 & 2 & -1 \\ 0 & \cdots & 0 & -1 & 2 \end{bmatrix}, \quad \boldsymbol{K}(\boldsymbol{z}) = \beta \begin{pmatrix} K(z_1) \\ K(z_2) \\ \vdots \\ K(z_M) \end{pmatrix}, \quad \beta = \frac{\mu}{\Delta x^2}.$$

Let us define

$$L(u) := \begin{cases} K(u)/u & \text{if } u > 0, \\ 0 & \text{if } u = 0. \end{cases}$$

With the requirements in (1.12), L is continuous in $[0,\infty)$ and $L(u), K(u) \geq 0$ for $u \geq 0$. Furthermore, we define

$$\boldsymbol{E}(\boldsymbol{z}) := \beta \operatorname{diag}\big(L(z_1),\dots,L(z_M)\big) = \operatorname{diag}(\vartheta_1,\dots,\vartheta_M),$$

where

$$\vartheta_i := \beta L(z_i) \geq 0 \quad \text{for} \quad i = 1,\dots,M,$$

then $\boldsymbol{K}(\boldsymbol{z}) = \boldsymbol{E}(\boldsymbol{z})\boldsymbol{z}$. Assume that $\boldsymbol{z} \geq \boldsymbol{0}$. Then $\boldsymbol{I} + \boldsymbol{G}\boldsymbol{E}(\boldsymbol{z})$ is a strictly diagonally dominant matrix (by columns) with positive diagonal entries and non-positive off-diagonal entries:

$$\boldsymbol{I} + \boldsymbol{G}\boldsymbol{E}(\boldsymbol{z}) = \begin{bmatrix} 1+2\vartheta_1 & -\vartheta_2 & 0 & \cdots & 0 \\ -\vartheta_1 & 1+2\vartheta_2 & -\vartheta_3 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & -\vartheta_{M-2} & 1+2\vartheta_{M-1} & -\vartheta_M \\ 0 & \cdots & 0 & -\vartheta_{M-1} & 1+2\vartheta_M \end{bmatrix},$$

and therefore $(\boldsymbol{I} + \boldsymbol{G}\boldsymbol{E}(\boldsymbol{z}))^{-1}$ is a non-negative matrix and it is a continuous function of $\boldsymbol{z}$. Then, the solution of equation (1.29) is reduced to finding fixed points of the mapping

$$\boldsymbol{z} \mapsto \boldsymbol{\varphi}(\boldsymbol{z}) = \big(\boldsymbol{I} + \boldsymbol{G}\boldsymbol{E}(\boldsymbol{z})\big)^{-1}\boldsymbol{c}.$$

To assess existence of fixed points, we aim to apply Brouwer's theorem to $\boldsymbol{\varphi}$ and the compact and convex set

$$\mathcal{K} := \{\boldsymbol{z} \in \mathbb{R}^M \mid \boldsymbol{z} \geq \boldsymbol{0} \text{ and } \|\boldsymbol{z}\|_1 \leq \|\boldsymbol{c}\|_1\}.$$

Clearly, $(\boldsymbol{I} + \boldsymbol{G}\boldsymbol{E}(\boldsymbol{z}))^{-1} \geq \boldsymbol{0}$ and $\boldsymbol{c} \geq \boldsymbol{0}$ immediately yield $\boldsymbol{\varphi}(\boldsymbol{z}) \geq \boldsymbol{0}$ for all $\boldsymbol{z} \in \mathcal{K}$, so, to prove that $\boldsymbol{\varphi}(\mathcal{K}) \subseteq \mathcal{K}$, there only remains to prove that

$$\big\|\boldsymbol{\varphi}(\boldsymbol{z})\big\|_1 \leq \|\boldsymbol{c}\|_1 \quad \text{for all } \boldsymbol{z} \in \mathcal{K}. \tag{1.30}$$

To this end, we take into account that

$$\big\|\boldsymbol{\varphi}(\boldsymbol{z})\big\|_1 \leq \big\|\big(\boldsymbol{I} + \boldsymbol{G}\boldsymbol{E}(\boldsymbol{z})\big)^{-1}\big\|_1 \|\boldsymbol{c}\|_1.$$

Thus, to establish (1.30) it is sufficient to prove that

$$\big\|\big(\boldsymbol{I} + \boldsymbol{G}\boldsymbol{E}(\boldsymbol{z})\big)^{-1}\big\|_1 \leq 1 \quad \text{for all } \boldsymbol{z} \in \mathcal{K}.$$

For this purpose, we use the auxiliary matrix

$$\tilde{\boldsymbol{G}} := \begin{bmatrix} 1 & -1 & 0 & \cdots & 0 \\ -1 & 2 & -1 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & -1 & 2 & -1 \\ 0 & \cdots & 0 & -1 & 1 \end{bmatrix}$$

and the notation $\boldsymbol{H} := \boldsymbol{I} + \boldsymbol{G}\boldsymbol{E}(\boldsymbol{z})$ and $\boldsymbol{M} := \boldsymbol{I} + \tilde{\boldsymbol{G}}\boldsymbol{E}(\boldsymbol{z})$. The same previous argument yields that $\boldsymbol{M}^{-1} \geq \boldsymbol{0}$. Now, for $\boldsymbol{e} := (1, \dots, 1)^{\mathrm{T}} \in \mathbb{R}^M$ it follows that $\boldsymbol{e}^{\mathrm{T}}\tilde{\boldsymbol{G}} = \boldsymbol{0}$, so $\boldsymbol{e}^{\mathrm{T}}\boldsymbol{M} = \boldsymbol{e}^{\mathrm{T}}$ and $\boldsymbol{e}^{\mathrm{T}}\boldsymbol{M}^{-1} = \boldsymbol{e}^{\mathrm{T}}$. If we assume that $\boldsymbol{H}^{-1} = (\bar{\eta}_{ij})_{1\leq i,j\leq M}$ and $\boldsymbol{M}^{-1} = (\bar{\mu}_{ij})_{1\leq i,j\leq M}$, then this is equivalent to

$$\sum_{i=1}^{M} \bar{\mu}_{ij} = 1, \quad j = 1, \dots, M.$$

Furthermore, since $\boldsymbol{H}^{-1} \geq \boldsymbol{0}$, $\boldsymbol{M}^{-1} \geq \boldsymbol{0}$ and $\boldsymbol{H} - \boldsymbol{M} = \beta \operatorname{diag}(L(z_1), 0, \dots, 0, L(z_M)) \geq \boldsymbol{0}$ for $\boldsymbol{z} \in \mathcal{K}$ and

$$\boldsymbol{H}^{-1} = \boldsymbol{M}^{-1} - \boldsymbol{H}^{-1}(\boldsymbol{H} - \boldsymbol{M})\boldsymbol{M}^{-1},$$

it follows that $\boldsymbol{H}^{-1} \leq \boldsymbol{M}^{-1}$. This yields that

$$\|\boldsymbol{H}^{-1}\|_1 = \max_{1\leq j\leq M} \sum_{i=1}^{M} \bar{\eta}_{ij} \leq \max_{1\leq j\leq M} \sum_{i=1}^{M} \bar{\mu}_{ij} = 1.$$

Applying Brouwer's fixed point theorem to the continuous function $\boldsymbol{\varphi}\colon \mathcal{K} \to \mathcal{K}$ we deduce the existence of a fixed point of $\boldsymbol{\varphi}$, i.e. a non-negative solution to equation (1.28).

For uniqueness, we adapt an argument that can be found in [49] and define

$$\Psi(\boldsymbol{z}) := \beta \sum_{i=1}^{M} N\big(z_i\big), \quad N(u) = \int_0^{|u|} K(s)ds,$$

and

$$f(\boldsymbol{z}) := \frac{1}{2}\boldsymbol{z}^{\mathrm{T}}\boldsymbol{G}^{-1}\boldsymbol{z} + \Psi(\boldsymbol{z}) - \boldsymbol{z}^{\mathrm{T}}\boldsymbol{G}^{-1}\boldsymbol{c}.$$

Since $K(0) = 0$, it follows from the definition that $N'(u) = \operatorname{sgn}(u)K(|u|)$ and $N''(u) = K'(|u|)$ for any $u \in \mathbb{R}$, so $N \in \mathcal{C}^2(\mathbb{R})$. Therefore, Ψ is twice continuously differentiable. Therefore f is also twice continuously differentiable and its gradient $f'(\boldsymbol{z})$ and Hessian $f''(\boldsymbol{z})$ are given by the respective expressions

$$f'(\boldsymbol{z})^{\mathrm{T}} = \boldsymbol{G}^{-1}\boldsymbol{z} + \beta \begin{pmatrix} \operatorname{sgn}(z_1)K(|z_1|) \\ \vdots \\ \operatorname{sgn}(z_M)K(|z_M|) \end{pmatrix} - \boldsymbol{G}^{-1}\boldsymbol{c}, \quad f''(\boldsymbol{z}) = \boldsymbol{G}^{-1} + \beta \operatorname{diag}\big(K'(|z_1|), \dots, K'(|z_M|)\big).$$

Since $\boldsymbol{G}^{-1}$ is symmetric and positive definite and $\beta K'(|z_i|) \geq 0$, it follows that $f''(\boldsymbol{z})$ is symmetric and positive definite, therefore f is strictly convex, so any critical point (at which $f'(\boldsymbol{z}) = \boldsymbol{0}$) is the unique global minimum. Now, if $\boldsymbol{z} + \boldsymbol{G}\boldsymbol{K}(\boldsymbol{z}) = \boldsymbol{c}$ with $\boldsymbol{z} \geq \boldsymbol{0}$, then $f'(\boldsymbol{z}) = \boldsymbol{0}$ and $\boldsymbol{z} \in \mathcal{K}$, so positive solutions of (1.29) are critical points of f, so uniqueness is proven. □

Theorem 1.3. *If K satisfies the conditions (1.12) and*

$$\frac{\Delta t}{\Delta x} \max_j |v_{j+1/2}| \leq 1/2 \tag{1.31}$$

then the Euler IMEX method

$$\boldsymbol{u}^{n+1} = \boldsymbol{u}^n + \Delta t\big(\boldsymbol{\mathcal{C}}(\boldsymbol{u}^n) + \boldsymbol{\mathcal{D}}(\boldsymbol{u}^{n+1})\big)$$

is a positivity preserving scheme.

Proof. Let $\boldsymbol{b}^n := \boldsymbol{u}^n + \Delta t\boldsymbol{\mathcal{C}}(\boldsymbol{u}^n)$. Then it follows from Theorem 1.1 with $K = 0$ that $\boldsymbol{b}^n \geq \boldsymbol{0}$. Since the equation $\boldsymbol{u}^{n+1} = \boldsymbol{b}^n + \Delta t\boldsymbol{\mathcal{D}}(\boldsymbol{u}^{n+1})$ can be rewritten as $\boldsymbol{z} - \Delta t\boldsymbol{\mathcal{D}}(\boldsymbol{z}) = \boldsymbol{c}$ for $\boldsymbol{z} = \boldsymbol{u}^{n+1}$, $\boldsymbol{c} = \boldsymbol{b}^n$, Theorem 1.2 yields that a unique non-negative solution $\boldsymbol{z}$ exists. This concludes the proof. □

Unfortunately, this result cannot be directly applied to higher-order RK-IMEX schemes, since there cannot be implicit Runge-Kutta schemes in SSP form of order higher than one (see [32]), so Theorem 1.2 cannot be in principle applied for second-order accuracy in time. We have nevertheless used Newton-Raphson method, together with a line search algorithm (see [20]) to solve (1.26). At each step of this algorithm a tridiagonal system is solved. We have not experienced any troubles in solving these systems under a stability restriction as (1.31).

1.3 Numerical results

1.3.1 Preliminaries.

In the following examples, we solve (1.3) numerically for $0 \leq t \leq T$ and $-L \leq x \leq L$. We compare numerical results obtained by the IMEX-RK scheme proposed herein with those obtained by the explicit scheme of [21]. For each time step, $\Delta t = \Delta t_n$ is determined by the formula

$$\frac{\Delta t}{\Delta x}\max_j\big|v^n_{j+1/2}\big| + \frac{\Delta t}{\Delta x^2}\max_u\big|K'(u)\big| = C_{\text{cfl}_1} \tag{1.32}$$

for the explicit scheme and by

$$\frac{\Delta t}{\Delta x}\max_j\big|v^n_{j+1/2}\big| = C_{\text{cfl}_2} \tag{1.33}$$

for the IMEX-RK scheme. In the numerical examples we choose C_{cfl_1} and C_{cfl_2} in the respective cases as the largest multiple of 0.05 that yields oscillation-free reference solutions.

For comparison purposes, we compute reference solutions for numerical tests with $M_{\text{ref}} = 12800$ cells with the IMEX-RK scheme for Examples 1, 2, and 3, and with the first-order scheme of [7] for Example 4 (see Section 1.3.4 for a description of that scheme). We compute approximate L^1 errors at different times for each scheme as follows. We denote by $(u^M_j(t))_{j=1}^M$ and $(u^{\text{ref}}_j(t))_{j=1}^{M_{\text{ref}}}$ the numerical solution at time t calculated with M and M_{ref} cells, respectively.

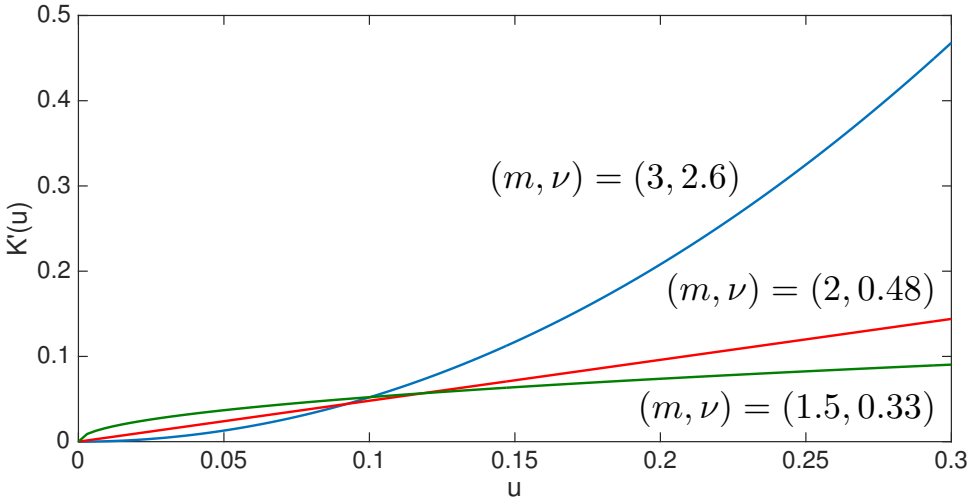

Figure 1.1: Example 1.1: nonlinear diffusion functions $K'(u) = uH''(u)$ for $H(u) = (\nu/m)u^m$ for the indicated pairs (m, ν) (figure produced by author).

We assume that $R := M_{\text{ref}}/M$ is an integer and compute the projection of the reference solution $\tilde{u}_j^{\text{ref},M}(t)$, $j = 1, \dots, M$, by

$$\tilde{u}_j^{\text{ref},M}(t) = \frac{1}{R} \sum_{k=1}^{\text{R}} u_{R(j-1)+k}^{\text{ref}}(t).$$

The approximate L^1 error $e_M(t)$ associated with the numerical solution on the mesh with M cells at time t is then given by

$$e_M(t) := \frac{2}{M} \sum_{j=1}^{M} \left| \tilde{u}_j^{\text{ref},M}(t) - u_j^M(t) \right|.$$

A numerical order of convergence can be calculated from pairs $e_{M/2}(t)$ and $e_M(t)$ by

$$\vartheta_M(t) := \log_2\big(e_M(t)/e_{2M}(t)\big).$$

In our simulations we limit ourselves to the second-order IMEX-RK scheme defined by the pair of Butcher arrays given by (0.5).

1.3.2 Examples 1.1 and 1.2

In Examples 1 and 2 we consider the numerical experiment proposed in [21]. Example 1 corresponds to equation (1.3) where the initial condition u_0 along with the functions W and H

are given by the respective expressions

$$u_0(x) = \frac{1}{\sqrt{8\pi}}\big(\exp(-0.5(x-3)^2) + \exp(-0.5(x+3)^2)\big), \quad W(x) = \frac{1}{\sqrt{2\pi\sigma}}\exp\left(-\frac{|x|^2}{2\sigma}\right),$$
$$H(u) = \frac{\nu}{m}u^m, \quad m \geq 1. \tag{1.34}$$

It is worth pointing out that the function $H(u)$ given by (1.34) for $m \geq 1$, which is used in Examples 1 to 3, satisfies the requirements in (1.12) (and, therefore, the assumptions of Theorems 1.1 and 1.2), since

$$K'(u) = uH''(u) = \nu(m-1)u^{m-1}, \; K(u) = \int_0^u K'(s)ds = \nu\frac{m-1}{m}u^m, K''(u) = \nu(m-1)^2u^{m-2},$$

yields that $K \in \mathcal{C}^1([0,\infty)) \cap \mathcal{C}^2((0,\infty))$, $K(0) = K'(0) = 0$, $K''(u) \geq 0$ for $u \in (0,\infty)$.

In Example 1 we employ the pairs $(m,\nu) = (1.5, 0.33)$, $(2, 0.48)$, and $(3, 2.6)$ that cover the cases $0 < m < 2$, $m = 2$, and $m > 2$, respectively. The corresponding functions $K'(u)$, that is the nonlinear diffusion coefficients, for these three cases are plotted in Figure 1.1. The simulations are run on the computational domain $[-L, L] = [-10, 10]$. Numerical solutions for the final times $T = 250$ and $T = 1250$ for each of the three parameter choices, and obtained with a relatively coarse discretization of $M = 200$ subintervals, are shown in Figure 1.2. Table 1.1 displays the approximate L^1 errors, corresponding convergence rates, and CPU times for each of these pairs of parameters for a number of final times and successive discretizations. The information of Table 1.1 is plotted in Figure 1.3 in terms of efficiency, that is, in terms of reduction of numerical error versus CPU time.

First of all, we observe that for $(m,\nu) = (1.5, 0.33)$, the reference solution is smooth while for $(m,\nu) = (2, 0.48)$ and $(3, 2.6)$, the solution has "kinks" at the basis of the "peak" that is forming. Moreover, we observe that the numerical errors produced by both the IMEX-RK and the explicit schemes roughly reduce at an observed second-order rate of convergence, which is also the theoretical order of accuracy (both in space and time). For a given discretization M, the explicit scheme produces significantly smaller approximate errors than the corresponding IMEX-RK scheme for the cases $(m,\nu) = (1.5, 0.33)$ and $(m,\nu) = (2, 0.48)$, while for $(m,\nu) = (3, 2.6)$ and $M \geq 400$ the error produced by the IMEX-RK scheme are smaller than those of the explicit scheme. While the results do not favor one or the other scheme in terms of accuracy, CPU times for the IMEX-RK scheme are substantially smaller than for the explicit scheme. This observation gives rise to the question of efficiency, that is which of the schemes turns out to be more efficient in terms of reduction of error per CPU time. Figure 1.3 indicates that for $(m,\nu) = (1.5, 0.33)$ and $(m,\nu) = (2, 0.48)$ the explicit scheme is most efficient for $M \leq 800$ but outperformed by the IMEX-RK scheme for even finer discretizations, while for $(m,\nu) = (3, 2.6)$, the IMEX-RK scheme is most efficient in almost all instances. In fact, in light of Figure 1.1 and the range of solution values attained, the third case is the one that involves the highest values of $K'(u)$, that is where diffusion is most dominant and therefore the gain in efficiency by using an IMEX-RK scheme instead of a comparable explicit scheme is most significant. Related

findings have been obtained for systems of convection-diffusion equations modeling equilibrium chromatography [19].

In Example 2 we consider the same function H as in Example 1 but now choose $m = 3$ and $\nu = 1.48$. The functions W and u_0 are given by

$$W(x) = \begin{cases} 1 - |x| & \text{if } |x| \leq 1, \\ 0 & \text{otherwise,} \end{cases} \qquad u_0(x) = \begin{cases} 0.05 & \text{if } x \in [-3, 3], \\ 0 & \text{otherwise.} \end{cases}$$

The numerical simulation are run over the computational domain $[-6, 6]$ until final time $T = 105$. Numerical results at $T = 0$ (the initial condition), $T = 45$, $T = 75$ and $T = 105$ are shown in Figure 1.4. Table 1.2 provides the corresponding approximate L^1 errors, convergence rates, and CPU times, and Figure 1.5 shows the efficiency plots for $T = 45$ and $T = 75$. First of all, it is interesting to note that the model predicts that one initial block of "mass" is split into three separate portions of which the middle one disappears later and two portions remain. For this model, the IMEX-RK scheme produces slightly larger errors but is considerably faster than the explicit scheme, and therefore turns out significantly more efficient (see Figure 1.5).

1.3.3 Example 1.3

This example represents a slight modification of Example 1, namely we choose W and H as in Example 1, but we utilize the initial function

$$u_0(x) = \frac{1}{\sqrt{8\pi}}\big(\exp(-0.2(x+7)^2) + \exp(-0.2x^2) + \exp(-0.2(x-7)^2)\big).$$

Moreover, we consider the pairs of parameter values $(m, \nu) = (3, 2.6)$ and $(m, \nu) = (3, 3)$ and with a CFL condition (1.32) with $C_{\text{cfl}_1} = 0.5$ for the explicit scheme. The numerical results are displayed in Figures 1.6 and 1.7, the approximate errors, convergence rates and CPU times are provided in Table 1.3, and Figure 1.8 contains the efficiency plots for two of the six end times T for which the errors are measured. According to Table 1.3, for $(m, \nu) = (3, 2.6)$ the IMEX-RK produces smaller errors (for $M \geq 200$) and occupies less CPU time than the explicit scheme, while for $(m, \nu) = (3, 3)$ the same happens for sufficiently fine discretizations. Summarizing, we can say that according to Table 1.3 for $M \geq 400$ (but in many runs even for coarser discretizations) the IMEX-RK scheme is more efficient than the explicit version.

1.3.4 Example 1.4

In this example we come back to the one-dimensional aggregation model outlined in Section 1.1.2, and present a numerical example of (1.3), (1.4) that is also a solution to the aggregation equation (1.5). We consider the numerical experiment proposed in [7], and first specify

the initial condition

$$u_0(x) = \begin{cases} 5 & \text{for } 0.1 \le x \le 0.2, \\ 8 & \text{for } 0.6 \le x \le 0.7, \\ 7 & \text{for } 0.8 \le x \le 0.9, \\ 0 & \text{otherwise,} \end{cases}$$

such that $C_0 = 2$. Then we set $\Phi(q) = -(1-q)^2$, which ensures that (1.7) recovers (1.5) (with $k = 1$), and correspondingly, $W(x) = |x|$. Moreover, the numerical experiment of [7] stipulates the strongly degenerating diffusion function

$$a(u) = \begin{cases} 0 & \text{for } u \le u_{\mathrm{c}}, \\ a_0 & \text{for } u > u_{\mathrm{c}}, \end{cases}$$

where $a_0 = 0.1$ and $u_{\mathrm{c}} = 10$ is a critical density value.

The numerical scheme for the initial value problem (1.8), analyzed in [7] and which is utilized in this example to calculate the reference solution, is defined as follows. We first set

$$q_j^0 := q_0(x_j) = \int_0^{x_j} u_0(x)\,\mathrm{d}x, \tag{1.35}$$

and then utilize the explicit marching formula

$$\begin{aligned} q_j^{n+1} = q_j^n - \frac{\Delta t}{\Delta x}\Bigg[& h(q_j^n, q_{j-1}^n) - h(q_{j-1}^n, q_j^n) \\ & - \left(A\left(\frac{q_{j+1}^n - q_j^n}{\Delta x}\right) - A\left(\frac{q_j^n - q_{j-1}^n}{\Delta x}\right)\right)\Bigg], \quad j = 1, \dots, M, \quad n \in \mathbb{N}_0, \end{aligned} \tag{1.36}$$

where Δt and Δx are subject to the CFL condition

$$\frac{\Delta t}{\Delta x} \max_{q\in[0,C_0]} \big|\Phi'(q)\big| + \frac{\Delta t}{\Delta x^2} \max_{u\in\mathbb{R}} \big|a(u)\big| \le \frac{1}{2}, \tag{1.37}$$

and h is the Engquist-Osher flux [31], a monotone numerical flux [26] consistent with Φ that is given by

$$h(q,r) = \Phi(0) + \int_0^q \max\big\{0, \Phi'(s)\big\}\,\mathrm{d}s + \int_0^r \min\big\{0, \Phi'(s)\big\}\,\mathrm{d}s.$$

To recover u from the numerical solution of (1.8) we use the divided difference

$$u_j^n = \frac{(q_{j+1}^n - q_j^n)}{\Delta x},$$

which gives a numerical method for equation (1.7) that converges as $\Delta t \to 0$ to the unique entropy solution of (1.7), (1.4), as is proven in [7]. Here we employ the aforementioned scheme

with $M_{\text{ref}} = 12800$ to calculate a reference solution to compare the performances of the IMEX-RK and explicit schemes.

Numerical results are shown in Figure 1.9. The approximate errors and CPU times are displayed in Table 1.4, and Figure 1.10 contains the corresponding efficiency plots. The results of Figure 1.9 alert to the fact that solutions of this model are in general discontinuous due to the strong degeneracy of the diffusion and the imposition of discontinuous initial data. In this case the ingredients of the equation have been designed such that all "mass" ("animals") move to the center of mass and eventually form one single group ("herd"). We observe slight "kinks" in the solution profiles near $u = u_{\text{c}} = 10$. Above this value of density the repulsive effect of degenerate diffusion sets on, and it is precisely this effect which prevents the model from forming unbounded densities (at least, in finite time [7]). In Table 1.4, we do not show the order of convergence because it is well known that in the presence of discontinuities this order is much lower than the formal order (in this case, second order) of accuracy of the scheme. Nevertheless, it appears that both the solution of the IMEX-RK and the explicit scheme converge to the reference solution produced by (1.35), (1.36), and the IMEX-RK occupies only a fraction of the CPU time in comparison with the explicit scheme (for instance, less than 1% for $M = 1600$), and therefore turns out more efficient than the explicit scheme, as is reconfirmed by Figure 1.10.

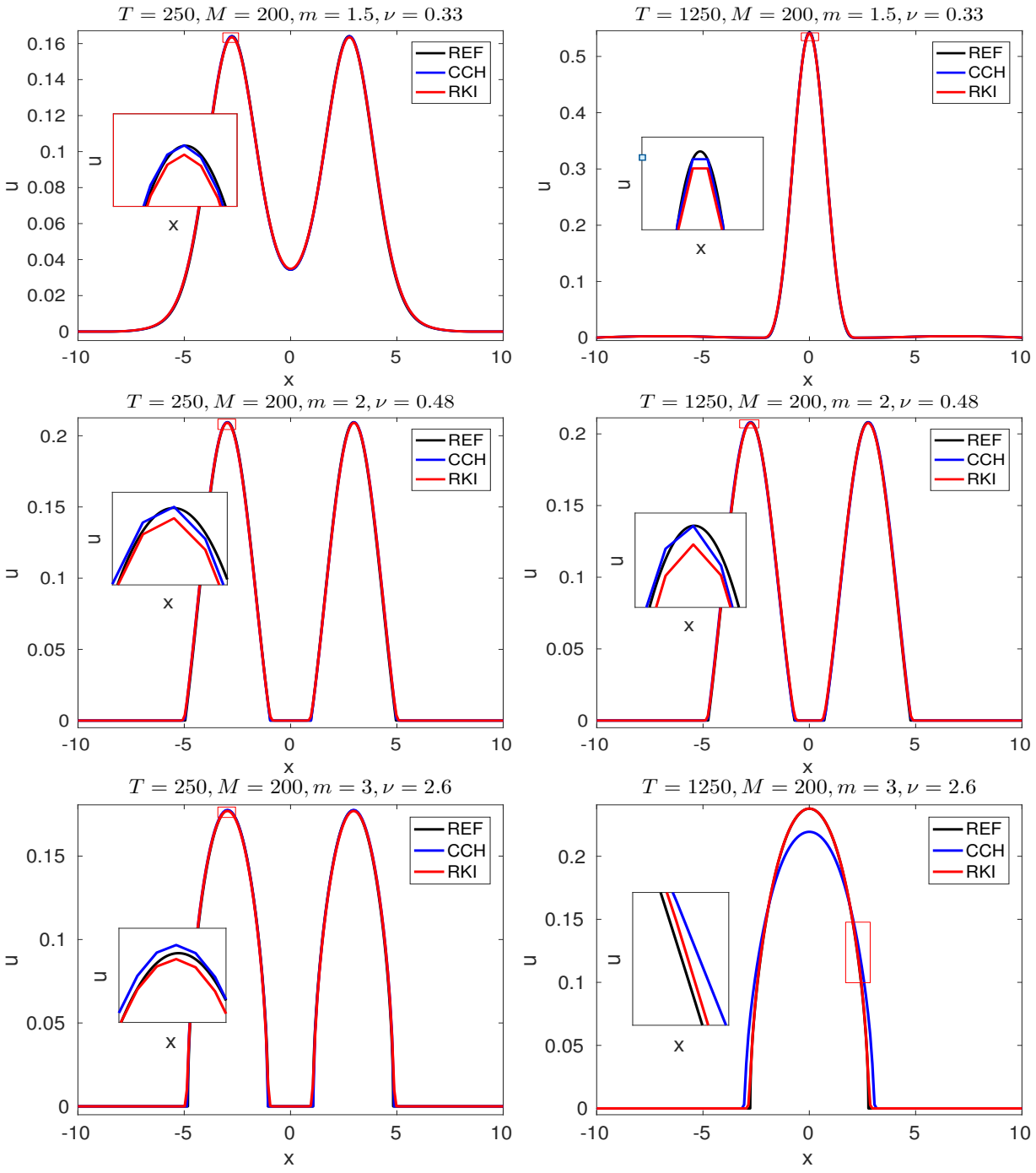

Figure 1.2: Example 1.1: numerical solutions with $\Delta x = 2L/M$, $L = 10$ and $M = 200$ for (top) $m = 1.5$, $\nu = 0.33$, (middle) $m = 2$, $\nu = 0.48$, (bottom) $m = 3$, $\nu = 2.6$ at simulated time (left) $T = 250$, (right) $T = 1250$.

