

**UNIVERSIDAD DE CONCEPCION
DIRECCION DE POSTGRADO
CONCEPCION-CHILE**



**Una contribución al estudio de métodos numéricos eficientes
para algunos modelos multi-especies en una dimensión**

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

Lihki José Rubio Ortega

**FACULTAD DE CIENCIAS FISICAS Y MATEMATICAS
DEPARTAMENTO DE INGENIERIA MATEMATICA**

2017

A contribution to the study of efficient numerical methods for some multi-species models in one dimension

Lihki José Rubio Ortega

Profesor guía: Dr. Raimund Bürger, Universidad de Concepción, Chile

Co-tutor: Dr. Pep Mulet, Universitat de València, España

Director de Programa: Dr. Raimund Bürger, Universidad de Concepción, Chile

COMISION EVALUADORA

Dr. _____

Dr. _____

Dr. _____

Dr. _____

Dr. Raimund Bürger
Universidad de Concepción, Chile

Dr. Pep Mulet
Universitat de València, España

*Mis padres que me dieron la vida y apoyo que siempre necesité.
Mis hermanos y Ángela mi esposa quien siempre me motivó a culminar este proceso
Cada persona que siempre me brindó apoyo en especial mis tutores de tesis.*

AGRADECIMIENTOS

Este trabajo no habría sido posible sin el apoyo y la supervisión de mis directores de tesis, profesor Raimund Bürger y Pep Mulet, les doy las gracias por aceptar trabajar conmigo, al profesor Pep le agradezco mucho su hospitalidad en mis dos estadias en la Universidad de Valencia y las importantes herramientas computacionales que me entregó para poder desarrollar esta tesis. A ambos les agradezco mucho su paciencia, apoyo y buena disposición que siempre tuvieron para compartir conmigo sus conocimientos durante todo este proceso.

A los miembros del Departamento de Ingeniería Matemática, en particular, a los profesores que orientaron los cursos y seminarios en el doctorado le doy las gracias por todo lo que me enseñaron.

A todos los compañeros y personal administrativo con los que compartí en el CI2MA, gracias por su excelente compañerismo durante todos estos años. A su director Gabriel Gatica por permitirme tener un espacio en el centro de investigación.

No puedo terminar sin agradecer a mi familia, en cuyo estímulo constante y amor he confiado a lo largo de mis años en la Academia. Gracias por estar siempre ahí brindándome esa voz de apoyo y motivación. A Ángela mi esposa que siempre estuvo apoyándome en los momentos mas difíciles. Es a ellos a quienes dedico este trabajo.

El desarrollo de esta tesis y las estadias en el Departamento de Matemática Aplicada de la Universidad de Valencia fueron financiadas gracias al apoyo de la Red Doctoral REDOC.CTA y CONICYT(Chile), project PAI-MEC, folio 80150006.

Lihki José Rubio Ortega.

Contents

List of Figures	vi
List of Tables	ix
Abstract	xi
Resumen	xii
1 Introduction	1
1.1 Multi-species kinematic flow models	1
1.2 Settling of dispersions of droplets and colloidal particles	3
1.3 Linearly implicit IMEX schemes for the equilibrium dispersive model of chromatography	5
2 Introducción	6
2.1 Modelos de flujo cinemático multiespecies	6
2.2 Asentamiento de dispersiones de gotas y partículas coloidales	8
3 Polynomial viscosity matrix (PVM) schemes for Multi-Species Kinematic flow Models	12
3.1 Introduction	12
3.1.1 Scope	12
3.1.2 Related work	13
3.1.3 Multi-Species kinematic flow models	14
3.1.4 Polydisperse sedimentation models	15
3.1.5 Multiclass Lighthill-Whitham-Richards (MCLWR) traffic model	17
3.2 Numerical schemes	18

3.2.1	Roe matrices	18
3.2.2	Polynomial viscosity matrices	19
3.2.3	Some PVM methods	20
3.3	PVM methods for kinematic flow models	26
3.3.1	MUSCL extrapolation	28
3.4	Numerical results	29
3.4.1	Preliminaries	29
3.4.2	Examples 3.1, 3.2 and 3.3 (MLB model, $N = 2, 4, 11$)	30
3.4.3	Example 3.4 (MCLWR model, $N = 9$)	31
3.4.4	Efficiency plots	38
3.5	Conclusions of Chapter 3	39
4	Implicit-Explicit Methods for the efficient simulation of the settling of dispersions of droplets and colloidal particles	40
4.1	Introduction	40
4.1.1	Scope	40
4.1.2	Outline of this chapter	41
4.2	Governing model	42
4.3	Numerical methods	43
4.3.1	Spatial discretization	43
4.3.2	Time discretization	45
4.4	Numerical results	49
4.4.1	Preliminaries	49
4.4.2	Examples 4.1 and 4.2.	49
4.5	Conclusions of Chapter 4	54
5	Linearly implicit IMEX schemes for the equilibrium dispersive model of chromatography	57
5.1	Introduction	57
5.1.1	Scope	57
5.1.2	The Equilibrium Dispersive (ED) model of chromatography	60
5.2	Mathematical structure of the ED model	61
5.3	Linearly implicit IMEX schemes	64
5.3.1	Method of lines approach	64
5.4	Numerical Examples	68
5.4.1	Preliminaries	68

Contents

5.4.2	Three-component displacement tests	69
5.4.3	Numerical error and efficiency	71
5.5	Conclusions of Chapter 5	86
6	Conclusions and outlook	87
7	Conclusiones y trabajos futuros	90

List of Figures

3.1	Function $V(\phi)$ defined by (3.7).	17
3.2	Function $P_0(x) = S_0$ corresponding to PVM-0(S_0) method.	21
3.3	Function $P_1(x) = \alpha_0 + \alpha_1 x$ corresponding to PVM-1(S_L, S_R) method.	22
3.4	Function $P_2(x) = \alpha_0 + \alpha_2 x^2$ corresponding to PVM-2(S_0) method.	23
3.5	Function $P_4(x) = \alpha_0 + \alpha_2 x^2 + \alpha_4 x^4$ corresponding to PVM-4(S_0) method.	24
3.6	Oscillatory numerical solution produced by method PVM-4(S_M, S_I) with $N = 11$ species and $cfl = 0,5$. The computational domain is subdivided into $M = 400$ cells. (a) Snapshot of the numerical solution, (b) enlarged view.	25
3.7	Comparison of P_4 , defined by (3.18), (3.19), for $ S_M = A = 1$ and different choices of $S_I = B$	26
3.8	Example 3.1 (MLB model, $N = 2$): numerical solution for ϕ, ϕ_1, ϕ_2 at $T = 50$ s computed (a) by SPEC-INT method with $M = M_{\text{ref}}=12800$ (reference solution), (b, c, d) by PVM methods with $M = 1600$, including the reference solution.	34
3.9	Example 3.2 (MLB model, $N = 4$): numerical solution for $\phi, \phi_1, \phi_2, \phi_3, \phi_4$ at $T = 50$ s computed (a) by SPEC-INT with $M_{\text{ref}}=12800$ (reference solution), (b, c, d, e, f) by PVM methods with $M = 1600$, including the reference solution.	35
3.10	Example 3.3 (MLB model, $N = 11$): numerical solution for $\phi, \phi_1, \dots, \phi_{11}$ at $T = 50$ s (a) computed by SPEC-INT with $M_{\text{ref}}=12800$ (reference solution), (b, c, d, e, f) by PVM methods with $M = 1600$, including the reference solution.	36

3.11	Example 3.4 (MCLWR model, $N = 9$): numerical solution (a) for ϕ and (b, c, d, e, f) for selected classes at $T = 0,015$ h computed by SPEC-INT with $M_{\text{ref}}=12800$ (reference solution), and by PVM methods with $M = 1600$, including the reference solution.	37
3.12	Examples 3.1 to 3.4: (MLB models $N = 2, 4, 11$, MCLWR model $N = 9$): Efficiency plot based on numerical solutions for $\Delta x = 1/M$ with $M = 100, 200, 400, 800$ and 1600 : (a) Example 3.1, (b) Example 3.2, (c) Example 3.3, (d) Example 3.4.	38
4.1	Examples 4.1 and 4.2: numerical solution for ϕ at the indicated times for (a) Example 4.1, (b) Example 4.2.	50
4.2	Example 4.1: numerical solution for $\phi_1, \dots, \phi_8$ at (a) $t = 30$ s, (b) $t = 200$ s, (c) $t = 300$ s, (d) $t = 1000$ s computed by PVM4-LIMEX with $M_{\text{ref}}=12800$	53
4.3	Example 4.2: numerical solution for $\phi_1, \dots, \phi_8$ at (a) $t = 30$ s, (b) $t = 100$ s, (c) $t = 200$ s, (d) $t = 1000$ s computed by PVM4-LIMEX with $M_{\text{ref}}=12800$	54
4.4	Examples 4.1 and 4.2: efficiency plots (approximate L^1 error versus CPU time) for (a, c, e) Example 4.1, (b, d, f) Example 4.2 at (a, b) $t = 30$ s, (c, d) $t = 200$ s, (e, f) $t = 1000$ s.	55
5.1	Physical Process of Liquid Chromatography. Source: www.pharmaguideline.com	59
5.2	Eigenvalues of the Jacobian matrix $\mathbf{W}$	62
5.3	Positive roots of the rational function $R_{\mathbf{w}}(y)$	63
5.4	Reference numerical solution obtained by IMEX-RK2 at final times $T = 1, 4, 8, 12$ for (a, b) Example 5.1, (c, d) Example 5.2 and (e, f) Example 5.3.	70
5.5	Example 5.1: numerical solutions obtained by IMEX-RK2 and explicit methods with $D_a = 10^{-4}$ at final time $T = 4$. The spatial resolution corresponds to $M = 400$, results are labeled by “IMEX” and “EX”, and the reference solution (“REF-SOL”) is defined by $M_{\text{ref}} = 12800$	72
5.6	Example 5.1: numerical solutions for $D_a = 10^{-4}$ and final time $T = 12$. The spatial resolution corresponds to $M = 400$, results are labeled by “IMEX” and “EX”, and the reference solution (“REF-SOL”) is defined by $M_{\text{ref}} = 12800$	79

5.7	Example 5.2: numerical solutions for $D_a = 10^{-4}$ and final time $T = 4$. The spatial resolution corresponds to $M = 400$, results are labeled by “IMEX” and “EX”, and the reference solution (“REF-SOL”) is defined by $M_{\text{ref}} = 12800$	80
5.8	Example 5.2: numerical solutions for $D_a = 10^{-4}$ and final time $T = 12$. The spatial resolution corresponds to $M = 400$, results are labeled by “IMEX” and “EX”, and the reference solution (“REF-SOL”) is defined by $M_{\text{ref}} = 12800$	81
5.9	Example 5.3: numerical solutions for $D_a = 10^{-4}$ and final time $T = 4$. The spatial resolution corresponds to $M = 400$, results are labeled by “IMEX” and “EX”, and the reference solution (“REF-SOL”) is defined by $M_{\text{ref}} = 12800$	82
5.10	Example 5.3: numerical solutions for $D_a = 10^{-4}$ and final time $T = 12$. The spatial resolution corresponds to $M = 400$, results are labeled by “IMEX” and “EX”, and the reference solution (“REF-SOL”) is defined by $M_{\text{ref}} = 12800$	83
5.11	Examples 5.1, 5.2, 5.3: Efficiency plots (approximate L^1 error versus CPU time). The final is $T = 4$. Left: $D_a = 10^{-4}$, right: $D_a = 10^{-3}$. . .	84
5.12	Examples 5.1, 5.2, 5.3: Efficiency plots (approximate L^1 error versus CPU time). The final is $T = 12$. Left: $D_a = 10^{-4}$, right: $D_a = 10^{-3}$. . .	85

List of Tables

3.1	Example 3.3 (MLB model, $N = 11$): initial concentrations ϕ_i^0 , diameters D_i and normalized particle sizes δ_i [71].	30
3.2	Examples 3.1, 3.2: approximate L^1 errors (e_M^{tot} , figures to be multiplied by 10^{-3}), convergence rates (θ_M) and CPU times (cpu). The Roe matrix was approximated by the midpoint rule (3.15).	31
3.3	Examples 3.3, 3.4: approximate L^1 errors (e_M^{tot} , figures to be multiplied by 10^{-3}), convergence rates (θ_M) and CPU times (cpu). The Roe matrix was approximated by the midpoint rule (3.15).	32
3.4	Example 3.3: approximate L^1 errors (e_M^{tot} , figures to be multiplied by 10^{-3}), convergence rates (θ_M) and CPU times (cpu) for approximation of the Roe matrix by Gaussian quadrature with two nodes (3.22) or three nodes (3.23).	33
4.1	Examples 4.1 and 4.2: particle diameters d_i and initial concentrations ϕ_i^0	50
4.2	Approximate L^1 errors (e_M^{tot} , figures to be multiplied by 10^6), convergence rates (θ_M) and CPU times (cpu). We use PVM4 with $B = 0.3873$ for to approximate the convective term and Finite Difference for diffusive term.	51
4.3	Approximate L^1 errors (e_M^{tot} , figures to be multiplied by 10^6), convergence rates (θ_M) and CPU times (cpu). We use PVM4 with $B = 0.3873$ to approximate the convective term.	51
4.4	Example 4.2: Approximate L^1 errors (e_M^{tot} , figures to be multiplied by 10^{-5}), convergence rates (θ_M) and CPU times (cpu). We use PVM4 with $B = 0.3873$ to approximate the convective term.	52
4.5	Example 4.2: Approximate L^1 errors (e_M^{tot} , figures to be multiplied by 10^{-5}), convergence rates (θ_M) and CPU times (cpu). We use PVM4 with $B = 0.3873$ to approximate the convective term.	52

5.1	Approximate L^1 errors (e_M^{tot} , figures to be multiplied by 10^{-3}), convergence rates (θ_M) and CPU times (cpu). We use Upwind Method to approximate the convective term and Finite Difference for diffusive term. The diffusion coefficient used was $D_a = 10^{-4}$	73
5.2	Approximate L^1 errors (e_M^{tot} , figures to be multiplied by 10^{-3}), convergence rates (θ_M) and CPU times (cpu). We use Upwind Method to approximate the convective term and Finite Difference for diffusive term. The diffusion coefficient used was $D_a = 10^{-4}$	74
5.3	Approximate L^1 errors (e_M^{tot} , figures to be multiplied by 10^{-3}), convergence rates (θ_M) and CPU times (cpu). We use Upwind Method to approximate the convective term and Finite Difference for diffusive term. The diffusion coefficient used was $D_a = 10^{-4}$	75
5.4	Approximate L^1 errors (e_M^{tot} , figures to be multiplied by 10^{-3}), convergence rates (θ_M) and CPU times (cpu). We use Upwind Method to approximate the convective term and Finite Difference for diffusive term. The diffusion coefficient used was $D_a = 10^{-3}$	76
5.5	Approximate L^1 errors (e_M^{tot} , figures to be multiplied by 10^{-3}), convergence rates (θ_M) and CPU times (cpu). We use Upwind Method to approximate the convective term and Finite Difference for diffusive term. The diffusion coefficient used was $D_a = 10^{-3}$	77
5.6	Approximate L^1 errors (e_M^{tot} , figures to be multiplied by 10^{-3}), convergence rates (θ_M) and CPU times (cpu). We use Upwind Method to approximate the convective term and Finite Difference for diffusive term. The diffusion coefficient used was $D_a = 10^{-3}$	78

Abstract

High order numerical methods are developed in this work to approximate the solution of multi-species kinematic flow models and convection-diffusion systems. Specifically, numerical schemes are proposed for vehicular traffic models, polydisperse sedimentation and liquid chromatography. The following are the objectives that are pursued in this thesis. The figures with the exception of Figure 5.1 are own preparation.

The first objective of this thesis is to show the advantages in terms of efficiency that the PVM (Polynomial Viscosity Matrix) method presents when simulating multi-species kinematic flow models under a suitable selection of the approximate viscosity of Roe's numerical flux by using different types of Gaussian integration.

The second objective of this thesis is to propose a new PVM method which does not present oscillations when the viscosity matrix is approximated by using fourth degree polynomials, as for example in the Masliyah-Lockett-Bassoon (MLB) model when taking a large number of species and the second eigenvalue is very close to zero with respect to the largest eigenvalue. This model arises from the continuity and linear momentum balance equations for the solid species and the fluid, for equal-density particles the velocities of this model are given by (3.5).

The third objective of this work is to use the techniques mentioned above to approximate the convective term of convection-diffusion systems in the simulation of sedimentation of droplets of different diameters dispersed in a viscous fluid. It is proposed to use Runge-Kutta Implicit-Explicit schemes to obtain an efficient solution of these convection-diffusion systems.

Finally we use Runge-Kutta Implicit-Explicit schemes with the method of lines approach to efficiently obtain approximate solutions of a liquid chromatography model, which is a powerful tool for separation of complex mixtures.

Keywords. PVM (Polynomial viscosity matrix), Roe matrix, Conservation law systems, Multi-species kinematic flow models, Polydispersed sedimentation, Vehicle traffic models.

Resumen

En este trabajo se desarrollan métodos numéricos de alto orden para aproximar la solución de modelos de flujo cinemático multiespecies y sistemas de convección-difusión. Precisamente, se plantean esquemas numéricos para los modelos de tráfico vehicular, sedimentación polidispersa y cromatografía de líquidos. Los siguientes son los objetivos que nos planteamos en esta tesis. Las figuras con la excepción de Figure 5.1 son de elaboración propia.