		IMEX-RK			Explicit			IMEX-RK			Explicit		
	M	e_M	ϑ_M	cpu [s]	e_M	ϑ_M	cpu [s]	e_M	ϑ_M	cpu [s]	e_M	ϑ_M	cpu [s]
$m = 1.5$,		$T = 250$						$T = 1000$					
$\nu = 0.33$	100	813.43	—	0.02	90.11	—	0.02	672.99	—	0.14	166.74	—	0.10
	200	200.03	2.02	0.12	22.94	1.97	0.11	174.22	1.95	0.69	42.00	1.99	0.60
	400	49.15	2.03	0.41	5.82	1.98	1.34	46.98	1.89	3.05	10.84	1.95	7.18
	800	12.12	2.02	1.93	1.49	1.96	11.96	13.13	1.84	14.25	2.71	2.00	65.57
	1600	2.98	2.03	8.59	0.41	1.88	103.98	3.88	1.76	64.94	0.73	1.89	562.33
		$T = 1250$						$T = 1500$					
	100	664.34	—	0.20	166.36	—	0.12	654.96	—	0.24	165.64	—	0.15
	200	174.50	1.93	0.93	41.93	1.99	0.80	174.39	1.91	1.16	41.73	1.99	1.01
	400	47.18	1.89	4.25	10.87	1.95	9.53	46.60	1.90	5.46	10.72	1.96	11.88
	800	13.48	1.81	19.85	2.71	2.00	87.38	13.32	1.81	25.50	2.70	1.99	109.32
	1600	4.10	1.72	90.79	0.75	1.86	746.51	4.05	1.72	116.89	0.75	1.85	930.66
$m = 2$,		$T = 250$						$T = 1000$					
$\nu = 0.48$	100	550.43	—	0.02	52.32	—	0.03	1326.63	—	0.08	170.40	—	0.10
	200	125.23	2.14	0.08	53.62	-0.04	0.12	291.96	2.18	0.32	81.76	1.06	0.44
	400	33.27	1.91	0.43	4.79	3.49	1.58	69.68	2.07	1.74	7.54	3.44	6.18
	800	7.82	2.09	2.11	2.78	0.78	14.43	17.06	2.03	8.42	5.36	0.49	58.24
	1600	1.91	2.03	9.47	0.51	2.44	125.89	4.10	2.06	38.53	1.01	2.41	507.48
		$T = 1250$						$T = 1500$					
	100	2440.04	—	0.10	351.76	—	0.12	9490.83	—	0.13	900.67	—	0.13
	200	503.08	2.38	0.40	139.83	1.33	0.55	1353.28	2.81	0.48	295.33	1.61	0.65
	400	123.00	2.03	2.18	8.43	4.05	7.71	308.88	2.13	2.67	23.82	3.63	9.23
	800	29.57	2.06	10.52	8.07	0.06	72.78	74.10	2.06	12.60	16.97	0.49	87.23
	1600	7.08	2.06	48.32	1.71	2.24	634.09	17.69	2.07	58.52	4.16	2.03	759.88
$m = 3$,		$T = 250$						$T = 1000$					
$\nu = 2.6$	100	760.70	—	0.03	878.66	—	0.03	6169.47	—	0.11	3103.46	—	0.11
	200	205.09	1.89	0.09	265.34	1.73	0.23	534.51	3.53	0.39	2630.90	0.24	0.75
	400	84.90	1.27	0.48	147.96	0.84	2.48	100.05	2.42	1.92	1491.90	0.82	9.93
	800	27.88	1.61	2.23	32.92	2.17	23.91	25.66	1.96	8.92	431.48	1.79	95.54
	1600	8.31	1.75	10.07	10.62	1.63	204.71	12.86	1.00	40.69	102.78	2.07	818.11
		$T = 1250$						$T = 1500$					
	100	280.86	—	0.13	7868.90	—	0.13	437.82	—	0.15	239.35	—	0.16
	200	113.89	1.30	0.49	4804.94	0.71	0.91	146.85	1.58	0.61	72.66	1.72	1.21
	400	46.04	1.31	2.44	1052.02	2.19	12.27	40.12	1.87	3.04	54.92	0.40	16.67
	800	14.58	1.66	11.41	149.30	2.82	119.93	17.51	1.20	14.23	11.10	2.31	163.30
	1600	5.19	1.49	51.99	29.97	2.32	1040.56	5.20	1.75	64.76	9.93	0.16	1413.58

Table 1.1: Example 1.1: approximate L^1 errors (e_M, figures to be multiplied by 10^{-6}), convergence rates (ϑ_M), and CPU times (cpu).

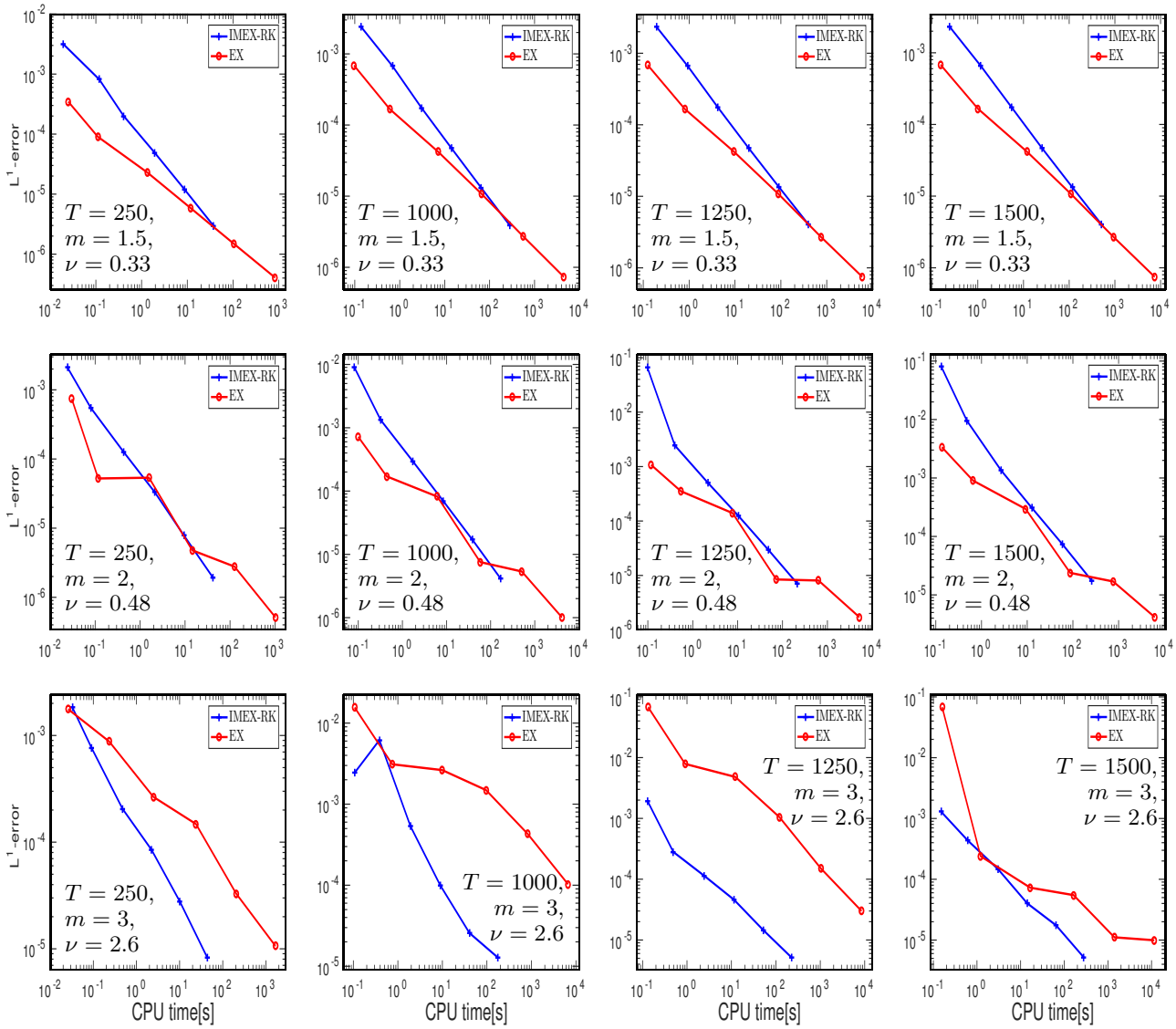

Figure 1.3: Example 1.1: efficiency plots: approximate L^1 errors versus CPU times for three pairs (m, ν), corresponding to four simulated times (figure produced by author).

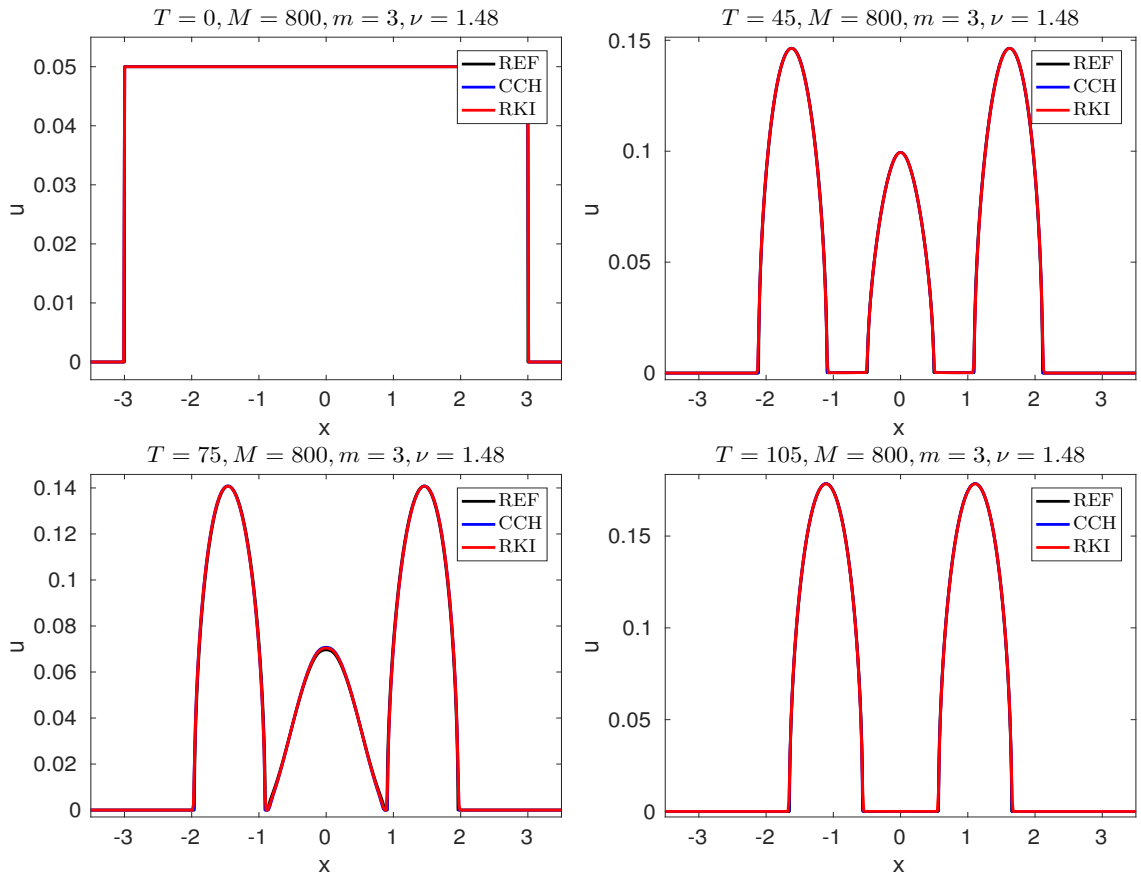

Figure 1.4: Example 1.2: numerical solution for $m = 3, \nu = 1.48$, $\Delta x = 2L/M$ with $L = 6$ and $M = 800$ (figure produced by author).

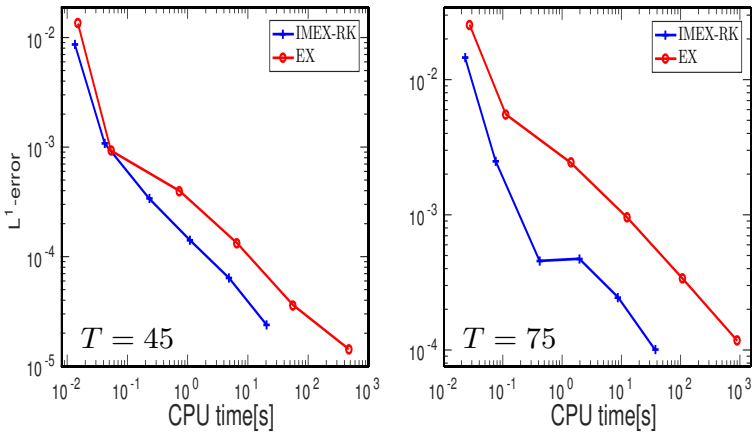

Figure 1.5: Example 1.2: efficiency plots based on numerical solutions for $\Delta x = 2L/M$ with $L = 6$ and $M = 100, 200, 400, 800$ and 1600 (figure produced by author).

	IMEX-RK			Explicit			IMEX-RK			Explicit		
M	e_M	ϑ_M	cpu [s]	e_M	ϑ_M	cpu [s]	e_M	ϑ_M	cpu [s]	e_M	ϑ_M	cpu [s]
	$T = 30$						$T = 45$					
100	969.59	—	0.03	488.36	—	0.03	1080.25	—	0.04	927.04	—	0.05
200	323.50	1.58	0.14	150.69	1.70	0.40	337.41	1.68	0.23	398.01	1.22	0.74
400	101.33	1.67	0.65	71.40	1.08	3.50	142.30	1.25	1.08	133.36	1.58	6.58
800	30.74	1.72	2.94	25.67	1.48	29.96	63.65	1.16	4.89	36.11	1.88	56.38
1600	10.91	1.49	12.40	7.01	1.87	250.92	23.69	1.43	20.63	14.33	1.33	472.33
	$T = 75$						$T = 105$					
100	2504.19	—	0.08	5528.90	—	0.11	1922.98	—	0.12	1343.35	—	0.17
200	456.17	4.46	0.42	2429.61	1.19	1.40	688.71	1.48	0.63	679.48	0.98	2.20
400	472.92	-0.05	1.95	952.62	1.35	12.61	210.39	1.71	2.91	100.70	2.75	20.20
800	245.74	0.94	8.78	337.33	1.50	108.16	71.85	1.55	13.15	66.79	0.59	174.81
1600	99.88	1.30	37.35	117.18	1.53	903.16	20.34	1.82	56.10	7.02	3.25	1495.84

Table 1.2: Example 1.2: approximate L^1 errors (e_M, figures to be multiplied by 10^{-6}), convergence rates (ϑ_M), and CPU times (cpu) for $m = 3$ and $\nu = 1.48$.

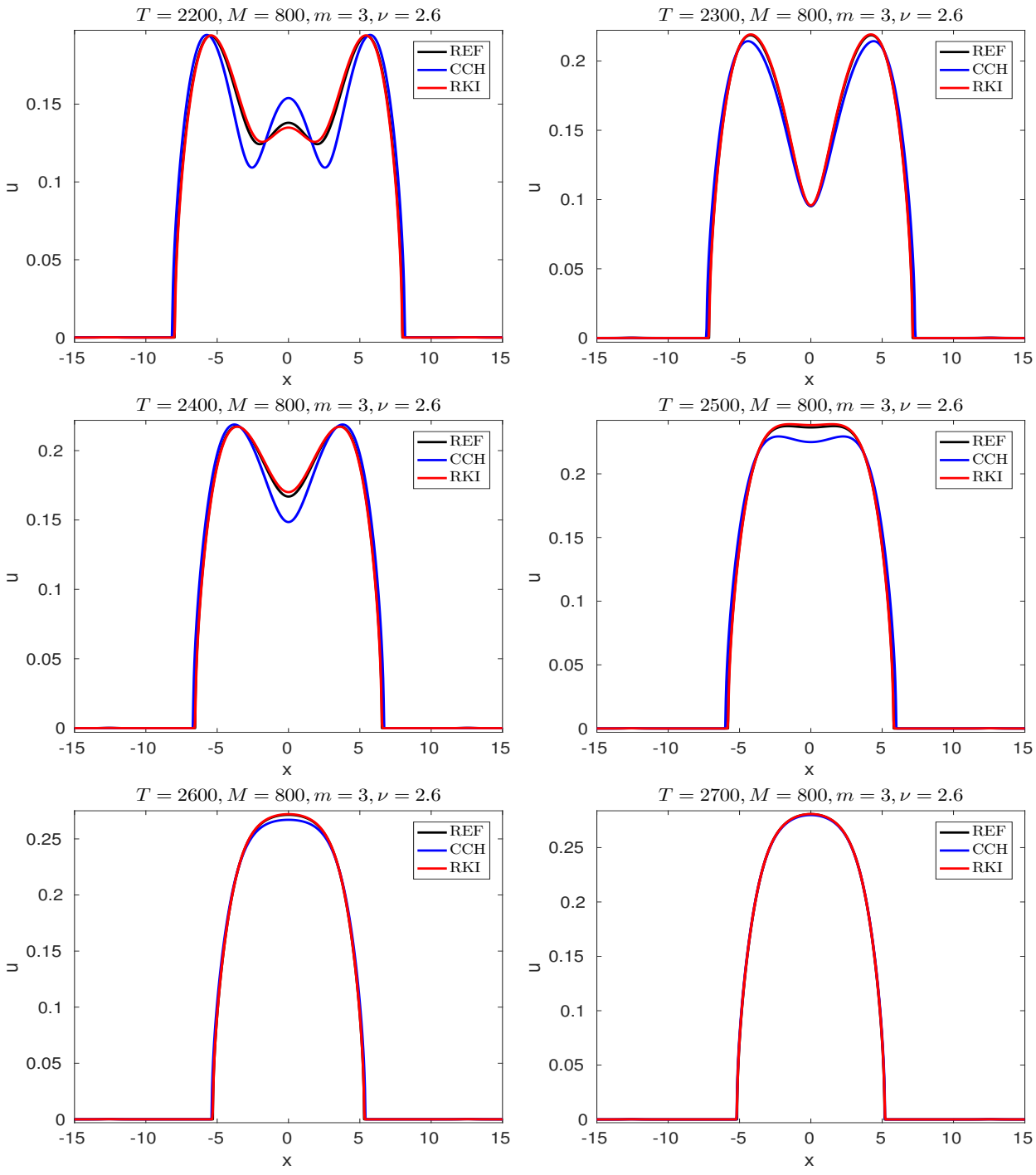

Figure 1.6: Example 1.3: numerical solution for $m = 3$, $\nu = 2.6$, $\Delta x = 2L/M$ with $L = 15$ and $M = 800$ (figure produced by author).

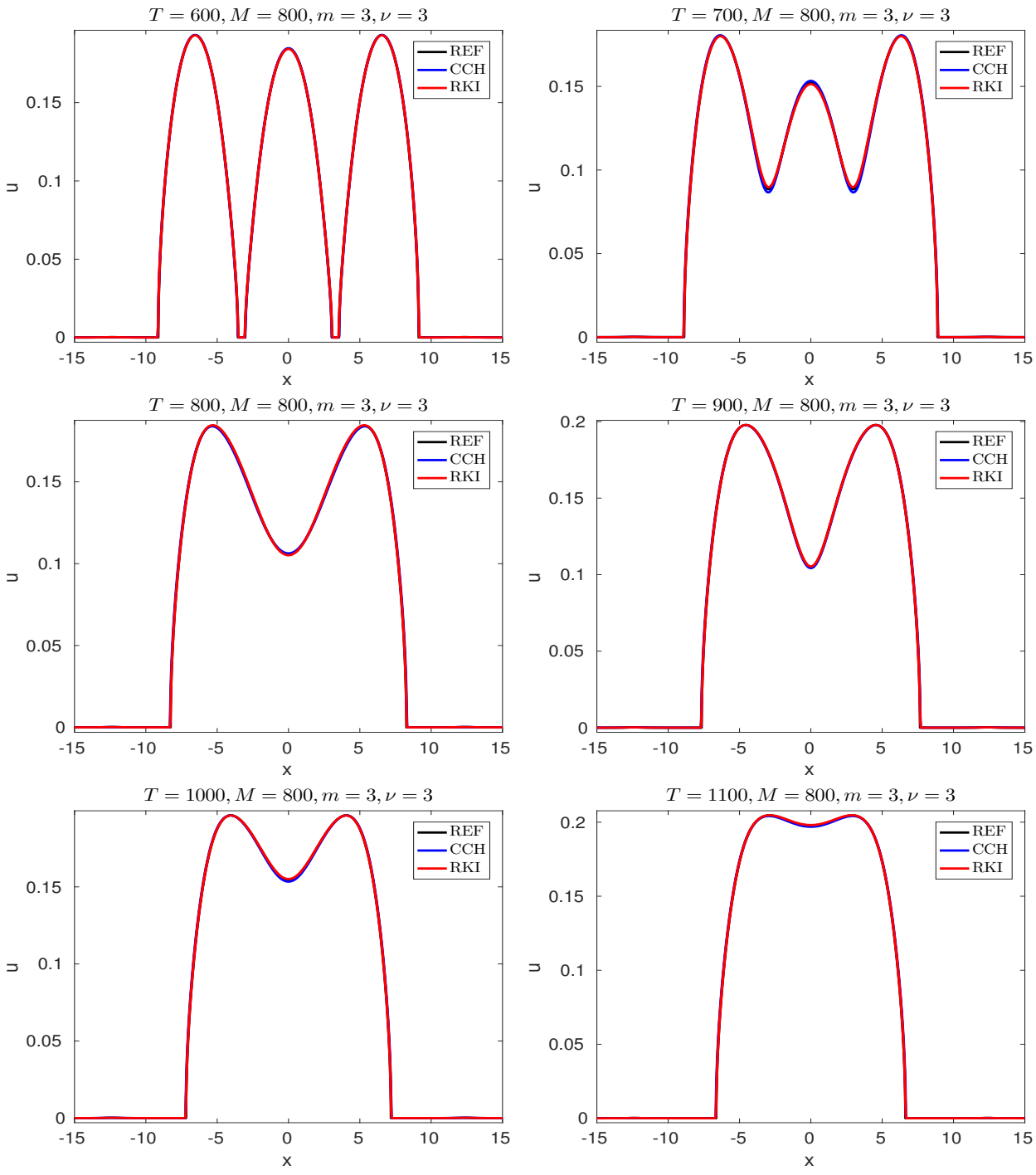

Figure 1.7: Example 1.3: numerical solution for $m = 3$, $\nu = 3$, $\Delta x = 2L/M$ with $L = 15$ and $M = 800$ (figure produced by author).

		IMEX-RK			Explicit			IMEX-RK			Explicit		
	M	e_M	ϑ_M	cpu [s]	e_M	ϑ_M	cpu [s]	e_M	ϑ_M	cpu [s]	e_M	ϑ_M	cpu [s]
$m = 3$,		$T = 2000$						$T = 2200$					
$\nu = 2.6$	100	25373.15	—	0.38	21792.46	—	0.63	28245.49	—	0.42	55270.59	—	0.70
	200	4019.83	2.66	2.00	21466.65	0.02	8.83	16656.50	0.76	2.21	55102.84	0.00	9.67
	400	1013.94	1.99	9.86	2628.76	3.03	91.53	5490.79	1.60	10.89	18401.83	1.58	100.24
	800	227.43	2.16	45.09	1029.62	1.35	801.49	1289.15	2.09	49.89	6699.83	1.46	875.41
	1600	76.02	1.58	194.65	267.19	1.95	6468.44	416.88	1.63	214.64	1650.75	2.02	7044.59
		$T = 2400$						$T = 2600$					
	100	27600.67	—	0.46	80217.69	—	0.76	2900.47	—	0.50	97113.69	—	0.83
	200	11820.57	1.22	2.42	80035.00	0.00	10.51	2046.62	0.50	2.65	97071.21	0.00	11.36
	400	3459.49	1.77	11.94	11016.22	2.86	108.51	806.98	1.34	13.06	5338.06	4.18	118.52
	800	765.42	2.18	54.73	4172.68	1.40	950.33	189.64	2.09	59.89	1459.04	1.87	1043.83
	1600	244.95	1.64	235.22	1060.75	1.98	7651.39	65.09	1.54	257.04	327.07	2.16	8422.95
		$T = 2700$						$T = 2900$					
	100	506.29	—	0.52	99018.36	—	0.86	578.80	—	0.56	99364.83	—	0.92
	200	296.52	0.77	2.76	98926.90	0.00	11.78	160.35	1.85	2.98	99273.19	0.00	12.63
	400	132.14	1.17	13.62	1177.50	6.39	125.50	77.46	1.05	14.73	86.15	10.17	140.78
	800	32.99	2.00	62.44	287.70	2.03	1107.61	20.00	1.95	67.57	16.84	2.36	1241.75
	1600	11.32	1.54	267.94	62.18	2.21	8938.30	9.20	1.12	290.11	5.94	1.50	10008.14
$m = 3$,		$T = 400$						$T = 600$					
$\nu = 3$	100	2360.54	—	0.07	1234.68	—	0.12	12307.41	—	0.11	3938.74	—	0.18
	200	588.13	2.00	0.37	507.41	1.28	1.63	1976.95	2.64	0.57	1351.61	1.54	2.45
	400	192.11	1.61	1.77	126.38	2.01	16.78	475.35	2.06	2.71	503.28	1.43	25.19
	800	58.33	1.72	8.34	53.63	1.24	146.45	138.80	1.78	12.65	171.59	1.55	219.82
	1600	19.63	1.57	35.31	13.49	1.99	1201.02	43.02	1.69	53.03	50.80	1.76	1795.61
		$T = 800$						$T = 1000$					
	100	9829.85	—	0.14	12258.41	—	0.23	8640.38	—	0.18	8186.72	—	0.28
	200	3098.79	1.67	0.74	3561.14	1.78	3.18	2138.93	2.01	0.94	2719.87	1.59	3.98
	400	800.36	1.95	3.62	1237.34	1.53	32.60	499.20	2.10	4.55	982.36	1.47	40.89
	800	226.01	1.82	16.95	407.87	1.60	284.42	135.84	1.88	21.40	329.44	1.58	357.06
	1600	68.59	1.72	71.10	119.90	1.77	2317.33	40.22	1.76	89.83	97.92	1.75	2904.90
		$T = 1000$						$T = 1300$					
	100	6613.15	—	0.20	8544.06	—	0.31	834.95	—	0.24	2942.87	—	0.38
	200	1750.78	1.92	1.03	2650.78	1.69	4.39	295.21	1.50	1.23	797.33	1.88	5.42
	400	398.99	2.13	5.02	936.43	1.50	45.20	78.21	1.92	6.03	258.01	1.63	56.12
	800	106.99	1.90	23.63	311.77	1.59	394.82	20.99	1.90	28.32	85.78	1.59	490.56
	1600	31.13	1.78	99.26	92.82	1.75	3210.16	6.13	1.78	118.78	25.74	1.74	3990.47

Table 1.3: Example 1.3: approximate L^1 errors (e_M, figures to be multiplied by 10^{-6}), convergence rates (ϑ_M), and CPU times (cpu).

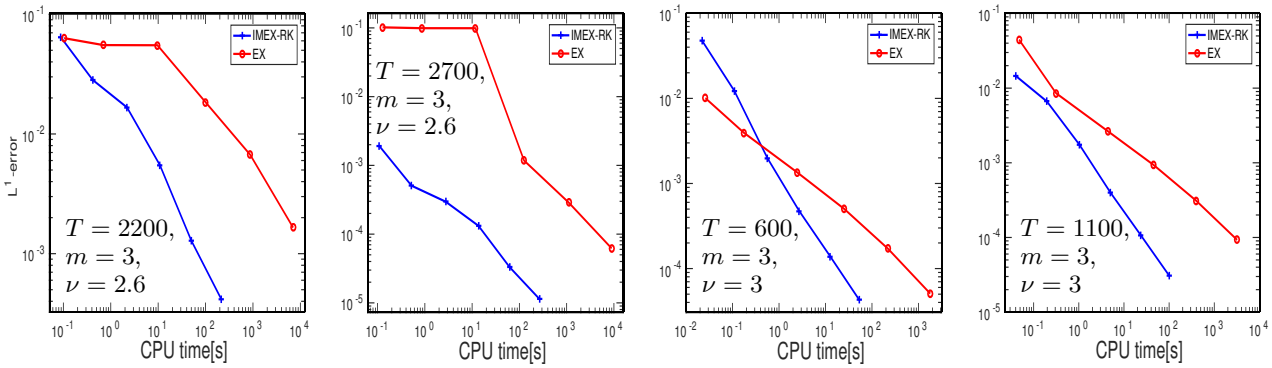

Figure 1.8: Example 1.3: efficiency plots based on numerical solutions for $\Delta x = 2L/M$ with $L = 15$ and $M = 100$, 200, 400, 800, and 1600 (figure produced by author).

	IMEX-RK		Explicit		IMEX-RK		Explicit	
M	e_M	cpu [s]	e_M	cpu [s]	e_M	cpu [s]	e_M	cpu [s]
	$T = 0.1$				$T = 0.2$			
50	212.98	0.01	233.41	0.03	216.19	0.01	322.13	0.06
100	78.58	0.03	124.21	0.15	95.25	0.05	184.56	0.31
200	49.13	0.13	69.55	2.27	55.49	0.25	98.93	4.70
400	20.25	0.63	25.86	19.87	27.99	1.25	58.03	42.03
800	10.71	3.04	13.84	180.65	19.77	6.07	32.69	380.37
1600	6.81	13.83	7.56	1469.53	15.37	27.78	19.71	3162.07
	$T = 0.3$				$T = 0.35$			
50	135.82	0.02	310.02	0.09	116.01	0.02	262.17	0.10
100	73.90	0.07	119.81	0.48	77.97	0.08	85.25	0.57
200	39.87	0.38	62.97	7.14	44.10	0.44	107.80	8.36
400	28.08	1.87	46.22	64.28	28.49	2.17	28.58	75.40
800	22.90	9.04	30.06	586.95	25.07	10.45	25.09	693.39
1600	20.49	41.54	23.72	4969.15	23.75	47.87	24.19	5914.65

Table 1.4: Example 1.4: approximate L^1 errors e_M (figures to be multiplied by 10^{-3}) and CPU times (cpu).

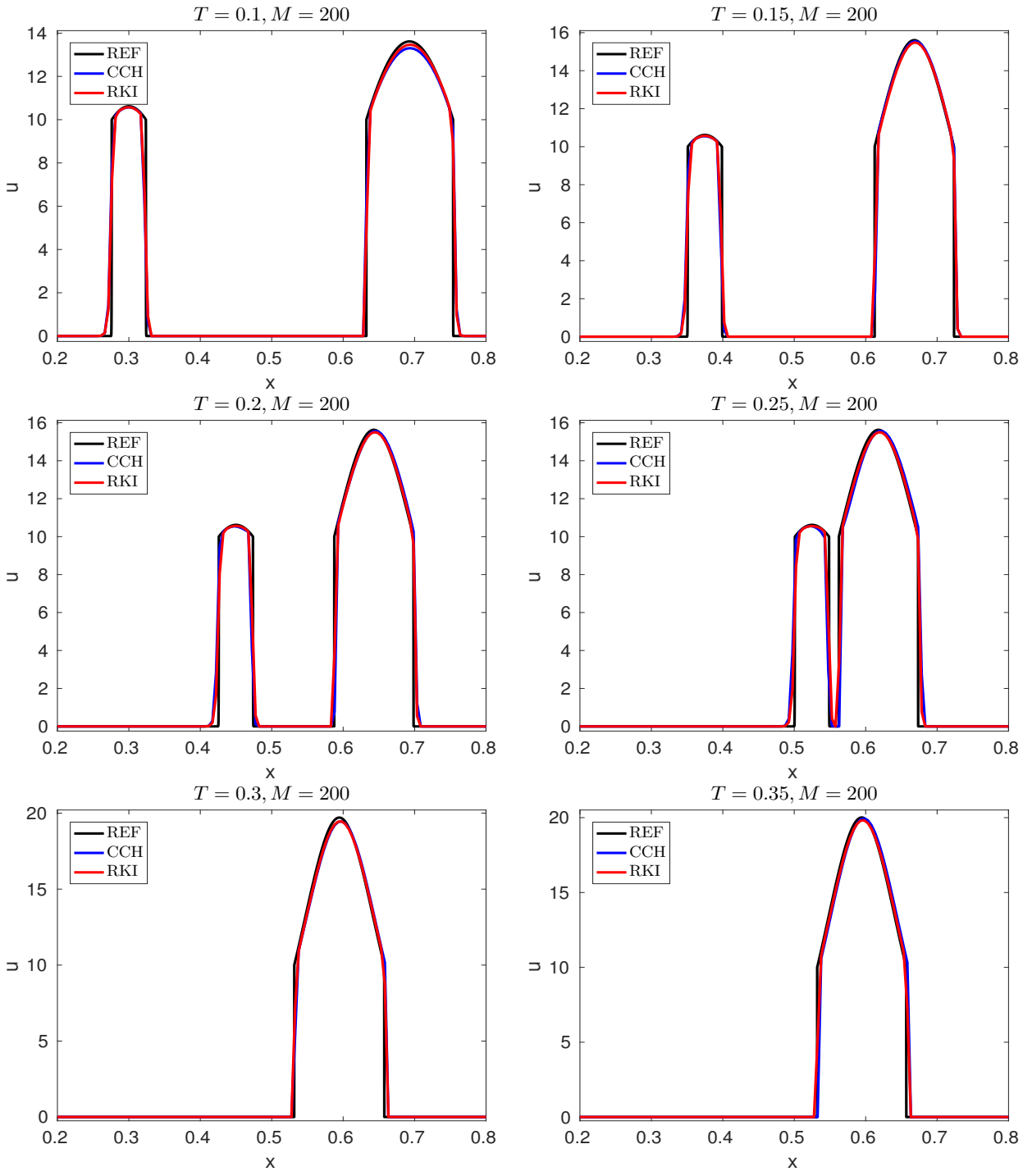

Figure 1.9: Example 1.4: numerical solution for $\Delta x = L/M$ and $M = 200$ (figure produced by author).

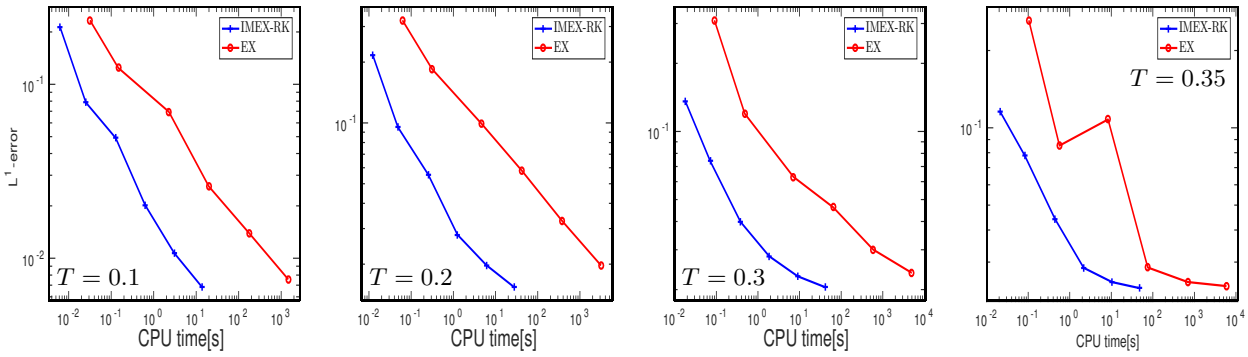

Figure 1.10: Example 1.4: efficiency plot based on numerical solutions for $\Delta x = L/M$ with $M = 100$, 200, 400, 800, and 1600 (figure produced by author).

CHAPTER 2

Implicit-explicit methods for a class of nonlinear nonlocal gradient flow equations modelling collective behaviour

2.1 Introduction

We are concerned with the efficient numerical solution of the following initial value problem for a nonlinear nonlocal partial differential equation (PDE) with a gradient flow structure in d space dimensions:

$$u_t = \nabla \cdot \big(u \nabla (H'(u) + V(\boldsymbol{x}) + W * u)\big), \quad \boldsymbol{x} = (x_1, \dots, x_d) \in \mathbb{R}^d, \quad t > 0, \tag{2.1}$$

$$u(\boldsymbol{x}, 0) = u_0(\boldsymbol{x}), \quad \boldsymbol{x} \in \mathbb{R}^d. \tag{2.2}$$

Here the sought solution $u = u(\boldsymbol{x}, t)$ is a probability distribution function or population density, $H(u)$ is a density of internal energy, $V(\boldsymbol{x})$ is a confinement potential, $W(\boldsymbol{x})$ is an interaction potential (which we assume to be symmetric), given by (0.11).

The PDE (2.1) includes, for instance, the classical heat equation and the porous medium equation under appropriate choices of W, V and H (see [18] for references). In general, it is useful to recall that (2.1) can be written as a nonlinear, nonlocal convection-diffusion equation

$$u_t + \nabla \cdot \big(u \boldsymbol{v}[u]\big) = \Delta \Phi(u),$$

where

$$\Phi(u) = \int_0^u s H''(s) \, \mathrm{d}s, \tag{2.3}$$

so that $\Phi'(u) = u H''(u)$ and $\boldsymbol{v}[u](\boldsymbol{x}) = -\nabla (W * u + V)(\boldsymbol{x})$. Here the notation $\boldsymbol{v}[u] = \boldsymbol{v}[u(\cdot, t)]$ means that the velocity $\boldsymbol{v}$ depends on $u(\cdot, t)$ as a function of $\boldsymbol{x}$ as a whole.

It is the purpose of this work to demonstrate the advantages of applying implicit-explicit (IMEX) schemes for the solution of (2.1), (2.2) (under specific assumptions on H or equivalently, Φ). The proposed schemes, based on IMEX Runge-Kutta (IMEX-RK) time discretizations, turn

out to be more efficient, in terms of error reduction versus CPU time, than the explicit scheme of [21].

2.2 Numerical method

2.2.1 Some assumptions and notation

We assume that $H''(u) \geq 0$ for all $u \in (0,\infty)$, $H'' \in C^1(\mathbb{R} \setminus \{0\})$, so that

$$\begin{aligned} &\Phi \in C^1([0,\infty)) \cap C^2((0,\infty)), \\ &\Phi(0) = \Phi'(0) = 0, \\ &\Phi(u) \geq 0, \Phi'(u) \geq 0 \text{ for } u \in [0,\infty). \end{aligned} \tag{2.4}$$

We limit the treatment to the spatial domain given by the d-dimensional open interval

$$\Omega := (-L_1, L_1) \times \cdots \times (-L_d, L_d) \tag{2.5}$$

and denote by $u : \Omega \times (0,\infty) \to [0,\infty)$ the solution of (2.1). Each coordinate interval $(-L_l, L_l)$, $l = 1,\ldots,d$, is subdivided into M_l subintervals of size $\Delta x_l = 2L_l/M_l$. This creates a number $M_* := M_1 M_2 \cdots M_d$ of finite volumes or cells $C_{\boldsymbol{i}}$, which we indicate by $\boldsymbol{i} = (i_1,\ldots,i_d) \in \mathcal{M}$, where we define the index set

$$\mathcal{M} := \{1,\ldots,M_1\} \times \cdots \times \{1,\ldots,M_d\} \subset \mathbb{N}^d. \tag{2.6}$$

The center of $C_{\boldsymbol{i}}$ is denoted by $\boldsymbol{x}_{\boldsymbol{i}}$, so the coordinates of $\boldsymbol{x}_{\boldsymbol{i}}$ are

$$\boldsymbol{x}_{\boldsymbol{i}} = (x_{1,i_1},\ldots,x_{d,i_d}) = \big((i_1 - o_1)\Delta x_1,\ldots,(i_d - o_d)\Delta x_d\big), \tag{2.7}$$

where $o_l = (M_l + 1)/2$ for $l = 1,\ldots,d$, such that $x_{l,1} = -L_l + \Delta x_l/2$ and $x_{L,M_l} = L_l - \Delta x_l/2$, $l = 1,\ldots,d$, and we utilize d-dimensional unit vectors $\boldsymbol{e}_1 = (1,0,\ldots,0)$ to $\boldsymbol{e}_d = (0,\ldots,0,1)$ to address neighboring grid points, for instance $\boldsymbol{x}_{\boldsymbol{i}+\boldsymbol{e}_1} = \boldsymbol{x}_{i_1+1,i_2,\ldots,i_d}$. A similar notation is used to address flux values associated with cell interfaces. Furthermore, we assume that the velocity vector $\boldsymbol{v}$ is given by components $\boldsymbol{v} = (v^1,\ldots,v^d)^{\mathrm{T}}$.

2.2.2 Spatial semi-discretization

To define the spatial (semi-)discretization, assume first that at an instant t the solution is given through the cell averages $u_{\boldsymbol{i}} = u_{\boldsymbol{i}}(t)$ for $\boldsymbol{i} \in \mathcal{M}$. As in the one-dimensional treatment [18], we use MUSCL reconstructions [57], which amounts for each cell $C_{\boldsymbol{i}}$ to calculating the following reconstructed values at the boundaries of $C_{\boldsymbol{i}}$:

$$u_{\boldsymbol{i}}^{l,\pm} = u_{\boldsymbol{i}} \pm \frac{\Delta x_l}{2}\sigma_{\boldsymbol{i}}^{(l)}, \quad \boldsymbol{i} \in \mathcal{M}, \quad l = 1,\ldots,d, \tag{2.8}$$

where the slope $\sigma_{\boldsymbol{i}}^{(l)}$ for the extrapolation of $u_{\boldsymbol{i}}$ in coordinate direction l is defined by using a so-called slope limiter that guarantees that the reconstructed point values are nonnegative as long as the cell averages $u_{\boldsymbol{i}}$ are nonnegative. Specifically, we utilize the slope limiter defined by

$$\sigma_{\boldsymbol{i}}^{(l)} = \begin{cases} \dfrac{1}{2\Delta x_l}\left(u_{\boldsymbol{i}+\boldsymbol{e}_l} - u_{\boldsymbol{i}-\boldsymbol{e}_l}\right) & \text{if } u_{\boldsymbol{i}} \geq |u_{\boldsymbol{i}+\boldsymbol{e}_l} - u_{\boldsymbol{i}-\boldsymbol{e}_l}|/4, \\ \vartheta \operatorname{minmod}\left\{\dfrac{1}{\Delta x_l}\left(u_{\boldsymbol{i}+\boldsymbol{e}_l} - u_{\boldsymbol{i}}\right), \dfrac{1}{\Delta x_l}\left(u_{\boldsymbol{i}} - u_{\boldsymbol{i}-\boldsymbol{e}_l}\right)\right\} & \text{otherwise,} \end{cases}$$

where the standard minmod function is given by

$$\operatorname{minmod}\{z_1, z_2\} := \begin{cases} \operatorname{sgn}(z_1)\min\{|z_1|, |z_2|\} & \text{if} \quad \operatorname{sgn}(z_1) = \operatorname{sgn}(z_2), \\ 0 & \text{otherwise.} \end{cases}$$

The parameter $\vartheta \in (0, 2]$ is used to control the numerical viscosity of the scheme. The value $\vartheta = 2$ is used in [18, 21], and we adopt it here in all numerical examples.

To approximate $\boldsymbol{v}[u](x_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}_l})$, we use the formula

$$\left.\frac{\partial z[u]}{\partial x_l}\right|_{\boldsymbol{x}=\boldsymbol{x}_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}_l}} \approx \frac{1}{\Delta x_l}\big(z[u](\boldsymbol{x}_{\boldsymbol{i}+\boldsymbol{e}_l}) - z[u](\boldsymbol{x}_{\boldsymbol{i}})\big). \tag{2.9}$$

If we assume that $u(t)$ is compactly supported in $(-L, L)^d$, then the discrete approximations of the convolutions

$$z[u](\boldsymbol{x}_{\boldsymbol{i}}) := (W * u + V)(\boldsymbol{x}_{\boldsymbol{i}}),$$

shifted by the function V, are given by

$$(W * u + V)(\boldsymbol{x}_{\boldsymbol{i}}) \approx \tilde{z}[u]_{\boldsymbol{i}} := \prod_{l=1}^{d} \Delta x_l \sum_{-\rho \leq p_1, \ldots, p_d \leq \rho} W_{\boldsymbol{p}}\, u^*_{\boldsymbol{i}-\boldsymbol{p}} + V_{\boldsymbol{i}}, \tag{2.10}$$

where $V_{\boldsymbol{i}} = V(\boldsymbol{x}_{\boldsymbol{i}})$ for $\boldsymbol{p} = (p_1, \ldots, p_d) \in \mathbb{Z}^d$, we define $W_{\boldsymbol{p}} := W(p_1\Delta x_1, \ldots, p_d\Delta x_d)$, and

$$u^*_{\boldsymbol{i}-\boldsymbol{p}} := \begin{cases} u_{\boldsymbol{i}-\boldsymbol{p}} & \text{if } \boldsymbol{i}-\boldsymbol{p} \in \mathcal{M}, \\ 0 & \text{otherwise,} \end{cases}$$

where the radius of the stencil $\rho \in \mathbb{N}_0$ is computed to retain second-order accuracy. To select ρ, we proceed as in [18] by choosing ρ as the smallest integer such that

$$1 - \frac{\sum_{n_1=-\rho}^{\rho} \cdots \sum_{n_d=-\rho}^{\rho} W(n_1\Delta x_1, \ldots, n_d\Delta x_d)}{\sum_{n_1=-\infty}^{\infty} \cdots \sum_{n_d=-\infty}^{\infty} W(n_1\Delta x_1, \ldots, n_d\Delta x_d)} \leq \xi \Delta x^2, \quad \Delta x = \min\{\Delta x_1, \ldots, \Delta x_d\},$$

where we have taken $\xi = 10^{-8}$ in all our numerical examples and the term

$$\sum_{n_1=-\infty}^{\infty} \cdots \sum_{n_d=-\infty}^{\infty} W(n_1\Delta x_1, \ldots, n_d\Delta x_d)$$

is approximated by

$$\sum_{n_1=-N_1}^{N_1} \cdots \sum_{n_d=-N_d}^{N_d} W(n_1\Delta x_1, \dots, n_d\Delta x_d)$$

for very large $N_1, \dots, N_d$. Here we used that W is a symmetric function. Clearly, the discrete convolution in (2.10) causes a computational bottleneck. This is a classical problem in scientific computing that is effectively handled by fast convolution algorithms, mainly based on Fast Fourier Transforms [59].

Moreover, we apply upwinding based on the sign of the l-th component $\hat{v}^l_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}_l}$ of the vector

$$\boldsymbol{v}_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}_l} = \frac{1}{\Delta x}\big(\boldsymbol{v}[u]_{\boldsymbol{i}+\boldsymbol{e}_l} - \boldsymbol{v}[u]_{\boldsymbol{i}}\big) = \big(\hat{v}^1_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}_l}, \dots, \hat{v}^d_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}_l}\big)^{\mathrm{T}},$$

where, in agreement with (2.9), the l-th component of $\hat{\boldsymbol{v}}[u]$, the discrete version of $\boldsymbol{v}[u]$, is given by

$$\hat{v}^l_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}_l} = \frac{1}{\Delta x_l}\big(\tilde{z}[u]_{\boldsymbol{i}+\boldsymbol{e}_l} - \tilde{z}[u]_{\boldsymbol{i}}\big).$$

The upwind procedure now consists in choosing the u-value associated with the cell interface $\boldsymbol{i} + \frac{1}{2}\boldsymbol{e}_l$ as follows:

$$u_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}_l} = \begin{cases} u^{l,+}_{\boldsymbol{i}} & \text{if } \hat{v}^l_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}_l} \geq 0, \\ u^{l,-}_{\boldsymbol{i}+\boldsymbol{e}_l} & \text{if } \hat{v}^l_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}_l} < 0. \end{cases} \tag{2.11}$$

Solution values are extended by zero outside the domain, i.e., we set $u_{\boldsymbol{i}} := 0$ for $\boldsymbol{i} \in \mathbb{Z}^d \backslash \mathcal{M}$.

Combining all ingredients, we may write the semidiscrete scheme in compact form as (0.4), where $\boldsymbol{u} : [0, \infty) \to \mathbb{R}^{M_*}$ and we recall that $\boldsymbol{\mathcal{C}}(\boldsymbol{u})$ and $\boldsymbol{\mathcal{D}}(\boldsymbol{u})$ represent the spatial discretizations of the convective and the diffusive terms, i.e., the respective entries of $\boldsymbol{\mathcal{C}}(\boldsymbol{u}) = (C(\boldsymbol{u})_{\boldsymbol{i}})_{\boldsymbol{i}\in\mathcal{M}}$ and $\boldsymbol{\mathcal{D}}(\boldsymbol{u}) = (D(\boldsymbol{u})_{\boldsymbol{i}})_{\boldsymbol{i}\in\mathcal{M}}$ are given by

$$C(\boldsymbol{u})_{\boldsymbol{i}} = -\sum_{l=1}^{d} \frac{1}{\Delta x_l}\big(u_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}_l}\hat{v}^l_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}_l} - u_{\boldsymbol{i}-\frac{1}{2}\boldsymbol{e}_l}\hat{v}^l_{\boldsymbol{i}-\frac{1}{2}\boldsymbol{e}_l}\big), \tag{2.12}$$

$$D(\boldsymbol{u})_{\boldsymbol{i}} = \sum_{l=1}^{d} \frac{1}{\Delta x_l^2}\big(\Phi(u_{\boldsymbol{i}+\boldsymbol{e}_l}) - 2\Phi(u_{\boldsymbol{i}}) + \Phi(u_{\boldsymbol{i}-\boldsymbol{e}_l})\big). \tag{2.13}$$

In closed form the finite volume semidiscretization on cells centered at $\boldsymbol{x}_{\boldsymbol{i}}$ can be written as the system of ODEs

$$\frac{\mathrm{d}u_{\boldsymbol{i}}}{\mathrm{d}t} = -\sum_{l=1}^{d} \frac{1}{\Delta x_l}\big(u_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}_l}\hat{v}^l_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}_l} - u_{\boldsymbol{i}-\frac{1}{2}\boldsymbol{e}_l}\hat{v}^l_{\boldsymbol{i}-\frac{1}{2}\boldsymbol{e}_l}\big) + \sum_{l=1}^{d} \frac{1}{\Delta x_l^2}\big(\Phi(u_{\boldsymbol{i}+\boldsymbol{e}_l}) - 2\Phi(u_{\boldsymbol{i}}) + \Phi(u_{\boldsymbol{i}-\boldsymbol{e}_l})\big).$$

Theorem 2.1. *If $\Phi' \geq 0$ on $(0,\infty)$, $u_{\boldsymbol{i}} \geq 0$ for all $\boldsymbol{i} \in \mathcal{M}$ and the CFL condition*

$$\frac{1}{d} \max_{0\leq u\leq \eta(\boldsymbol{u})} \Phi'(u) \sum_{q=1}^{d} \frac{\Delta t}{\Delta x_q^2} + \max_{1\leq l\leq d} \left\{ \frac{\Delta t}{\Delta x_l} \max_{\boldsymbol{k}\in\mathcal{M}} \left| v^l_{\boldsymbol{k}+\frac{1}{2}\boldsymbol{e}_l} \right| \right\} \leq \frac{1}{2d}, \quad \eta(\boldsymbol{u}) := \max_{\boldsymbol{j}\in\mathcal{M}} u_{\boldsymbol{j}} \tag{2.14}$$

is satisfied, then the quantity

$$\mathcal{E}(\boldsymbol{u})_{\boldsymbol{i}} := u_{\boldsymbol{i}} + \Delta t \big(C(\boldsymbol{u})_{\boldsymbol{i}} + D(\boldsymbol{u})_{\boldsymbol{i}} \big) \tag{2.15}$$

satisfies $\mathcal{E}(\boldsymbol{u})_{\boldsymbol{i}} \geq 0$ for all $\boldsymbol{i} \in \mathcal{M}$, i.e., the explicit Euler method applied to the semi-discrete scheme (0.4) yields a fully discrete positivity preserving scheme.

Proof. From (2.8) there hold

$$\begin{aligned} &\frac{1}{2d} \sum_{l=1}^{d} \big(u_{\boldsymbol{i}}^{l,+} + u_{\boldsymbol{i}}^{l,-} \big) = u_{\boldsymbol{i}} \quad \text{for } \boldsymbol{i} \in \mathcal{M}, \\ &u_{\boldsymbol{i}}^{l,\pm} \geq 0 \quad \text{for } \boldsymbol{i} \in \mathcal{M},\ l = 1,\dots,d. \end{aligned} \tag{2.16}$$

Moreover for all $\boldsymbol{i} \in \mathcal{M}$ there exist convex combinations

$$\hat{u}_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}_l} = \vartheta_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}_l} u_{\boldsymbol{i}} + \big(1 - \vartheta_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}_l}\big) u_{\boldsymbol{i}+\boldsymbol{e}_l}, \quad \vartheta_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}_l} \in (0,1), \quad l = 1,\dots,d,$$

such that for all $\boldsymbol{i} \in \mathcal{M}$ and $l = 1,\dots,d$,

$$\Phi(u_{\boldsymbol{i}+\boldsymbol{e}_l}) - \Phi(u_{\boldsymbol{i}}) = \Delta x_l^2 \beta_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}_l} (u_{\boldsymbol{i}+\boldsymbol{e}_l} - u_{\boldsymbol{i}}), \quad \beta_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}_l} = \Phi'(\hat{u}_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}_l}) / \Delta x_l^2. \tag{2.17}$$

Then $D(\boldsymbol{u})_{\boldsymbol{i}}$, given by (2.13), can be written as

$$D(\boldsymbol{u})_{\boldsymbol{i}} = \sum_{l=1}^{d} \big(\beta_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}_l} u_{\boldsymbol{i}+\boldsymbol{e}_l} + \beta_{\boldsymbol{i}-\frac{1}{2}\boldsymbol{e}_l} u_{\boldsymbol{i}-\boldsymbol{e}_l} \big) + 2d\gamma_{\boldsymbol{i}} u_{\boldsymbol{i}}, \tag{2.18}$$

where we use the notation

$$\gamma_{\boldsymbol{i}} := \frac{1}{2d} \sum_{l=1}^{d} \big(\beta_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}_l} + \beta_{\boldsymbol{i}-\frac{1}{2}\boldsymbol{e}_l} \big). \tag{2.19}$$

Therefore, we obtain from (2.19) and (2.17) that

$$\max_{\boldsymbol{j}\in\mathcal{M}} \gamma_{\boldsymbol{j}} \leq \frac{1}{d} \max_{0\leq u\leq \eta(\boldsymbol{u})} \Phi'(u) \sum_{q=1}^{d} \frac{1}{\Delta x_q^2}. \tag{2.20}$$

From (2.11) we obtain

$$u_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}_l} \hat{v}^l_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}_l} = u_{\boldsymbol{i}}^{l,+} \hat{v}^{l,+}_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}_l} + u_{\boldsymbol{i}+\boldsymbol{e}_l}^{l,-} \hat{v}^{l,-}_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}_l}, \quad \hat{v}^{l,\pm}_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}_l} = \big(\hat{v}^l_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}_l} \big)^{\pm}.$$

By (2.15), (2.16), (2.12) and (2.18) we may write $\mathcal{E}(\boldsymbol{u})_{\boldsymbol{i}}$ as follows:

$$\begin{aligned}\mathcal{E}(\boldsymbol{u})_{\boldsymbol{i}} = \frac{1-2d\Delta t\gamma_{\boldsymbol{i}}}{2d}\sum_{l=1}^{d}\big(u_{\boldsymbol{i}}^{l,+}+u_{\boldsymbol{i}}^{l,-}\big)\\ +\Delta t\sum_{l=1}^{d}\frac{1}{\Delta x_l}\big(-u_{\boldsymbol{i}}^{l,+}v_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}_l}^{l,+}-u_{\boldsymbol{i}+\boldsymbol{e}_l}^{l,-}v_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}_l}^{l,-}+u_{\boldsymbol{i}-\boldsymbol{e}_l}^{l,+}v_{\boldsymbol{i}-\frac{1}{2}\boldsymbol{e}_l}^{l,+}+u_{\boldsymbol{i}}^{l,-}v_{\boldsymbol{i}-\frac{1}{2}\boldsymbol{e}_l}^{l,-}\big)\\ +\Delta t\sum_{l=1}^{d}\big(\beta_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}_l}u_{\boldsymbol{i}+\boldsymbol{e}_l}+\beta_{\boldsymbol{i}-\frac{1}{2}\boldsymbol{e}_l}u_{\boldsymbol{i}-\boldsymbol{e}_l}\big).\end{aligned}$$

Taking into account that $u_{\boldsymbol{i}}\geq 0$, $u_{\boldsymbol{i}}^{l,\pm}\geq 0$, $v_{\boldsymbol{i}\pm\frac{1}{2}\boldsymbol{e}_l}^{l,\pm}\geq 0$, and (2.20), we deduce

$$\begin{aligned}\mathcal{E}(\boldsymbol{u})_{\boldsymbol{i}} &\geq \left(\frac{1}{2d}-\Delta t\gamma_{\boldsymbol{i}}\right)\sum_{l=1}^{d}\big(u_{\boldsymbol{i}}^{l,+}+u_{\boldsymbol{i}}^{l,-}\big)+\Delta t\sum_{l=1}^{d}\frac{1}{\Delta x_l}\big(-u_{\boldsymbol{i}}^{l,+}v_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}_l}^{l,+}+u_{\boldsymbol{i}}^{l,-}v_{\boldsymbol{i}-\frac{1}{2}\boldsymbol{e}_l}^{l,-}\big)\\ &\geq \sum_{l=1}^{d}\left(\left(\frac{1}{2d}-\Delta t\gamma_{\boldsymbol{i}}-\frac{\Delta t}{\Delta x_l}\big|v_{\boldsymbol{i}-\frac{1}{2}\boldsymbol{e}_l}^{l}\big|\right)u_{\boldsymbol{i}}^{l,-}+\left(\frac{1}{2d}-\Delta t\gamma_{\boldsymbol{i}}-\frac{\Delta t}{\Delta x_l}\big|v_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}_l}^{l}\big|\right)u_{\boldsymbol{i}}^{l,+}\right)\\ &\geq \sum_{l=1}^{d}\left(\frac{1}{2d}-\Delta t\max_{\boldsymbol{j}\in\mathcal{M}}\gamma_{\boldsymbol{j}}-\frac{\Delta t}{\Delta x_l}\max_{\boldsymbol{j}\in\mathcal{M}}\big|v_{\boldsymbol{j}-\frac{1}{2}\boldsymbol{e}_l}^{l}\big|\right)\big(u_{\boldsymbol{i}}^{l,-}+u_{\boldsymbol{i}}^{l,+}\big)\\ &\geq \left(\frac{1}{2d}-\frac{1}{d}\max_{0\leq u\leq\eta(\boldsymbol{u})}\Phi'(u)\sum_{q=1}^{d}\frac{\Delta t}{\Delta x_q^2}+\max_{1\leq l\leq d}\left\{\frac{\Delta t}{\Delta x_l}\max_{\boldsymbol{k}\in\mathcal{M}}\big|v_{\boldsymbol{k}+\frac{1}{2}\boldsymbol{e}_l}^{l}\big|\right\}\right)\sum_{l=1}^{d}\big(u_{\boldsymbol{i}}^{l,+}+u_{\boldsymbol{i}}^{l,-}\big)\geq 0.\end{aligned}$$

This concludes the proof. □

2.2.3 Time discretization

As in [18] we will use IMEX-RK integration for the system of ODEs (0.4), and where the components of $\boldsymbol{\mathcal{C}}(\boldsymbol{u})$ and $\boldsymbol{\mathcal{D}}(\boldsymbol{u})$ are given by (2.12) and (2.13), respectively.

Algorithm 3.1 (see section 1.2.4) requires solving for the vector $\boldsymbol{u}=\boldsymbol{u}^{(m)}$ a nonlinear system of scalar equations of the form

$$\boldsymbol{F}_m(\boldsymbol{u}) := \boldsymbol{u}-a_{mm}\Delta t\boldsymbol{\mathcal{D}}(\boldsymbol{u})-\boldsymbol{r}_m=\boldsymbol{0},\quad m=1,\dots,s. \tag{2.21}$$

The solution of (2.21) is positive as long as $\boldsymbol{r}_m$ is positive. The following result, which is a generalization of Theorem 1.2 in Chapter 1, deals with the solution of (2.21). (This theorem is formulated for a general system of nonlinear equations of a particular form; in the subsequent Corollary 2.3, we will apply it to the special case of (2.21).)