El primer objetivo de esta tesis es mostrar las ventajas en términos de eficiencia que el método PVM (Polynomial Viscosity Matrix) presenta al simular modelos de flujo cinemático multiespecies bajo una selección adecuada de la forma en que aproximamos la viscosidad del flujo numérico de Roe usando diferentes tipos de integración Gaussiana.

El segundo objetivo de esta tesis es proponer un nuevo método PVM el cual no presente oscilaciones cuando aproximamos la matriz de viscosidad usando polinomios de grado cuatro, como por ejemplo en el modelo de Masliyah-Lockett-Bassoon (MLB) al tomar un gran número de especies y cuando el segundo autovalor es muy cercano a cero con respecto al mayor autovalor. Este modelo surge de las ecuaciones de balance del momento lineal y de continuidad para la especie sólida y el fluido respectivamente, para partículas de igual densidad las velocidades de este modelo están dadas por (3.5).

El tercer objetivo de este trabajo es utilizar las técnicas mencionadas arriba para aproximar el termino convectivo de sistemas de convección-difusión en la simulación de sedimentación de gotas de diferentes diámetros dispersas en un fluido viscoso. Se propone utilizar esquemas Linealmente Implícitos-Explícitos Runge-Kutta para obtener una solución eficiente de estos sistemas de convección-difusión.

Finalmente hacemos uso del método de líneas y de esquemas Linealmente Implícitos-Explícitos Runge-Kutta para obtener una solución rápida de un modelo de cromatografía líquida, la cual es una poderosa herramienta para la separación de mezclas complejas.

Palabras claves. PVM (Polynomial viscosity matrix), Matriz de Roe, Sistemas de leyes de conservación, Modelos de flujo cinemático multiespecies, Sedimentación

RESUMEN

polidispersa, Modelos de trafico vehicular

Chapter 1

Introduction

1.1 Multi-species kinematic flow models

Multi-species kinematic flow models are defined by systems of strongly coupled, non-linear first-order conservation laws. They arise in various applications including sedimentation of polydisperse suspensions and multiclass vehicular traffic. The velocity v_i of species $i \in \{1, \dots, N\}$ is an explicit function of the vector of the unknown concentrations

$$\Phi(x, t) = (\phi_1(x, t), \dots, \phi_N(x, t))^T \in \mathbb{R}^N,$$

where t is time and x is the spatial coordinate. Multi-species kinematic flow models are given by systems of conservation laws of the type

$$\partial_t \Phi + \partial_x \mathbf{f}(\Phi) = \mathbf{0}, \quad \mathbf{f}(\Phi) := (f_1(\Phi), \dots, f_N(\Phi))^T, \quad (1.1)$$

$$f_i(\Phi) := \phi_i v_i(\Phi), \quad i = 1, \dots, N, \quad (1.2)$$

In this work we consider two applications of these models.

Polydisperse suspensions model. Polydisperse suspensions consist of small solid particles that belong to a number N of species that differ in size, and which are dispersed in a viscous fluid. The diameters are assumed to be ordered as

$$D_1 > D_2 > \dots > D_N.$$

The sedimentation of such a mixture of given initial concentration $\Phi_0(x)$ in a column of depth L can be then described by (1.1), where ϕ_i denotes the local volume fraction of particle species i having diameter D_i , supplied with the initial condition

$$\Phi(x, 0) = \Phi_0(x), \quad x \in (0, L) \quad (1.3)$$

and zero-flux boundary conditions

$$\mathbf{f}|_{x=0} = \mathbf{f}|_{x=L} = \mathbf{0}, \quad t > 0. \quad (1.4)$$

Multiclass Lighthill-Whitham-Richards (MCLWR) traffic model. The LWR kinematic traffic model describes the evolution of vehicle density $\phi(x, t)$ by a scalar conservation law

$$\partial_t \phi + \partial_x (\phi v(\phi)) = 0,$$

where the velocity function $v(\phi)$ is nonnegative and nonincreasing with respect to density ($v' \leq 0$). In [4, 76] this model is generalized to multiple classes of drivers, with individual densities $\phi_i(x, t)$ ($i = 1, \dots, N$) evolving by LWR-like equations

$$\partial_t \phi_i + \partial_x f_i(\phi_1, \dots, \phi_N) = 0, \quad f_i(\phi_1, \dots, \phi_N) = \phi_i v_i(\phi), \quad (1.5)$$

coupled via $\phi = \phi_1 + \dots + \phi_N$.

The numerical approximation of multi-species kinematic flow models is a challenge since the eigenvalues and eigenvectors of the corresponding flux Jacobian matrix have no closed algebraic form. It is demonstrated that a recently introduced class of fast first-order finite volume solvers, called PVM (polynomial viscosity matrix) methods [23], can be adapted to multi-species kinematic flows. PVM methods have the advantage that they only need some information about the eigenvalues of the flux Jacobian $\mathcal{J}_{\mathbf{f}}(\Phi)$, and no spectral decomposition of a Roe matrix is needed. In fact, the so-called interlacing property (of eigenvalues with known velocity functions), which holds for several important multi-species kinematic flow models, provides sufficient information for the implementation of PVM methods. Several variants of PVM methods (differing in polynomial degree and the underlying quadrature formula to approximate the Roe matrix) are compared by numerical experiments. It turns out that PVM methods are competitive in accuracy and efficiency with several existing methods, including the HLL method of Harten, Lax and van Leer [44] and a spectral WENO (weighted ENO) reconstruction procedure [15] that is based on the same interlacing property.

The corresponding HLL intercell flux is given by

$$\hat{\mathbf{f}}_{i+1/2} = \begin{cases} \mathbf{f}_L & \text{if } S_L \geq 0, \\ \frac{1}{S_R - S_L} (S_R \mathbf{f}_L - S_L \mathbf{f}_R + S_R S_L (\Phi_R - \Phi_L)) & \text{if } S_L \leq 0 \leq S_R, \\ \mathbf{f}_R & \text{if } S_R \leq 0. \end{cases}$$

given the interlacing property to approximate the speeds S_L, S_R we can obtain the corresponding HLL intercell flux.

A component-wise WENO5 scheme is defined by the numerical flux

$$\hat{f}_{j+1/2,k} = \mathcal{R}^+ (f_{j-2,k}^+, \dots, f_{j+2,k}^+; x_{j+1/2}) + \mathcal{R}^- (f_{j-1,k}^-, \dots, f_{j+3,k}^-; x_{j+1/2})$$

where $\mathcal{R}$ is the mapped WENO5 reconstructor operator [45] and $\mathbf{f}_{j,k}^\pm$ are given by the global Lax-Friedrichs flux splitting

$$(f_{j,1}^\pm, \dots, f_{j,N}^\pm)^\top = \mathbf{f}(\Phi_j) \pm \alpha \Phi_j, \quad j \in \mathbb{Z}$$

with α defined in (4.12).

In **Chapter 3** the PVM method for the two kinematic flow models mentioned above is applied with the MUSCL technique [74] to obtain second-order schemes. The approximate solutions thus obtained are compared with those provided by a scheme that uses the eigenstructure of the Jacobian matrix.

The contents of **Chapter 3** correspond to the following published article [18]:

- Bürger R, Mulet P, Rubio L. Polynomial viscosity methods for multi-species kinematic flow models. *Numerical Methods for Partial Differential Equations* 2016; **32**:1265–1288.

1.2 Settling of dispersions of droplets and colloidal particles

The settling of a dispersion of droplets [1, 66] or that of a suspension of colloidal solid particles [28] dispersed in a fluid can be modelled by systems of convection-diffusion equations of the form

$$\partial_t \Phi + \partial_x \mathbf{f}(\Phi) = \partial_x (\mathbf{B}(\phi) \partial_x \Phi), \quad 0 < x < L, \quad t > 0, \quad (1.6)$$

where

$$\Phi(x, t) = (\phi_1(x, t), \dots, \phi_N(x, t))^\top$$

is the sought unknown, namely the vector of volume fractions of droplets or particles of class i , and $\phi = \phi_1 + \dots + \phi_N$. Here we assume that particles of class i have size D_i , where

$$D_1 > D_2 > \dots > D_N$$

Thus, the corresponding settling velocities v_i of individual particles of class i in an unbounded fluid satisfy $v_1 > v_2 > \dots > v_N$. Moreover, we assume that

$$\mathbf{f}(\Phi) = V(\phi)(v_1\phi_1, \dots, v_N\phi_N)^T,$$

where $V(\phi)$ is a given function, and that $\mathbf{B}(\phi)$ is a given diffusion matrix that depends on the total concentration ϕ of the disperse phase.

IMEX-RK methods. Due to the diffusion term in the equations, an explicit solver for the system (4.10) would require $\Delta t \sim \Delta x^2$, a time step much smaller than what it would be required by accuracy alone. An alternative is represented by semi-implicit IMEX-RK schemes, where the convective term is treated explicitly and the diffusive term is treated implicitly. The corresponding pair of Butcher arrays of IMEX-RK methods is given by

$$\begin{array}{c|c} \tilde{\mathbf{c}} & \tilde{\mathbf{A}} \\ \hline & \tilde{\mathbf{b}}^T \end{array} \quad \begin{array}{c|c} \mathbf{c} & \mathbf{A} \\ \hline & \mathbf{b}^T \end{array}, \quad (1.7)$$

where the $s \times s$ lower triangular matrices $\tilde{\mathbf{A}} = (\tilde{a}_{ij})$ (with $\tilde{a}_{ij} = 0$ for all $j \geq i$) and $\mathbf{A} = (a_{ij})$ are the matrices of the explicit and implicit parts of the method, respectively, while

$$\tilde{\mathbf{b}} = (\tilde{b}_1, \dots, \tilde{b}_s)^T, \quad \tilde{\mathbf{c}} = (\tilde{c}_1, \dots, \tilde{c}_s)^T, \quad \mathbf{b} = (b_1, \dots, b_s)^T \text{ and } \mathbf{c} = (c_1, \dots, c_s)^T$$

are s -dimensional vectors of real coefficients, and $\tilde{\mathbf{c}}$ and $\mathbf{c}$ are given by the usual relations

$$\tilde{c}_i = \sum_{j=1}^{i-1} \tilde{a}_{ij}, \quad c_i = \sum_{j=1}^i a_{ij}, \quad i = 1, \dots, s.$$

In **Chapter 4** techniques for the efficient approximate solution of systems of convection-diffusion partial differential equations modelling the sedimentation of droplets of different sizes in a viscous fluid are introduced. These techniques comprise the use of Polynomial Viscosity Matrix (PVM) methods for the convective numerical fluxes and implicit treatment of the nonlinear diffusion terms.

The contents of **Chapter 4** correspond to the following article:

- Bürger R, Mulet P, Rubio L. Implicit-Explicit methods for the efficient simulation of the settling of dispersions of droplets and colloidal particles. Submitted to *Advances in Applied Mathematics and Mechanics*.

1.3 Linearly implicit IMEX schemes for the equilibrium dispersive model of chromatography

Numerical schemes for the nonlinear equilibrium dispersive (ED) model for chromatographic processes with adsorption isotherms of Langmuir type are proposed. This model consists of a system of nonlinear, convection-dominated partial differential equations. The nonlinear convection gives rise to sharp moving transitions between concentrations of different solute components. This property calls for numerical methods with shock capturing capabilities. Based on results by [33], conservative, shock capturing, numerical schemes can be designed for this chromatography model. Since explicit schemes for diffusion problems can pose severe stability restrictions on the time step, the novel schemes treat diffusion implicitly and convection explicitly. To avoid the need to solve the nonlinear systems appearing in the implicit treatment of the nonlinear diffusion, second-order linearly implicit implicit-explicit (IMEX) Runge-Kutta schemes are employed. Numerical experiments demonstrate that the schemes give accurate numerical solutions with the same stability restrictions as in the purely hyperbolic case. In **Chapter 5** techniques for the efficient approximate solution of systems of convection-diffusion partial differential equations modelling the equilibrium dispersive model of chromatography are introduced. These techniques comprise the use of the method of lines approach with an implicit treatment of the nonlinear diffusion terms.

The contents of **Chapter 5** correspond to the article:

- Bürger R, Mulet P, Rubio L, Sepúlveda M. Linearly implicit IMEX schemes for the equilibrium dispersive model of chromatography. Pre-print [20] CI²MA submitted to *Applied Mathematics and Computation*.

Capítulo 2

Introducción

2.1 Modelos de flujo cinemático multiespecies

Modelos de flujo cinemático multiespecies son definidos por sistemas de leyes de conservación de primer orden no lineales fuertemente acopladas. Surgen en diversas aplicaciones, entre ellas la sedimentación de suspensiones polidispersas y tráfico vehicular de múltiples clases. La velocidad v_i de la especie $i \in \{1, \dots, N\}$ es una función explícita del vector de concentraciones desconocido

$$\Phi(x, t) = (\phi_1(x, t), \dots, \phi_N(x, t))^T \in \mathbb{R}^N,$$

donde t es el tiempo y x es la coordenada espacial. Modelos de flujo cinemático multiespecies están dados por sistemas de leyes de conservación del tipo

$$\begin{aligned} \partial_t \Phi + \partial_x \mathbf{f}(\Phi) &= 0, \quad \mathbf{f}(\Phi) := (f_1(\Phi), \dots, f_N(\Phi))^T, \\ f_i(\Phi) &:= \phi_i v_i(\Phi), \quad i = 1, \dots, N, \end{aligned}$$

En este trabajo consideraremos dos aplicaciones de estos modelos.

Modelo de suspensión polidispersa. Suspensiones polidispersas consisten en pequeñas partículas que pertenecen a un número N de especies que difieren en tamaño y las cuales están dispersas en un fluido viscoso. Suponemos que los diámetros están ordenados como

$$D_1 > D_2 > \dots > D_N.$$

La sedimentación de tal mezcla de concentración inicial $\Phi_0(x)$ en una columna de profundidad L puede ser descrita por (1.1), donde ϕ_i es la fracción local de volumen de la

especie i con diámetro D_i , con la condición inicial

$$\Phi(x, 0) = \Phi_0(x), \quad x \in (0, L)$$

y condiciones de borde de flujo nulo.

$$\mathbf{f}|_{x=0} = \mathbf{f}|_{x=L} = \mathbf{0}, \quad t > 0.$$

Modelo de trafico multiclase Lighthill-Whitham-Richards (MCLWR). El modelo cinemático de trafico LWR describe la evolución de la densidad de vehículos $\phi(x, t)$ mediante una ley de conservación escalar

$$\partial_t \phi + \partial_x (\phi v(\phi)) = 0,$$

donde la función velocidad $v(\phi)$ es decreciente y no negativa con respecto a la densidad ($v' \leq 0$). En [4, 76] este modelo es generalizado a múltiples clases de conductores, con densidades individuales $\phi_i(x, t)$ ($i = 1, \dots, N$) evolucionando mediante ecuaciones LWR como

$$\partial_t \phi_i + \partial_x f_i(\phi_1, \dots, \phi_N) = 0, \quad f_i(\phi_1, \dots, \phi_N) = \phi_i v_i(\phi),$$

acopladas a través de $\phi = \phi_1 + \dots + \phi_N$.

Las aproximaciones numéricas para modelos de flujo cinemático multiespecies suponen un reto debido a que los autovalores y autovectores de la correspondiente matriz Jacobiana de la función flujo no tienen una forma algebraica cerrada. Se demostró que una recientemente introducida clase de resolvedores rápidos en volúmenes finitos de primer orden, llamados métodos PVM (Matrices de viscosidad polinomiales) [23], puede ser adaptada a modelos de flujo cinemático multiespecies. Los métodos PVM tienen la ventaja de solo necesitar cotas de los autovalores de la matriz Jacobiana del flujo $\mathcal{J}_f(\Phi)$, y no la descomposición espectral de una matriz de Roe. En efecto, la llamada propiedad de interrelación (de autovalores con velocidades conocidas), la cual se satisface para varios importantes modelos de flujo cinemático multiespecies, provee suficiente información para la implementación del método PVM. Distintas variantes del método PVM (diferentes grados del polinomio y la fórmula subyacente de cuadratura para aproximar la matriz de Roe) son comparados mediante experimentos numéricos. Esto produce métodos PVM que son competitivos en precisión y eficiencia en comparación con otros métodos existentes, incluyendo el método HLL de Harten, Lax y van Leer [44] y un procedimiento de reconstrucción espectral WENO (weighted ENO) [15] espectral basado en la misma propiedad de interrelación.

El correspondiente flujo numérico del método HLL esta dado por

$$\hat{\mathbf{f}}_{i+1/2} = \begin{cases} \mathbf{f}_L & \text{si } S_L \geq 0, \\ \frac{1}{S_R - S_L} (S_R \mathbf{f}_L - S_L \mathbf{f}_R + S_R S_L (\Phi_R - \Phi_L)) & \text{si } S_L \leq 0 \leq S_R, \\ \mathbf{f}_R & \text{si } S_R \leq 0. \end{cases}$$

dada la propiedad de interrelación para aproximar las velocidades S_L, S_R podemos obtener el flujo numérico del método HLL sobre la interfaz.

Un esquema WENO5 esta definido por el flujo numérico

$$\hat{f}_{j+1/2,k} = \mathcal{R}^+ (f_{j-2,k}^+, \dots, f_{j+2,k}^+; x_{j+1/2}) + \mathcal{R}^- (f_{j-1,k}^-, \dots, f_{j+3,k}^-; x_{j+1/2})$$

donde $\mathcal{R}$ es el operador reconstructor WENO5 [45] y $\mathbf{f}_{j,k}^\pm$ están dados por la separación de flujos de Lax-Friedrichs

$$(f_{j,1}^\pm, \dots, f_{j,N}^\pm)^T = \mathbf{f}(\Phi_j) \pm \alpha \Phi_j, \quad j \in \mathbb{Z}$$

con α definido en (4.12).

En el **Capítulo 2** el método PVM para modelos de flujo cinemático multiespecies mencionado arriba es aplicado junto con la técnica MUSCL [74] para obtener esquemas de segundo orden. Las aproximaciones así obtenidas se comparan con las obtenidas con esquemas que usan la autoestructura de la matriz Jacobiana.

El contenido del **Capítulo 2** corresponde al siguiente artículo publicado [18]:

- Bürger R, Mulet P, Rubio L. Polynomial viscosity methods for multi-species kinematic flow models. *Numerical Methods for Partial Differential Equations* 2016; **32**:1265–1288.

2.2 Asentamiento de dispersiones de gotas y partículas coloidales

El asentamiento de gotas [1,66] o partículas sólidas coloidales [28] dispersas en un fluido se puede modelar mediante sistemas de ecuaciones de convección-difusión de la forma:

$$\partial_t \Phi + \partial_x \mathbf{f}(\Phi) = \partial_x (\mathbf{B}(\phi) \partial_x \Phi), \quad 0 < x < L, \quad t > 0,$$

donde el vector de fracciones de volumen de gotas o partículas de la clase i

$$\Phi(x, t) = (\phi_1(x, t), \dots, \phi_N(x, t))^T \in \mathbb{R}^N$$

es la variable desconocida y $\phi = \phi_1 + \dots + \phi_N$. Aquí asumimos que las partículas de la clase i tienen tamaños d_i , donde

$$D_1 > D_2 > \dots > D_N.$$

Por lo tanto, las correspondientes velocidades de sedimentación v_i de partículas individuales de la clase i en un fluido no acotado satisface $v_1 > v_2 > \dots > v_N$. Además suponemos que

$$\mathbf{f}(\Phi) = V(\phi)(v_1\phi_1, \dots, v_N\phi_N)^T,$$

donde $V(\phi)$ es una función dada, y que $\mathbf{B}(\phi)$ es una matriz de difusión que depende de la concentración total ϕ de la fase dispersa.

Métodos IMEX-RK. Debido al termino de difusión en la Ecuación 1.6, un resolvidor explícito para el sistema (4.10) requeriría $\Delta t \sim \Delta x^2$, un paso temporal mucho menor que el que se requeriría solo por precisión. Una alternativa es el uso de esquemas semi-implícitos IMEX-RK, donde el termino convectivo es tratado explícitamente y el termino difusivo implícitamente. El correspondiente par de matrices Butcher del método IMEX-RK está dado por

$$\begin{array}{c|c} \tilde{\mathbf{c}} & \tilde{\mathbf{A}} \\ \hline & \tilde{\mathbf{b}}^T \end{array} \quad \begin{array}{c|c} \mathbf{c} & \mathbf{A} \\ \hline & \mathbf{b}^T \end{array},$$

donde las matrices triangulares inferiores $s \times s$ $\tilde{\mathbf{A}} = (\tilde{a}_{ij})$ (con $\tilde{a}_{ij} = 0$ para todo $j \geq i$) y $\mathbf{A} = (a_{ij})$ son las matrices de la parte explícita e implícita del método respectivamente, mientras que

$$\tilde{\mathbf{b}} = (\tilde{b}_1, \dots, \tilde{b}_s)^T, \tilde{\mathbf{c}} = (\tilde{c}_1, \dots, \tilde{c}_s)^T, \mathbf{b} = (b_1, \dots, b_s)^T \text{ y } \mathbf{c} = (c_1, \dots, c_s)^T$$

son vectores s -dimensionales de coeficientes reales, y $\tilde{\mathbf{c}}, \mathbf{c}$ están dados por la relación usual

$$\tilde{c}_i = \sum_{j=1}^{i-1} \tilde{a}_{ij}, \quad c_i = \sum_{j=1}^i a_{ij}, \quad i = 1, \dots, s.$$

En el **Capítulo 3** se introducen técnicas para la aproximación eficiente de la solución de sistemas de ecuaciones en derivadas parciales de convección-difusión que modelan la sedimentación de gotas de diferentes tamaños dispersas en un fluido viscoso. Estas técnicas comprenden el uso del método de Matrices de Viscosidad Polinomiales (PVM) para el flujo numérico convectivo y tratamiento implícito de los términos de difusión no lineales.

El contenido del **Capítulo 3** corresponde al artículo:

- Bürger R, Mulet P, Rubio L. Implicit-Explicit methods for the efficient simulation of the settling of dispersions of droplets and colloidal particles. Preprint enviado a *Advances in Applied Mathematics and Mechanics*.

Esquemas linealmente implícitos IMEX para el modelo dispersivo de equilibrio para procesos de cromatografía

Se proponen esquemas numéricos para el modelo dispersivo de equilibrio no lineal (ED) para procesos cromatográficos con isotermas de adsorción de tipo Langmuir. Este modelo consta de un sistema de ecuaciones diferenciales parciales con convección no lineal dominante. La convección no lineal da lugar a transiciones abruptas y móviles entre concentraciones de diferentes componentes de soluto. Esta propiedad demanda el uso de métodos numéricos de captura de ondas de choque. Basado en los resultados de [R. Donat, F. Guerrero y P. Mulet (2017); sometido], se pueden diseñar esquemas numéricos conservativos (de captura de ondas de choque) para estos modelos de cromatografía. Dado que los esquemas explícitos para la difusión pueden plantear restricciones severas de estabilidad en el paso temporal, nuevos esquemas tratan el término difusivo de manera implícita y el término convectivo de manera explícita. Para evitar la necesidad de resolver los sistemas no lineales que aparecen en el tratamiento implícito de la difusión no lineal, se usan esquemas linealmente implícitos-explícitos (IMEX) Runge-Kutta de segundo orden. Los experimentos numéricos demuestran que estos esquemas proporcionan soluciones numéricas precisas con la misma restricción de estabilidad del caso puramente hiperbólico.

En el **Capítulo 4** se introducen técnicas para la aproximación eficiente de la solución del sistema de ecuaciones diferenciales parciales de difusión-convección del modelo dispersivo de equilibrio no lineal (ED) para procesos cromatográficos. Estas técnicas

comprenden el uso de métodos Upwind para los flujos numéricos convectivos y el tratamiento implícito de los términos de difusión no lineal.

El contenido del **Capítulo 4** corresponde al artículo:

- Bürger R, Mulet P, Rubio L, Sepúlveda M. Linearly implicit IMEX schemes for the equilibrium dispersive model of chromatography. Preprint enviado a *Applied Mathematics and Computation*.

Chapter 3

Polynomial viscosity matrix (PVM) schemes for Multi-Species Kinematic flow Models

3.1 Introduction

3.1.1 Scope

This work concerns high-resolution numerical schemes for systems of conservation laws that arise as one-dimensional multiclass kinematic flow models describing the flow of one disperse substance through a continuous phase. In many cases, the disperse substance consists of small particles of different species that differ in size or density, segregate, and create areas of different composition. Such models include a model of polydisperse sedimentation of solid-liquid suspensions with solid particles of different sizes, and also certain continuum approximations of traffic flow of vehicles on a highway with drivers having different preferential velocities (see section 1.1). Numerical methods for hyperbolic conservation laws need some sort of upwinding to be stable. For scalar conservation laws, upwinding is a well-established issue. For hyperbolic systems of conservation laws, there are at least two alternatives for upwinding: one may locally project the solution information onto characteristic fields, and then apply the usual techniques for scalar conservation laws to each field; or alternatively, one can use flux-splitting methods with a high amount of numerical viscosity. The first alternative, that yields characteristic-wise numerical schemes, gives better resolution than the second, since the

amount of numerical viscosity is much smaller. The prototype method of this kind of is Roe's method [65] (see also [24]), for which the first-order numerical flux, specified here for two adjacent states Φ_L and Φ_R to the left and right of a cell boundary, is given by

$$\hat{\mathbf{f}}(\Phi_L, \Phi_R) = \frac{1}{2} \left(\mathbf{f}(\Phi_L) + \mathbf{f}(\Phi_R) - |\mathbf{A}(\Phi_L, \Phi_R)|(\Phi_R - \Phi_L) \right), \quad (3.1)$$

where $\mathbf{A}(\Phi_L, \Phi_R)$ is a Roe matrix (related to the flux Jacobian $\mathcal{J}_{\mathbf{f}} = (\partial f_i / \partial \phi_j)_{1 \leq i, j \leq N}$) for the flux $\mathbf{f}$ and $|\mathbf{A}| \in \mathbb{R}^{N \times N}$ denotes a real diagonalizable matrix that is computed through the eigenvectors and eigenvalues of $\mathbf{A}$ (not to be confused with the matrix of absolute values of the entries of $\mathbf{A}$). On the other hand, the second alternative, which gives rise to component-wise schemes, tends to yield faster methods. Thus, the relative efficiency of one alternative with respect to the other is problem dependent. Roughly speaking, if the characteristic information (eigenstructure of $\mathcal{J}_{\mathbf{f}}$) is available in closed form, then the first alternative tends to be more efficient; otherwise, if the characteristic information requires a high computational effort, then the second may turn out to be more efficient. Some kinematic flow models (1.1), (1.2) have flux Jacobians of arbitrary size N , without readily available characteristic information, but can be proved to be hyperbolic by other means [14]. For some of these models characteristic-wise schemes are more efficient than some component-wise schemes [14, 32], essentially because the component-wise schemes used in the comparison prescribe a global numerical viscosity that tends to smear out relevant singularities of the solution.

The purpose of this chapter is to explore a class of numerical methods, so-called Polynomial Viscosity Matrix (PVM) methods [30], whose aim is to mimic Roe's method without using the characteristic information. These methods approximate $|\mathbf{A}|$ by $P(\mathbf{A})$ for some suitably defined polynomial P that approximates the function $x \mapsto |x|$ in some interval containing the eigenvalues of $\mathbf{A}$. The main difficulties that are encountered in this work are the unavailability of a Roe matrix, at least in closed form, and that the characteristic fields may not be genuinely nonlinear (for scalar conservation laws this corresponds to fluxes with inflection points).

3.1.2 Related work

The PVM methods were studied in [23]. That work includes numerous applications of PVM methods to hyperbolic systems of conservation laws with source terms or nonconservative products. These methods have the advantage that they only need

partial information on the eigenvalues of the system, and no spectral decomposition of a Roe matrix is required. We therefore propose to apply PVM methods to multi-species kinematic flow models [17], which give rise to systems of conservation laws (1.1), (1.2) that are constructed in a systematic way but for which the spectral information of $\mathcal{J}_f(\Phi)$ is, in general, not available in closed form. The best-known model of this type is the multi-class Lighthill-Whitham-Richards (MCLWR) kinematic traffic model, which extends the well-known Lighthill-Whitham-Richards (LWR) model [55, 63] to vehicles with drivers having different preferential velocities, and which was proposed by Benzoni-Gavage and Colombo [4] and Wong and Wong [76]. A very similar model, posed with zero-flux boundary conditions, describes the settling of emulsions [66]. The velocity functions v_i in these applications have an algebraically simple structure in the sense that the resulting flux Jacobian matrix $\mathcal{J}_f(\Phi)$ is a rank- m perturbation of a diagonal matrix with $m = 1$. More involved velocity functions, for which the corresponding rank is $m \geq 2$, arise in models of sedimentation of polydisperse suspensions, see [14, 15] and the references cited in these papers.

3.1.3 Multi-Species kinematic flow models

Kinematic flow models are given by systems of conservation laws (1.1) with fluxes given by (1.2) for sufficiently smooth functions v_i . However, in most applications these functions do not depend individually on each of the densities $\phi_1, \dots, \phi_N$ but rather on a small number $m \ll N$ of functions $p_1, \dots, p_m$ of $\phi_1, \dots, \phi_N$, i.e.,

$$v_i = v_i(p_1, \dots, p_m), \quad p_l = p_l(\phi_1, \dots, \phi_N), \quad i = 1, \dots, N, \quad l = 1, \dots, m, \quad m \ll N.$$

Under this assumption the Jacobian is a matrix of the form

$$\mathcal{J}_f = \mathcal{J}_f(\Phi) = \mathbf{D} + \mathbf{B}\mathbf{C}^T, \tag{3.2}$$

where $\mathbf{D} := \text{diag}(v_1, \dots, v_N)$,

$$\mathbf{B} := (b_{il}) = \left(\phi_i \frac{\partial v_i}{\partial p_l} \right), \quad \mathbf{C} := (c_{ij}) = \left(\frac{\partial p_l}{\partial \phi_j} \right), \quad 1 \leq i, j \leq N, \quad 1 \leq l \leq m. \tag{3.3}$$

The following theorem is proved in [2]. It is applied in [14, 31, 32] to prove hyperbolicity of selected multi-species kinematic flow models and for the construction of spectral numerical schemes.

Theorem 3.1.1 (Secular Equation [2, 31]) *Assume that $\mathbf{D}$ is a diagonal matrix as given by (3.2) with $v_i > v_j$ for $i < j$ and that $\mathbf{C}$ and $\mathbf{B}$ have the formats specified in (3.3). Let $\lambda \neq v_i$ for $i = 1, 2, \dots, N$. Then λ is an eigenvalue of $\mathbf{D} + \mathbf{B}\mathbf{C}^T$ if and only if*

$$R(\lambda) := \det \mathcal{M}_\lambda = 1 + \sum_{i=1}^N \frac{\gamma_i}{v_i - \lambda} = 0,$$

where $\mathcal{M}_\lambda = \mathbf{I} + \mathbf{C}^T(\mathbf{D} - \lambda\mathbf{I})^{-1}\mathbf{B}$.

The coefficients $\gamma_1, \dots, \gamma_N$ are given by the expression

$$\gamma_i = \sum_{r=1}^{\min\{N, m\}} \sum_{i \in I \in S_r^N, J \in S_r^m} \frac{\det \mathbf{C}^{I, J} \det \mathbf{B}^{I, J}}{\prod_{l \in I, l \neq i} (v_l - v_i)},$$

where

$$I := \{i_1 < \dots < i_k\} \in S_k^N \text{ and } J := \{j_1 < \dots < j_l\} \in S_l^m$$

are index sets and we denote by S_r^p the set of all (ordered) subsets of r elements taken from a set of p elements.

Corollary 3.1.1 (Interlacing property [14]) *With the notation of Theorem 3.1.1, assume that $\gamma_i \gamma_j > 0$ for $i, j = 1, \dots, N$. Then $\mathbf{D} + \mathbf{B}\mathbf{C}^T$ is diagonalizable with real eigenvalues $\lambda_1, \dots, \lambda_N$. If $\gamma_1, \dots, \gamma_N < 0$, the interlacing property*

$$M_1 := v_N + \gamma_1 + \dots + \gamma_N < \lambda_N < v_N < \lambda_{N-1} < \dots < \lambda_1 < v_1 \quad (3.4)$$

holds, while for $\gamma_1, \dots, \gamma_N > 0$, the following analogous property holds:

$$v_N < \lambda_N < v_{N-1} < \lambda_{N-1} < \dots < v_1 < \lambda_1 < M_2 := v_1 + \gamma_1 + \dots + \gamma_N$$

3.1.4 Polydisperse sedimentation models

A widely used velocity model for polydisperse sedimentation (see section 1.1) is the Masliyah-Lockett-Bassoon (MLB) model [56, 57]. This model arises from the continuity and linear momentum balance equations for the solid species and the fluid through

suitable constitutive assumptions and simplifications [16]. For equal-density particles, the velocities $v_1(\Phi), \dots, v_N(\Phi)$ are given by

$$v_i(\Phi) = C(1 - \phi)V(\phi) \left(d_i^2 - \sum_{m=1}^n \phi_m d_m^2 \right), \quad i = 1, \dots, N; \quad C = \frac{(\varrho_s - \varrho_f)gD_1^2}{18\mu_f}, \quad (3.5)$$

where $d_i := D_i/D_1$, $\phi := \phi_1 + \dots + \phi_N$, ϱ_s and ϱ_f are the solid and fluid densities, g is the acceleration of gravity, μ_f is the fluid viscosity and V is an empirical hindrance function assumed to satisfy $V(0) = 1$, $V(\phi_{\max}) = 0$ and $V'(\phi) \leq 0$ for $\phi \in [0, \phi_{\max}]$. A standard choice for $V(\phi)$ is the Richardson-Zaki expression [64]:

$$V(\phi) = \begin{cases} (1 - \phi)^{n_{\text{RZ}}-2} & \text{for } 0 < \phi < \phi_{\max}, \\ 0 & \text{otherwise,} \end{cases} \quad n_{\text{RZ}} > 2, \quad (3.6)$$

which we use in the following version with “soft cutoff” to avoid the discontinuity in the definition (3.6) for $\phi = \phi_{\max}$:

$$V(\phi) = \begin{cases} (1 - \phi)^{n_{\text{RZ}}-2} & \text{for } 0 < \phi < \phi_*, \\ \tau(\phi) & \text{for } \phi_* \leq \phi \leq \phi_{\max}, \\ 0 & \text{otherwise,} \end{cases} \quad n_{\text{RZ}} > 2, \quad (3.7)$$

where

$$\phi \mapsto \tau(\phi) := V(\phi_*) + V'(\phi_*)(\phi - \phi_*)$$

is the tangent line to $V(\phi)$ at $(\phi_*, V(\phi_*))$, with ϕ_* being chosen such that $\tau(\phi_{\max}) = 0$. From this equation, and for $V(\phi)$ defined as in (3.6) we have

$$\phi_* = \frac{(n_{\text{RZ}} - 2)\phi_{\max} - 1}{n_{\text{RZ}} - 3}.$$

The MLB model (1.1), (1.2), (3.5) is strictly hyperbolic whenever $\phi_i > 0$ and $\phi < \phi_{\max}$ [14]. The eigenvalues $\lambda_i(\Phi)$ of $\mathcal{J}_{\mathbf{f}}(\Phi)$ satisfy the interlacing property (3.4) with the lower bound

$$M_1(\Phi) = C \left(d_N^2 V(\phi) + ((1 - \phi)V'(\phi) - 2V(\phi)) \sum_{m=1}^N d_m^2 \phi_m \right). \quad (3.8)$$

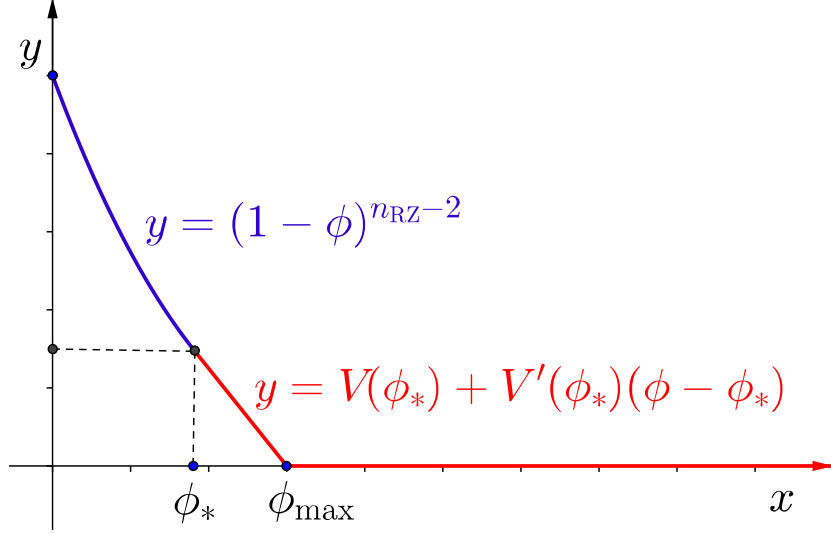


Figure 3.1: Function $V(\phi)$ defined by (3.7).