Theorem 2.2. *Let $\boldsymbol{G}$ be a symmetric invertible diagonally dominant $M\times M$ matrix, with positive diagonal entries and non-positive off-diagonal entries and $\boldsymbol{w}\in\mathbb{R}^M$, $\boldsymbol{w}\geq\boldsymbol{0}$, where such*

inequalities for vectors and matrices are understood in the component-wise sense. If Φ *satisfies* (2.4) *and* $\boldsymbol{\Phi}$ *denotes its vectorial component-wise extension* $\boldsymbol{\Phi}(\boldsymbol{u})_i = \Phi(u_i)$*, then the equation*

$$\boldsymbol{z} + \boldsymbol{G}\boldsymbol{\Phi}(\boldsymbol{z}) = \boldsymbol{w} \tag{2.22}$$

has a unique solution $\boldsymbol{z} \in \mathbb{R}^M$ *satisfying* $\boldsymbol{z} \geq \boldsymbol{0}$*.*

Proof. We define the function

$$L(u) := \begin{cases} \Phi(u)/u & \text{if } u > 0, \\ 0 & \text{if } u = 0. \end{cases}$$

In view of the requirements in (2.4), L is continuous in $[0, \infty)$ and $L(u), \Phi(u) \geq 0$ for $u \geq 0$. Let

$$\boldsymbol{E}(\boldsymbol{z}) := \operatorname{diag}\big(L(z_1), \ldots, L(z_M)\big),$$

then $\boldsymbol{\Phi}(\boldsymbol{z}) = \boldsymbol{E}(\boldsymbol{z})\boldsymbol{z}$. We denote by $\boldsymbol{I}$ the $M \times M$ identity matrix. For $\boldsymbol{z} \geq \boldsymbol{0}$, the matrix $\boldsymbol{I} + \boldsymbol{G}\boldsymbol{E}(\boldsymbol{z})$ is strictly diagonally dominant (by columns) with positive diagonal entries and non-positive off-diagonal entries, and therefore $(\boldsymbol{I} + \boldsymbol{G}\boldsymbol{E}(\boldsymbol{z}))^{-1}$ is a non-negative matrix and it is a continuous function of $\boldsymbol{z}$. Then, the solution of equation (2.22) is reduced to finding fixed points of the mapping $\boldsymbol{z} \mapsto \boldsymbol{\varphi}(\boldsymbol{z}) = (\boldsymbol{I} + \boldsymbol{G}\boldsymbol{E}(\boldsymbol{z}))^{-1}\boldsymbol{w}$. To assess existence of fixed points, we aim to apply Brouwer's theorem to $\boldsymbol{\varphi}$ and the compact and convex set $\mathcal{K} := \{\boldsymbol{z} \in \mathbb{R}^M \mid \boldsymbol{z} \geq \boldsymbol{0} \text{ and } \|\boldsymbol{z}\|_1 \leq \|\boldsymbol{w}\|_1\}$. Clearly, $(\boldsymbol{I} + \boldsymbol{G}\boldsymbol{E}(\boldsymbol{z}))^{-1} \geq \boldsymbol{0}$ and $\boldsymbol{w} \geq \boldsymbol{0}$ immediately yield $\boldsymbol{\varphi}(\boldsymbol{z}) \geq \boldsymbol{0}$ for all $\boldsymbol{z} \in \mathcal{K}$, so, to prove that $\boldsymbol{\varphi}(\mathcal{K}) \subseteq \mathcal{K}$, there only remains to prove that

$$\big\|\boldsymbol{\varphi}(\boldsymbol{z})\big\|_1 \leq \|\boldsymbol{w}\|_1 \quad \text{for all } \boldsymbol{z} \in \mathcal{K}. \tag{2.23}$$

To this end, we take into account that

$$\big\|\boldsymbol{\varphi}(\boldsymbol{z})\big\|_1 \leq \big\|\big(\boldsymbol{I} + \boldsymbol{G}\boldsymbol{E}(\boldsymbol{z})\big)^{-1}\big\|_1 \|\boldsymbol{w}\|_1.$$

Thus, to establish (2.23) it is sufficient to prove that

$$\big\|\big(\boldsymbol{I} + \boldsymbol{G}\boldsymbol{E}(\boldsymbol{z})\big)^{-1}\big\|_1 \leq 1 \quad \text{for all } \boldsymbol{z} \in \mathcal{K}.$$

For this purpose, we use the auxiliary matrix $\tilde{\boldsymbol{G}} = (\tilde{G}_{ij})_{1 \leq i,j \leq M}$ defined by

$$\tilde{G}_{ij} := \begin{cases} G_{ij} & \text{if } i \neq j, \\ -\sum_{k \neq i} G_{ik} & \text{if } i = j \end{cases}$$

and the notation $\boldsymbol{H} := \boldsymbol{I} + \boldsymbol{G}\boldsymbol{E}(\boldsymbol{z})$ and $\tilde{\boldsymbol{H}} := \boldsymbol{I} + \tilde{\boldsymbol{G}}\boldsymbol{E}(\boldsymbol{z})$. Since $\tilde{\boldsymbol{H}}$ is also a strictly diagonally dominant matrix (by columns) with positive diagonal entries and non-positive off-diagonal entries, $\tilde{\boldsymbol{H}}^{-1} \geq \boldsymbol{0}$. Now, for $\boldsymbol{e} := (1, \ldots, 1)^{\mathrm{T}} \in \mathbb{R}^M$ it follows that $\boldsymbol{e}^{\mathrm{T}}\tilde{\boldsymbol{G}} = \boldsymbol{0}$, so $\boldsymbol{e}^{\mathrm{T}}\tilde{\boldsymbol{H}} = \boldsymbol{e}^{\mathrm{T}}$

and $\boldsymbol{e}^{\mathrm{T}}\tilde{\boldsymbol{H}}^{-1} = \boldsymbol{e}^{\mathrm{T}}$. If we assume that $\boldsymbol{H}^{-1} = (\bar{\eta}_{ij})_{1\le i,j\le M}$ and $\tilde{\boldsymbol{H}}^{-1} = (\bar{\mu}_{ij})_{1\le i,j\le M}$, then this is equivalent to $\bar{\mu}_{1j} + \cdots + \bar{\mu}_{Mj} = 1$ for $j = 1, \dots, M$. Furthermore, since $\boldsymbol{H}^{-1} \ge \boldsymbol{0}$, $\tilde{\boldsymbol{H}}^{-1} \ge \boldsymbol{0}$,

$$\boldsymbol{H} - \tilde{\boldsymbol{H}} = (\boldsymbol{G} - \tilde{\boldsymbol{G}})\boldsymbol{E}(\boldsymbol{z}) = \operatorname*{diag}_{1\le i\le M}\left(\left(G_{ii} - \sum_{j\ne i} G_{ij}\right)L(\boldsymbol{z}_i)\right) \ge 0 \quad \text{for } \boldsymbol{z} \in \mathcal{K}$$

and $\boldsymbol{H}^{-1} = \tilde{\boldsymbol{H}}^{-1} - \boldsymbol{H}^{-1}(\boldsymbol{H} - \tilde{\boldsymbol{H}})\tilde{\boldsymbol{H}}^{-1}$, it follows that $\boldsymbol{H}^{-1} \le \tilde{\boldsymbol{H}}^{-1}$. This yields that

$$\|\boldsymbol{H}^{-1}\|_1 = \max_{1\le j\le M}\sum_{i=1}^{M}\bar{\eta}_{ij} \le \max_{1\le j\le M}\sum_{i=1}^{M}\bar{\mu}_{ij} = 1.$$

Applying Brouwer's fixed point theorem to the continuous function $\boldsymbol{\varphi}\colon \mathcal{K} \to \mathcal{K}$ we deduce the existence of a fixed point of $\boldsymbol{\varphi}$, i.e. a non-negative solution to equation (2.22).

For uniqueness, we adapt an argument that can be found in [49] and define

$$\Psi(\boldsymbol{z}) := \sum_{i=1}^{M} N\big(z_i\big), \quad N(u) := \int_0^{|u|} \Phi(s)\,\mathrm{d}s, \quad \text{and} \quad f(\boldsymbol{z}) := \frac{1}{2}\boldsymbol{z}^{\mathrm{T}}\boldsymbol{G}^{-1}\boldsymbol{z} + \Psi(\boldsymbol{z}) - \boldsymbol{z}^{\mathrm{T}}\boldsymbol{G}^{-1}\boldsymbol{w}.$$

Since $\Phi(0) = 0$, it follows from the definition that $N'(u) = \operatorname{sgn}(u)\Phi(|u|)$ and $N''(u) = \Phi'(|u|)$ for any $u \in \mathbb{R}$, so $N \in C^2(\mathbb{R})$. Therefore, Ψ is twice continuously differentiable. Thus, f is also twice continuously differentiable and its gradient $f'(\boldsymbol{z})$ and Hessian $f''(\boldsymbol{z})$ are given by the respective expressions

$$\begin{aligned} f'(\boldsymbol{z})^{\mathrm{T}} &= \boldsymbol{G}^{-1}\boldsymbol{z} + \big(\operatorname{sgn}(z_1)\Phi(|z_1|), \dots, \operatorname{sgn}(z_M)\Phi(|z_M|)\big)^{\mathrm{T}} - \boldsymbol{G}^{-1}\boldsymbol{w}, \\ f''(\boldsymbol{z}) &= \boldsymbol{G}^{-1} + \operatorname{diag}\big(\Phi'(|z_1|), \dots, \Phi'(|z_M|)\big). \end{aligned}$$

Since $\boldsymbol{G}^{-1}$ is symmetric and positive definite and $\Phi'(|z_i|) \ge 0$, it follows that $f''(\boldsymbol{z})$ is symmetric and positive definite, therefore f is strictly convex, so any critical point (at which $f'(\boldsymbol{z}) = \boldsymbol{0}$) is the unique global minimum. Now, if $\boldsymbol{z} + \boldsymbol{G}\boldsymbol{\Phi}(\boldsymbol{z}) = \boldsymbol{w}$ with $\boldsymbol{z} \ge \boldsymbol{0}$, then $f'(\boldsymbol{z}) = \boldsymbol{0}$ and $\boldsymbol{z} \in \mathcal{K}$, so positive solutions of (2.22) are critical points of f, so uniqueness is proven. □

Corollary 2.3. *If Φ satisfies the conditions (2.4) and*

$$\max_{1\le l\le d}\left\{\frac{\Delta t}{\Delta x_l}\max_{\boldsymbol{k}\in\mathcal{M}}\big|v^l_{\boldsymbol{k}+\frac{1}{2}\boldsymbol{e}_l}\big|\right\} \le \frac{1}{2d}, \tag{2.24}$$

then the Euler IMEX method

$$\boldsymbol{u}^{n+1} = \boldsymbol{u}^n + \Delta t\big(\boldsymbol{\mathcal{C}}(\boldsymbol{u}^n) + \boldsymbol{\mathcal{D}}(\boldsymbol{u}^{n+1})\big) \tag{2.25}$$

is a positivity-preserving scheme.

Proof. To be able to apply Theorem 2.2 to the situation at hand, we must be explicit on how we write the semi-discrete formulation as a system of ordinary differential equations (ODEs). To this end, we assume that $\boldsymbol{u}(t)$ is an M_*-dimensional vector that represents an arrangement of

the M_* unknown functions $u_{\boldsymbol{i}}$, $\boldsymbol{i} \in \mathcal{M}$. Specifically, we fix a bijective map $\nu : \mathcal{M} \ni \boldsymbol{i} \mapsto \nu(\boldsymbol{i}) \in \{1, \dots, M_*\}$, with inverse $\eta : \{1, \dots, M_*\} \ni m \mapsto \eta(m) \in \mathcal{M}$, that maps the d-dimensional index $\boldsymbol{i}$ to the corresponding position within the vector $\boldsymbol{u}(t)$. Now the nonlinear equation (2.21) can be written in the form (2.22) for $M = M_*$ as follows. Suppose that $\boldsymbol{u}^n = (u^n_{\boldsymbol{i}})_{\boldsymbol{i}\in\mathcal{M}}$, then we set

$$\boldsymbol{z} = (z_1, \dots, z_{M_*})^{\mathrm{T}} = \big(u^{n+1}_{\eta(1)}, \dots, u^{n+1}_{\eta(M_*)}\big)^{\mathrm{T}}.$$

Moreover, if we set correspondingly

$$\boldsymbol{\Phi}(\boldsymbol{z}) = \big(\Phi\big(u^{n+1}_{\eta(1)}\big), \dots, \Phi\big(u^{n+1}_{\eta(M_*)}\big)\big)^{\mathrm{T}},$$

then (2.13) stipulates that (2.25) or equivalently,

$$\boldsymbol{u}^{n+1} - \Delta t \boldsymbol{\mathcal{D}}(\boldsymbol{u}^{n+1}) = \boldsymbol{u}^n + \Delta t \boldsymbol{\mathcal{C}}(\boldsymbol{u}^n),$$

can be written as (2.22) if we define $\boldsymbol{G} = (G_{ij})_{1\le i,j\le M_*}$ by

$$G_{ij} := \Delta t \cdot \begin{cases} 2d & \text{if } i = j, \\ -1 & \text{if } i \neq j \text{ and } j = \nu(\eta(i) \pm \boldsymbol{e}_l) \text{ for } l \in \{1, \dots, d\}, \\ 0 & \text{otherwise,} \end{cases}$$

and analogously define the entries of $\boldsymbol{w} = (w_1, \dots, w_{M_*})$ by

$$w_i = u^n_{\eta(i)} + \Delta t C(\boldsymbol{u}^n)_{\eta(i)}, \quad i = 1, \dots, M_*.$$

By Theorem 2.1 with $\Phi = 0$, condition (2.24) guarantees that $w_i \ge 0$ for $i = 1, \dots, M_*$, so the statement of the corollary follows if we apply Theorem 2.2 to the system (2.22) under the present interpretations of $\boldsymbol{z}$, $\boldsymbol{G}$ and $\boldsymbol{w}$. □

The status of Corollary 2.3 is similar to that of Theorem 2.3 in [18]; it cannot be directly applied to higher-order IMEX-RK schemes, since Runge-Kutta implicit schemes in SSP form of order higher than one cannot exist (see [32]). We have nevertheless used Newton-Raphson method, together with a line search algorithm (see [20]) to solve (2.21). At each step of this algorithm a particular sparse system is solved (apart from the diagonal entry in each row, only $2d$ off-diagonal entries are occupied; details in Section 2.2.4). We have not experienced problems in solving these systems whenever a stability restriction as (2.24) was in effect.

2.2.4 Linear solver

The (damped) Newton's method applied to (2.22) or equivalently, to

$$\boldsymbol{F}(\boldsymbol{z}) := \boldsymbol{z} + \boldsymbol{G}\boldsymbol{\Phi}(\boldsymbol{z}) - \boldsymbol{w} = \boldsymbol{0}, \tag{2.26}$$

consists in the iteration

$$\boldsymbol{F}'(\boldsymbol{z}^\nu)\boldsymbol{\delta}^\nu = -\boldsymbol{F}(\boldsymbol{z}^\nu), \quad \boldsymbol{z}^{\nu+1} = \boldsymbol{z}^\nu + \alpha^\nu \boldsymbol{\delta}^\nu, \quad \nu = 0, 1, 2, \dots,$$

where the scalar α^ν is selected using a line-search algorithm that enforces sufficient decrease of the function (see [28])

$$\alpha \mapsto \|\boldsymbol{F}(\boldsymbol{z}^{\boldsymbol{\nu}} + \alpha \boldsymbol{\delta}^\nu)\|_2^2.$$

The structure of the Jacobian matrix associated with (2.26) is particularly simple, namely

$$\boldsymbol{J} := \boldsymbol{F}'(\boldsymbol{z}) := \boldsymbol{I} + \boldsymbol{G}\boldsymbol{D},$$

where $\boldsymbol{D} = \operatorname{diag}(\boldsymbol{\Phi}'(\boldsymbol{z}))$. To get a favorable structure when solving $\boldsymbol{J}\boldsymbol{\delta} = -\boldsymbol{F}(\boldsymbol{z})$, we notice that the columns corresponding to $z_k = 0$ are zero. In particular, the equations for those k are explicit, so the only equations to be solved are those for $z_k \neq 0$. In algebraic terms, we define $M_* := M_1 M_2 \cdots M_d$ and assume that the vector $\boldsymbol{z} \in (\mathbb{R}_0^+)^{M_*}$ is an arrangement of $\{u_{\boldsymbol{i}}\}_{\boldsymbol{i}\in\mathcal{M}}$, taking into account (2.7) and (2.6). For a fixed vector $\boldsymbol{z} = (z_1, \dots, z_{M_*})^{\mathrm{T}}$ of this dimension, we define the index set

$$\mathcal{I} = \mathcal{I}(\boldsymbol{z}) := \{k \mid \Phi'(z_k) > 0\} = \{k_1 < k_2 < \cdots < k_r\} \subseteq \{1, \dots, M_*\}$$

and its complement

$$\mathcal{L} := \{1, \dots, M_*\} \backslash \mathcal{I}(\boldsymbol{z}) = \{k \mid \Phi'(\boldsymbol{z}_k) = 0\} = \{j_1 < j_2 < \cdots < j_{\bar{r}}\},$$

and consider the permutation of $(1, \dots, M_*)$ given by $(k_1, k_2, \dots, k_r, j_1, \dots, j_{\bar{r}})$, with associated permutation matrix $\boldsymbol{P}$. If $\boldsymbol{\mathcal{A}} = (\mathcal{A}_{ij})_{1\le i,j\le M_*}$ is any $M_* \times M_*$ matrix, then

$$\boldsymbol{P}\boldsymbol{\mathcal{A}}\boldsymbol{P}^{\mathrm{T}} = \begin{bmatrix} \boldsymbol{\mathcal{A}}_{\mathcal{I},\mathcal{I}} & \boldsymbol{\mathcal{A}}_{\mathcal{I},\mathcal{L}} \\ \boldsymbol{\mathcal{A}}_{\mathcal{L},\mathcal{I}} & \boldsymbol{\mathcal{A}}_{\mathcal{L},\mathcal{L}} \end{bmatrix}$$

with the submatrices

$$(\boldsymbol{\mathcal{A}}_{\mathcal{I},\mathcal{I}})_{p,q} = \mathcal{A}_{k_p,k_q}, \quad (\boldsymbol{\mathcal{A}}_{\mathcal{I},\mathcal{L}})_{p,m} = \mathcal{A}_{k_p,j_m}, \quad (\boldsymbol{\mathcal{A}}_{\mathcal{L},\mathcal{I}})_{l,q} = \mathcal{A}_{j_l,k_q}, \quad (\boldsymbol{\mathcal{A}}_{\mathcal{L},\mathcal{L}})_{l,m} = \mathcal{A}_{j_l,j_m}.$$

Consequently, the matrix $\boldsymbol{P}\boldsymbol{J}\boldsymbol{P}^{\mathrm{T}}$ has the following block structure, where $\boldsymbol{I}_r$ and $\boldsymbol{I}_{\bar{r}}$ denote the $r \times r$ and $\bar{r} \times \bar{r}$ identity matrix, respectively:

$$\begin{aligned} \boldsymbol{P}\boldsymbol{J}\boldsymbol{P}^{\mathrm{T}} = \boldsymbol{I} + \boldsymbol{P}\boldsymbol{G}\boldsymbol{P}^{\mathrm{T}}\boldsymbol{P}\boldsymbol{D}\boldsymbol{P}^{\mathrm{T}} &= \begin{bmatrix} \boldsymbol{I}_r & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I}_{\bar{r}} \end{bmatrix} + \begin{bmatrix} \boldsymbol{G}_{\mathcal{I},\mathcal{I}} & \boldsymbol{G}_{\mathcal{I},\mathcal{L}} \\ \boldsymbol{G}_{\mathcal{L},\mathcal{I}} & \boldsymbol{G}_{\mathcal{L},\mathcal{L}} \end{bmatrix} \begin{bmatrix} \boldsymbol{D}_{\mathcal{I},\mathcal{I}} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \\ &= \begin{bmatrix} \boldsymbol{I}_r + \boldsymbol{G}_{\mathcal{I},\mathcal{I}}\boldsymbol{D}_{\mathcal{I},\mathcal{I}} & \boldsymbol{0} \\ \boldsymbol{G}_{\mathcal{L},\mathcal{I}}\boldsymbol{D}_{\mathcal{I},\mathcal{I}} & \boldsymbol{I}_{\bar{r}} \end{bmatrix}. \end{aligned}$$

Therefore, the solution of $\boldsymbol{J\delta} = -\boldsymbol{F}(\boldsymbol{z})$ can be obtained by solving $\boldsymbol{PJP}^{\mathrm{T}}\boldsymbol{P\delta} = -\boldsymbol{PF}(\boldsymbol{z})$, that is,

$$\begin{bmatrix} \boldsymbol{I}_r + \boldsymbol{G}_{\mathcal{I},\mathcal{I}}\boldsymbol{D}_{\mathcal{I},\mathcal{I}} & 0 \\ \boldsymbol{G}_{\mathcal{L},\mathcal{I}}\boldsymbol{D}_{\mathcal{I},\mathcal{I}} & \boldsymbol{I}_{\bar{r}} \end{bmatrix} \begin{pmatrix} \boldsymbol{\delta}_{\mathcal{I}} \\ \boldsymbol{\delta}_{\mathcal{L}} \end{pmatrix} = - \begin{pmatrix} \boldsymbol{F}(\boldsymbol{z})_{\mathcal{I}} \\ \boldsymbol{F}(\boldsymbol{z})_{\mathcal{L}} \end{pmatrix},$$

which means that in each iteration, we first determine $\boldsymbol{\delta}_{\mathcal{I}}$ by solving

$$\left(\boldsymbol{I}_r + \boldsymbol{G}_{\mathcal{I},\mathcal{I}}\boldsymbol{D}_{\mathcal{I},\mathcal{I}}\right)\boldsymbol{\delta}_{\mathcal{I}} = -\boldsymbol{F}(\boldsymbol{z})_{\mathcal{I}}, \tag{2.27}$$

and then calculate $\boldsymbol{\delta}_{\mathcal{L}}$ by evaluating

$$\boldsymbol{\delta}_{\mathcal{L}} = -\boldsymbol{F}(\boldsymbol{z})_{\mathcal{L}} - \boldsymbol{G}_{\mathcal{L},\mathcal{I}}\boldsymbol{D}_{\mathcal{I},\mathcal{I}}\boldsymbol{\delta}_{\mathcal{I}}.$$

The matrix of the system (2.27) can be written as $\hat{\boldsymbol{J}} = \boldsymbol{I}_r + \hat{\boldsymbol{G}}\hat{\boldsymbol{D}}$, where $\hat{\boldsymbol{G}}$ and $\hat{\boldsymbol{D}}$ are the corresponding submatrices of $\boldsymbol{G}$ and $\boldsymbol{D}$, respectively. Since the diagonal entries of $\hat{\boldsymbol{D}}$ are positive, $\hat{\boldsymbol{J}}$ can be transformed into a symmetric and positive definite matrix by

$$\hat{\boldsymbol{D}}^{1/2}\hat{\boldsymbol{J}}\hat{\boldsymbol{D}}^{-1/2} = \boldsymbol{I}_r + \hat{\boldsymbol{D}}^{1/2}\hat{\boldsymbol{G}}\hat{\boldsymbol{D}}^{1/2}.$$

Therefore system (2.27) can be solved by solving first

$$\left(\boldsymbol{I}_r + \hat{\boldsymbol{D}}^{1/2}\hat{\boldsymbol{G}}\hat{\boldsymbol{D}}^{1/2}\right)\hat{\boldsymbol{\delta}} = -\hat{\boldsymbol{D}}^{1/2}\boldsymbol{F}(\boldsymbol{z})_{\mathcal{I}} \tag{2.28}$$

and then evaluating

$$\boldsymbol{\delta}_{\mathcal{I}} = \hat{\boldsymbol{D}}^{1/2}\hat{\boldsymbol{\delta}}.$$

The solution of (2.28) can be obtained by applying the conjugate gradient method.

2.3 Numerical examples

2.3.1 IMEX-RK schemes and CFL condition

We solve numerically (2.1), (2.2) for $0 \le t \le T$ and $\boldsymbol{x} \in \Omega$ (see (2.5)). We compare numerical results obtained by the IMEX approach with those obtained by the explicit scheme of [21]. To demonstrate that the IMEX schemes are more efficient than the explicit one independently of the particular choice of the specific IMEX-RK scheme as given by its pair of Butcher arrays (1.25), we utilize and partly compare results produced by three different IMEX-RK schemes, given in (0.5), (0.6) and (0.7) respectively.

For each iteration, the time step $\Delta t = \Delta t_n$ is determined by the formula

$$\Delta t\left(\frac{1}{d}\max_{0\le u\le \eta(\boldsymbol{u})}\Phi'(u)\sum_{q=1}^{d}\frac{1}{\Delta x_q^2} + \max_{1\le l\le d}\left\{\frac{1}{\Delta x_l}\max_{\boldsymbol{k}\in\mathcal{M}}\left|v^{l,n}_{\boldsymbol{k}+\frac{1}{2}\boldsymbol{e}_l}\right|\right\}\right) = C_{\mathrm{cfl}_1}, \quad \eta(\boldsymbol{u}) := \max_{\boldsymbol{j}\in\mathcal{M}} u_{\boldsymbol{j}}, \tag{2.29}$$

for the explicit scheme and by

$$\Delta t \max_{1\le l\le d}\left\{\frac{1}{\Delta x_l}\max_{\boldsymbol{k}\in\mathcal{M}}\left|v^{l,n}_{\boldsymbol{k}+\frac{1}{2}\boldsymbol{e}_l}\right|\right\} = C_{\mathrm{cfl}_2} \tag{2.30}$$

for the IMEX-RK scheme. (Note that the left-hand sides of (2.29) and (2.30) are identical to those of (2.14) and (2.24), respectively, for $u_{\boldsymbol{i}} = u^n_{\boldsymbol{i}}$ for all $\boldsymbol{i} \in \mathcal{M}$.) In the numerical examples and for each time discretization we choose C_{cfl_1} and C_{cfl_2} as the largest multiple of 0.05 that yields oscillation-free solutions. This strategy leads to $C_{\mathrm{cfl}_1} = 0.25$ for the explicit scheme, $C_{\mathrm{cfl}_2} = 0.25$ for the H-CN(2,2,2) and IMEX-SSP3(4,3,3) schemes, and $C_{\mathrm{cfl}_1} = 0.2$ for the IMEX-SSP2(3,3,2) scheme, respectively.

2.3.2 Approximate numerical error

The approximate numerical error is measured for four of the five numerical examples, all of which are defined on a square domain with $M_1 = M_2 =: M$. In all these cases we compute a reference solution with $M = M_{\mathrm{ref}}$ utilizing the IMEX-RK H-CN(2,2,2) scheme. In each case the reference solution allows us to compute approximate L^1 errors at different times as follows. We have $\mathcal{M} = \{1,\dots,M\}^2$, define $\mathcal{M}_{\mathrm{ref}} := \{1,\dots,M_{\mathrm{ref}}\}^2$, and denote by

$$\left(u^M_{i,j}(t)\right)_{(i,j)\in\mathcal{M}} \quad \text{and} \quad \left(u^{M_{\mathrm{ref}}}_{i,j}(t)\right)_{(i,j)\in\mathcal{M}_{\mathrm{ref}}}$$

the numerical solution at time t calculated with M^2 and M^2_{ref} cells, respectively. We assume that $R := M_{\mathrm{ref}}/M$ is an integer and compute the projection of the reference solution

$$\tilde{u}^{\mathrm{ref},M}_{j,k}(t) = \frac{1}{R^2}\sum_{p,q=1}^{R} u^{M_{\mathrm{ref}}}_{R(j-1)+p,R(k-1)+q}(t).$$

The approximate L^1 error $e_M(t)$ associated with the numerical solution on the mesh with M^2 cells at time t is given by

$$e_M(t) = \frac{1}{M^2}\sum_{j,k=1}^{M}\left|\tilde{u}^{\mathrm{ref},M}_{j,k}(t) - u^M_{j,k}(t)\right|.$$

A numerical order of convergence can be calculated from pairs $e_{M/2}(t)$ and $e_M(t)$ by

$$\vartheta_M(t) := \log_2\left(e_M(t)/e_{M/2}(t)\right).$$

	IMEX-RK H-CN(2,2,2)			Explicit			IMEX-RK H-CN(2,2,2)			Explicit		
M	e_M	ϑ_M	cpu [s]	e_M	ϑ_M	cpu [s]	e_M	ϑ_M	cpu [s]	e_M	ϑ_M	cpu [s]
	$T = 0.5$						$T = 1$					
40	6923	—	0.50	6183	—	0.44	14391	—	0.69	13817	—	0.95
80	3437	1.01	0.73	3104	0.99	3.26	7089	1.02	1.80	6973	0.99	7.70
160	1595	1.11	5.50	1443	1.11	35.5	3355	1.08	12.0	3292	1.08	80.5
320	709.9	1.17	53.3	651.1	1.15	577	1507	1.15	103	1480	1.15	1267
640	290.3	1.29	466	270.8	1.27	10010	636.7	1.24	985	625.8	1.24	21849
	$T = 4.5$						$T = 5$					
40	35067	0.60	2.30	41592	0.72	12.2	40057	—	2.58	47500	—	14.7
80	23158	0.88	14.3	25320	0.95	135	24684	0.70	16.5	26910	0.82	164
160	12619	1.02	87.1	13137	1.05	1562	12629	0.97	100	13225	1.02	1945
320	6226	1.18	716	6340	1.20	21548	6032	1.07	821	6171	1.10	26936
640	2739	0.00	8065	2766	0.00	338985	2614	1.21	8982	2649	1.22	422759
	$T = 6.5$						$T = 7$					
40	269041	—	3.80	275375	—	25.9	172173	—	4.70	183920	0.00	36.2
80	165465	0.70	24.2	177743	0.63	252	178006	-0.05	31.3	183770	0.15	362
160	63488	1.38	141	65775	1.43	3050	163004	0.13	170	165729	0.93	3771
320	26180	1.28	1159	26637	1.30	42585	85463	0.93	1374	86850	1.31	48479
640	10577	1.31	12377	10691	1.32	665871	34566	1.31	14069	34931	0.00	741821

Table 2.1: Example 2.1: approximate L^1 errors (e_M, figures to be multiplied by 10^{-6}), convergence rates (ϑ_M), and CPU times (cpu).

2.3.3 Numerical examples

Example 2.1

Following a numerical experiment proposed in [21], we solve (2.1) for

$$u_0(\boldsymbol{x}) = 0.25\chi_{[-3,3]\times[-3,3]}(\boldsymbol{x}), \quad W(\boldsymbol{x}) = -\frac{1}{\pi}\exp\bigl(-|\boldsymbol{x}|^2\bigr), \quad V \equiv 0,$$
$$H(u) = \frac{\nu}{m}u^m \Rightarrow \Phi(u) = \frac{\nu(m-1)}{m}u^m, \tag{2.31}$$

where χ_A is the characteristic function of a set A and $\Phi(u)$ is defined in (2.3). In this example we choose $\nu = 0.1$ and $m = 2.1$, and limit ourselves to the IMEX-RK scheme H-CN(2,2,2) given by (0.5). The numerical results for various values of M are displayed in Figure 2.1. The approximate errors, convergence rates, and CPU times are provided in Table 2.1, where we compare the error of approximation with respect to a reference solution with $M_{\text{ref}} = 2560$ cells per direction. Figure 2.2 contains the efficiency plots for the end times T for which the errors are measured. According to Table 2.1, for $T \geq 4.5$ the errors and CPU times produced by

	IMEX-RK H-CN(2,2,2)			IMEX-RK IMEX-SSP2(3,3,2)			IMEX-RK IMEX-SSP3(4,3,3)			Explicit		
M	e_M	ϑ_M	cpu [s]	e_M	ϑ_M	cpu [s]	e_M	ϑ_M	cpu [s]	e_M	ϑ_M	cpu [s]
	$T = 0.01$											
75	27430	—	1.84	28858	—	0.87	30323	—	1.96	29394	—	1.49
150	19532	0.49	1.78	14933	0.95	1.57	10070	1.59	2.90	17487	0.75	14.9
300	34651	-0.83	17.6	26977	-0.85	28.6	15781	-0.65	21.7	19571	-0.16	185
600	49383	-0.51	381	12475	1.11	484	23922	-0.60	386	22320	-0.19	2626
	$T = 0.7$											
75	10480	—	12.5	10496	—	8.64	10545	—	15.5	11106	—	20.6
150	6787	0.63	56.8	6818	0.62	57.3	6833	0.63	72.8	7052	0.66	265
300	3188	1.09	594	3207	1.09	503	3218	1.09	629	3294	1.10	3647
600	1096	1.54	4878	1105	1.54	7367	1112	1.53	4822	1138	1.53	47957
	$T = 2.8$											
75	14212	—	39.4	14217	—	28.6	14231	—	52.3	14481	—	40.1
150	9044	0.65	182	9052	0.65	197	9057	0.65	236	9165	0.66	523
300	4244	1.09	1675	4249	1.09	1457	4252	1.09	1839	4296	1.09	7359
600	1461	1.54	11360	1463	1.54	15493	1465	1.54	11235	1482	1.54	123643
	$T = 11.2$											
75	17527	—	134	17528	—	128	17532	—	182	17631	—	72.4
150	11171	0.65	749	11173	0.65	853	11174	0.65	1034	11223	0.65	946
300	5264	1.09	4840	5265	1.09	5565	5266	1.09	5915	5288	1.09	13319
600	1817	1.53	27290	1818	1.53	44158	1818	1.53	31011	1827	1.53	288459

Table 2.2: Example 2.2: approximate L^1 errors (e_M, figures to be multiplied by 10^{-6}), convergence rates (ϑ_M) and CPU times (cpu).

IMEX-RK scheme are smaller than for the explicit version, and Figure 2.2 indicates that for these simulated times the IMEX-RK scheme is more efficient than the explicit version.

Example 2.2

We consider the numerical example in two dimensions proposed in [55]. Here u represents the population density and $W * u$ is the velocity. Specifically, we choose the functions

$$W(\boldsymbol{x}) = -\frac{1}{2\pi}\exp(-|\boldsymbol{x}|), \quad V \equiv 0,$$

and $\Phi(u)$ given by (2.31) with $\nu = 1/2$ and $m = 3$. The initial datum u_0 consists of two disjoint discs of radius 5 and centered at $(7, 0)$ and $(-7, 0)$, both with randomly distributed population with values between 0 and 1 such that the total population size is given by $\int_{\mathbb{R}^2} u_0(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} = 600$. The initial condition is defined for a 75×75 discretization, which is also utilized for finer

	IMEX-RK H-CN(2,2,2)			IMEX-RK IMEX-SSP2(3,3,2)			IMEX-RK IMEX-SSP3(4,3,3)			Explicit		
M	e_M	ϑ_M	cpu [s]	e_M	ϑ_M	cpu [s]	e_M	ϑ_M	cpu [s]	e_M	ϑ_M	cpu [s]
	$T = 10$											
40	4276	—	1.23	4294	—	4.62	4264	—	7.37	6528	—	11.0
80	3428	0.32	5.91	3417	0.33	42.7	3419	0.32	84.5	4110	0.67	118
160	2142	0.68	44.5	2140	0.68	310	2140	0.68	547	2262	0.86	1423
320	1042	1.04	301	1041	1.04	2797	1042	1.04	4325	1071	1.08	19170
	$T = 20$											
40	14161	—	2.60	14433	—	10.7	14202	—	16.9	20045	—	28.9
80	5298	1.42	15.2	5566	1.37	105	5352	1.41	213	4128	2.28	349
160	960	2.46	118	959	2.54	788	961	2.48	1299	1130	1.87	4738
320	580	0.73	819	571	0.75	7236	580	0.73	11774	735	0.62	67428
	$T = 40$											
40	2109	—	6.42	2139	—	26.6	2216	—	40.4	24943	—	64.4
80	3031	-0.52	38.0	3130	-0.55	273	3071	-0.47	578	2226	3.49	1148
160	1228	1.30	277	1235	1.34	1896	1233	1.32	3052	1170	0.93	15782
320	496	1.31	2000	497	1.31	16784	498	1.31	27367	506	1.21	228229
	$T = 80$											
40	2109	—	15.0	2076	—	63.8	2133	—	96.2	32239	—	135.3
80	1912	0.14	86.0	1919	0.11	626	1988	0.10	1306	2562	3.65	2892
160	1152	0.73	593	1154	0.73	4061	1163	0.77	6546	1154	1.15	39366
320	567	1.02	4335	567	1.03	35241	568	1.03	45427	537	1.10	571621

Table 2.3: Example 2.3: approximate L^1 errors (e_M, figures to be multiplied by 10^{-6}), convergence rates (ϑ_M), and CPU times (cpu).

discretizations with $M = 150, 300$ and 600, so the initial condition is exactly the same in all cases. The numerical solution for three different discretizations at four different times is displayed in Figure 2.3. The approximate errors, convergence rates, and CPU times for these times are provided in Table 2.2. The reference solution is computed with $M_{\text{ref}} = 2400$ cells per dimension. Figure 2.4 contains the efficiency plots for three different end times. We observe that for a given discretization M and with the exception of simulated time $T = 0.01$, the errors produced by the IMEX-RK schemes are smaller than those of the explicit scheme. Also, with the exception of some of the cases of $M = 75$, the CPU times for the IMEX-RK schemes are substantially smaller than for the explicit scheme. Thus, the IMEX-RK schemes are most efficient in all instances.

Example 2.3

This example is a 2D version of [21, Example 2]. More precisely, we utilize

$$u_0(\boldsymbol{x}) = 0.05\chi_{[-3,3]\times[-3,3]}(\boldsymbol{x}), \quad W(\boldsymbol{x}) = -\left(1 - |\boldsymbol{x}|\right)_+, \quad V \equiv 0,$$

and $\Phi(u)$ given by (2.31) with $\nu = 1.48$ and $m = 3$. The numerical solution for three different discretizations at four different times is displayed in Figure 2.5. The approximate errors, convergence rates, and CPU times for these times are provided in Table 2.3, where we compare the error of approximation with respect to the reference solution with $M_{\text{ref}} = 1280$ cells per dimension, and Figure 2.6 contains the efficiency plots for three different end times. For this example, the IMEX-RK schemes produce slightly smaller errors but are faster than the explicit scheme, and therefore turns out significantly more efficient.

Example 2.4

In this example we utilize the functions

$$W(\boldsymbol{x}) = \frac{1}{2\pi}\big(\exp(-|\boldsymbol{x} - \boldsymbol{x}_1|) + \exp(-|\boldsymbol{x} - \boldsymbol{x}_2|)\big),$$

where $\boldsymbol{x}_1 = (7, 0)$ and $\boldsymbol{x}_2 = (-7, 0)$, $V \equiv 0$, and $\Phi(u)$ given by (2.31) with $\nu = 1$ and $m = 4$. The initial condition u_0 is a random function with values between 0 and 1 distributed over the (x_1, x_2)-square $[-30, 30] \times [-30, 30]$. The initial condition is defined for a discretization with $M = 40$, which is also used for finer discretizations as in Example 2. Numerical solutions for three different discretizations at four different simulations times are displayed in Figure 2.7. For this example the initial solution evolves until a steady state that consists of vertical stripes. The numerical solutions for the different discretizations converge to the same steady state solution. This can be observed in Table 2.4, where we compare the error of approximation with respect the reference solution which is computed with $M_{\text{ref}} = 1280$. Figure 2.8 contains the efficiency plots for three different end times. Table 2.4 indicates that the errors produced by IMEX-RK schemes are slightly smaller than those of the explicit scheme, but again the IMEX-RK schemes use less CPU time than the explicit scheme, and we conclude that the IMEX-RK schemes turn out more efficient than the explicit version.

It is instructive to compare the gain in efficiency of this example with that of Example 2. In view of the CFL conditions (2.29) and (2.30), the gain in efficiency by IMEX-RK schemes with respect to their explicit counterpart is likely to appear earlier (when discretization is successively refined) whenever the diffusion term is dominant, that is $\max_{0\le u\le \eta(\boldsymbol{u}^n)} \Phi'(u)$ (arising in (2.29)) is large in comparison with the maximum on the convective velocities (the term that arises in both (2.29) and (2.30)). In Example 2 we have $\Phi'(u) = u^2$ and in Example 4 there holds $\Phi'(u) = u^3$, with u-values ranging between 0 and 1 in Example 2 (see Figure 2.3) but only between 0 and 0.6 for $M = 160$ and $M = 320$ in Example 4 (see Figure 2.7). Since the dimensions of the domain of Examples 2 and 4 are the same and the interaction potential W

produces similar values in both cases, one can roughly say that Example 2 is more diffusion dominant than Example 4, which explains why the gain in efficiency is better visible for the discretizations considered in the plots of Figure 2.4 (for Example 2) than for those of Figure 2.8.

Example 2.5

Finally, we present one additional example without error analysis but to demonstrate that (2.1), (2.2), and numerical methods developed for the approximation of its solutions, capture a model of swarming with diffusion [52]. In this context (2.1) represents the Fokker-Planck equation of the space-homogeneous version of a swarming model whose solution $u = u(\boldsymbol{x}, t)$ is the density distribution of individuals having velocity $\boldsymbol{x} \in \mathbb{R}^d$ at time $t > 0$. For our example, the functions

$$W(\boldsymbol{x}) = \frac{1}{|\boldsymbol{x}|}, \quad V(\boldsymbol{x}) = \alpha \left(\frac{|\boldsymbol{x}|^4}{4} - \frac{|\boldsymbol{x}|^2}{2} \right), \quad \text{and } \Phi(u) = \nu u,$$

the parameters $\alpha = 2$ or $\alpha = 4$ and $\nu = 0.3$, 0.1 and 0.5, and the initial condition is given by

$$u_0(\boldsymbol{x}) = \frac{1}{\pi} \exp(-|\boldsymbol{x} - \boldsymbol{x}_3|^2), \quad \text{where } \boldsymbol{x}_3 = (2, 2), \tag{2.32}$$

are chosen precisely as in [52, Example 3]. We obtained numerical solutions for $M_1 = M_2 = 80$ cells at four different simulated times shown in Figure 2.9. The pairs $(\alpha, \nu) = (2, 0.3)$, $(4, 0.1)$, and $(4, 0.5)$, corresponding to the top, middle, and bottom rows of Figure 2.9, respectively, correspond to the scenarios for which the corresponding steady-state solution is shown in plot (e), (g), and (i) of [52, Figure 5], respectively. It is worth noting that only in the case $(\alpha, \nu) = (4, 0.5)$, due to the relative high value of ν (cf. [52, Th. 4]), the steady-state solution will be radially symmetric with mean $(0, 0)$ so the swarm will not propel into any preferential direction while in the two other cases that mean will have two equal positive components, as is stipulated by the initial condition (2.32).

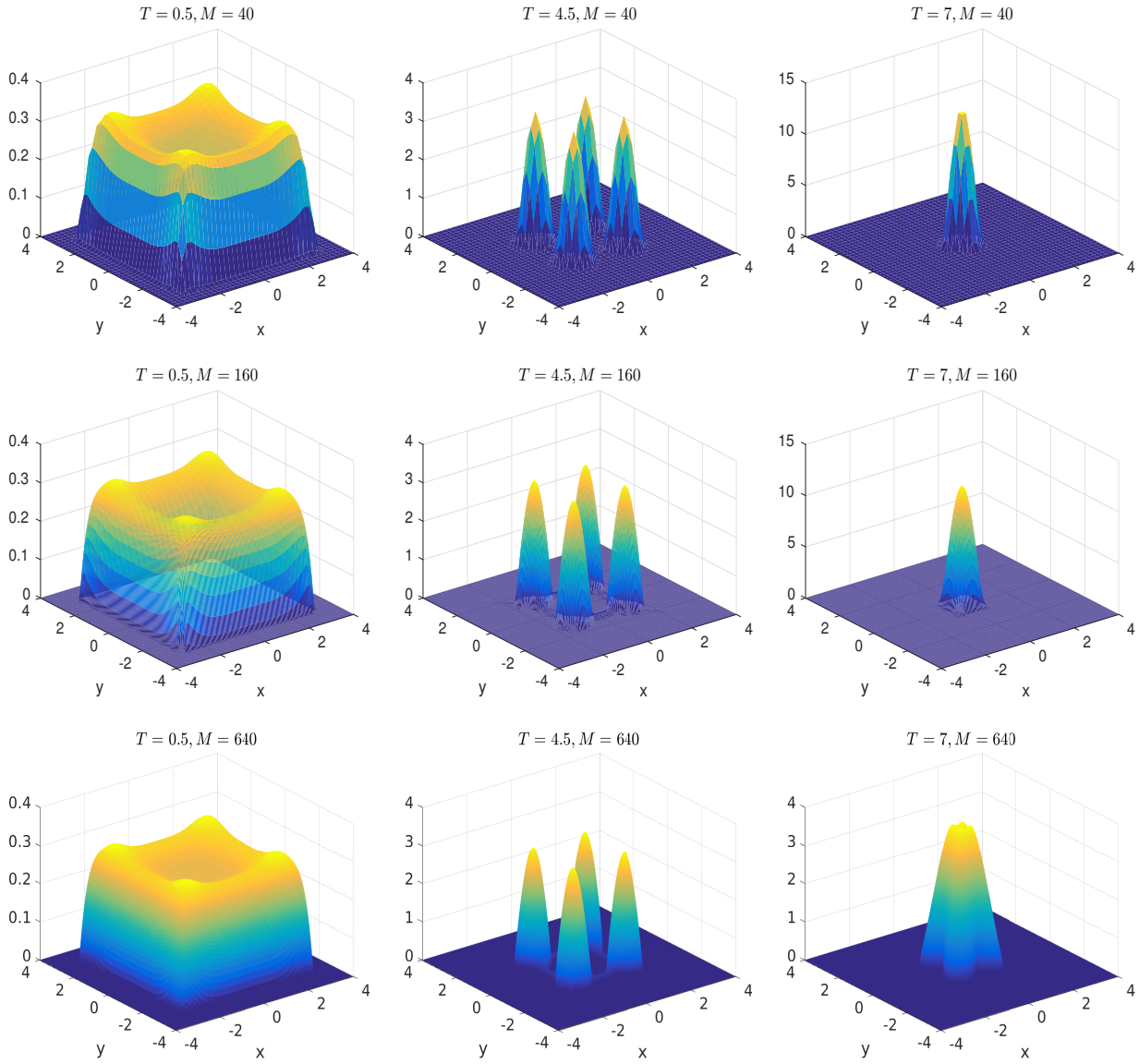

Figure 2.1: Example 2.1: numerical solutions with $\Delta x = 2L/M$ and $L = 4$ for (top) $M = 40$, (middle) $M = 160$, and (bottom) $M = 640$, at simulated times $T = 0.5$, 4.5, and 7. The IMEX-RK scheme used is H-CN(2,2,2) given by (0.5) (figure produced by author).

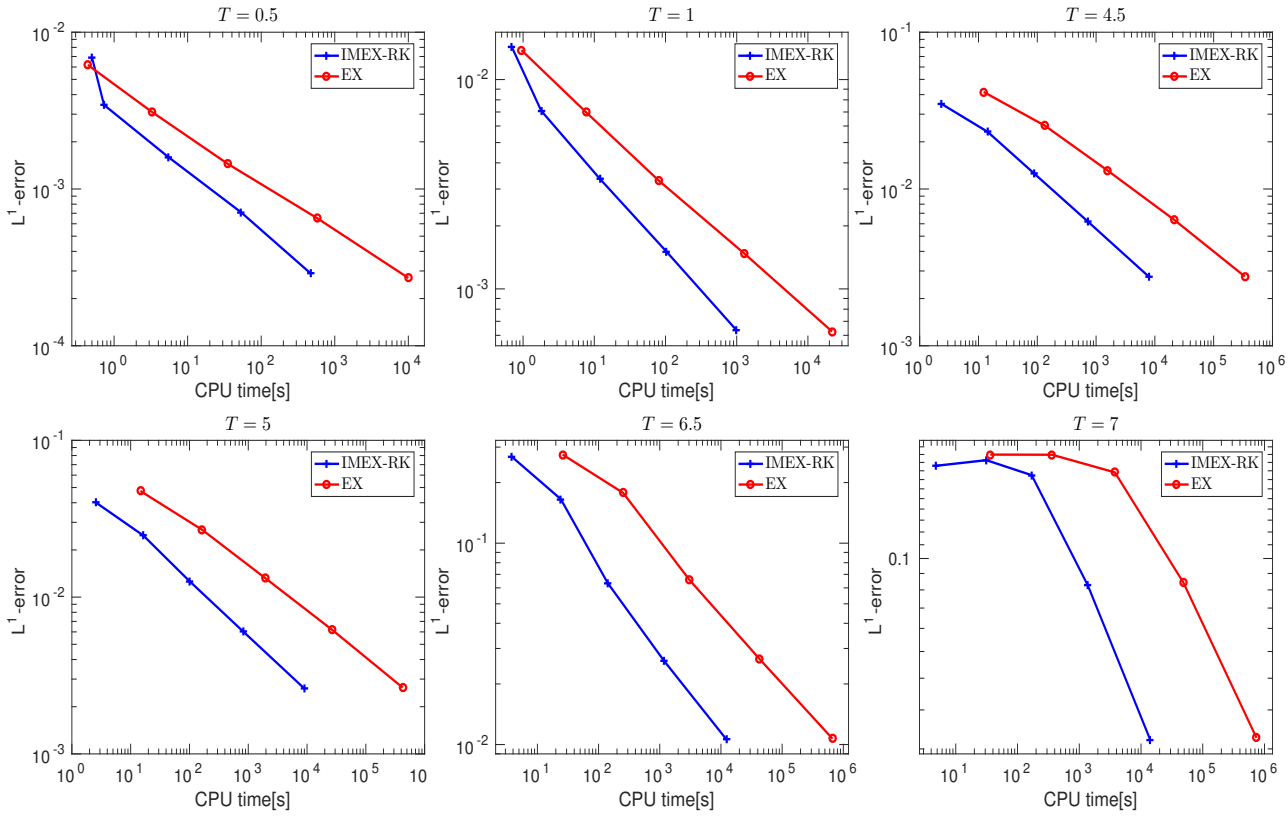

Figure 2.2: Example 2.1: efficiency plots corresponding to six simulated times. The IMEX-RK scheme employed is the scheme H-CN(2,2,2) given by (0.5) (figure produced by author).

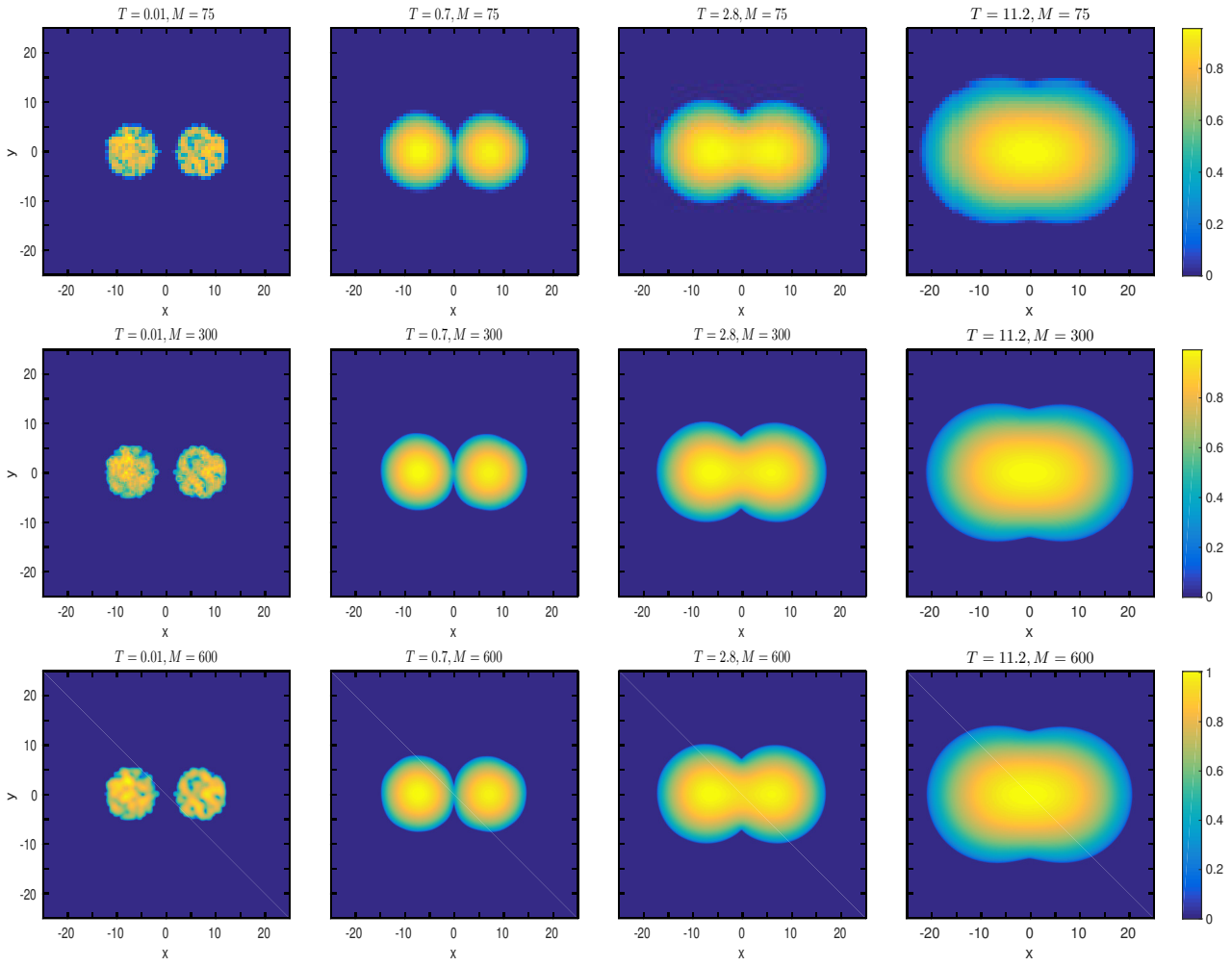

Figure 2.3: Example 2.2: numerical solutions with $\Delta x = 2L/M$ and $L = 20$ for (top) $M = 75$, (middle) $M = 300$ and (bottom) $M = 600$, at simulated times $T = 0.01$, 0.7, 2.8, and 11.2. The IMEX-RK scheme is H-CN(2,2,2) given by (0.5) (figure produced by author).

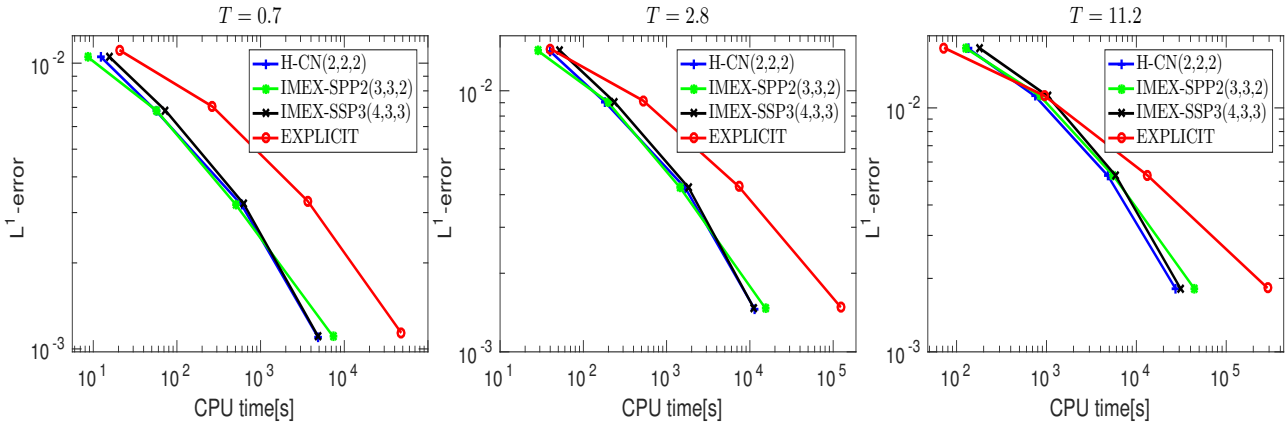

Figure 2.4: Example 2.2: efficiency plots based on numerical solutions for $M = 75, 150, 300, 600$ (figure produced by author).

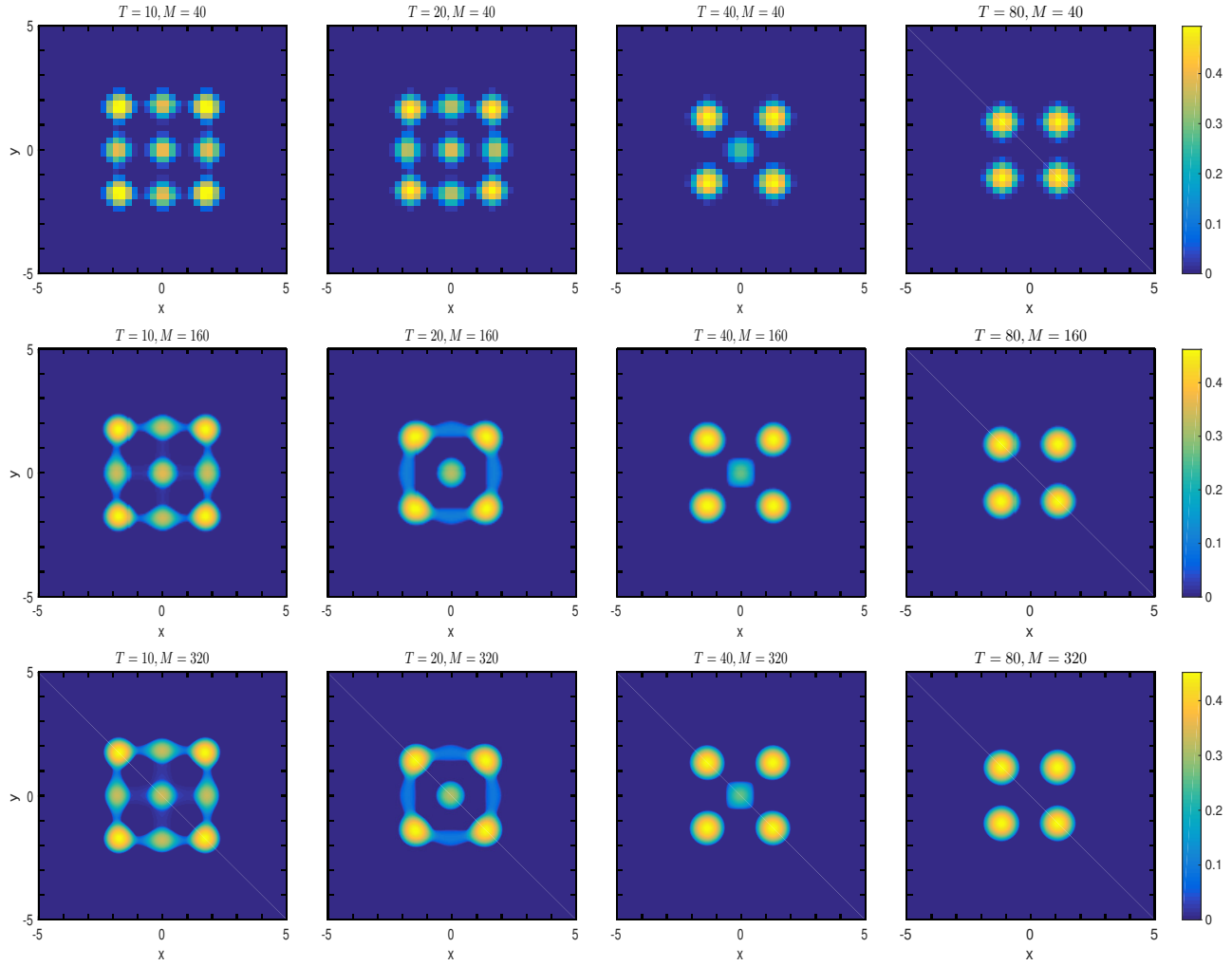

Figure 2.5: Example 2.3: numerical solutions with $\Delta x = 2L/M$ and $L = 5$ for (top) $M = 40$, (middle) $M = 160$ and (bottom) $M = 320$, at simulated times $T = 10$, 20, 40, and 80. The IMEX-RK scheme is given by (0.5) (figure produced by author).

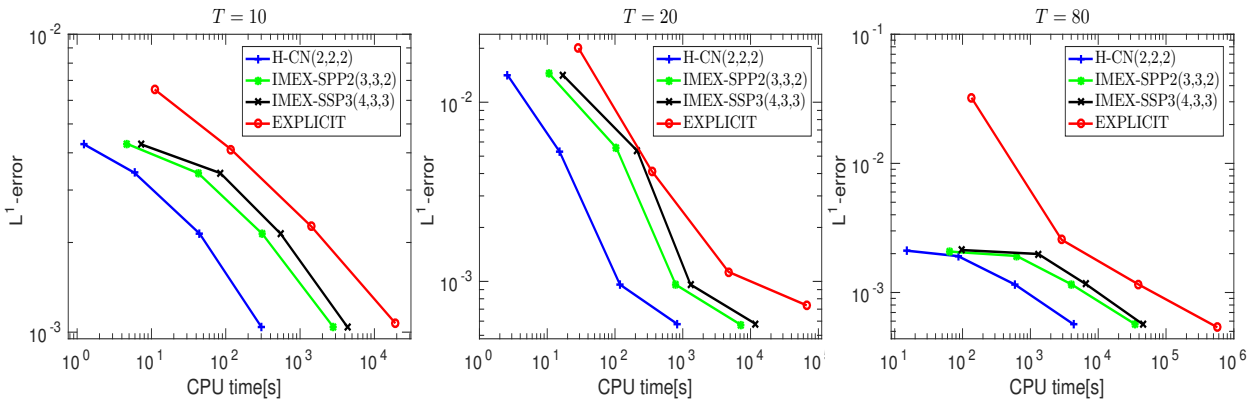

Figure 2.6: Example 2.3: efficiency plots based on numerical solutions for $\Delta x = 2L/M$ with $M = 40, 80, 160, 320$ (figure produced by author).