3.1.5 Multiclass Lighthill-Whitham-Richards (MCLWR) traffic model

The basic assumption of the MCLWR model (see section 1.1) is that

$$v_i(\phi) = \beta_i v(\phi), \quad i = 1, \dots, N, \quad (3.9)$$

i.e., drivers of different classes adjust their speed to the total traffic density ϕ through the same function $v(\phi)$, and β_i is the free-flowing speed of users in class i on an empty highway. The behavioral law $\phi \mapsto v(\phi)$ may be taken from standard speed-density relations like the Greenshields model $v(\phi) = 1 - \phi/\phi_{\max}$ [41], where $\phi_{\max}$ denotes a maximal car density, or the Drake model [37]

$$v(\phi) = \exp(-(\phi/\phi_{\text{opt}})^2/2), \quad (3.10)$$

where the parameter ϕ_{opt} is an “optimal” density (that maximizes $\phi v(\phi)$). It is further assumed that $\beta_1 > \dots > \beta_N > 0$. It can be seen in [32, 77] that the MCLWR model (1.1), (1.2), (3.9) is strictly hyperbolic whenever $\phi_i > 0$ and $\phi < \phi_{\max}$. The eigenvalues $\lambda_i = \lambda_i(\Phi)$ of the Jacobian $\mathcal{J}_f(\Phi)$ satisfy the interlacing property (3.4), where the lower bound is given by

$$M_1(\Phi) = v_N(\phi) + v'(\phi) \sum_{i=1}^N \beta_i \phi_i. \quad (3.11)$$

It is common in traffic flow modeling to study the evolution of vehicular traffic on a circular road of length L , so for this application (1.1), (1.2) is studied along with the initial condition (1.3) and the periodic boundary condition

$$\Phi(0, t) = \Phi(L, t), \quad t > 0. \quad (3.12)$$

3.2 Numerical schemes

3.2.1 Roe matrices

To focus on the main idea, let us concentrate for the moment on the discretization for the initial value problem of (1.1) for $x \in \mathbb{R}$. Assume that Δx and Δt are the spatial mesh width and time step, respectively, of a standard Cartesian grid on $\mathbb{R} \times [0, \infty)$. We define $x_i = (i + 1/2)\Delta x$, $i \in \mathbb{Z}$, and $t_n = n\Delta t$, $n \in \mathbb{N}_0$. Then the finite volume formulation of (1.1) is

$$\frac{d\bar{\Phi}_i(t)}{dt} = -\frac{1}{\Delta x} (\mathbf{f}(\Phi(x_{i+1/2}, t)) - \mathbf{f}(\Phi(x_{i-1/2}, t))), \quad (3.13)$$

where we define

$$\bar{\Phi}_i(t) = \frac{1}{\Delta x} \int_{x_{i-1/2}}^{x_{i+1/2}} \Phi(\xi, t) d\xi, \quad i \in \mathbb{Z}.$$

This leads to the following semi-discretization of (3.13) for $\Phi_i(t) \approx \bar{\Phi}_i(t)$:

$$\frac{d\Phi_i(t)}{dt} = -\frac{1}{\Delta x} (\hat{\mathbf{f}}_{i+1/2} - \hat{\mathbf{f}}_{i-1/2}), \quad i \in \mathbb{Z}, \quad (3.14)$$

where $\hat{\mathbf{f}}_{i+1/2}$ is a numerical flux that depends on values of Φ on a stencil around $x_{i+1/2}$. The ordinary differential equations (ODEs) (3.14) can be solved by a suitable ODE integrator (we use here Runge-Kutta methods), which gives the final fully discrete form

$$\Phi_i^{n+1} = \Phi_i^n - \frac{\Delta t}{\Delta x} (\tilde{\mathbf{f}}_{i+1/2}^n - \tilde{\mathbf{f}}_{i-1/2}^n), \quad i \in \mathbb{Z}, \quad n = 0, 1, 2, \dots,$$

with numerical fluxes $\tilde{\mathbf{f}}_{i+1/2}^n$ obtained from $\hat{\mathbf{f}}_{i+1/2}$. For the first-order version, we use Euler's method:

$$\Phi_i^{n+1} = \Phi_i^n - \frac{\Delta t}{\Delta x} (\hat{\mathbf{f}}_{i+1/2}^n - \hat{\mathbf{f}}_{i-1/2}^n), \quad i \in \mathbb{Z}, \quad n = 0, 1, 2, \dots,$$

where $\hat{\mathbf{f}}_{i+1/2}^n = \hat{\mathbf{f}}(\Phi_i^n, \Phi_{i+1}^n)$. One can obtain high-order numerical methods from suitable reconstruction techniques (MUSCL, ENO, WENO), first-order methods such as Roe's method [65], and higher-order ODE integrators.

The numerical flux function for Roe's solver is based on a matrix $\mathbf{A} = \mathbf{A}(\Phi_L, \Phi_R)$ that satisfies the following:

1. Hyperbolicity of the system: $\mathbf{A}$ is diagonalizable with real eigenvalues.
2. Consistency with the exact Jacobian:

$$\mathbf{A}(\Phi_L, \Phi_R) \rightarrow \mathcal{J}_{\mathbf{f}}(\Phi) \text{ smoothly as } \Phi_L, \Phi_R \rightarrow \Phi.$$

3. Conservation across discontinuities:

$$\mathbf{f}(\Phi_R) - \mathbf{f}(\Phi_L) = \mathbf{A}(\Phi_R - \Phi_L).$$

The numerical flux function is then given by (3.1), where the matrix $|\mathbf{A}|$ is defined through the eigendecomposition $\mathbf{A} = \mathbf{R}\mathbf{\Lambda}\mathbf{R}^{-1}$, where $\mathbf{\Lambda} := \text{diag}(\lambda_1, \dots, \lambda_N)$ and $\lambda_1, \dots, \lambda_N$ are the eigenvalues of $\mathbf{A}$, as follows:

$$|\mathbf{A}| := \mathbf{R}|\mathbf{\Lambda}|\mathbf{R}^{-1}, \quad |\mathbf{\Lambda}| = \text{diag}(|\lambda_1|, \dots, |\lambda_N|).$$

One way to guarantee that both conditions (1) and (2) are satisfied is to take

$$\mathbf{A} = \mathcal{J}_{\mathbf{f}} \left(\frac{1}{2}(\Phi_R + \Phi_L) \right). \quad (3.15)$$

Unfortunately, this simple choice of $\mathbf{A}$ does not satisfy (3) in general. Roe matrices (that satisfy (1), (2) and (3)) can be obtained for some equations, such as Euler equations for gas dynamic, shallow waters equations, etc. For the general case, Harten et al. proved in [44] that a Roe matrix exists if the system is provided with an entropy. It is known that the MCLWR model (1.5) possesses an entropy [4], but it is not clear whether the MLB model does.

3.2.2 Polynomial viscosity matrices

PVM methods [30] constitute an alternative to obtain numerical methods for hyperbolic conservation laws when the computational cost for obtaining the eigenstructure of $\mathbf{A}$ is

high. PVM methods replace $|\mathbf{A}(\Phi_L, \Phi_R)|$ in (3.1) by an approximation, based on some polynomial $P(x) = P(x; \Phi_L, \Phi_R)$,

$$|\mathbf{A}(\Phi_L, \Phi_R)| \approx P(\mathbf{A}(\Phi_L, \Phi_R)) \quad (3.16)$$

to define the numerical flux function:

$$\hat{\mathbf{f}}(\Phi_L, \Phi_R) = \frac{1}{2}(\mathbf{f}(\Phi_L) + \mathbf{f}(\Phi_R) - P(\mathbf{A})(\Phi_R - \Phi_L)). \quad (3.17)$$

Here it is understood that if

$$P(x) = \alpha_0 + \alpha_1 x + \alpha_2 x^2 + \cdots + \alpha_d x^d \text{ for } x \in \mathbb{R},$$

where d is the degree of the polynomial and $\alpha_0, \dots, \alpha_d$ are real coefficients, then

$$P(\mathbf{A}) = \alpha_0 \mathbf{I} + \alpha_1 \mathbf{A} + \alpha_2 \mathbf{A}^2 + \cdots + \alpha_d \mathbf{A}^d$$

for any square matrix $\mathbf{A}$, where $\mathbf{I}$ denotes the identity matrix.

The goal in the design of the polynomial P is to use some knowledge on the eigenvalues of $\tilde{\mathbf{A}}$ to get (3.16) as accurate as possible. The key point is to achieve that $P(x) \approx |x|$ for x in some interval $\mathcal{I}$ containing all the eigenvalues of $\mathbf{A}$. If $\mathcal{I}$ is tight, then a better approximation is obtained. But, of course, $\mathcal{I}$ should be obtained without explicit knowledge of the eigenvalues. For instance, it is easy to see that (3.16) is exact if P interpolates all the eigenvalues of $\mathbf{A}$, but this would not be of practical use. It can be seen [23] that the scheme is L^∞ -stable if

$$|x| \leq P(x; \Phi_i, \Phi_{i+1}) \leq \frac{\Delta x}{\Delta t} \quad \text{for all } x \in \mathcal{I},$$

where $\mathcal{I}$ is an interval containing the eigenvalues of $\mathbf{A}(\Phi_i, \Phi_{i+1})$.

3.2.3 Some PVM methods

The following PVM methods can be obtained from bounds on the eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_N$ of the Roe matrix (see [23]).

Rusanov method

This method is based on a bound $S_0 \geq \max_j |\lambda_j|$, so that we can take $\mathcal{I} = [-S_0, S_0]$ and the zero-degree polynomial that interpolates $|x|$ at S_0 is simply $P_0(x) = S_0$ (see Figure 3.2).

The corresponding numerical flux obtained from setting $P(\mathbf{A}) = S_0 \mathbf{I}$ in (3.17) is that of the well-known Rusanov method [68, 73].

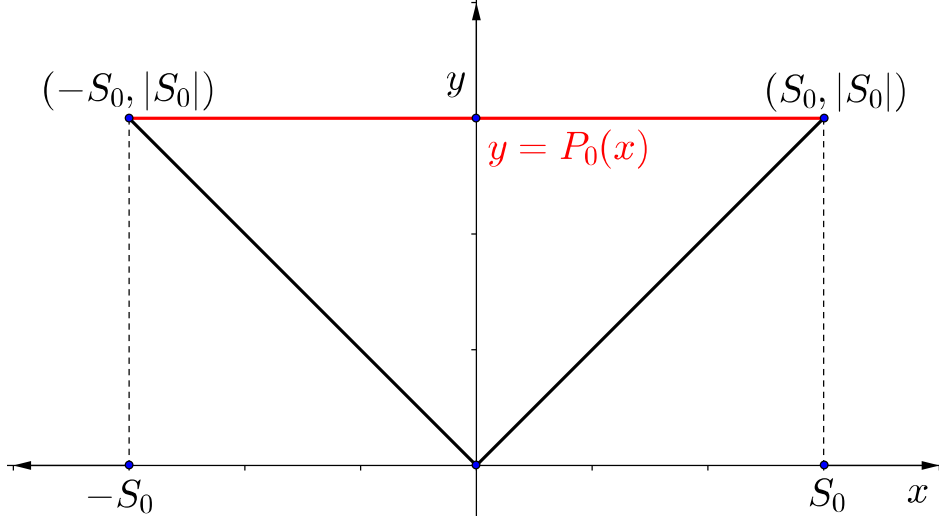


Figure 3.2: Function $P_0(x) = S_0$ corresponding to PVM-0(S_0) method.

HLL method

Assume that bounds $S_L \leq \lambda_j \leq S_R$, $j = 1, \dots, N$, are available, so that we can take $\mathcal{I} = [S_L, S_R]$, and define

$$P_1(x) = \alpha_0 + \alpha_1 x$$

to be the first-degree polynomial that interpolates $|x|$ at S_L, S_R (see Figure 3.3).

This yields the numerical flux of the known Harten, Lax and van Leer (HLL) method [44], namely

$$\hat{f}_{i+1/2}^{\text{HLL}} = \frac{1}{S_R - S_L} (S_R \mathbf{f}(\Phi_i) - S_L \mathbf{f}(\Phi_{i+1}) + S_R S_L (\Phi_{i+1} - \Phi_i)).$$

This flux can be modified when S_L and S_R have the same sign to give the upwind numerical flux, such that

$$\hat{f}_{i+1/2} = \begin{cases} \mathbf{f}(\Phi_i) & \text{if } S_L \geq 0, \\ \hat{f}_{i+1/2}^{\text{HLL}} & \text{if } S_L \leq 0 \leq S_R, \\ \mathbf{f}(\Phi_{i+1}) & \text{if } S_R \leq 0. \end{cases}$$

PVM-2(S_0) method or FORCE-type methods

This method is based on a bound $S_0 \geq \max_j |\lambda_j|$ and takes a second-degree polynomial

$$P_2(x) = \alpha_0 + \alpha_2 x^2 \text{ such that } P_2(S_0) = S_0 \text{ and } P_2'(S_0) = 1$$

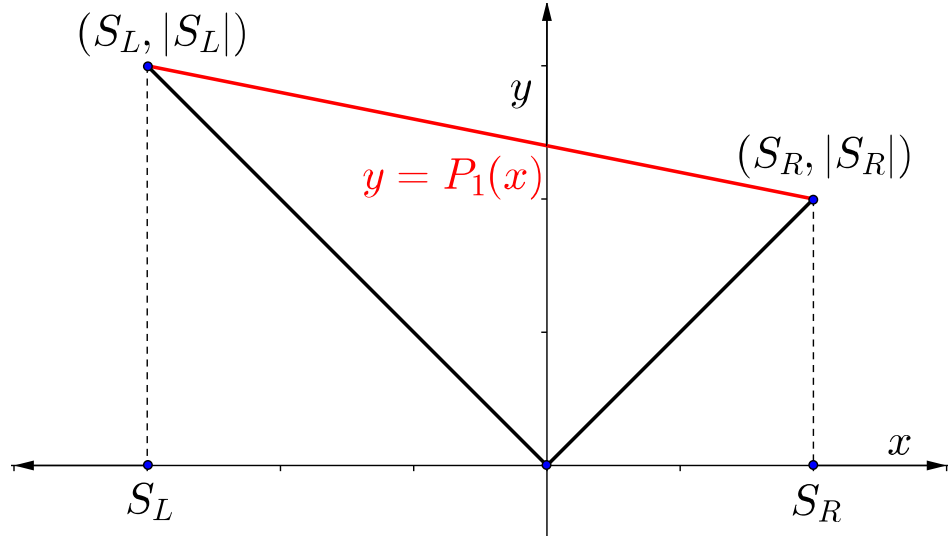


Figure 3.3: Function $P_1(x) = \alpha_0 + \alpha_1 x$ corresponding to PVM-1(S_L, S_R) method.

to obtain the numerical flux (see Figure 3.4).

PVM-2(S_M, S_m) method

This method is based on bounds $S_L \leq \lambda_j \leq S_R$, $j = 1, \dots, N$ and considers a second-degree polynomial

$$P_2(x) = \alpha_0 + \alpha_1 x + \alpha_2 x^2$$

such that

$$P_2(S_m) = |S_m|, \quad P_2(S_M) = |S_M|, \quad P_2'(S_M) = \text{sgn}(S_M)$$

where S_m , respectively, S_M are the elements with the minimum, respectively, maximum absolute value of $\{S_L, S_R\}$.