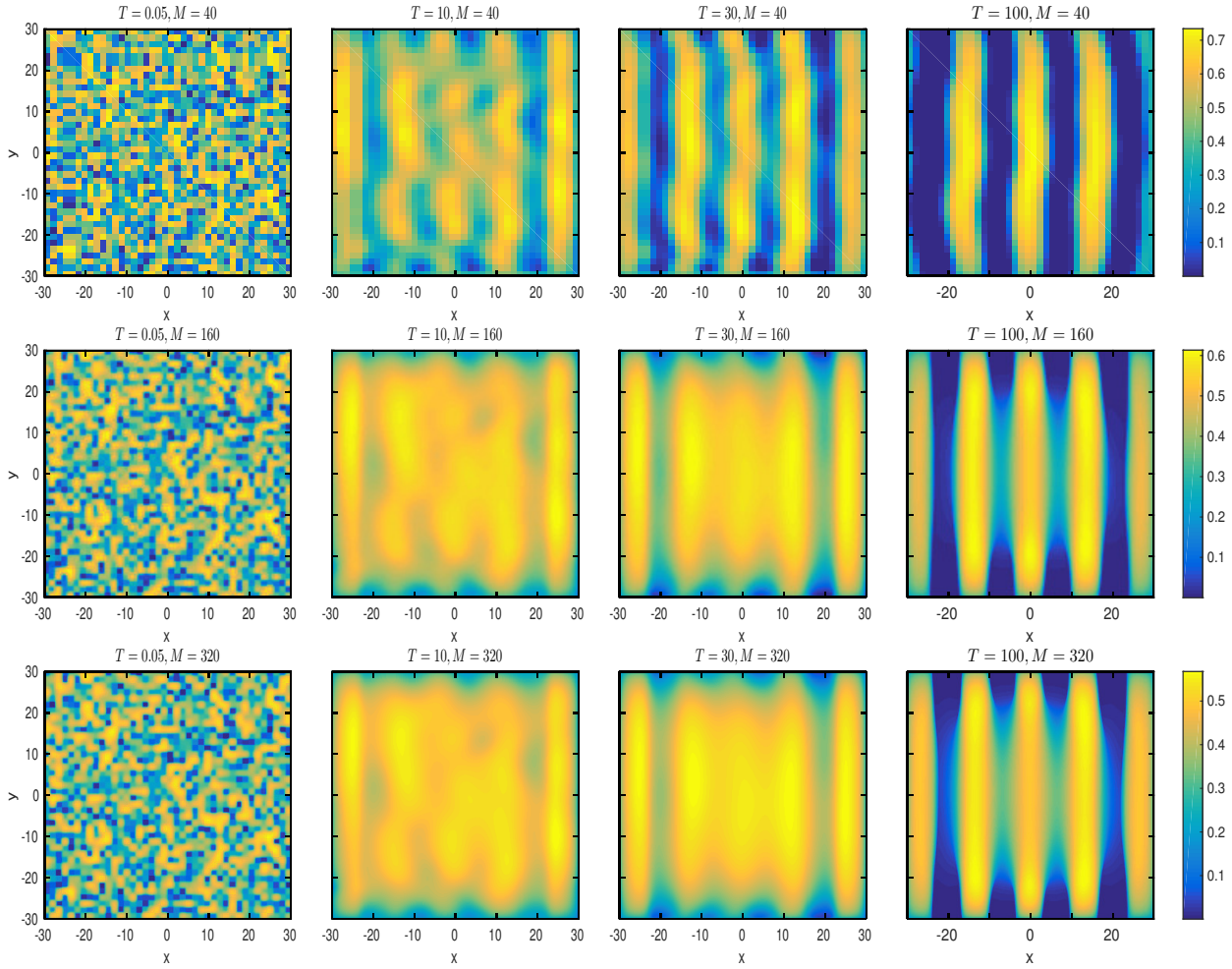

Figure 2.7: Example 2.4: numerical solutions with $\Delta x = 2L/M$ and $L = 30$ for (top) $M = 40$, (middle) $M = 160$ and (bottom) $M = 320$, at simulated times $T = 0.05$, 10, 30, and 100. The IMEX-RK scheme employed is IMEX-SSP2(3,3,2) given by (0.6) (figure produced by author).

	IMEX-RK H-CN(2,2,2)			IMEX-RK IMEX-SSP2(3,3,2)			IMEX-RK IMEX-SSP3(4,3,3)			Explicit		
M	e_M	ϑ_M	cpu [s]	e_M	ϑ_M	cpu [s]	e_M	ϑ_M	cpu [s]	e_M	ϑ_M	cpu [s]
	$T = 0.05$											
40	55627	—	1.05	55685	—	0.55	55661	—	0.54	51846	—	0.19
80	34136	0.70	0.82	34446	0.69	0.83	34462	0.69	0.72	33340	0.64	0.87
160	17462	0.97	14.3	15969	1.11	8.32	13870	1.31	6.63	12353	1.43	11.8
320	9056	0.95	83.6	3392	2.24	104	3664	1.92	75.3	2921	2.08	190
	$T = 10$											
40	55446	—	2.44	55572	—	1.97	55472	—	1.55	58363	—	3.47
80	19039	1.54	30.7	19020	1.55	27.4	19017	1.54	19.7	19283	1.60	41.3
160	7408	1.36	614	7406	1.36	466	7406	1.36	337	7539	1.35	595
320	2384	1.64	4511	2383	1.64	4974	2383	1.64	3625	2477	1.61	9817
	$T = 30$											
40	168099	—	5.23	170601	—	4.83	170061	—	3.44	184135	—	10.7
80	41992	2.00	72.0	42051	2.02	63.7	42052	2.02	45.6	42685	2.11	102
160	13931	1.59	1455	13930	1.59	968	13932	1.59	797	13983	1.61	1380
320	4191	1.73	10804	4191	1.73	11134	4191	1.73	7890	4231	1.72	20746
	$T = 60$											
40	222432	—	10.5	224421	—	10.2	224201	—	6.99	250099	—	27.7
80	101733	1.13	132	102380	1.13	118	102265	1.13	90.2	104855	1.25	209
160	32695	1.64	2451	32743	1.64	1578	32736	1.64	1335	32847	1.67	2427
320	9388	1.80	18426	9391	1.80	18536	9390	1.80	13027	9389	1.81	34878
	$T = 100$											
40	213524	—	16.9	214386	—	16.7	214315	—	11.3	220105	—	49.2
80	135826	0.65	233	136635	0.65	211	136502	0.65	164	142007	0.63	407
160	56102	1.28	3790	56290	1.28	2612	56259	1.28	2133	57156	1.31	4359
320	17536	1.68	26139	17569	1.68	27088	17563	1.68	19782	17668	1.69	55105

Table 2.4: Example 2.4: approximate L^1 errors (e_M, figures to be multiplied by 10^{-6}), convergence rates (ϑ_M), and CPU times (cpu).

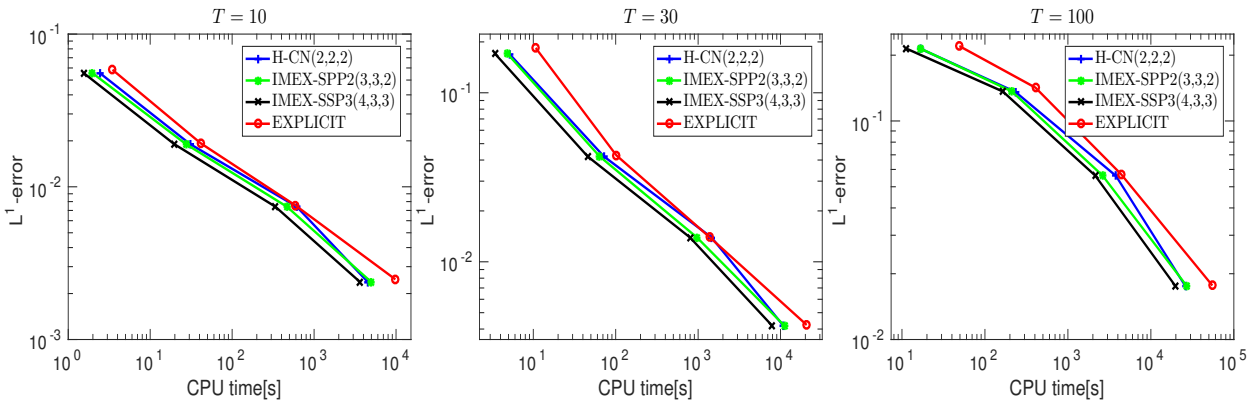

Figure 2.8: Example 2.4: efficiency plots based on numerical solution for $\Delta x = 2L/M$ with $M = 40, 80, 160, 320$ (figure produced by author).

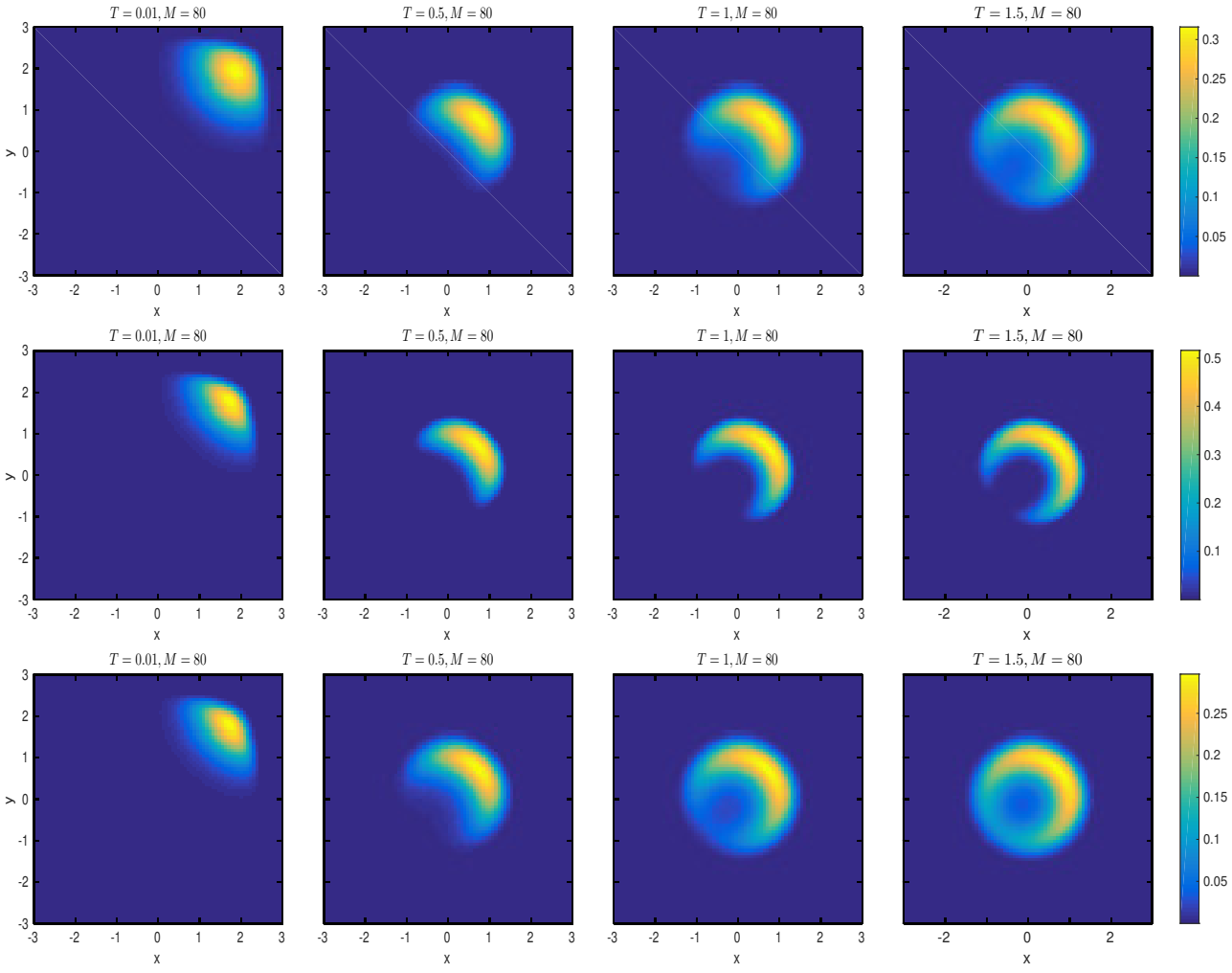

Figure 2.9: Example 2.5: numerical solutions with $\Delta x = 2L/M$ and $L = 3$ for $M = 80$ at simulated times $T = 0.01$, 0.5, 1, and 1.5 produced by the H-CN(2,2,2) scheme for (top) $\alpha = 2$, $\nu = 0.3$, (middle) $\alpha = 4$, $\nu = 0.1$, and (bottom) $\alpha = 4$, $\nu = 0.5$ (figure produced by author).

CHAPTER 3

High-order finite-difference WENO schemes for models of crowd dynamics

3.1 Introduction

3.1.1 Scope

We are concerned with the numerical approximation of a class of nonlocal systems of conservation laws in two space dimension for the macroscopic modelling of pedestrian flow. The pedestrian position are described through a time t and space $\boldsymbol{x}$ dependent density function $\rho = \rho(\boldsymbol{x}, t)$.

We subdivide the crowd moving in $\Omega \subset \mathbb{R}^2$ into N populations differing in their destinations or behaviors. The average density of the k-th population at time $t > 0$ and the position $\boldsymbol{x} \in \Omega$ is $\rho^k(\boldsymbol{x}, t)$. The crowd movement is described by the following system of non-local conservation laws in two space dimensions:

$$\partial_t \rho^k + \operatorname{div}_{\boldsymbol{x}} \boldsymbol{f}^k(t, \boldsymbol{x}, \rho^k, \eta * \boldsymbol{\rho}) = 0, \qquad k = 1, \dots, N \tag{3.1}$$

where the flux function $\boldsymbol{f} := (f^1, \dots, f^N)$ depends on t, $\boldsymbol{x}$ and also on the overall crowd distribution $\boldsymbol{\rho} := (\rho^1, \dots, \rho^N)$, and the coupling is due to the nonlocal terms $\eta * \boldsymbol{\rho}$ which represents the usual convolution product.

3.2 Numerical Method

We consider the system (3.1) in a rectangular domain $\Omega = [x_{\min}, x_{\max}] \times [y_{\min}, y_{\max}]$, we introduce a Cartesian grid with nodes (x_i, y_j), $i = 1, \dots, M_x$ and $j = 1, \dots, M_y$ such that $x_i = (i-1/2)h_x$, $y_j = (j-1/2)h_y$, $h_x = (x_{\max}-x_{\min})/M_x$ and $h_y = (y_{\max}-y_{\min})/M_y$. We then advance a semi-discretization scheme in which spatial derivatives in $\operatorname{div}_{\mathbf{x}} \mathbf{f}$ are discretized first, resulting

in a system of ordinary differential equations whose numerical solution is iteratively updated in time. First, we define a matrix $\boldsymbol{u}(t)$ of unknowns approximations $\boldsymbol{u}_{k,i,j}(t) \approx \rho^k(t, x_i, y_j)$. To solve (3.1), we utilize the Shu-Osher finite difference scheme with upwind spatial reconstructions of the flux function. Then WENO reconstruction [36] of order $2r+1$ are considered. To specify the time discretization, we write the semi-discrete scheme as

$$\boldsymbol{u}' = \mathcal{L}(\boldsymbol{u}) \tag{3.2}$$

where $\mathcal{L}(\boldsymbol{u})_{k,i,j} \approx -\operatorname{div}_{\mathbf{x}} \boldsymbol{f}^k(t, x_i, y_j, \boldsymbol{\rho}^k(t, x_i, y_j), (\eta * \boldsymbol{\rho})(t, x_i, y_j))$. To discretize this divergence for the approximation $\boldsymbol{u}$, we first approximate the terms corresponding to the convolution $\eta * \boldsymbol{\rho}$ using the technique in Chapter 1 expounded, then the discrete convolution obtained is performed by FFTs. Thus, the discretization obtained is

$$\tilde{\boldsymbol{f}}^k(\boldsymbol{u})_{i,j} = \boldsymbol{f}^k(t, x_i, y_j, \boldsymbol{u}_{k,i,j}, (\eta *_h \boldsymbol{u})(t, x_i, y_j)).$$

We introduce the following notation

$$\left(f^x_{i,j}, f^y_{i,j}\right) := \tilde{\boldsymbol{f}}(\boldsymbol{u})_{i,j},$$

where we have dropped the k index for obtaining clearer expressions.
Our purpose is to use a r order, $r = 3, 5, 7$, WENO finite difference discretization [36,40,53,54] of $\operatorname{div}_{\boldsymbol{x}} \boldsymbol{f}$ for which

$$\operatorname{div}_{\boldsymbol{x}} \boldsymbol{f}(x_i, y_j) \approx \mathcal{L}(\boldsymbol{u})_{i,j} := \frac{\hat{f}^x_{i+1/2,j} - \hat{f}^x_{i-1/2,j}}{h_x} + \frac{\hat{f}^y_{i,j+1/2} - \hat{f}^y_{i,j-1/2}}{h_y}, \tag{3.3}$$

for suitable numerical fluxes $\hat{f}^x_{i+1/2,j}, \hat{f}^y_{i,j+1/2}$ obtained by WENO reconstructions of split fluxes. For the numerical flux in the x-direction, the Lax-Friedrichs-type flux splitting $f^{x,\pm}$ is given by

$$f^{x,\pm}_{i,j} = \frac{1}{2}\left(f^x_{i,j} \pm \alpha^x \boldsymbol{u}_{k,i,j}\right), \quad \alpha^x = \|\partial_{\boldsymbol{\rho}} f^x\|_{\mathbf{L}^\infty}.$$

Likewise, the numerical flux $\hat{f}^y_{i,j+1/2}$ is obtained by WENO reconstructions of split fluxes given by

$$f^{y,\pm}_{i,j} = \frac{1}{2}\left(f^y_{i,j} \pm \alpha^y \boldsymbol{u}_{k,i,j}\right), \quad \alpha^y = \|\partial_{\boldsymbol{\rho}} f^y\|_{\mathbf{L}^\infty}.$$

If $\mathcal{R}^\pm$ denotes $(2r-1)$th-order WENO upwind biased reconstructions for $r = 2, 3, 4$, then

$$\begin{aligned}
\hat{f}^x_{i+1/2,j} &= \mathcal{R}^+\left(f^{x,+}_{i-r:i+r,j}\right) + \mathcal{R}^-\left(f^{x,-}_{i-r+1:i+r+1,j}\right), \\
\hat{f}^y_{i,j+1/2} &= \mathcal{R}^+\left(f^{y,+}_{i,j-r:j+r}\right) + \mathcal{R}^-\left(f^{y,-}_{i,j-r+1:j+r+1}\right),
\end{aligned}$$

where we have used Matlab-type notation for submatrices.

For the time discretization, we use the third-order Runge-Kutta TVD scheme proposed in [40].

3.3 Numerical Examples

For the following numerical examples, we compare numerical approximation obtained with the high-order numerical scheme described in Section 3.2. We denote by FD-WENOr the numerical method of r order for $r = 3, 5$ or 7. In order to validate the numerical results, we compare these approximations with respect to first-order Lax-Friedrich (LxF) scheme proposed in [1]. For all numerical schemes, we use a uniform mesh with $h_x = h_y = h$, these values will be specified for each test.

3.3.1 Example 3.1: A crowd dynamics sample integration, $N = 1$

We consider the example proposed in [1] in order to compare L^1-error between LxF and FD-WENO schemes and evacuation times.

$$\begin{cases} \partial_t \rho + \operatorname{div}_{\boldsymbol{x}} \left(\rho(1-\rho)(1-\eta * \rho)\boldsymbol{v}(\boldsymbol{x})\right) = 0, \quad \boldsymbol{x} \in \Omega \subset \mathbb{R}^2 \\ \rho(0, \boldsymbol{x}) = \rho_0(\boldsymbol{x}) \end{cases} \tag{3.4}$$

where one group of pedestrians, described through its density ρ, moves along a corridor defined by $\Omega = [0, 7] \times [-1, 1]$ and the pedestrian may exit along the segments $\{7\} \times [-1, 1]$. The vector field $\boldsymbol{v} = (v^1(x, y), v^2(x, y))$ describes the path followed by pedestrians and the smooth, nonnegative and compactly supported function η models the way in which each individual averages density around her/his position to adjust her/his speed. In order to compare the numerical solution obtained with the first-order and high-order schems, we consider the following parameters,

$$\boldsymbol{v}(x, y) = \begin{pmatrix} (1-y^2)^3 \exp\left(-1/(x-9.5)^2\right)\chi_{[\infty, 9.5]\times[-1,1]}(x, y) \\ -2y \exp(1 - 1/y^2)\chi_{[-1,1]}(y) \end{pmatrix}, \tag{3.5}$$

$$\eta(\boldsymbol{x}) = \frac{2}{\pi\, l^8}(l^2 - \|\boldsymbol{x}\|^2)^3 \chi_{[0,l]}(\|\boldsymbol{x}\|), \qquad l = 0.4, \tag{3.6}$$

and the initial datum

$$\rho_0(\boldsymbol{x}) = \chi_{[1,4]\times[0.1,0.8]}(\boldsymbol{x}) + \chi_{[2,5]\times[-0.8,-0.1]}(\boldsymbol{x}). \tag{3.7}$$

Numerical approximations for different simulation times $T = 1$, $T = 2$ and $T = 3$ are displayed in Figure 3.2 for $h = 1/10$. We observe that the numerical approximation for the LxF scheme is more diffusive than the numerical approximation obtained with the FD-WENO schemes. This behavior can be better appreciated in Figure 3.3 where we display the numerical approximation with $h = 1/80$ for FD-WENO schemes and with $h = 1/360$ for the LxF scheme, we observe that qualitatively, the approximation obtained by the FD-WENO schemes on the coarse mesh can be compared with the approximation obtained by the LxF scheme on the finest mesh. Taking

as reference solution the approximation obtained with LxF with $h = 1/640$, in Table 3.1 we compute the L^1-error for different discretizations with $h = 1/10, 1/20, 1/40$ and $h = 1/60$. We observe than for a certain error the approximation obtained by the FD-WENO 3 for a discretization is compared with the error obtained by the LxF scheme with a discretization twice as fine, and the CPU time is approximately ten times smaller. Similarly, a error obtained by the schemes FD-WENO5 and FD-WENO7 for a discretization level are compared with the error obtained by the LxF with a discretization 3 times finer and the this obtained in a shorter time.

Now we consider a group of pedestrain leaving the room by the exit at $\{7\} \times [-1, 1]$. In Figure 3.1, we display the evacuation times for different discretizations. Since the numerical approximations obtained by FD-WENO schemes are less diffusive, the evacuation times with these schemes are larger than the evacuation times obtained with by the LxF scheme. We observe that the required time by the LxF scheme with $h = 1/80$ is equivalent to the time required by the FD-WENO3 scheme with $h = 1/20$, while for the required time by the LxF scheme with $h = 1/160$ is comparable to the time required by the FD-WENO5 and FD-WENO7 with $h = 1/20$.

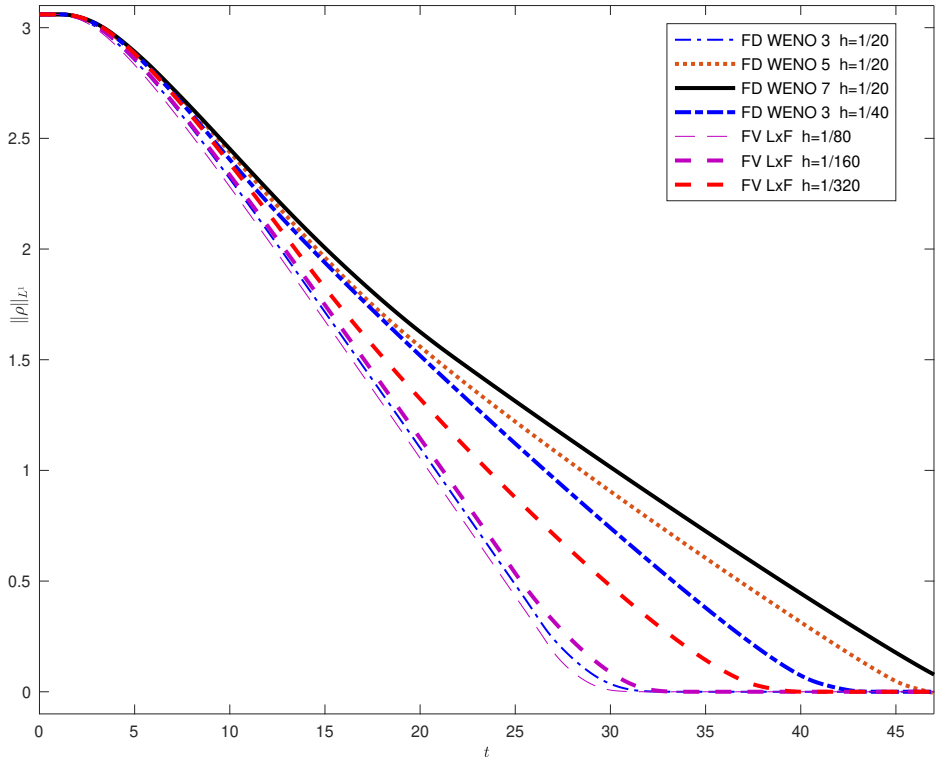

Figure 3.1: Example 3.1: Evacuation times for different discretizations (figure produced by author).

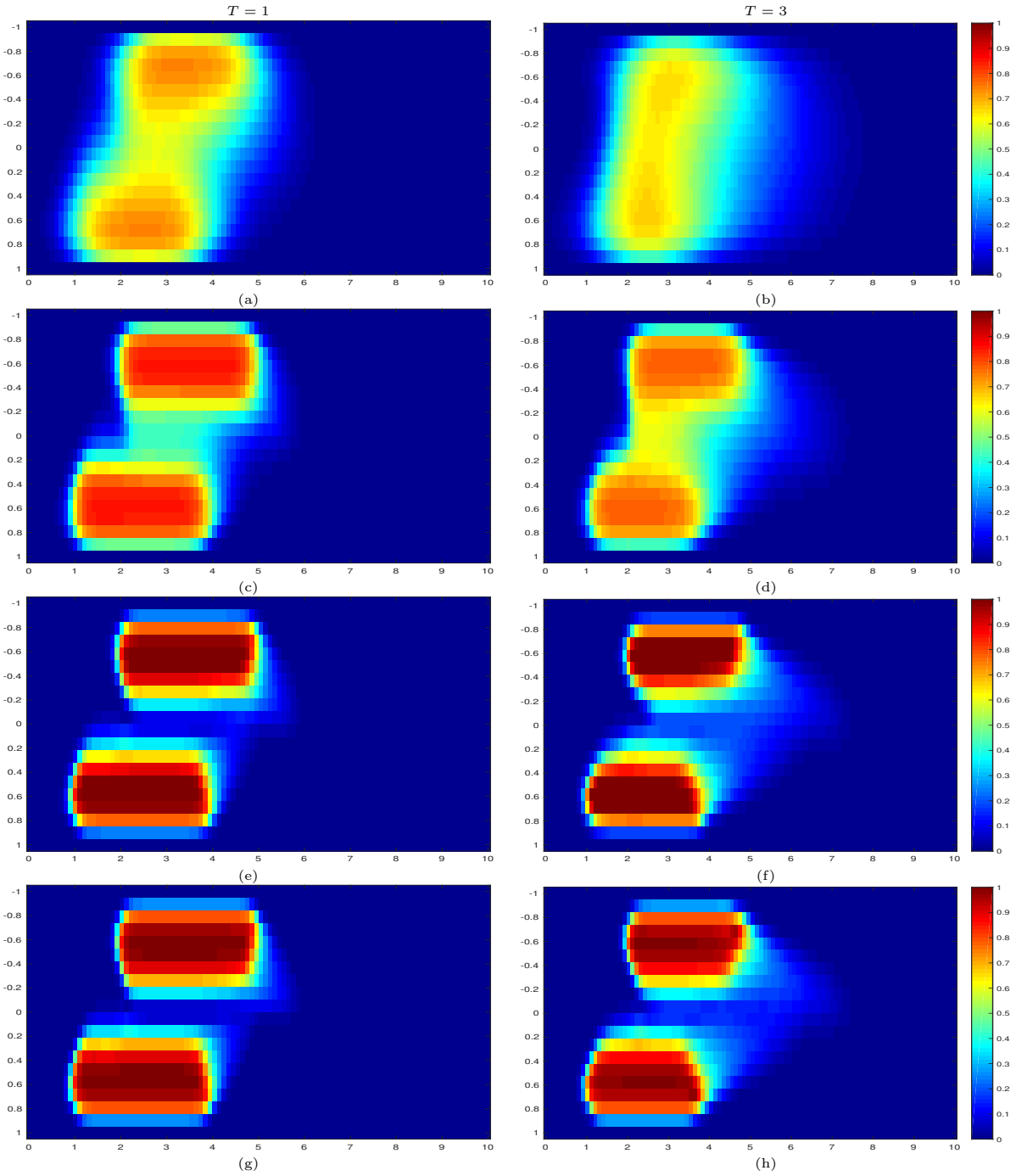

Figure 3.2: Example 3.1: Numerical solution computed with $h = 1/10$ at simulated times $T = 1$ and $T = 3$ for (a-b) Lx-F, (c-d) FD-WENO3, (e-f) FD-WENO5, (g-h) FD-WENO7. Vector field is obtained from vector field function (3.5) (figure produced by author).

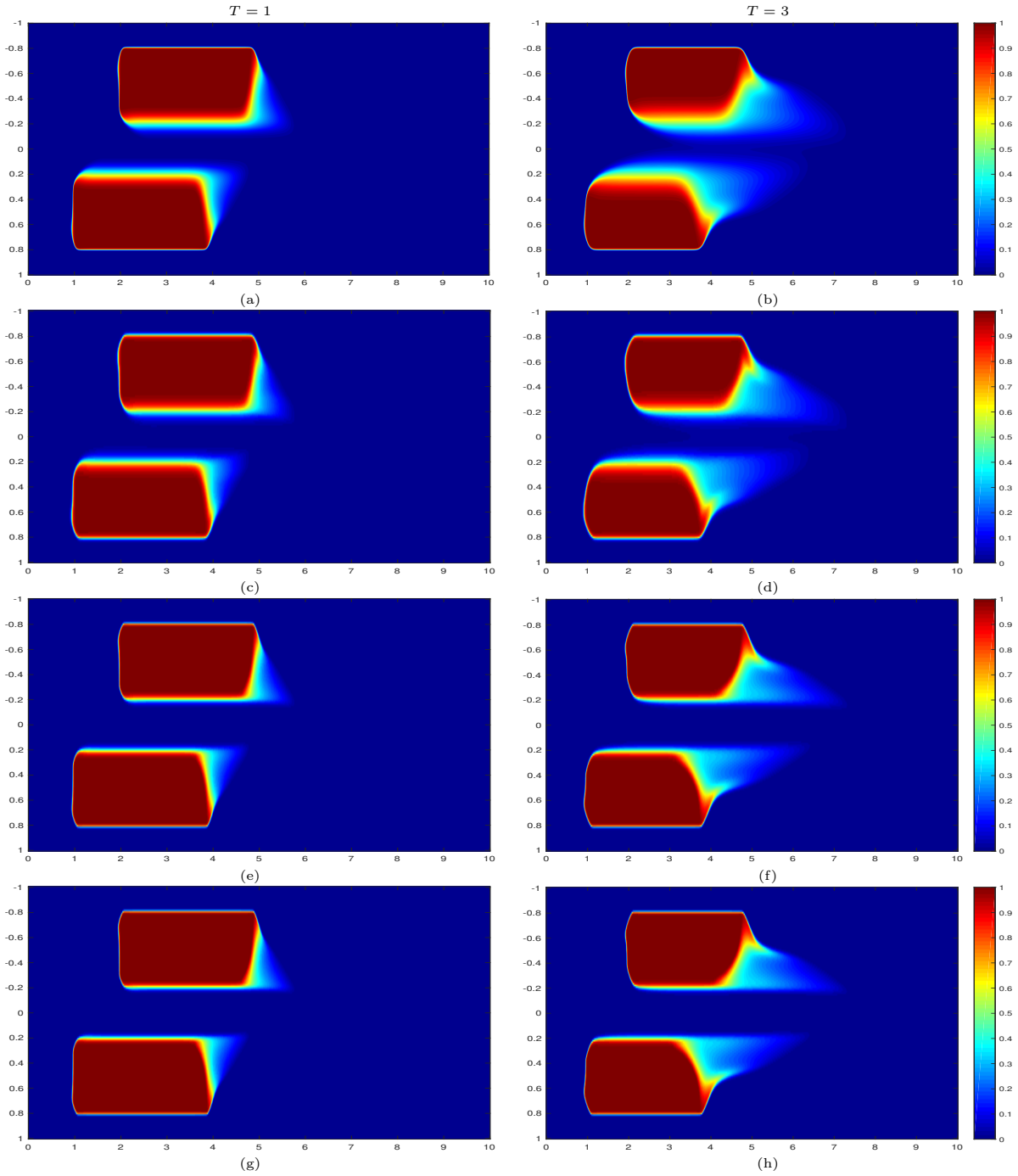

Figure 3.3: Example 3.1: Numerical solution computed at simulated times $T = 1$ and $T = 3$ with $h = 1/640$ for (a-b) Lx-F and computed with $h = 1/80$ for (c-d) FD-WENO3, (e-f) FD-WENO5, (g-h) FD-WENO7. Vector field is obtained from vector field function (3.5) (figure produced by author).

	LxF		FD-WENO 3		FD-WENO 5		FD-WENO 7	
M	e_M	cpu [s]	e_M	cpu [s]	e_M	cpu [s]	e_M	cpu [s]
	$T=1$							
100	125.63	0.63	81.65	0.92	44.83	0.98	41.89	6.05
200	103.76	5.99	42.96	17.71	23.28	22.37	19.22	39.15
400	72.49	38.16	19.42	87.12	8.12	101.15	6.14	213.93
800	45.71	215.08	6.62	481.76	2.79	610.84	3.30	1532.37
	$T=2$							
100	133.28	1.12	89.44	1.75	44.96	1.84	41.97	11.82
200	113.43	11.87	50.24	34.74	22.81	43.18	18.46	77.85
400	85.10	77.31	21.18	175.31	7.86	200.78	8.79	426.99
800	57.03	435.99	6.77	957.85	4.40	1179.98	5.10	2996.02
	$T=3$							
100	138.05	1.62	92.70	2.57	47.12	2.71	43.58	17.53
200	117.03	17.79	54.61	52.23	22.43	64.09	18.37	116.73
400	90.80	115.54	23.00	261.11	9.95	300.97	11.35	640.25
800	63.86	653.04	7.91	1438.98	5.76	1731.90	6.64	4469.25

Table 3.1: Example 3.1: approximate L^1 errors (e_h, figures to be multplied by 10^{-3}).

3.3.2 Example 3.2: Two Groups of people crossing, $N = 2$.

In this example, we consider a model of crowd dynamics proposed in [24]

$$\begin{cases} \partial_t \rho^1 + \operatorname{div}_{\boldsymbol{x}} \left(\rho^1 v(\rho^1) \left(\boldsymbol{v}^1(\boldsymbol{x}) + \mathcal{I}^{11}(\rho^1) + \mathcal{I}^{12}(\rho^2) \right) \right) &= 0 \\ \partial_t \rho^2 + \operatorname{div}_{\boldsymbol{x}} \left(\rho^2 v(\rho^2) \left(\boldsymbol{v}^2(\boldsymbol{x}) + \mathcal{I}^{21}(\rho^1) + \mathcal{I}^{22}(\rho^2) \right) \right) &= 0. \end{cases} \tag{3.8}$$

where the ρ^1 population moves to the right and the ρ^2 population moves to the left. The vector field $\boldsymbol{v}^i$ describes the preferred path. The term

$$\mathcal{I}^{ij}\left[\,\rho(\cdot,t)\,\right] = -\varepsilon_{ij} \frac{\nabla(\eta * \rho)}{\sqrt{1 + \|\nabla(\eta * \rho)\|^2}}. \tag{3.9}$$

describes how the ij-population deviates from its preferred trajectory due to the interaction among individuals, both of the same and the other population. In [23, 25] the authors consider this term as a nonlocal functional, since the value at any point $\boldsymbol{x}$ depends on the population densities average over a neighborhood of $\boldsymbol{x}$.

We consider the situation where the domain corresponds to a corridor defined by $\Omega = [-8, 8] \times [-4, 4]$ and pedestrians may exit along the segments $\{-8\} \times [-3, 3]$ and $\{8\} \times [-3, 3]$. The other parameters are defined by

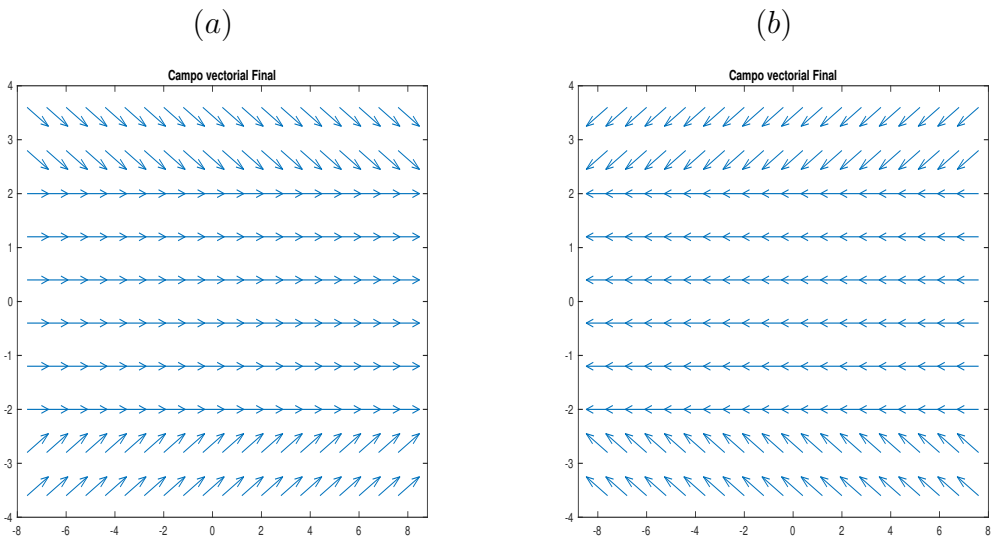

Figure 3.4: Example 3.2: Vector field $\boldsymbol{v}^1$ (a) and $\boldsymbol{v}^2$ (b) of corredor with two exits (figure produced by author).

$$\begin{aligned} \boldsymbol{v}^1 &= \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \delta, \qquad \eta(x,y) = \left((1-4x^2)(1-4y^2)\right)^3 \chi_{[-0.5,0.5]^2}(x,y) \\ \boldsymbol{v}^2 &= \begin{pmatrix} -1 \\ 0 \end{pmatrix} + \delta, \quad v(\rho) = 4(1-\rho), \qquad \varepsilon_{11} = \varepsilon_{22} = 0.3, \quad \varepsilon_{12} = \varepsilon_{21} = 0.7. \end{aligned} \tag{3.10}$$

where the vector δ, given in Figure 3.4, describes the disconfort of pedestrians when walking too near to a wall and is described more precisely in Example 3.3. The initial datum is given by

$$\rho_0^1(x,y) = 0.9\chi_{[-6.4,-3.2]\times[-2.4,2.4]}(x,y)\ , \qquad \rho_0^2(x,y) = 0.7\chi_{[3.2,6.4]\times[-2.4,2.4]}(x,y). \tag{3.11}$$

Numerical approximations are displayed in Figures 3.5 to Figures 3.8 for times $T = 1.5$, $T = 2.8$ and $T = 3.6$, for $h = 1/40$. As in Example 3.3.1, we observe that the numerical solution for LxF scheme 3.5 is more diffusive than the numerical solution for FD-WENO schemes.

In Figures 3.6 to 3.8 the numerical solution shows lane formation, in coarse resolution, due initially to the interaction between people of the same group and then to the interaction of the two populations in the central part of the corridor. We note that with the same resolution it is not possible that the LxF scheme captures the lane formation.

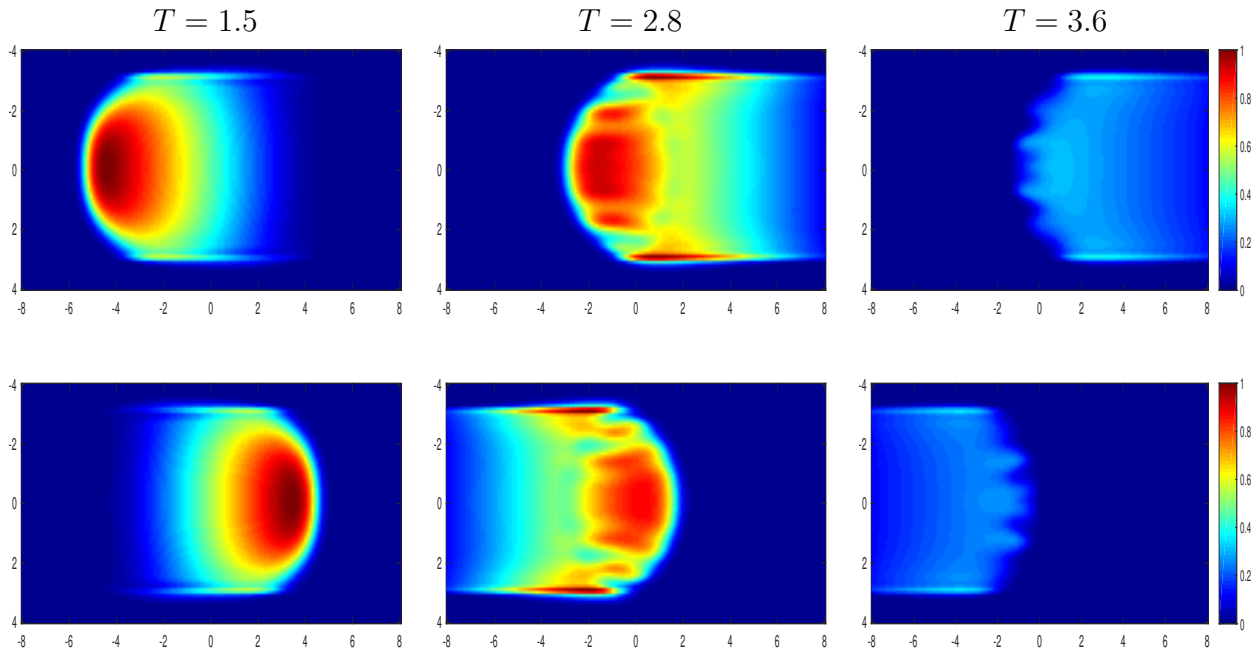

Figure 3.5: Example 3.2: Numerical Approximation of ρ^1 (bottom) and ρ^2 (top) obtained with LxF, at simulated times $T = 1.5, 2.8$ and 3.6 (figure produced by author).

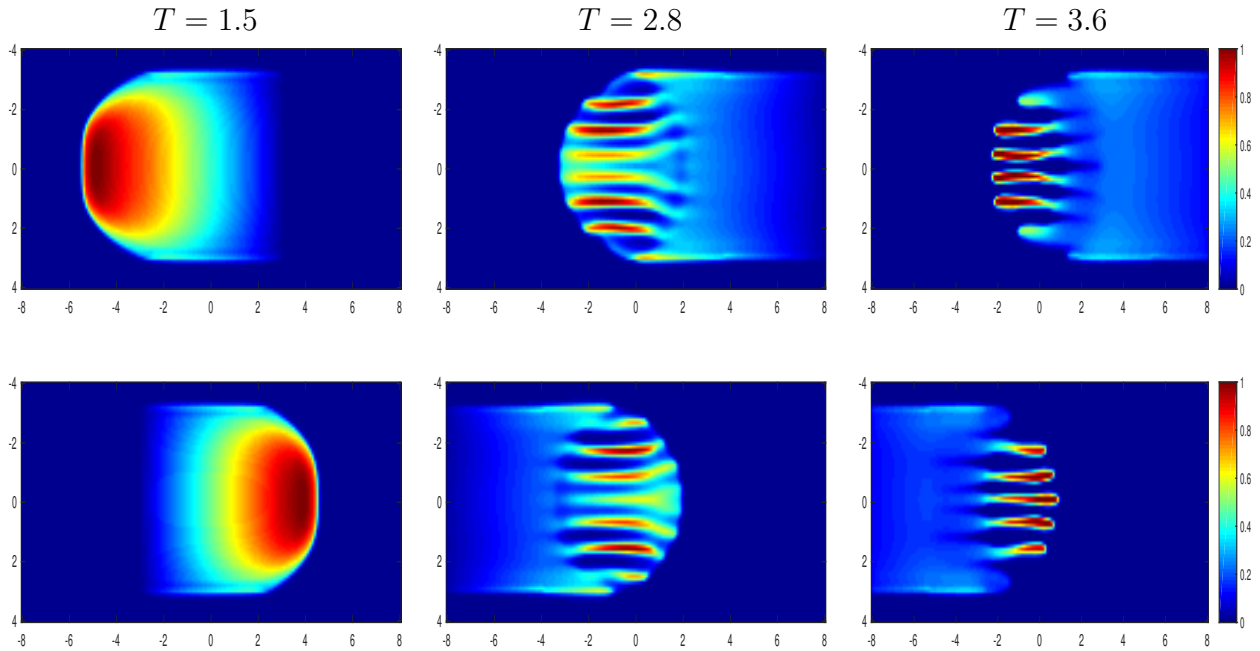

Figure 3.7: Example 3.2: Numerical Approximation of ρ^1 (bottom) and ρ^2 (top) obtained with FD-WENO5, at simulated times $T = 1.5, 2.8$ and 3.6 (figure produced by author).

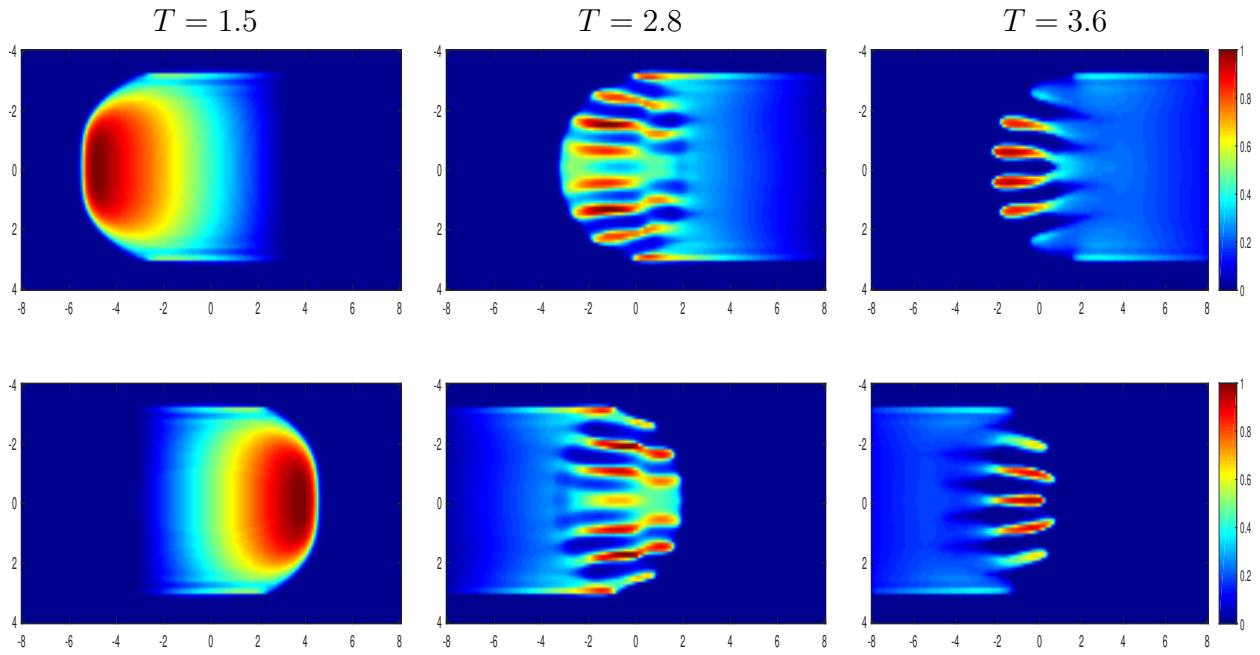

Figure 3.6: Example 3.2: Numerical Approximation of ρ^1 (bottom) and ρ^2 (top) obtained with FD-WENO3, at simulated times $T = 1.5, 2.8$ and 3.6 (figure produced by author).

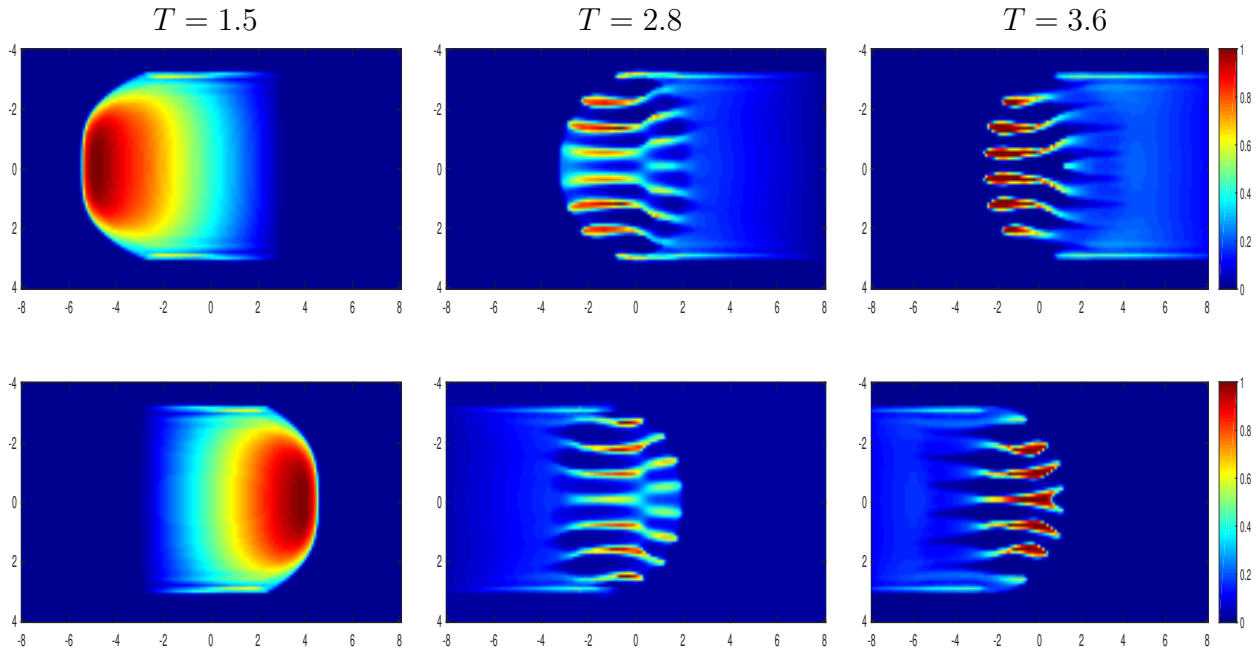

Figure 3.8: Exmple 3.2: Numerical Approximation of ρ^1 (bottom) and ρ^2 (top) obtained with FD-WENO7, at simulated times $T = 1.5, 2.8$ and 3.6 (figure produced by author).

3.3.3 Example 3.3: Evacuation from a room with obstacles, $N = 2$.

In this example, we focus on an evacuation problem of two groups of pedestrians with densities ρ_1 and ρ_2 in a room Ω defined in Figure 3.9, whose aim is to exit through Γ_0^1 and Γ_0^2 respectively. During the evacuation, the two species will cross on their way to the exit. Moreover, we consider the presence of obstacles in the room. According to [23], these obstacles may relieve the pressure on the exit, allowing a lower overall escape time. To this aim, we consider the nonlocal continuous equation (3.8) where the vector field $\boldsymbol{v}(\boldsymbol{x})$ is obtained as a sum of the unit vector tangent to the geodesic from $\boldsymbol{x}$ to the exit door and a discomfort vector field with maximal intensity along the walls.

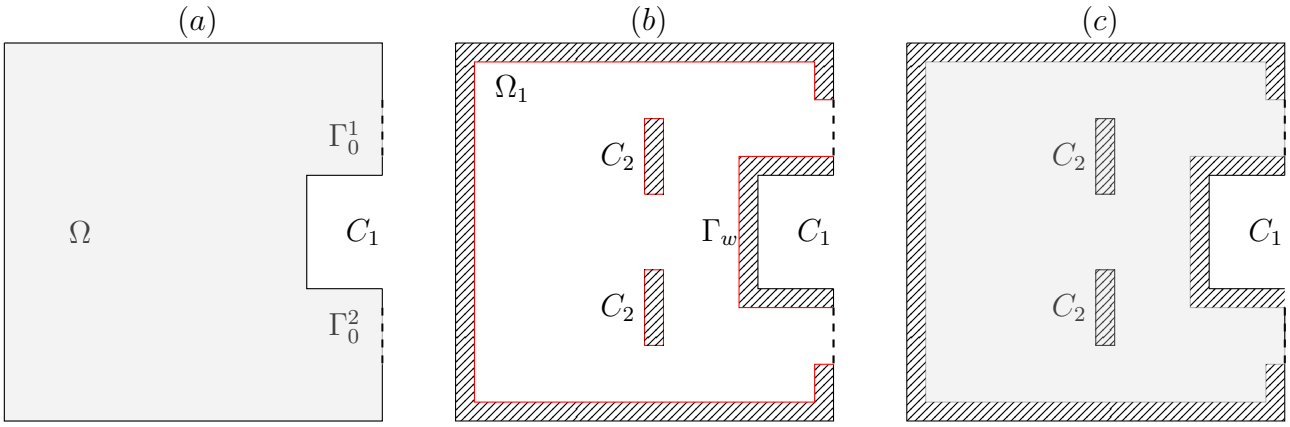

Figure 3.9: (a) Domain Ω with exit Γ_0, (b) Repulsion domain Ω_1 (black stripes) with exit Γ_w (red lines) and (c) union of both domains (figure produced by author).

The domain corresponds to a square room $\Omega = [0, 10] \times [-5, -5]$ contains columns $C_1 = [9, 10] \times [-2.5, 2.5]$ and $C_2 = C_1 \cup A$ where $A = [5, 5.5] \times ([-3, -1] \cup [1, 3])$. In all cases, the exit is given by $\Gamma_0 = \{10\} \times ([-4, -2.5] \cup [2.5, 4])$.

To obtain these vectors we solved the eikonal equation over a domain Ω_* with exit Γ_*, then the vector field $\boldsymbol{v}(\boldsymbol{x})$ can be modeled taking the normalized gradient of the solution:

$$\begin{aligned} |\nabla\varphi(\boldsymbol{x})| &= 1, \quad \boldsymbol{x} \in \Omega_*, \\ \varphi(\boldsymbol{x}) &= 0, \quad \boldsymbol{x} \in \Gamma_*. \end{aligned} \qquad \boldsymbol{v}(\boldsymbol{x}) = \frac{\nabla\varphi(\boldsymbol{x})}{\|\nabla\varphi(\boldsymbol{x})\|}. \tag{3.12}$$

In our example, we solve (3.12) following the description in [42, Appendix C] twice, one with $\Omega_* = \Omega_0$ and $\Gamma_* = \Gamma_0^1$ to obtain the direction vector field of ρ^1 and once with $\Omega_* = \Omega_1$ and $\Gamma_* = \Gamma_w$ to obtain the repulsion of the walls vector field of ρ^2 (see Figure 3.9).

The initial condition and the other parameters are defined as

$$\begin{gathered} \rho_0^1(\boldsymbol{x}) = 0.6\chi_{[0.5,2]\times[-4.5,-1]}(\boldsymbol{x}) \ , \qquad \rho_0^2(\boldsymbol{x}) = 0.9\chi_{[0.5,2]\times[1,4.5]}(\boldsymbol{x}) \\ \eta(\boldsymbol{x}) = \frac{315}{128\pi}\left(1 - \|\boldsymbol{x}\|^4\right)^4 \chi_{[0,1]}(\boldsymbol{x}). \end{gathered} \tag{3.13}$$

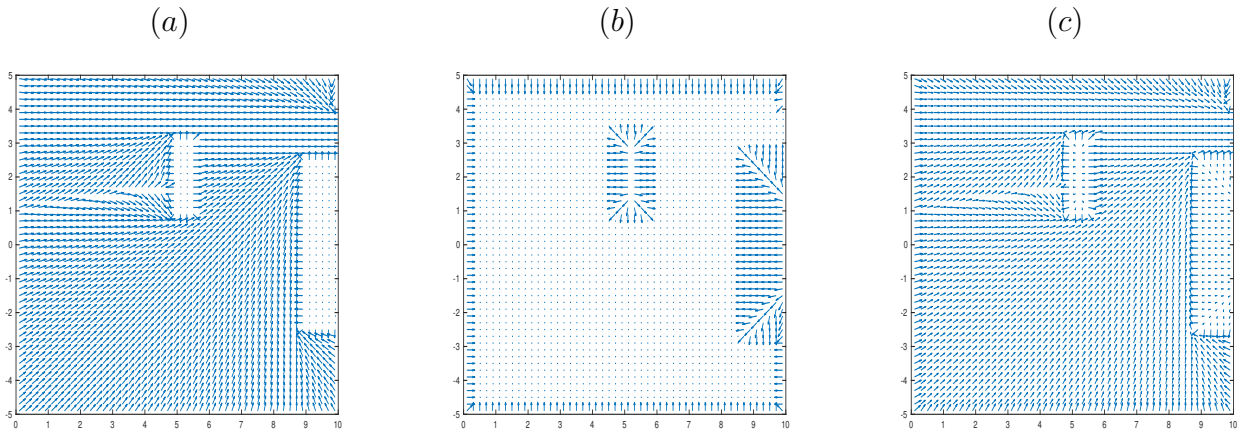

Figure 3.10: Example 3.3: (a) Vector field of domain, (b) Repulsion vector field and (c) Union of vector field (a) and (b). By symetry we obtain the vector field of the lower output (figure produced by author).

Numerical approximations are shown in Figures 3.11 and 3.12 for times $T = 1$, $T = 4$ and $T = 7$, for $h = 1/20$. In them we observe that for a coarse discretization, the solution displays the lane formation phenomenon, with pedestrian self-organized along lanes and is independent of the directions and obstacles of the problem.