PVM-4(S_M, S_I) and PVM-4(S_0) methods

These methods are based on utilizing

$$P_4(x) = \alpha_0 + \alpha_2 x^2 + \alpha_4 x^4, \quad \text{such that } P_4(S_M) = |S_M|, P_4(S_I) = S_I \text{ and } P_4'(S_I) = 1, \quad (3.18)$$

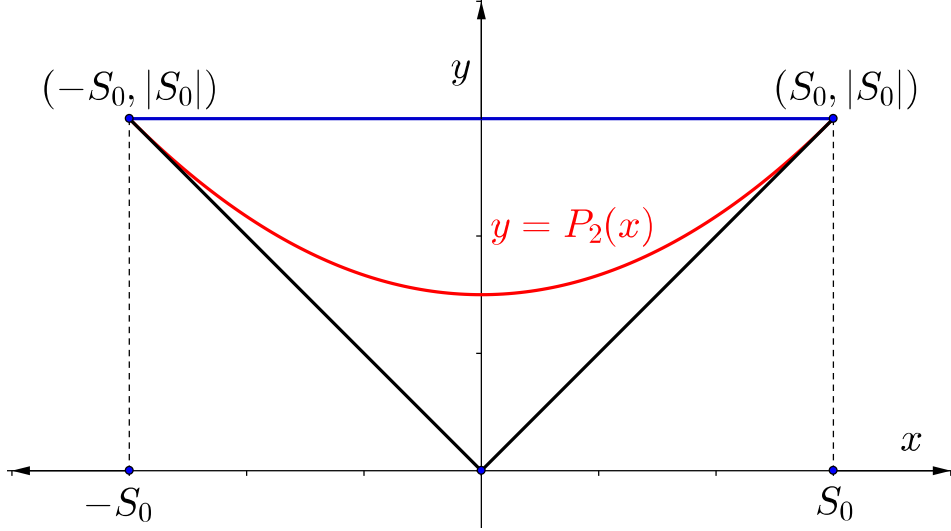


Figure 3.4: Function $P_2(x) = \alpha_0 + \alpha_2 x^2$ corresponding to PVM-2(S_0) method.

where S_M and S_I are the first and second elements of the list of eigenvalue ordered by decreasing absolute value. Then one can compute

$$\begin{aligned}\alpha_0 &= \frac{|S_M||S_I|(|S_I| + 2|S_M|)}{2(|S_I| + |S_M|)^2}, \\ \alpha_2 &= \frac{1}{2|S_M|} + \frac{|S_M|}{(|S_I| + |S_M|)^2}, \\ \alpha_4 &= -\frac{1}{2|S_M|(|S_I| + |S_M|)^2}.\end{aligned}\tag{3.19}$$

If $S_M = S_I = S_0$ then these coefficients reduce to

$$\alpha_0 = \frac{3S_0}{8}, \quad \alpha_2 = \frac{3}{4S_0}, \quad \alpha_4 = -\frac{1}{8S_0^3}.$$

This method will be denoted as PVM-4(S_0).

As we see in Figure 3.6, corresponding to a preliminary numerical experiment with $N = 11$ species MLB model, strong oscillations are generated in the region corresponding to clear liquid. These oscillations disappeared reducing the Courant number $|S_M|\Delta t/\Delta x$ to 0.01. This fact alerted us to a possible violation of the constraint

$$\frac{\|P_l\|_\infty \Delta t}{\Delta x} \leq 1,\tag{3.20}$$

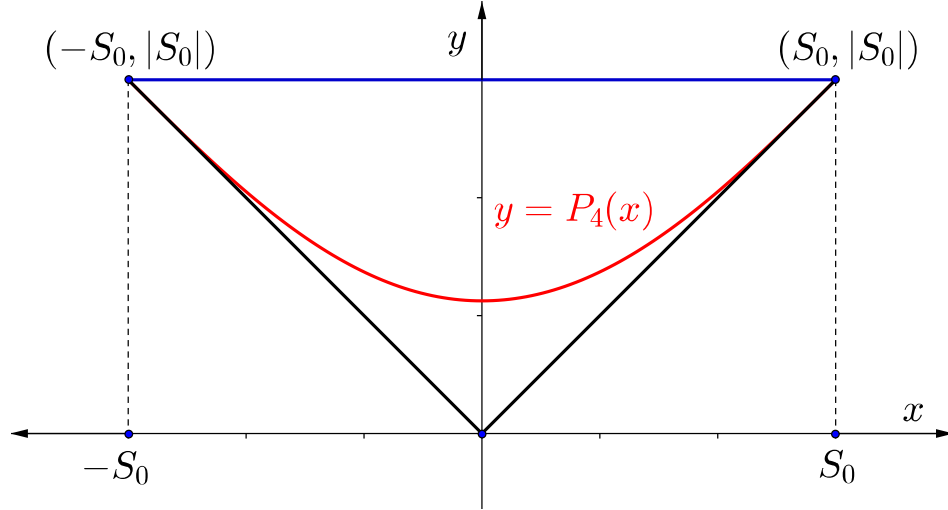


Figure 3.5: Function $P_4(x) = \alpha_0 + \alpha_2 x^2 + \alpha_4 x^4$ corresponding to PVM-4(S_0) method.

which is indeed the case, as we next detail. For notational simplicity, we write A, B instead of $|S_M|, S_I$, respectively. It can be proven that P_4 has a unique local maximum in $(0, \infty)$, given by

$$c = -\frac{\alpha_2}{2\alpha_4} = 2B^2 + (A+B)^2 \quad \text{and} \quad P_4(c) = B + \frac{(A+B)^2}{8B}.$$

If it happens that $B \ll A$ and $c \in (0, A)$, then it turns out that $P_4(c) \gg A = |S_M|$ and the restriction in (3.20) is much more severe than

$$\frac{|S_M|\Delta t}{\Delta x} \leq 1.$$

Since $B \in (0, A)$ is arbitrary we propose to set it such that

$$\max_{x \in [0, A]} P_4(x) = \max_{x \in [-A, A]} P_4(x) = P_4(A) = A,$$

which yields $B = A/3$.

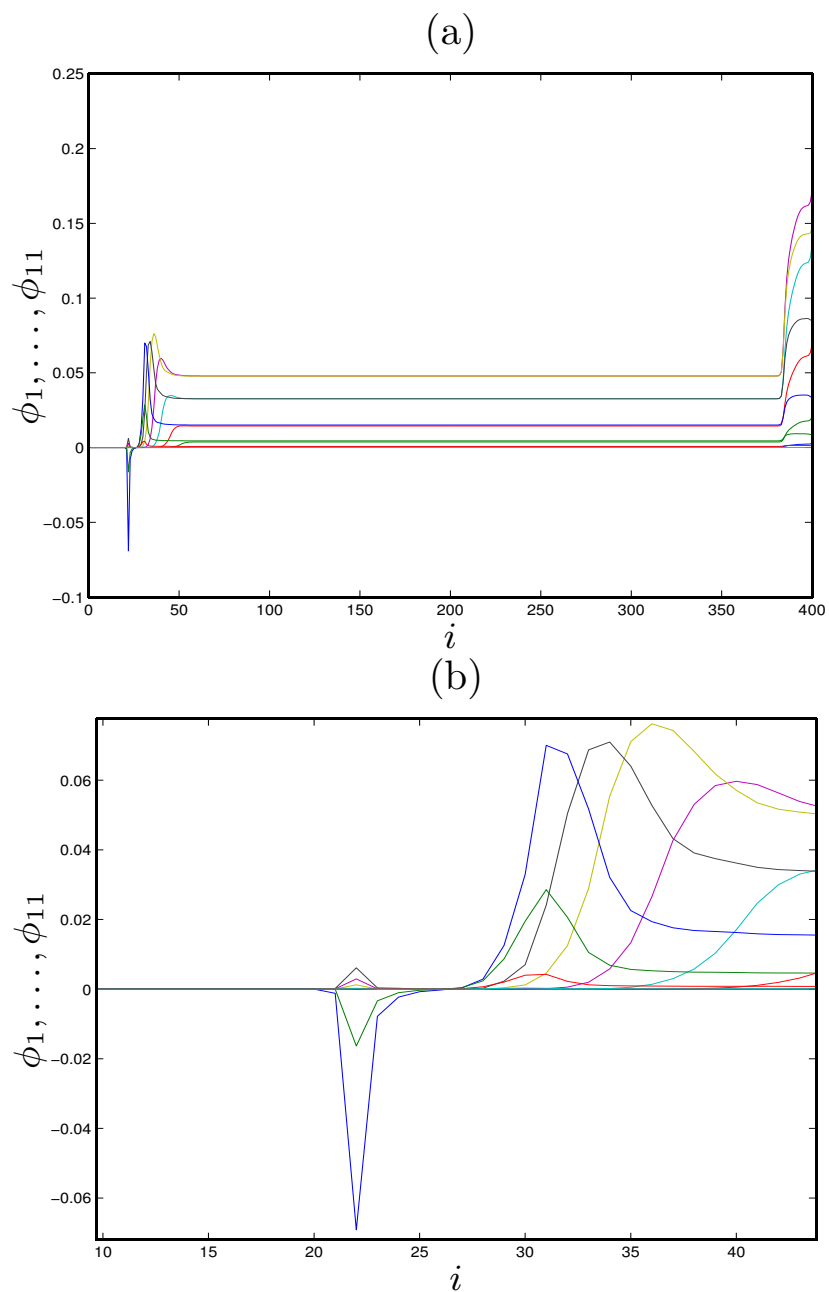


Figure 3.6: Oscillatory numerical solution produced by method PVM-4(S_M, S_I) with $N = 11$ species and $\text{cfl} = 0.5$. The computational domain is subdivided into $M = 400$ cells. (a) Snapshot of the numerical solution, (b) enlarged view.

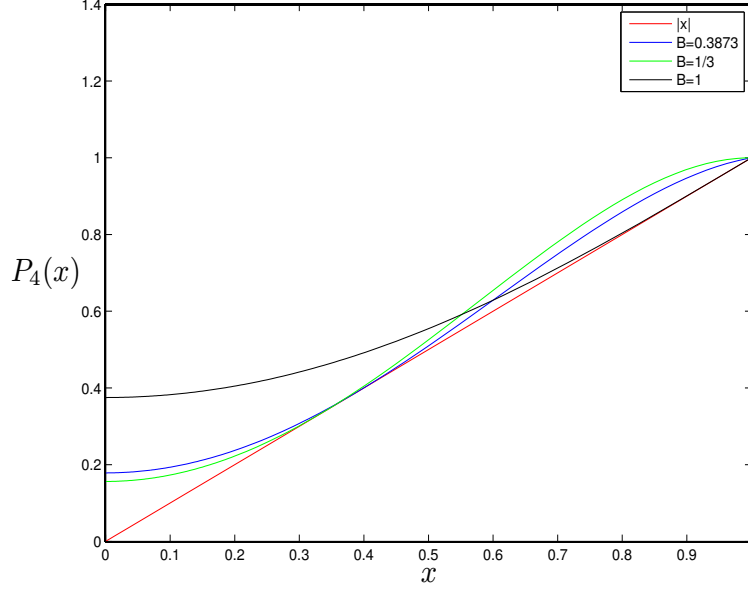


Figure 3.7: Comparison of P_4 , defined by (3.18), (3.19), for $|S_M| = A = 1$ and different choices of $S_I = B$.

We propose another option, which is considering $B \in [0, A]$ that minimizes

$$\Phi_A(B) = \int_0^A (P_4[A, B](x) - x)^2 dx.$$

Since $\lim_{B \rightarrow 0} \Phi_A(B) = \infty$, the minimum is attained either at $B = A$ or at $B \in (0, A)$ such that $\Phi'_A(B) = 0$, which can be written as:

$$0 = -16 + 33\frac{B}{A} + 66\left(\frac{B}{A}\right)^2 - 210\left(\frac{B}{A}\right)^3 + 42\left(\frac{B}{A}\right)^4 + 525\left(\frac{B}{A}\right)^5 =: q\left(\frac{B}{A}\right).$$

The only positive root of q is ≈ 0.3873 . The polynomials obtained with $A = 1$, $B = 0.3873, 1/3, 1$ appear in Figure 3.7.

3.3 PVM methods for kinematic flow models

There are specific issues about the implementation of PVM methods for kinematic flow models. The first one is the general unavailability of Roe matrices and the second one is

bounding eigenvalues. For the first issue, we propose to use approximate Roe matrices, i.e., matrices that satisfy the defining properties approximately. For this purpose, we consider the equation

$$\mathbf{f}(\Phi_R) - \mathbf{f}(\Phi_L) = \bar{\mathbf{A}}(\Phi_L, \Phi_R)(\Phi_R - \Phi_L), \quad \bar{\mathbf{A}} = \int_0^1 \mathcal{J}_{\mathbf{f}}(\Phi_L + s(\Phi_R - \Phi_L)) \, ds. \quad (3.21)$$

This definition satisfies the second and third properties of Roe matrices, but the real diagonalizability of $\mathcal{J}_{\mathbf{f}}(\Phi_L + s(\Phi_R - \Phi_L))$ does not need to carry over to $\bar{\mathbf{A}}$. Specifically, we will use matrices $\mathbf{A}$ that approximate $\bar{\mathbf{A}}$ by using the midpoint rule (3.15), a two-node Gaussian formula

$$\mathbf{A} = \frac{1}{2} \mathcal{J}_{\mathbf{f}} \left(\Phi_L + \frac{1 - 1/\sqrt{3}}{2} (\Phi_R - \Phi_L) \right) + \frac{1}{2} \mathcal{J}_{\mathbf{f}} \left(\Phi_L + \frac{1 + 1/\sqrt{3}}{2} (\Phi_R - \Phi_L) \right) \quad (3.22)$$

or a three-node Gaussian formula

$$\begin{aligned} \mathbf{A} = & \frac{5}{18} \mathcal{J}_{\mathbf{f}} \left(\Phi_L + \frac{1 - \sqrt{3}/\sqrt{5}}{2} (\Phi_R - \Phi_L) \right) + \frac{4}{9} \mathcal{J}_{\mathbf{f}} \left(\frac{\Phi_L + \Phi_R}{2} \right) \\ & + \frac{5}{18} \mathcal{J}_{\mathbf{f}} \left(\Phi_L + \frac{1 + \sqrt{3}/\sqrt{5}}{2} (\Phi_R - \Phi_L) \right) \end{aligned} \quad (3.23)$$

(at the risk of not being real diagonalizable, an issue that we have not experienced in our numerical tests).

For the second issue, in Section 3.2.3 we have not fully specified which eigenvalues are dealt with in the formulae defining each method. In any case, we can use $M_1(\Phi)$ in (3.8) or (3.11) and $M_2 = v_1(\Phi)$ to obtain the required bounds, as long as the referring matrix is a Jacobian of fluxes. But there is another point here that should be taken into account: the fluxes are not genuinely nonlinear, which in the scalar case corresponds to the flux having inflection points, and the interval where the eigenvalues lie should be considered more carefully. Our proposal is to consider not only the matrices $\mathbf{A}$, $\mathcal{J}_{\mathbf{f}}(\Phi_L)$ and/or $\mathcal{J}_{\mathbf{f}}(\Phi_R)$ but all the matrices

$$\mathcal{J}_{\mathbf{f}}(\Phi_L + s(\Phi_R - \Phi_L)), \quad s \in [0, 1]$$

to compute, for instance,

$$\min_{0 \leq s \leq 1} M_1(\Phi_L + s(\Phi_R - \Phi_L)) \quad \text{or} \quad \max_{0 \leq s \leq 1} M_2(\Phi_L + s(\Phi_R - \Phi_L))$$

as lower, respectively, upper bound for the eigenvalues.

3.3.1 MUSCL extrapolation

We use the MUSCL technique (see [74]) to obtain second-order schemes from the basic first-order PVM schemes. Following this technique, one defines the numerical flux as follows, where $\Phi = \{\Phi_i\}_{i \in \mathbb{Z}}$:

$$\hat{f}_{i+1/2} = \hat{f}_{i+1/2}(\Phi) = \hat{f}(\Phi_{i-1}, \Phi_i, \Phi_{i+1}, \Phi_{i+2}) = \hat{f}^{\text{PVM}}(\Phi_{i+1/2}^-, \Phi_{i+1/2}^+),$$

where $\hat{f}^{\text{PVM}}$ denotes one of the first-order PVM fluxes defined so far and the extrapolated solution vectors $\Phi_{i \mp 1/2}^\pm$ are defined by

$$\Phi_{i \mp 1/2}^\pm = \Phi_i \mp \frac{1}{2} \min\text{mod}(\Phi_i - \Phi_{i-1}, \Phi_{i+1} - \Phi_i),$$

where the standard minmod function is defined by

$$\min\text{mod}(a, b) = \frac{1}{2} (\text{sgn}(a) + \text{sgn}(b)) \min\{|a|, |b|\}.$$

We use Heun's method to obtain the following final fully discrete scheme, where $\Phi^n = \{\Phi_i^n\}_{i \in \mathbb{Z}}$ and $\Phi^{(k)} = \{\Phi_i^{(k)}\}_{i \in \mathbb{Z}}$ for $k = 1, 2$:

$$\begin{aligned} \Phi_i^{(1)} &= \Phi_i^n - \frac{\Delta t}{\Delta x} (\hat{f}_{i+1/2}(\Phi^n) - \hat{f}_{i-1/2}(\Phi^n)), \\ \Phi_i^{(2)} &= \Phi_i^{(1)} - \frac{\Delta t}{\Delta x} (\hat{f}_{i+1/2}(\Phi^{(1)}) - \hat{f}_{i-1/2}(\Phi^{(1)})), \\ \Phi_i^{n+1} &= \frac{1}{2} (\Phi_i^n + \Phi_i^{(2)}), \quad i \in \mathbb{Z}, \quad n = 0, 1, 2, \dots \end{aligned}$$

To complete the description, we mention that the boundary conditions are discretized by setting

$$\Phi_i^n = \mathbf{0} \quad \text{for } i = -1, -2, \quad \Phi_{M+i+1}^n = (\phi_{\max}, \dots, \phi_{\max})^T \quad \text{for } i = 1, 2 \quad (3.24)$$

for (1.4) and by

$$\Phi_i^n = \Phi_{M+i}^n, \quad \Phi_{M+2+i}^n = \Phi_{2+i}^n \quad \text{for } i = 1, 2$$

in the case of (3.12). We recall that a MUSCL scheme requires to consider two additional ghost cells on each boundary of the computational domain. The second condition of (3.24) ensures that the total concentration at the bottom is the maximum.