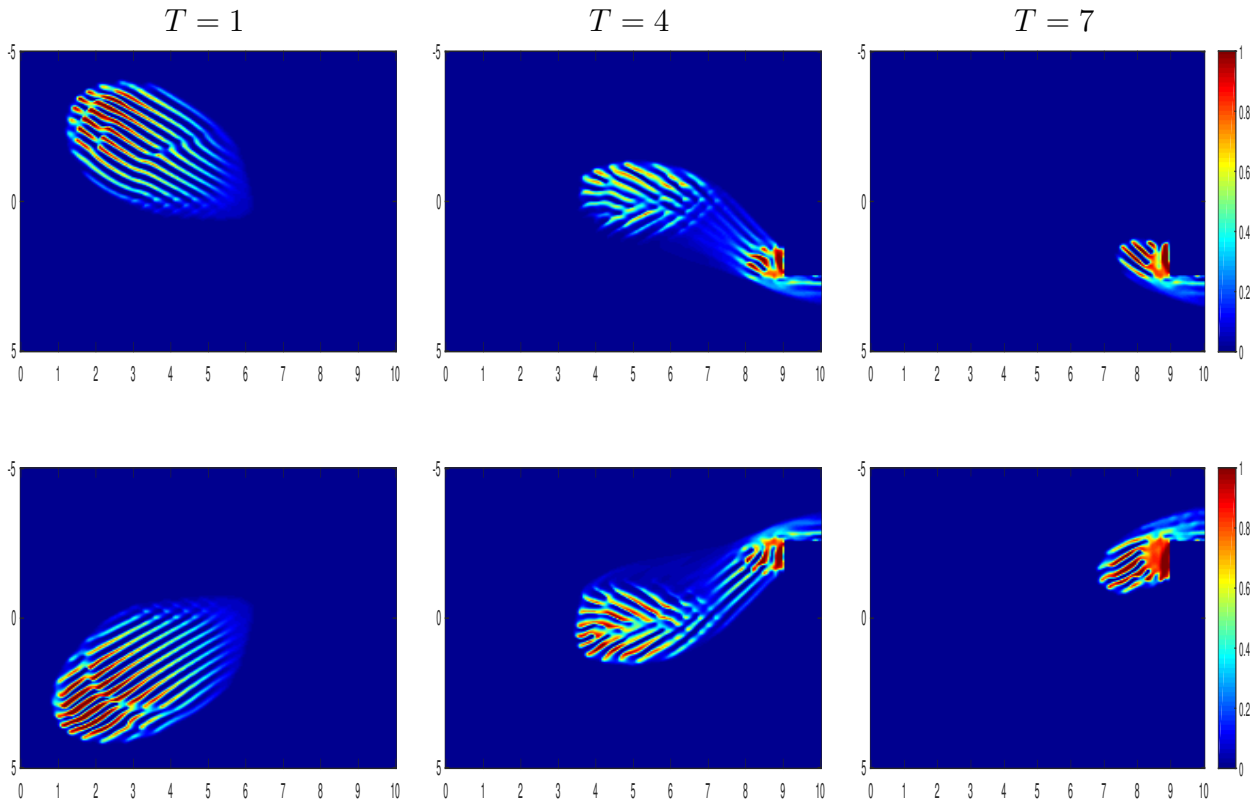

Figure 3.11: Example 3.3 (without obstacle): Numerical solution with $h = 1/20$ at simulated times $T = 1$, 2 and 7, produced by FD-WENO5 scheme (figure produced by author).

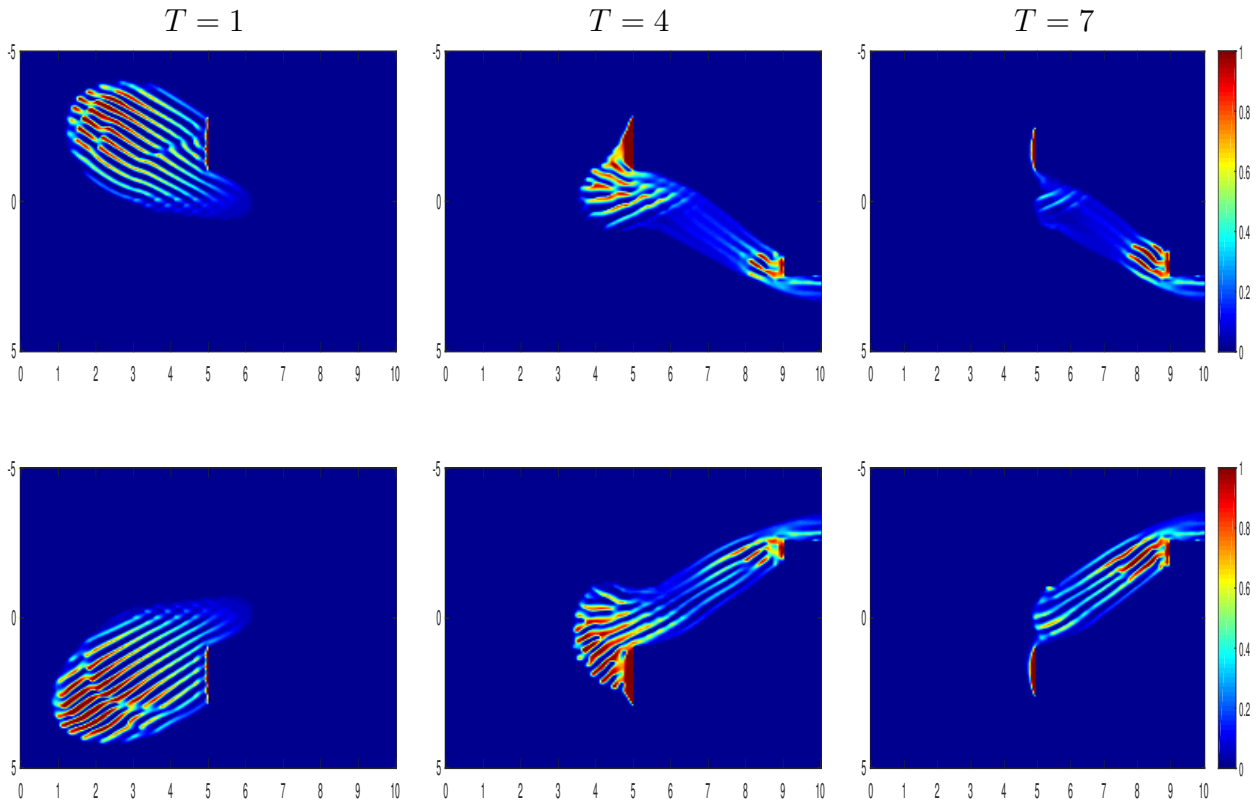

Figure 3.12: Example 3.3 (with obstacle): Numerical solution with $h = 1/20$ at simulated times $T = 1$, 2 and 7, produced by FD-WENO5 scheme (figure produced by author).

Conclusions and future works

Conclusions

Here we present a summary with the main contributions and conclusions of the thesis.

- In Chapter 1 we show that a particular IMEX-RK scheme represents an important and serious alternative to the explicit scheme introduced in [21] for the efficient numerical solution of the one-dimensional nonlinear nonlocal equation (0.1). At a fixed spatial discretization the explicit scheme is more accurate in most settings. However, the gain in CPU time (due to the less restrictive CFL condition) by the IMEX-RK scheme is in most circumstances, and in particular for fine discretizations, so significant that the IMEX-RK scheme turns out most efficient in terms of error reduction per CPU time. In this respect we mention that higher-order IMEX-RK schemes have also been tested, but with less significant gains of accuracy at least for the moderately fine discretizations used in this chapter. The gain of efficiency attainable by an IMEX-RK scheme depends on the relative magnitude of the diffusion versus convection terms, a parameter that we did not vary herein since we insist on adopting the test cases of [21] (Examples 1 to 3) and [7] (Example 4). The scenario of Example 4, and in particular the convenient computation of a reference solution by solving a local PDE, does not have a counterpart in two or three space dimensions since the aggregation model cannot be extended in a straightforward way to several dimensions. This has been our prime motivation to analyze the one-dimensional case separately.

- In Chapter 2 we naturally extend the results obtained in Chapter 1 and through a series of numerical examples we reconfirm that IMEX schemes, based on time discretizations of the type IMEX-RK, represent a serious alternative to the explicit scheme introduced in [21] for an efficient numerical solution of (2.1). Here, we observe for Examples 1 to 4 that, according to the Tables 2.1 to 2.4, the numerical approximation of the error for a given simulation time and discretization remain very close to each other for all the numerical schemes tested and the efficiency plots indicate that at least for fine discretizations, there is a clear efficiency gain of IMEX-RK schemes in comparison with their explicit counterpart. With respect to CPU times, we remark that the maxima arising obtained in (2.29) and

(2.30) have been evaluated in each iteration. Examples 2 and 5 deserve mention, which are particular cases of the equation(2.1), and which represent the behavior of swarm with diffusion, which have been able captured efficiently and cheap (computationally speaking) though IMEX-RK schemes.

- In Chapter 3 we confirm what is stated in Chapter 1. High-order schemes are an important alternative to first-order schemes, since they manage to capture more accurately the numerical solutions obtained and in some cases they manage to lower the CPU time. This can be seen in the middle part of the Table 3.1. In addition, it is possible to appreciate, with low resolution, and in the case of domains with obstacles, to the process of lane formation in numerical solutions (Examples 3.2 and 3.3).

Future Work

In general terms we can indicate that, further research is interesting about similar equations using techniques analogous to those used in this thesis. Especially tackle the different applications that have such equations and the importance of obtaining numerical solutions fast and efficiently (computationally speaking). For example, probable scenarios for future research are:

- Obtain high-order numerical methods for non-linear and non-local equations with cross-diffusion.
- Efficiently simulate epidemiological and dynamic models of the predator-prey type.
- Efficient numerical methods for advection-diffusion-reaction equations of the form

$$\begin{aligned} \frac{\partial u}{\partial t} + \boldsymbol{w}(\boldsymbol{x},t)\cdot\nabla u &= \nabla\cdot(K(u)\nabla u) + f(u,v,\boldsymbol{x}), \\ \frac{\partial v}{\partial t} &= g(u,v). \end{aligned} \tag{3.14}$$

  that model forest fires.

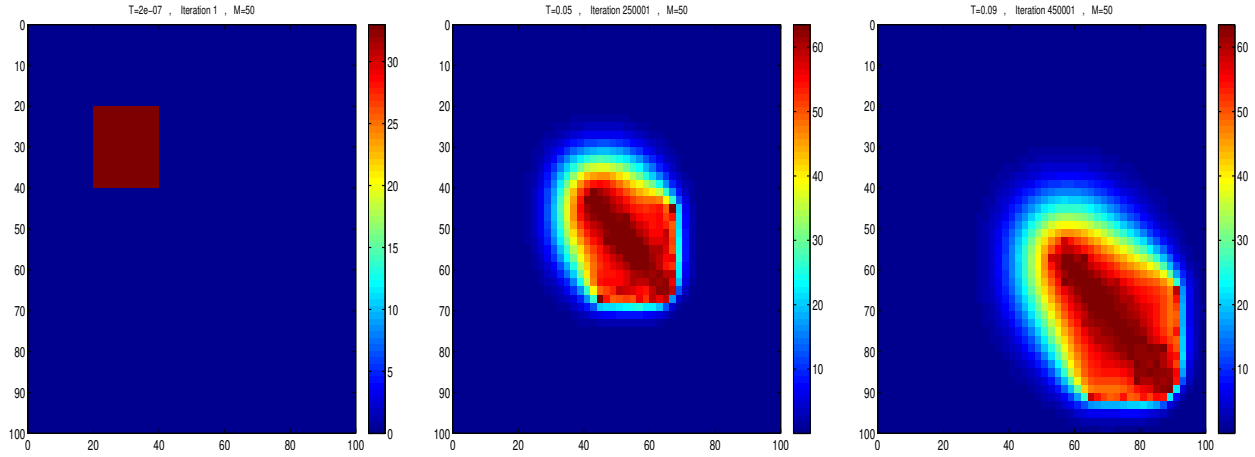

Figure 3.13: Numerical solution of (3.14) computed at simulated times $T = 2e-07$, $T = 0.05$ and $T = 0.09$ with on the domain $\Omega = (0, 100) \times (0, 100)$ in dimensionless variables, which corresponds to a square of side length $100l_0 = 89.94$ m. We choose the dimensionless wind vector $w = (w_1, w_2) = (100, 100)$, which blows in south-east direction at physical speed $\|\boldsymbol{v}\| = (l_0/t_0)\|\boldsymbol{w}\| \approx 0.0142\text{m}s^{-1}$. *Work in preparation* (figure produced by author).

- Maximum principle and positivity preserving high-order schemes solving crowd dynamics equations and pedestrian movement.
- Propose high-order numerical schemes that allow solving the eikonal equation, thereby generating the repulsion and vector fields associated with the movement of populations, and the control problem associated with the position on which must have an obstacle to reduce evacuation time.

Conclusiones Generales y Trabajos Futuros

Conclusiones

A continuación, se presenta un resumen con los principales aportes y conclusiones generadas en esta tesis.

- En el Capítulo 1 mostramos que un esquema IMEX-RK particular representa una alternativa importante y seria al esquema explícito introducido en [21] para la solución numérica eficiente de la ecuación no lineal no lineal unidimensional (0.1). Para una discretización espacial fija, el esquema explícito es más preciso en la mayoría de las configuraciones. Sin embargo, la ganancia en tiempo de CPU (debido a la condición CFL menos restrictiva) por el esquema IMEX-RK es en la mayoría de las circunstancias, y en particular para discretizaciones finas, tan significativa que el esquema IMEX-RK resulta más eficiente en términos de reducción del error por tiempo de CPU. Al respecto, mencionamos que también se han probado esquemas IMEX-RK de orden superior, pero con ganancias de precisión menos significativas al menos para las discretizaciones moderadamente finas utilizado en este capítulo. La ganancia de eficiencia alcanzable por un esquema IMEX-RK depende de la magnitud relativa de los términos de difusión versus convección, un parámetro que no variamos aquí ya que insistimos en adoptar los casos de prueba de [21] (Ejemplos 1.1 a 1.3) y [7] (Ejemplo 1.4). El escenario del Ejemplo 4, y en particular el cálculo conveniente de una solución de referencia al resolver un PDE local, no tiene una contraparte en dos o tres dimensiones espaciales ya que el modelo de agregación no puede extenderse de manera directa a varias dimensiones. Esta ha sido nuestra principal motivación para analizar el caso unidimensional por separado.

- En el Capítulo 2 extendemos de manera natural los resultados obtenidos en el Capítulo 1 y a través de una serie de ejemplos numéricos reconfirmamos que los esquemas IMEX, basados en discretizaciones del tiempo del tipo IMEX-RK, representan una seria alternativa al esquema explícito introducido en [21] para una solución numérica eficiente de (2.1). Aquí, observamos para los Ejemplos del 2.1 al 2.4 que, de acuerdo a las Tablas del 2.1 al 2.4, la aproximación numérica del error para un tiempo de simulación y discretización dada permanecen muy cercanas unas a otras para todos los esquemas numéricos testeados y los gráficos de eficiencia indican que al menos para discretizaciones finas, hay una

clara ganancia de eficiencia de los esquemas IMEX-RK en comparación con su contraparte explícita. Con respecto al tiempo de CPU, recalcamos que los valores máximos que se obtienen en (2.29) y en (2.30) han sido calculados en cada iteración. Mención aparte merecen los Ejemplos 2.2 y 2.5, los cuales son casos particulares de la ecuación (2.1), y que representan el comportamiento de enjambre con difusión, los cuales se han podido capturar de forma eficiente y barata (computacionalmente hablando) a través de los esquemas IMEX-RK.

- En el Capítulo 3 confirmamos lo planteado en el Capítulo 1. Los esquemas de alto orden son una alternativa importante a esquemas de primer orden, pues logran capturar de forma más precisa las soluciones numéricas obtenidas y en algunos casos logran bajar el tiempo de CPU. Esto se puede apreciar en la parte media de la Tabla 3.1. Además se logra apreciar, con baja resolución, aún en los casos de dominios con obstáculos el proceso de formación de lineas en las soluciones numéricas (Ejemplos 3.2 y 3.3).

Trabajo Futuro

En lineas generales podemos indicar que, son de interés realizar más investigaciones sobre ecuaciones similares empleando técnicas análogas a las utilizadas en esta tesis. En especial abordar las diferentes aplicaciones que tienen tales ecuaciones y la importancia que tiene obtener soluciones numéricas de forma rápida y eficiente (computacionalmente hablando). Por ejemplo, probables escenarios de interés para futuras investigaciones son:

- Obtener métodos numéricos de alto orden para ecuaciones no linealess y no locales con difusión cruzada.
- Simular de forma eficiente modelos epidemiológicos y dinámicas del tipo depredador-presa.
- Métodos numéricos eficientes para ecuaciones de advección-difusión-reacción de la forma

$$\begin{aligned} \frac{\partial u}{\partial t} + \boldsymbol{w}(\boldsymbol{x},t)\cdot\nabla u &= \nabla\cdot(K(u)\nabla u) + f(u,v,\boldsymbol{x}), \\ \frac{\partial v}{\partial t} &= g(u,v). \end{aligned} \tag{3.15}$$

  que modelan incendios forestales.

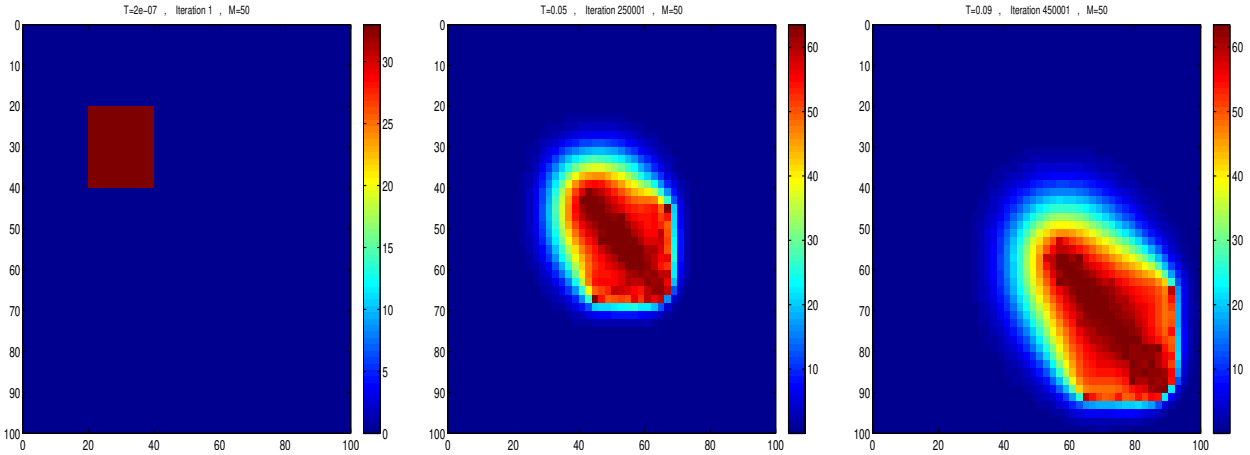

Figure 3.14: Solución numérica de (3.14) calculada en tiempos $T = 2e-07$, $T = 0.05$ y $T = 0.09$ con dominio $\Omega = (0, 100)\times(0, 100)$ con dimensiones variables, que corresponde a un cuadrado de largo $100l_0 = 89.94$ m. Elegimos el vector de viento no dimensional $w = (w_1, w_2) = (100, 100)$, que sopla en la dirección sureste con velocidad física $\|\boldsymbol{v}\| = (l_0/t_0)\|\boldsymbol{w}\| \approx 0.0142\text{m}s^{-1}$. *Trabajo en preparación* (Figura producida por el autor).

- Principio del máximo y esquemas de alto orden que preservan positividad y que resuelven dinámica de poblaciones y movimiento de peatones.

- Plantear esquemas numéricos de alto orden que permitan resolver la ecuación eikonal, con lo cual se generan la repulsión y campos de vectores asociados al movimiento de poblaciones, y el problema de control asociado a lo posición que debe tener un obstáculo de manera de disminuir el tiempo de evacuación.

References

[1] A. Aggarwal, R. M. Colombo, and P. Goatin, *Nonlocal systems of conservation laws in several space dimensions*, SIAM Journal on Numerical Analysis, 53 (2015), pp. 963–983.

[2] W. Alt, *Degenerate diffusion equations with drift functionals modelling aggregation*, Nonlinear Analysis: Theory, Method and Applications, 9 (1985), pp. 811–836.

[3] U. Ascher, S. Ruuth, and J. Spiteri, *Implicit-explicit runge-kutta methods for time dependent partial differential equations*, Applied Numerical Mathematics, 25 (1997), pp. 151–167.

[4] A. B. T. Barbaro, J. A. Cañizo, J. A. Carrillo, and P. Degond, *Phase transitions in a kinetic model of Cucker-Smale type*, Multiscale Modelling and Simulation, 14 (2016), pp. 1063–1088.

[5] D. Benedetto, E. Caglioti, and M. Pulvirenti, *A kinetic equation for granular media*, ESAIM: Mathematical Modelling and Numerical Analysis, 31 (1997), pp. 615–641.

[6] M. Bessemoulin-Chatard and F. Filbet, *A finite volume scheme for nonlinear degenerate parabolic equations*, SIAM Journal on Scientific Computing, 34 (2012), pp. B559–B583.

[7] F. Betancourt, R. Bürger, and K. H. Karlsen, *A strongly degenerate parabolic aggregation equation*, Communications in Mathematical Sciences, 9 (2011), pp. 711–742.

[8] F. Betancourt, R. Bürger, and K. H. Karlsen, *Well-posedness and travelling wave analysis for strongly degenerate parabolic aggregation equation*, Hyperbolic Problems: Theory, Numerics and Applications. Serie in Contemporary Applied Mathematics CAM 17/18, vol. 1. Higher Education Press, Beijing, China, 2012.

[9] S. Boscarino, R. Bürger, P. Mulet, G. Russo, and L. M. Villada, *Linearly implicit IMEX Runge-Kutta methods for a class of degenerate convection-diffusion problems*, SIAM Journal on Scientific Computing, 37 (2015), pp. B305–B331.

[10] S. Boscarino, R. Bürger, P. Mulet, G. Russo, and L. M. Villada, *On linearly implicit IMEX Runge-Kutta methods for degenerate convection-diffusion problems modelling polydisperse sedimentation*, Bulletin of the Brazilian Mathematical Society (New Series), 47 (2016), pp. 171–185.

[11] S. Boscarino, F. Filbet, and G. Russo, *High order semi-implicit schemes for time dependent differential equations*, Journal of Scientific Computing, 68 (2016), pp. 975–1001.

[12] S. Boscarino, P. G. LeFloch, and G. Russo, *High order asymptotic-preserving methods for fully nonlinear relaxation problems*, SIAM Journal on Scientific Computing, 36 (2014), pp. A377–A395.

[13] S. Boscarino and G. Russo, *On a class of uniformly accurate IMEX Runge-Kutta schemes and applications to hyperbolic systems with relaxation*, SIAM Journal on Scientific Computing, 31 (2009), pp. 1926–1945.

[14] S. Boscarino and G. Russo, *Flux-explicit IMEX Runge-Kutta schemes for hyperbolic to parabolic relaxation problems*, SIAM Journal on Numerical Analysis, 51 (2013), pp. 163–190.

[15] M. Braś, G. Izzo, and Z. Jackiewicz, *Accurate implicit-explicit general linear methods with inherent Runge-Kutta stability*, Journal of Scientific Computing, 70 (2017), pp. 1105–1143.

[16] M. Burger, J. A. Carrillo, and M. T. Wolfram, *A mixed finite element method for nonlinear diffusion equations*, Kinetic and Related Models, 3 (2010), pp. 59–83.

[17] R. Bürger, D. Inzunza, P. Mulet, and L. M. Villada, *Implicit-explicit methods for a class of nonlinear nonlocal gradient flow equations modelling collective behavior*, Applied Numerical Mathematics, 144 (2019), pp. 234–252.

[18] R. Bürger, D. Inzunza, P. Mulet, and L. M. Villada, *Implicit-explicit schemes for nonlinear nonlocal equations with a gradient flow structure in one space dimension*, Numerical Methods for Partial Differential Equations, 35 (2019), pp. 1008–1034.

[19] R. Bürger, P. Mulet, L. Rubio, and M. Sepulveda, *Linearly implicit-explicit schemes for the equilibrium dispersive model of chromatography*, Applied Mathematics and Computation, 317 (2018), pp. 172–186.

[20] R. Bürger, P. Mulet, and L. M. Villada, *Regularized nonlinear solvers for IMEX methods applied to diffusively corrected multi-species kinematics flow models*, SIAM Journal on Scientific Computing, 35 (2013), pp. B751–B777.

[21] J. A. Carrillo, A. Chertock, and Y. Huang, *A finite-volume method for nonlinear nonlocal with a gradient flow structure*, Communications in Computational Physics, 17 (2015), pp. 233–258.

[22] J. A. Carrillo and G. Toscani, *Asymptotic l^1-decay of solutions of the porous media equation to self-similarity*, Indiana University Mathematics Journal, 49 (2000), pp. 113–142.

[23] R. M. Colombo, M. Garavello, and M. Lécureux-Mercier, *A class of nonlocal models for pedestrian traffic*, Mathematical Models and Methods in Applied Sciences, 22 (2012), p. 1150023.

[24] R. M. Colombo and M. Lécureux-Mercier, *Nonlocal crowd dynamics models for several populations*, Acta Mathematica Scientia, 32 (2012), pp. 177–196.

[25] R. M. Colombo and E. Rossi, *Nonlocal conservation laws in bounded domains*, SIAM Journal on Mathematical Analysis, 50 (2018), pp. 4041–4065.

[26] M. G. Crandall and A. Majda, *Monotone difference approximations for scalar conservation laws*, Mathematics of Computation, 34 (1980), pp. 1–21.

[27] M. Crouzeix, *Une méthode multipas implicite-explicite pour l'approximation des équations d'évolution paraboliques*, Numerische Mathematik, 35 (1980), pp. 257–276.

[28] J. E. Dennis Jr. and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Classics in Applied Mathematics vol. 16, SIAM, 1996.

[29] R. Donat, F. Guerrero, and P. Mulet, *Implicit-explicit methods for models for vertical equilibrium multiphase flow*, Computers and Mathematics with Applications, 68 (2014), pp. 363–383.

[30] R. Donat and I. Higueras, *On stability issues for IMEX scheme applied to 1d scalar hyperbolic equations with stiff reaction terms*, Mathematics of Computation, 80 (2011), pp. 2097–2126.

[31] B. Engquist and S. Osher, *One-sided difference approximations for nonlinear conservative laws*, Mathematics of Computation, 36 (1981), pp. 312–351.

[32] S. Gottlieb, C. W. Shu, and E. Tadmor, *Strong stability-preserving high-order time discretization methods*, SIAM Review, 43 (2001), pp. 89–112.

[33] X. Y. Hu, N. A. Adams, and C.-W. Shu, *Positivity-preserving method for high-order conservative schemes solving compressible euler equations*, Journal of Computational Physics, 242 (2013), pp. 169 – 180.

[34] G. Izzo and Z. Jackiewicz, *Highly stable implicit-explicit runge-kutta methods*, Applied Numerical Mathematics, 113 (2017), pp. 71–92.

[35] Z. Jackiewicz, *General Linear Methods for Ordinary Differential Equations*, Wiley, Hoboken, NJ, 2009.

[36] G. S. Jiang and C. W. Shu, *Efficient implementation of weighted ENO schemes*, Journal of Computational Physics, 126 (1996), pp. 202–228.

[37] K. H. Karlsen and N. H. Risebro, *Convergence of finite difference schemes for viscous and inviscid conservation laws with rough coefficients*, ESAIM: Mathematical Modelling and Numerical Analysis, 35 (2001), pp. 239–269.

[38] E. F. Keller and L. A. Segel, *Initiation of slime mold aggregation viewed as an instability*, Journal of Theoretical Biology, 26 (1970), pp. 399–415.

[39] C. A. Kennedy and M. H. Carpenter, *Additive Runge-Kutta schemes for convection-diffusion-reaction equations*, Applied Numerical Mathematics, 44 (2003), pp. 139–181.

[40] X. D. Liu, S. Osher, and T. Chan, *Weighted essentially non-oscillatory schemes*, Journal of Computational Physics, 115 (1994), pp. 200–212.

[41] R. J. McCann, *A convexity principle for interacting gases*, Advances in Mathematics, 128 (1997), pp. 153–179.

[42] M. Mimault, *Crowd motion modeling by conservation laws*, PhD thesis, Université de Nice-Sophia Antipolis (2015).

[43] T. Nagai, *Some nonlinear degenerate diffusion equations with a nonlocally convective term in ecology*, Hiroshima Mathematical Journal, 13 (1983), pp. 165–202.

[44] T. Nagai and M. Mimura, *Asymptotic behavior for a nonlinear degenerate diffusion equation in population dynamics*, SIAM Journal on Applied Mathematics, 43 (1983), pp. 449–464.

[45] T. Nagai and M. Mimura, *Some nonlinear degenerate diffusion equations related to population dynamics*, Journal of Mathematical Society of Japan, 35 (1983), pp. 539–562.

[46] T. Nagai and M. Mimura, *Asymptotic behavior of the interface to a nonlinear degenerate diffusion equation in population dynamics*, Japan Journal of Applied Mathematics, 3 (1986), pp. 129–161.

[47] G. Naldi, L. Pareschi, and G. Toscani, *Mathematical Modeling of Collective Behavior in Socio-Economic and Life Science*, Birkhäuser, Boston, 2010.

[48] G. Naldi, L. Pareschi, and G. Toscani, *Interactive Multiagent Systems: Kinetic Equations and Monte Carlo Methods*, Oxford University Press, Oxford, 2013.

[49] J. M. Ortega and W. C. Rheinboldt, *Iterative solution of nonlinear equations in several variables*, Academic Press, New York-London, 1970.

[50] F. Otto, *The geometry of dissipative evolution equations: The porous medium equation*, Communications in Partial Differential Equations, 26 (2001), pp. 101–174.

[51] L. Pareschi and G. Russo, *Implicit-explicit Runge-Kutta schemes and applications to hyperbolic systems with relaxation*, Journal of Scientific Computing, 25 (2005), pp. 129–155.

[52] L. Pareschi and M. Zanella, *Structure preserving schemes for nonlinear Fokker-Planck equations and applications*, Journal of Scientific Computing, 74 (2018), pp. 1575–1600.

[53] C. W. Shu and S. Osher, *Efficient implementation of essentially non-oscillatory shock-capturing schemes*, Journal of Computational Physics, 77 (1988), pp. 439–447.

[54] C. W. Shu and S. Osher, *Efficient implementation of essentially non-oscillatory shock-capturing schemes II*, Journal of Computational Physics, 83 (1989), pp. 32–78.

[55] C. M. Topaz, A. L. Bertozzi, and M. A. Lewis, *A nonlocal continuum model for biological aggregation*, Bulletin of Mathematical Biology, 68 (2006), pp. 1601–1623.

[56] G. Toscani, *One-dimensional kinetic models of granular flows*, ESAIM: Mathematical Modelling and Numerical Analysis, 34 (2000), pp. 1277–1291.

[57] B. van Leer, *Towards the ultimate conservative finite difference scheme, V. A second sequel to Godunov's method*, Journal of Computational Physics, 32 (1979), pp. 101–136.

[58] J. L. Vázquez, *The Porous Media Equation*, University Press, Oxford, 2007.

[59] J. von Gathen and J. Gerhard, *Modern Computer Algebra*, Cambridge, second edition, 2003.

[60] H. Zhang, A. Sandu, and S. Blaise, *Partitioned and implicit-explicit general linear methods for ordinary differential equations*, Journal of Scientific Computing, 61 (2014), pp. 119–144.

[61] H. Zhao, *A fast sweeping method for eikonal equations*, Mathematics of Computation, 74 (2005), pp. 603–627.

[62] X. Zhong, *Additive semi-implicit Runge-Kutta methods for computing high-speed nonequilibrium*, Journal of Computational Physics, 128 (1996), pp. 19–31.