3.4 Numerical results

3.4.1 Preliminaries

In the following examples, we solve (1.1) numerically for $0 \leq t \leq T$ and $0 \leq x \leq L$ for the MLB polydisperse sedimentation model and the MCLWR traffic model. We compare numerical results obtained by some chosen PVM methods with a high-order characteristic-wise numerical method (SPEC-INT), based on fifth-order finite difference weighted essentially non-oscillatory (WENO) schemes, see [15] for the details.

For each model, the x -interval $[0, L]$ is subdivided into M subintervals of length $\Delta x = L/M$. We denote by Δt the time step used to advance the numerical solution from $t = t^n$ to $t^{n+1} = t^n + \Delta t$ and by Φ_i^n the vector of numerical solutions associated with cell $[i\Delta x, (i+1)\Delta x]$, $i = 0, \dots, M-1$, at time t^n . For each iteration, Δt is determined by the following formula (CFL condition):

$$\frac{\Delta t}{\Delta x} \max_{1 \leq j \leq M} \varrho(\mathcal{J}_f(\Phi_j^n)) = C_{\text{cfl}},$$

where $\varrho(\cdot)$ is the spectral radius (or an upper bound of it). In the numerical examples we choose $C_{\text{cfl}} = 0.6$.

For comparison purposes, we compute reference solutions for numerical tests by the SPEC-INT scheme with $M_{\text{ref}} = 12800$ cells. We compute approximate L^1 errors at different times for each scheme as follows. We denote by $(\phi_{i,j}^M(t))_{i=0}^{M-1}$ and $(\phi_{i,j}^{\text{ref}}(t))_{i=0}^{M_{\text{ref}}-1}$ the numerical solution for the i -th component at time t calculated with M and M_{ref} cells, respectively. We compute $\tilde{\phi}_{i,j}^{\text{ref}}(t)$ for $i = 0, \dots, M-1$ by

$$\tilde{\phi}_{i,j}^{\text{ref}}(t) = \frac{1}{R} \sum_{k=0}^{R-1} \phi_{R(i-1)+k,j}^{\text{ref}}(t), \quad R = M_{\text{ref}}/M. \quad (3.25)$$

The total approximate L^1 error of the numerical solution $(\phi_{i,j}^M(t))_{i=0}^{M-1}$ at time t is then given by

$$e_M^{\text{tot}}(t) := \frac{1}{M} \sum_{j=1}^N \sum_{i=0}^{M-1} |\tilde{\phi}_{i,j}^{\text{ref}}(t) - \phi_{i,j}^M(t)|. \quad (3.26)$$

Based on the approximate errors defined by (3.26), we may calculate a numerical order of convergence from pairs of total approximate L^1 errors $e_{M/2}^{\text{tot}}(t)$ and $e_M^{\text{tot}}(t)$ by

$$\theta_M(t) := \log_2(e_{M/2}^{\text{tot}}(t)/e_M^{\text{tot}}(t)). \quad (3.27)$$

3.4. Numerical results

i	1	2	3	4	5	6	7	8	9	10	11
$\phi_i^0[10^{-3}]$	0.435	3.747	14.420	32.603	47.912	47.762	32.663	15.104	4.511	0.783	0.060
$D_i[10^{-5}\text{m}]$	8.769	8.345	7.921	7.497	7.073	6.649	6.225	5.801	5.377	4.953	4.529
δ_i	1.000	0.952	0.903	0.855	0.807	0.758	0.710	0.662	0.613	0.565	0.516

Table 3.1: Example 3.3 (MLB model, $N = 11$): initial concentrations ϕ_i^0 , diameters D_i and normalized particle sizes δ_i [71].

3.4.2 Examples 3.1, 3.2 and 3.3 (MLB model, $N = 2, 4, 11$)

In Examples 3.1 to 3.3 we consider the standard test case of batch settling of an initially homogeneous suspension in a column. Example 3.1 corresponds to $N = 2$ species [70] with density $\varrho_s = 2790 \text{ kg/m}^3$ and different diameters $D_1 = 4.96 \times 10^{-4} \text{ m}$ and $D_2 = 1.25 \times 10^{-4} \text{ m}$, corresponding to $\delta_1 = 1$ and $\delta_2 = D_2/D_1 = 0.25202$. The depth of the vessel is $L = 0.3 \text{ m}$. The maximum total concentration is $\phi_{\max} = 0.6$ and the initial concentrations are $\Phi_0 = (\phi_1^0, \phi_2^0) = (0.2, 0.05)^T$. The hindered settling factor $V(\phi)$ is chosen according to (3.7) with the exponent $n_{\text{RZ}} = 4.7$. The remaining parameters are $g = 9.81 \text{ m/s}^2$, $\mu_f = 0.02416 \text{ Pa s}$ and $\varrho_f = 1208 \text{ kg/m}^3$. To compare the performance of PVM methods with that of SPEC-INT, we calculate numerical solutions for a sequence of spatial discretizations $\Delta x = L/M$, and compare the solutions with the above-mentioned reference solution. The variant of the PVM-4 method employed for simulations is the one that corresponds to the choice $|S_M| = A = 1$ and $S_I = B = 0.3873$ (see Section 3.2.3). These solutions are shown in Figure 3.8 for the simulated time $T = 50 \text{ s}$. We use the midpoint rule (3.15) to calculate the approximate Roe matrix $\mathbf{A}$. The approximate errors, convergence rates and CPU time for Examples 3.1 to 3.4 are shown in Table 3.2.

In Example 3.2 we consider the MLB model for $N = 4$ with parameters $\delta_1 = 1$, $\delta_2 = 0.8$, $\delta_3 = 0.6$, $d_4 = 0.4$, $\phi_{\max} = 0.6$, and $\phi_i^0 = 0.05$ for $i = 1, \dots, 4$. The other parameters are the same as in Example 3.1. Numerical results at $T = 50 \text{ s}$ are shown in Figure 3.9.

In Example 3.3 we simulate the MLB model for $N = 11$ species settling in a column of depth $L = 0.935 \text{ m}$, according to an experiment published in [71]. The initial concentrations ϕ_i^0 , diameters D_i and normalized $\delta_i = D_i/D_1$ are given in Table 3.1; we use again (3.7) with $n_{\text{RZ}} = 4.7$ and the maximum total concentration $\phi_{\max} = 0.641$. Numerical results are shown in Figure 3.10.

3.4. Numerical results

	SPEC-INT				HLL			PVM-2			PVM-4		
	M	e_M^{tot}	θ_M	cpu [s]	e_M^{tot}	θ_M	cpu [s]	e_M^{tot}	θ_M	cpu [s]	e_M^{tot}	θ_M	cpu [s]
Example 3.1	100	2.58	—	2.03	6.14	—	0.05	5.42	—	0.07	4.89	—	0.07
MLB model	200	1.62	0.67	3.75	3.76	0.70	0.21	3.35	0.69	0.28	3.06	0.67	0.29
$N = 2$	400	0.89	0.85	10.43	2.19	0.77	1.41	1.94	0.78	1.83	1.76	0.79	1.91
$T = 50$ s	800	0.44	0.99	38.04	1.18	0.89	6.20	1.05	0.89	8.30	0.96	0.87	8.52
	1600	0.19	1.17	101.00	0.61	0.94	23.89	0.54	0.94	32.37	0.50	0.92	33.13
	M	e_M^{tot}	θ_M	cpu [s]	e_M^{tot}	θ_M	cpu [s]	e_M^{tot}	θ_M	cpu [s]	e_M^{tot}	θ_M	cpu [s]
Example 3.2	100	5.10	—	5.97	10.43	—	0.07	9.40	—	0.10	8.68	—	0.1
MLB model	200	2.41	1.07	17.38	5.50	0.92	0.62	4.93	0.93	0.85	4.57	0.92	0.82
$N = 4$	400	1.27	0.92	43.40	3.01	0.86	2.21	2.71	0.86	3.06	2.55	0.84	3.15
$T = 50$ s	800	0.70	0.84	75.44	1.67	0.84	7.17	1.52	0.83	9.97	1.45	0.81	10.18
	1600	0.31	1.18	221.18	0.82	1.02	26.78	0.75	1.02	37.34	0.71	1.01	39.04

Table 3.2: Examples 3.1, 3.2: approximate L^1 errors (e_M^{tot} , figures to be multiplied by 10^{-3}), convergence rates (θ_M) and CPU times (cpu). The Roe matrix was approximated by the midpoint rule (3.15).

3.4.3 Example 3.4 (MCLWR model, $N = 9$)

We consider the numerical experiment proposed in [78] where the initial density distribution is given by an isolated platoon in the congested regime for the Drake model (3.10) with $\phi_{\text{opt}} = 50$ cars/km. We consider a nine-class system with

$$(\beta_1, \dots, \beta_9) = (60.0, 67.5, 75.0, \dots, 120.0) \text{ km/h}$$

and the initial datum

$$\Phi(x, 0) = p(x)\phi_0(0.04, 0.08, 0.12, 0.16, 0.2, 0.16, 0.12, 0.08, 0.04)^T,$$

corresponding to a “platoon” defined by $\phi = \phi_0 = 120$ cars/km (which is well over the optimal density ϕ_{opt} and leads to a congested traffic regime) and the shape function

$$p(x) = \begin{cases} 10x & \text{for } 0 < x < 0.1, \\ -10(x-1) & \text{for } 0.9 < x \leq 1, \\ 1 & \text{for } 0.1 < x \leq 0.9, \\ 0 & \text{otherwise.} \end{cases}$$

Numerical results are shown in Figure 3.11. Table 3.4 shows results corresponding to different forms of Gaussian quadrature applied to the integral in (3.21) to calculate

3.4. Numerical results

	SPEC-INT				HLL			PVM-2			PVM-4		
	M	e_M^{tot}	θ_M	cpu [s]	e_M^{tot}	θ_M	cpu [s]	e_M^{tot}	θ_M	cpu [s]	e_M^{tot}	θ_M	cpu [s]
Example 3.3	100	4.64	—	1.12	6.44	—	0.01	5.80	—	0.02	5.26	—	0.02
MLB model	200	2.98	0.64	5.16	4.52	0.51	0.06	4.13	0.48	0.09	3.88	0.44	0.10
$N = 11$	400	1.81	0.71	32.79	3.36	0.42	0.30	3.07	0.42	0.46	2.90	0.41	0.49
$T = 50$ s	800	1.14	0.66	118.66	2.34	0.51	1.22	2.15	0.51	1.79	2.04	0.50	1.98
	1600	0.57	0.99	444.04	1.42	0.71	4.63	1.30	0.72	6.79	1.24	0.72	7.47
	M	e_M^{tot}	θ_M	cpu [s]	e_M^{tot}	θ_M	cpu [s]	e_M^{tot}	θ_M	cpu [s]	e_M^{tot}	θ_M	cpu [s]
Example 3.4	100	1.69	—	0.26	3.46	—	0.02	3.46	—	0.02	3.49	—	0.02
MCLWR model	200	1.63	0.05	1.27	2.77	0.32	0.06	2.76	0.32	0.09	2.77	0.33	0.11
$N = 9$	400	1.13	0.53	4.16	1.93	0.52	0.27	1.90	0.53	0.39	1.88	0.56	0.43
$T = 0.01$ h	800	0.79	0.50	19.89	1.40	0.45	1.13	1.38	0.45	1.63	1.35	0.47	1.81
	1600	0.44	0.84	63.11	0.91	0.62	5.07	0.89	0.62	7.29	0.87	0.64	7.94

Table 3.3: Examples 3.3, 3.4: approximate L^1 errors (e_M^{tot} , figures to be multiplied by 10^{-3}), convergence rates (θ_M) and CPU times (cpu). The Roe matrix was approximated by the midpoint rule (3.15).

the approximate Roe matrix. No substantial difference is found in the approximate errors, whereas the CPU time required for the midpoint rule (3.15) is much smaller than for the Gaussian quadrature rules (3.22) and (3.23). This is due to the fact that multiplication by a single Jacobian can be performed fastly by exploiting its structure.

3.4. Numerical results

	Gaussian quadrature, 2 nodes							Gaussian quadrature, 3 nodes						
	PVM-2				PVM-4			PVM-2				PVM-4		
	M	e_M^{tot}	θ_M	cpu [s]	e_M^{tot}	θ_M	cpu [s]	e_M^{tot}	θ_M	cpu [s]	e_M^{tot}	θ_M	cpu [s]	
Example 3.3	100	5.80	—	0.03	5.26	—	0.04	5.80	—	0.05	5.26	—	0.06	
MLB model	200	4.13	0.48	0.15	3.88	0.44	0.18	4.13	0.48	0.25	3.88	0.44	0.28	
$N = 11$	400	3.07	0.42	0.74	2.90	0.41	0.83	3.07	0.42	1.20	2.90	0.41	1.29	
$T = 50\text{ s}$	800	2.15	0.51	2.93	2.04	0.50	3.34	2.15	0.51	4.68	2.04	0.50	5.17	
	1600	1.30	0.72	11.29	1.24	0.72	12.79	1.30	0.72	17.91	1.24	0.72	19.85	

Table 3.4: Example 3.3: approximate L^1 errors (e_M^{tot} , figures to be multiplied by 10^{-3}), convergence rates (θ_M) and CPU times (cpu) for approximation of the Roe matrix by Gaussian quadrature with two nodes (3.22) or three nodes (3.23).

3.4. Numerical results

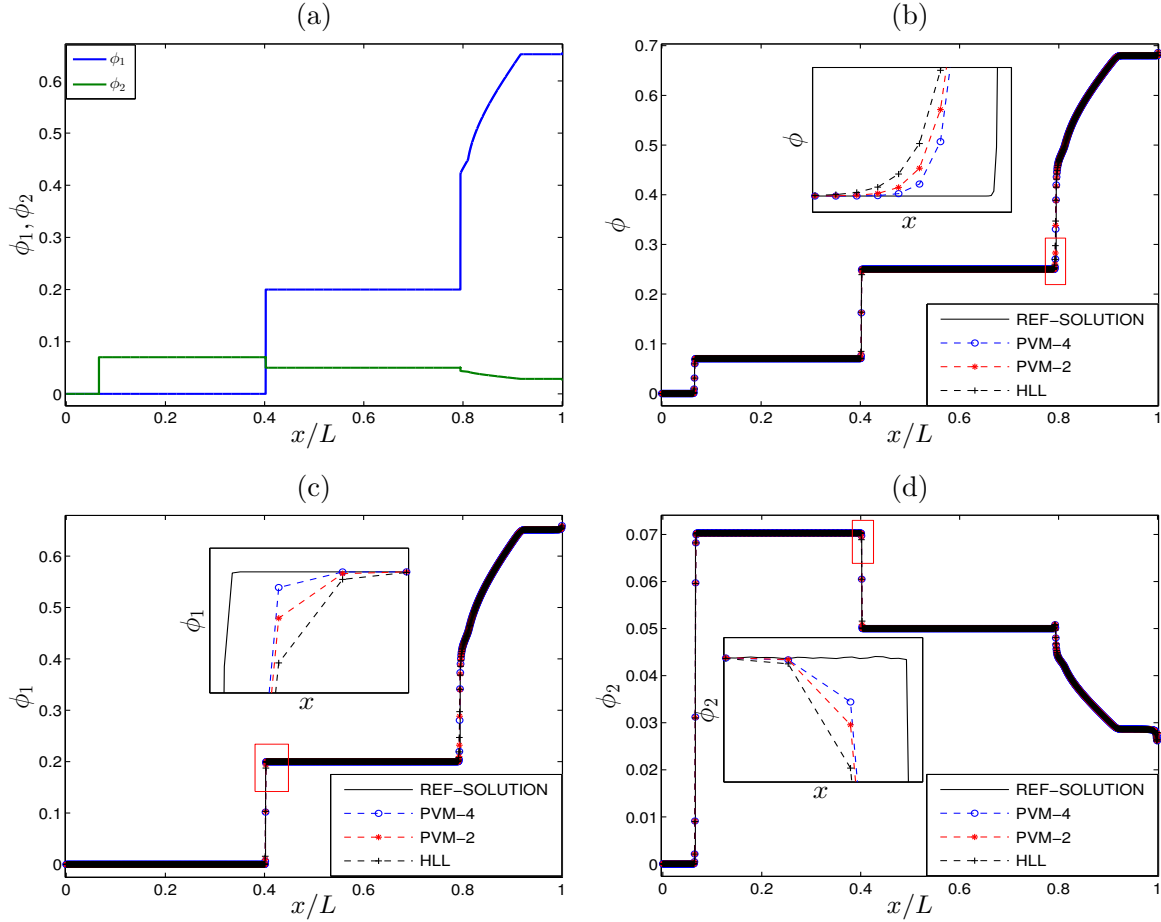


Figure 3.8: Example 3.1 (MLB model, $N = 2$): numerical solution for ϕ, ϕ_1, ϕ_2 at $T = 50$ s computed (a) by SPEC-INT method with $M = M_{\text{ref}} = 12800$ (reference solution), (b, c, d) by PVM methods with $M = 1600$, including the reference solution.

3.4. Numerical results

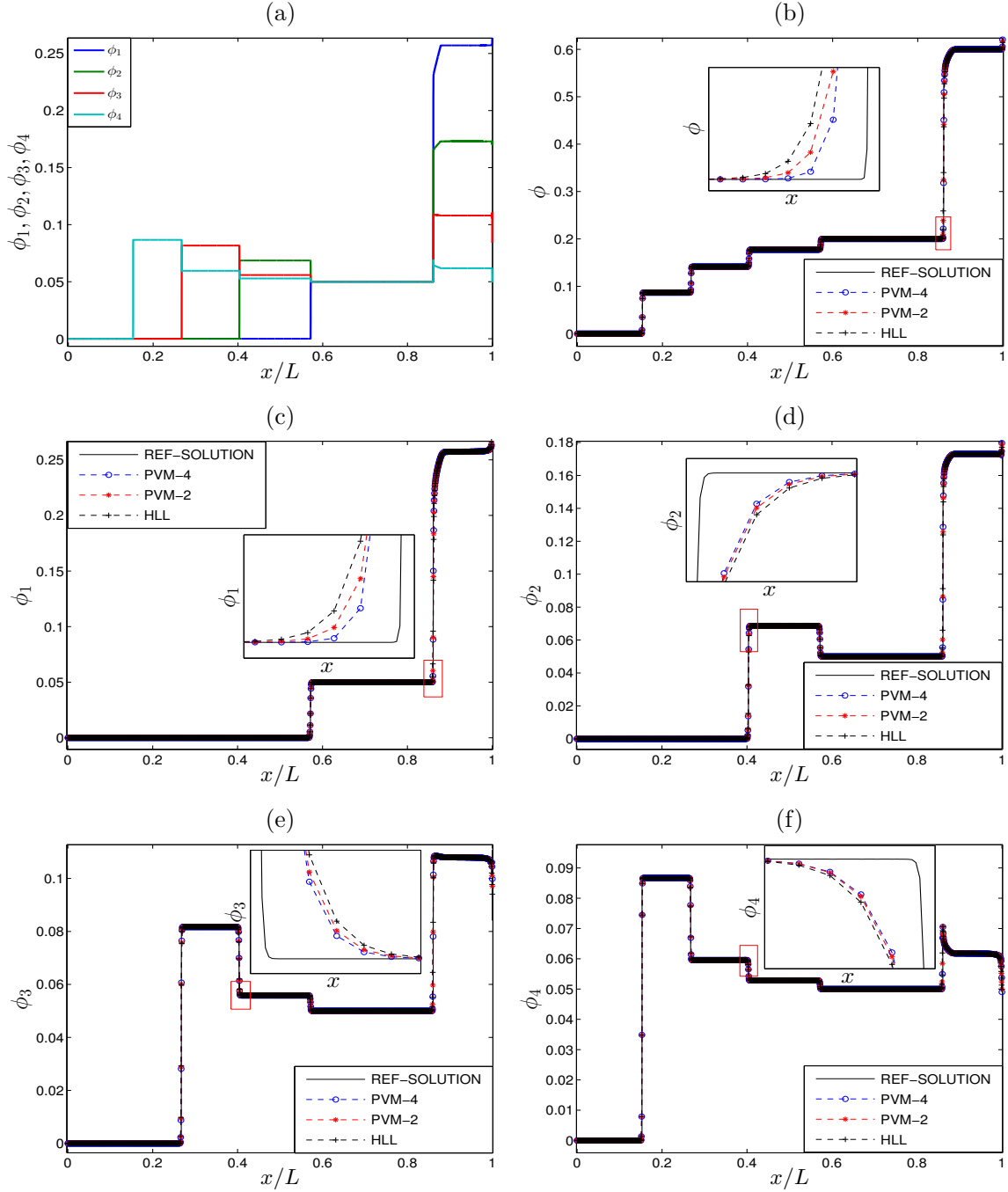


Figure 3.9: Example 3.2 (MLB model, $N = 4$): numerical solution for $\phi, \phi_1, \phi_2, \phi_3, \phi_4$ at $T = 50$ s computed (a) by SPEC-INT with $M_{\text{ref}}=12800$ (reference solution), (b, c, d, e, f) by PVM methods with $M = 1600$, including the reference solution.

3.4. Numerical results

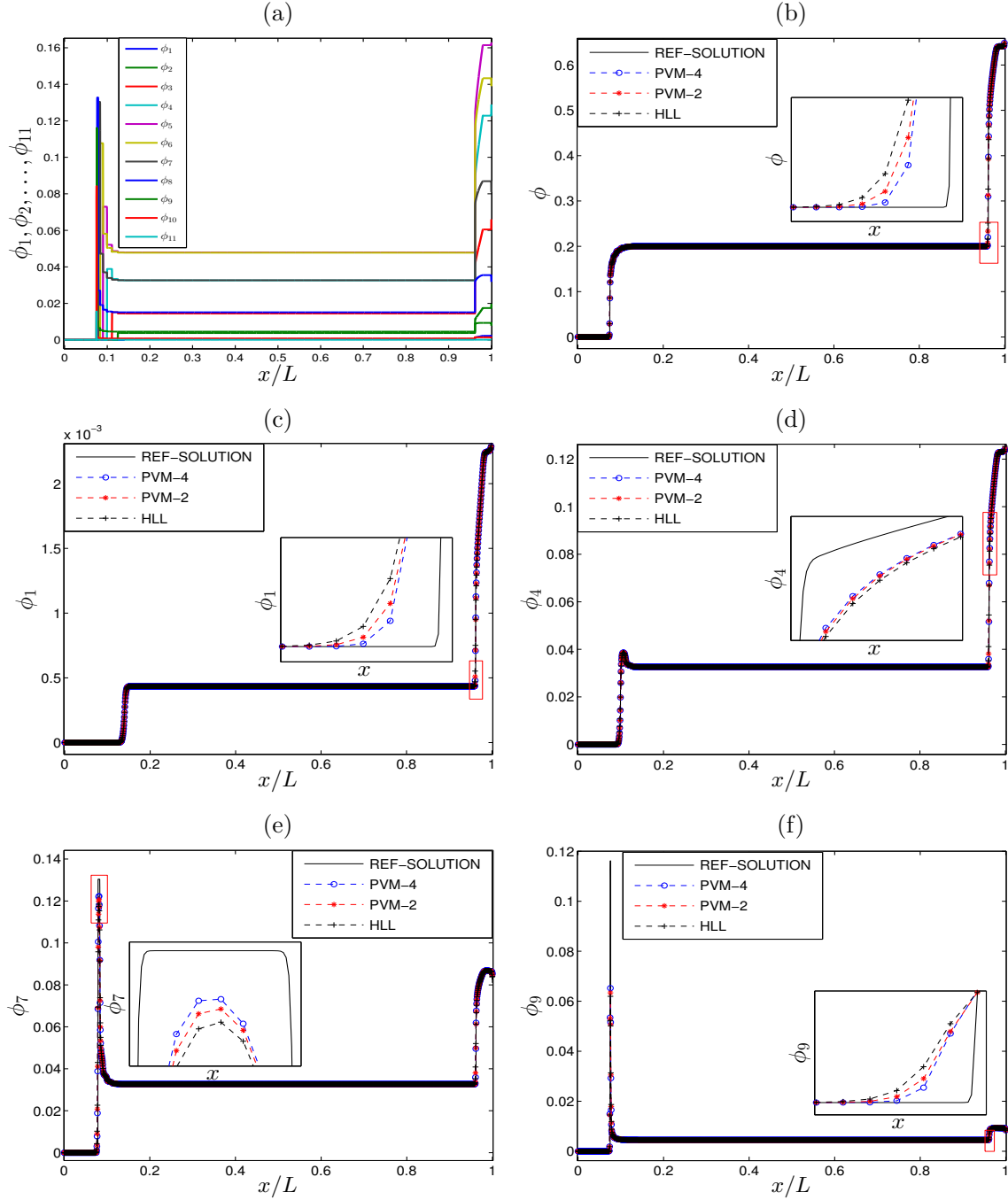


Figure 3.10: Example 3.3 (MLB model, $N = 11$): numerical solution for $\phi, \phi_1, \dots, \phi_{11}$ at $T = 50$ s (a) computed by SPEC-INT with $M_{\text{ref}}=12800$ (reference solution), (b, c, d, e, f) by PVM methods with $M = 1600$, including the reference solution.

3.4. Numerical results

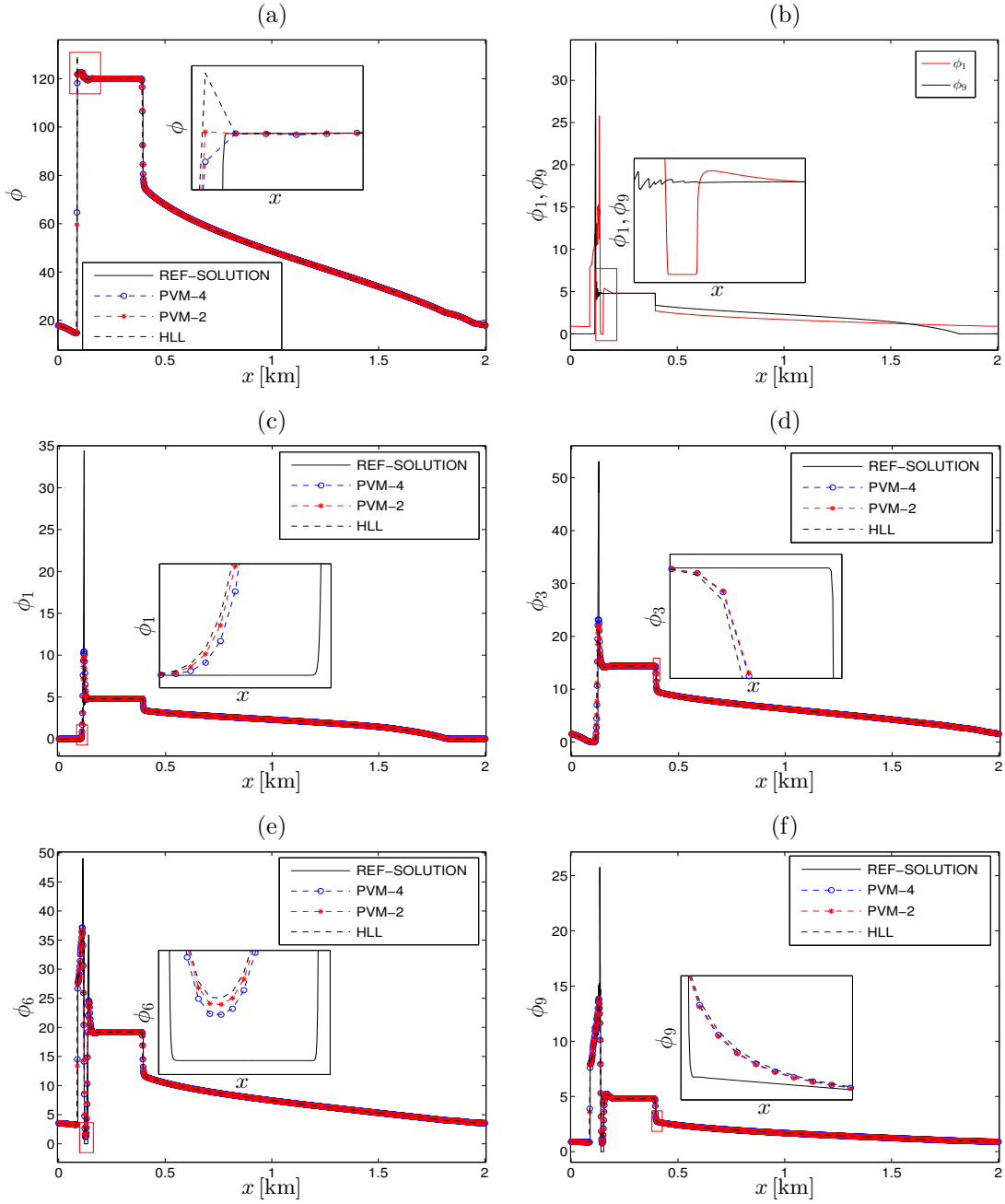


Figure 3.11: Example 3.4 (MCLWR model, $N = 9$): numerical solution (a) for ϕ and (b, c, d, e, f) for selected classes at $T = 0.015$ h computed by SPEC-INT with $M_{\text{ref}}=12800$ (reference solution), and by PVM methods with $M = 1600$, including the reference solution.

3.4.4 Efficiency plots

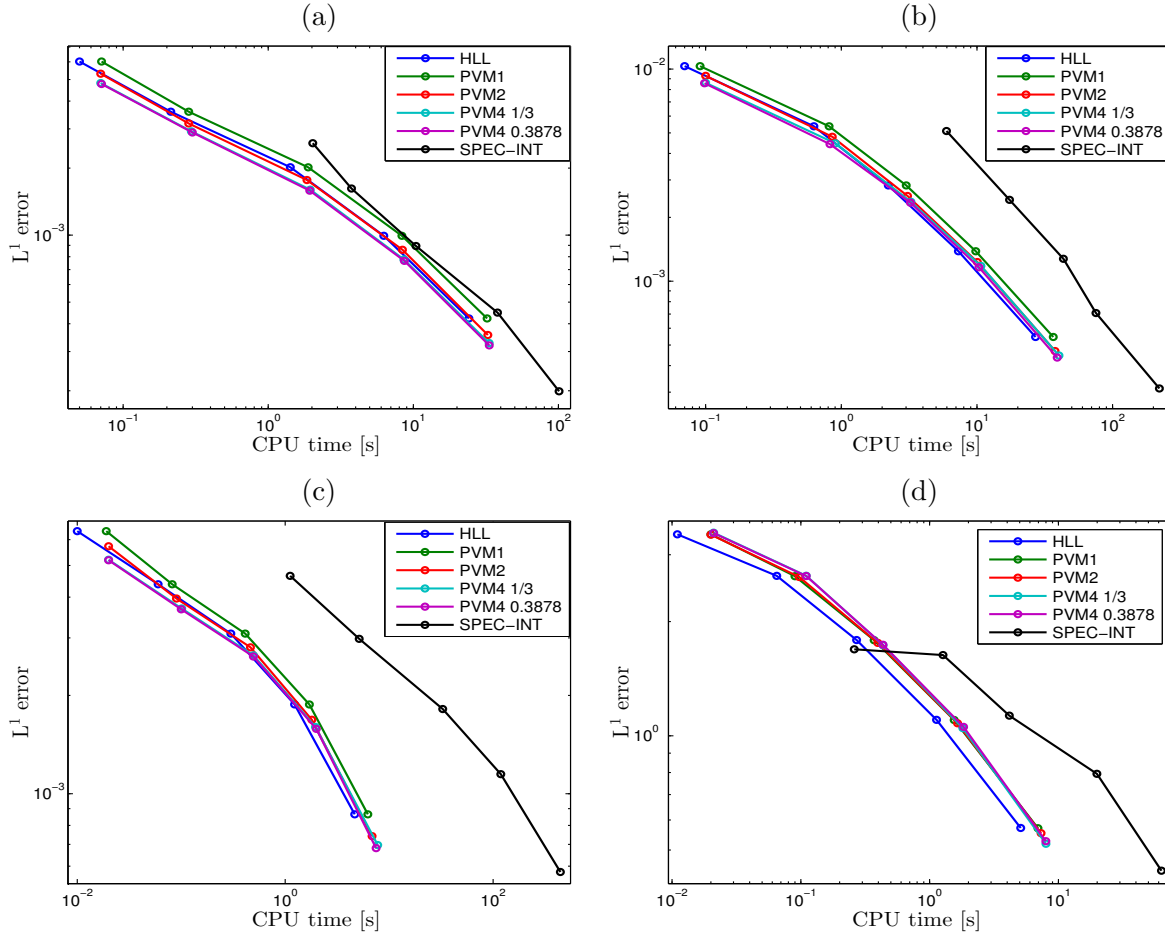


Figure 3.12: Examples 3.1 to 3.4: (MLB models $N = 2, 4, 11$, MCLWR model $N = 9$): Efficiency plot based on numerical solutions for $\Delta x = 1/M$ with $M = 100, 200, 400, 800$ and 1600 : (a) Example 3.1, (b) Example 3.2, (c) Example 3.3, (d) Example 3.4.

Based on the information of Table 3.2-3.3 and considering several variants of the PVM-4 method we plot in Figure 3.12 approximate L^1 errors versus CPU time to assess which of the methods examined is most efficient (in the usual sense of error versus CPU time). It turns out that the PVM methods are roughly comparable in efficiency, and in some cases and for coarse discretizations (e.g., for Example 3.3, as seen in Figure 3.12 (c)), the PVM-4 methods are more efficient. Moreover, for large values of N and fine discretizations, PVM methods are significantly more efficient than SPEC-INT.

3.5 Conclusions of Chapter 3

In this chapter we have proposed a fast numerical method that uses polynomial viscosity matrices to approximate Roe's numerical flux for polydisperse sedimentation simulations. We conclude that the best approximation is obtained by means of four degree polynomials (PVM-4) and an approximate Roe matrix given by the Jacobian of the fluxes computed at an average state. We have proposed two new PVM-4 schemes that improve the performance of the PVM-4 schemes proposed in [23], specially when the second largest eigenvalue is very close to zero with respect to the largest eigenvalue.

Another contribution of this work is the efficiency test of PVM schemes when the approximate Roe matrix is computed using different types of numerical integration. In this regard, we conclude that there are almost no differences between the results obtained with each quadrature rule, whereas the midpoint rule (3.15) is much faster, essentially because multiplying by flux Jacobian matrices can be performed relatively faster by exploiting its structure.

Our last contribution stems from the fact that the characteristic fields are neither genuinely nonlinear nor linearly degenerate, so the maximal characteristic speed in a segment that joins two adjacent states may be attained in a point strictly in the interior of the segment. To prevent possible entropy violations we bound all characteristic velocities at points in the segment by using the bounds provided by the interlacing property.

With respect to the quality of the numerical solutions, the best first order numerical scheme is, of course Roe's method if we could calculate a Roe matrix followed by PVM-4, PVM-2 and HLL method. PVM-2 is an interesting choice, as it is an upwind scheme and provides results as good as PVM-4. To conclude the PVM schemes are an excellent alternative to Roe methods to approximate time-dependent solutions when computing the spectral information of the flux Jacobian is very expensive.

Chapter 4

Implicit-Explicit Methods for the efficient simulation of the settling of dispersions of droplets and colloidal particles

4.1 Introduction

4.1.1 Scope

The settling of a dispersion of droplets [1, 66] or that of a suspension of colloidal solid particles [28] dispersed in a fluid can be modelled by systems of convection-diffusion equations of the form

$$\partial_t \Phi + \partial_x \mathbf{f}(\Phi) = \partial_x (\mathbf{B}(\phi) \partial_x \Phi), \quad 0 < x < L, \quad t > 0, \quad (4.1)$$

where t is time, x is depth, $\Phi(x, t) = (\phi_1(x, t), \dots, \phi_N(x, t))^T$ is the sought unknown, namely the vector of volume fractions of droplets or particles of class i , and $\phi = \phi_1 + \dots + \phi_N$ is the total volume fraction of the disperse phase. Here we assume that particles of class i have diameter D_i , where $D_1 > \dots > D_N$. Thus, the corresponding settling velocities v_i of individual particles of class i in an unbounded fluid satisfy $v_1 > v_2 > \dots > v_N$. Moreover, we assume that the flux vector $\mathbf{f}(\Phi)$ is of the form

$$\mathbf{f}(\Phi) = V(\phi)(v_1 \phi_1, \dots, v_N \phi_N)^T, \quad (4.2)$$

where $V(\phi)$ is a given function, and that $\mathbf{B}(\phi)$ is a given diffusion matrix that depends on ϕ . We assume that $\mathbf{B}$ is a positive semidefinite $N \times N$ matrix, and explicitly allow that $\mathbf{B} = \mathbf{0}$ on ϕ -intervals of positive length.

The settling of a dispersion of initial composition Φ_0 in a column of depth L is now described by the system (4.1) together with the zero-flux boundary conditions

$$\mathbf{f}(\Phi) - \mathbf{B}(\phi)\partial_x\Phi|_{x=0} = \mathbf{0}, \quad \mathbf{f}(\Phi) - \mathbf{B}(\phi)\partial_x\Phi|_{x=L} = \mathbf{0} \quad (4.3)$$

and the initial condition

$$\Phi(x, 0) = \Phi_0(x), \quad 0 \leq x \leq L, \quad (4.4)$$

where Φ_0 is the initial composition. This chapter is focused on new methods for the efficient numerical solution of the model (4.1)–(4.4) including a high-resolution discretization of the convective term.

Explicit numerical schemes for (4.1) on a uniform Cartesian grid of meshwidth Δx and time step Δt are easy to implement but are associated with a Courant-Friedrichs-Lewy (CFL) stability condition that requires the proportionality $\Delta t \sim \Delta x^2$, which makes long-term simulations of (4.1)–(4.4) on a uniform grid unacceptably slow. A well-known remedy consists in handling the diffusive term in (4.1) by an implicit discretization. The resulting semi-implicit or implicit-explicit (IMEX) schemes (cf., e.g., [3, 26]) are associated with an acceptable CFL condition $\Delta t \sim \Delta x$ but the case that $\mathbf{B}$ depends discontinuously on ϕ needs to be handled by special nonlinear solvers [21] or by solving linear problems in each time step that arise from carefully distinguishing between stiff and nonstiff unknowns in the discretized version of $\partial_x(\mathbf{B}(\phi)\partial_x\Phi)$ (see [8]). These variants will be addressed as *nonlinearly implicit* (NI-IMEX) and *linearly implicit* (LI-IMEX) schemes, respectively. On the other hand, so-called polynomial viscosity matrix (PVM) methods [23, 30] have turned out as efficient high-resolution schemes for the approximation of solutions of the first-order system of conservation laws $\partial_t\Phi + \partial_x\mathbf{f}(\Phi) = \mathbf{0}$ for models akin to (4.2) (see [18]). It is the purpose of this work to demonstrate that the combination of PVM methods with NI-IMEX and LI-IMEX discretizations provided an efficient solver for the model (4.1)–(4.4) under the assumption of a nonlinear, degenerate, and possibly discontinuous diffusion matrix $\mathbf{B}(\phi)$.

4.1.2 Outline of this chapter

The remainder of this chapter is organized as follows. In Section 4.2 a model for the polydisperse sedimentation of droplets is stated. In Section 4.3 the numerical methods

that will be used for the approximate solution of the sedimentations are presented. These methods are compared for some selected tests in Section 4.4. Some conclusions are drawn in Section 4.5.

4.2 Governing model

The model of settling of polydisperse liquid-liquid dispersions in [1] extends the treatment by Davis and Russel [28] of colloidal, monodisperse solid-liquid suspensions to the polydisperse case. According to [1], the model is given by (4.1), where v_i is the Stokes terminal velocity of the i -th class given by

$$v_i = \frac{(\rho_d - \rho_c)g\delta_i^2}{18\mu_c}, \quad i = 1, \dots, N,$$

where ρ and μ , respectively, denote density and viscosity, and the indices d and c, respectively, refer to the disperse or continuous phase. A common choice of the hindered settling function is given by the following expression:

$$V(\phi) = \begin{cases} (1 - \phi)^{n_{\text{RZ}}} & \text{for } \phi \leq 1, \\ 0 & \text{for } \phi > 1. \end{cases} \quad (4.5)$$

where the index n_{RZ} was given by Richardson and Zaki [64] as 4.65 for laminar flow conditions. The secular equation [2, 14, 31] can be used to prove that under the present assumptions, the eigenvalues $\lambda_i = \lambda_i(\Phi)$ of the Jacobian

$$\mathcal{J}_{\mathbf{f}}(\Phi) := (\partial f_i(\Phi)/\partial \phi_j)_{1 \leq i, j \leq N}$$

are all real when $\phi_i > 0$ and $\phi < \phi_{\text{max}}$ and satisfy the interlacing property

$$M_1 := v_N V(\phi) + V'(\phi) \sum_{i=1}^N v_i \phi_i < \lambda_N < v_N V(\phi) < \lambda_{N-1} < \dots < \lambda_1 < v_1 V(\phi) \quad (4.6)$$

Moreover, the diffusion matrix will always be defined by

$$\mathbf{B}(\phi) = \beta(\phi) \mathbf{I}_N, \quad (4.7)$$

where β is a non-negative function and $\mathbf{I}_N$ denotes the $N \times N$ identity matrix. The choice proposed in [1] for the settling of droplets in another liquid (specifically, of glycerol in biodiesel) is

$$\beta(\phi) := D_0 V(\phi) = D_0 (1 - \phi)^{n_{\text{RZ}}}, \quad D_0 > 0, \quad (4.8)$$

for some constant D_0 .

4.3 Numerical methods

4.3.1 Spatial discretization

We consider a spatial semi-discretization for the initial-boundary value problem (4.1)–(4.4) for $x \in [0, L]$. Assume that Δx and Δt are the spatial meshwidth and time step, respectively, of a standard Cartesian grid on

$$[0, L] \times [0, \infty), \text{ where } \Delta x = L/M$$

for some integer M . We define

$$x_i = (i + 1/2)\Delta x, \quad i = 0, 1, \dots, M - 1,$$

$$x_{i+1/2} = (i + 1)\Delta x, \quad i = -1, 0, \dots, M - 1,$$

and $t_n = n\Delta t$, $n \in \mathbb{N}_0$. Then the finite volume formulation of (4.1) that is

$$\frac{d\bar{\Phi}_i(t)}{dt} = \frac{1}{\Delta x} \begin{cases} -\mathbf{f}(\Phi(x_{1/2}, t)) + \mathbf{B}(\phi(x_{1/2}, t))\partial_x \Phi(x_{1/2}, t) & \text{for } i = 0, \\ -(\mathbf{f}(\Phi(x_{i+1/2}, t)) - \mathbf{f}(\Phi(x_{i-1/2}, t))) \\ \quad + (\mathbf{B}(\phi(x_{i+1/2}, t))\partial_x \Phi(x_{i+1/2}, t) \\ \quad \quad - \mathbf{B}(\phi(x_{i-1/2}, t))\partial_x \Phi(x_{i-1/2}, t)) & \text{for } i = 1, \dots, M - 2, \\ \mathbf{f}(\Phi(x_{M-3/2}, t)) - \mathbf{B}(\phi(x_{M-3/2}, t))\partial_x \Phi(x_{i-3/2}, t) & \text{for } i = M - 1, \end{cases} \quad (4.9)$$

where we define the vector of cell averages

$$\bar{\Phi}_i(t) := \frac{1}{\Delta x} \int_{x_{i-1/2}}^{x_{i+1/2}} \Phi(\xi, t) d\xi, \quad i = 0, \dots, M - 1.$$

This leads to the following semi-discretization of (4.9) for $\Phi_i(t) \approx \bar{\Phi}_i(t)$, $i = 0, \dots, M - 1$:

$$\frac{d\Phi_i}{dt} = \begin{cases} -\frac{1}{\Delta x} \hat{\mathbf{f}}_{1/2} + \frac{1}{\Delta x^2} \mathbf{B}_{3/2}(\Phi_1 - \Phi_0) & \text{for } i = 0, \\ -\frac{1}{\Delta x} (\hat{\mathbf{f}}_{i+1/2} - \hat{\mathbf{f}}_{i-1/2}) + \frac{1}{\Delta x^2} (\mathbf{B}_{i-1/2} \Phi_{i-1} \\ \quad - (\mathbf{B}_{i-1/2} + \mathbf{B}_{i+1/2}) \Phi_i + \mathbf{B}_{i+1/2} \Phi_{i+1}) & \text{for } i = 1, \dots, M - 2, \\ \frac{1}{\Delta x} \hat{\mathbf{f}}_{M-3/2} - \frac{1}{\Delta x^2} \mathbf{B}_{M-3/2}(\Phi_{M-1} - \Phi_{M-2}) & \text{for } i = M - 1, \end{cases} \quad (4.10)$$

where $\hat{\mathbf{f}}_{i+1/2}$ is a numerical flux that depends on values of Φ on a stencil around $x_{i+1/2}$ and

$$\mathbf{B}_{i+1/2} = \mathbf{B}_{i+1/2}(\phi_i, \phi_{i+1}) := \frac{1}{2}(\mathbf{B}(\phi_i) + \mathbf{B}(\phi_{i+1})). \quad (4.11)$$

With the notation

$$\Phi = (\Phi_1^T, \dots, \Phi_M^T)^T \in \mathbb{R}^{MN},$$

we can define the $M \times M$ block tridiagonal matrix $\mathcal{B} = \mathcal{B}(\Phi)$, with blocks of size $N \times N$, as

$$\mathcal{B} = \begin{bmatrix} -\mathbf{B}_{3/2} & \mathbf{B}_{3/2} & 0 & \cdots & 0 & 0 \\ \mathbf{B}_{3/2} & -(\mathbf{B}_{3/2} + \mathbf{B}_{5/2}) & \mathbf{B}_{5/2} & \cdots & \vdots & \vdots \\ 0 & \mathbf{B}_{5/2} & \ddots & \ddots & 0 & 0 \\ 0 & 0 & \ddots & \ddots & \mathbf{B}_{M-5/2} & 0 \\ \vdots & \vdots & 0 & \mathbf{B}_{M-5/2} & -(\mathbf{B}_{M-5/2} + \mathbf{B}_{M-3/2}) & \mathbf{B}_{M-3/2} \\ 0 & 0 & \cdots & 0 & \mathbf{B}_{M-3/2} & -\mathbf{B}_{M-3/2} \end{bmatrix}$$

for which we have,

$$(\mathcal{B}(\Phi)\Phi)_i = \mathbf{B}_{i-1/2}\Phi_{i-1} - (\mathbf{B}_{i-1/2} + \mathbf{B}_{i+1/2})\Phi_i + \mathbf{B}_{i+1/2}\Phi_{i+1}.$$

We propose a second order discretization of the convective term in (4.10) by using the techniques proposed in [18], which we briefly describe here for the sake of completeness. We use the MUSCL technique (see [74] and Section 3.3.1) to obtain second-order schemes from basic first-order schemes. The numerical flux function for the PVM method is based on a matrix $\mathbf{A} = \mathbf{A}(\Phi_L, \Phi_R)$ that satisfies certain consistency properties and is given by

$$\hat{\mathbf{f}}(\Phi_L, \Phi_R) = \frac{1}{2}(\mathbf{f}(\Phi_L) + \mathbf{f}(\Phi_R) - P(\mathbf{A})(\Phi_R - \Phi_L)).$$

Bounds for the eigenvalues of $\mathbf{A}(\Phi_i, \Phi_{i+1})$ can be obtained from the interlacing property (4.6). Our proposal is to consider not only the matrices $\mathbf{A}$, $\mathcal{J}_{\mathbf{f}}(\Phi_L)$ and/or $\mathcal{J}_{\mathbf{f}}(\Phi_R)$ but all the matrices $\mathcal{J}_{\mathbf{f}}(\Phi_i + s(\Phi_{i+1} - \Phi_i))$, $s \in [0, 1]$ to compute the interval containing the

eigenvalues as $\mathcal{I} = [-S_0, S_0]$ for an upper bound S_0 for absolute values of eigenvalues as follows:

$$S_0 = S_{0,i+\frac{1}{2}} = \max_{0 \leq s \leq 1} \max(|v_1(\Phi_i + s(\Phi_{i+1} - \Phi_i))|, |M_1(\Phi_i + s(\Phi_{i+1} - \Phi_i))|). \quad (4.12)$$

We use the fourth degree polynomials proposed in [18, 23] defined by utilizing

$$P_4(x) = \alpha_0 + \alpha_2 x^2 + \alpha_4 x^4, \quad \text{such that } P_4(S_0) = |S_0|, P_4(S_I) = S_I \text{ and } P_4'(S_I) = 1, \quad (4.13)$$

where $S_I = 0.3873S_0$, see [18, Theorem 3.1 and PVM-4(S_0) Section 3.2].

4.3.2 Time discretization

Second-order explicit time discretization

A fully discrete scheme arises from applying Heun's method. We define the vectors

$$\Phi^n := (\Phi_{-2}^n, \Phi_{-1}^n, \Phi_0^n, \dots, \Phi_M^n, \Phi_{M+1}^n, \Phi_{M+2}^n)^T, \quad \Phi^{(k)} := (\Phi_{-2}^{(k)}, \dots, \Phi_{M+2}^{(k)})^T, \quad k = 1, 2.$$

and according to (4.11),

$$B_{i+1/2}^n = B_{i+1/2}(\phi_i^n, \phi_{i+1}^n), \quad B_{i+1/2}^{(1)} = B_{i+1/2}(\phi_i^{(1)}, \phi_{i+1}^{(1)}) \quad \text{for } i = 0, \dots, M-1.$$

The zero-flux boundary conditions are incorporated by defining the total numerical flux

$$\mathbf{J}_{i+1/2}(\Phi^n) := \begin{cases} \hat{\mathbf{f}}_{i+1/2}(\Phi^n) - \frac{1}{\Delta x} B_{i+1/2}^n(\Phi_{i+1}^n - \Phi_i^n) & \text{for } i = 0, \dots, M-1, \\ 0 & \text{for } i = -1 \text{ or } i = M. \end{cases} \quad (4.14)$$

The total numerical fluxes $\mathbf{J}_{i+1/2}(\Phi^{(1)})$ are defined analogously, with Φ^n in (4.14) replaced by $\Phi^{(1)}$. This scheme is defined as follows:

$$\begin{aligned} \Phi_i^{(1)} &= \Phi_i^n - \frac{\Delta t}{\Delta x} (\mathbf{J}_{i+1/2}(\Phi^n) - \mathbf{J}_{i-1/2}(\Phi^n)), \\ \Phi_i^{(2)} &= \Phi_i^{(1)} - \frac{\Delta t}{\Delta x} (\mathbf{J}_{i+1/2}(\Phi^{(1)}) - \mathbf{J}_{i-1/2}(\Phi^{(1)})), \\ \Phi_i^{n+1} &= \frac{1}{2} (\Phi_i^n + \Phi_i^{(2)}), \quad i = 0, \dots, M-1, \quad n = 0, 1, 2, \dots \end{aligned}$$

We consider semi-implicit IMEX-RK schemes described in Section 1.2 and the following definitions of nonlinearly and linearly implicit IMEX-RK methods.

Nonlinearly implicit IMEX-RK methods

To describe the nonlinearly implicit IMEX-RK method for solving (4.10), we rewrite it in the form

$$\frac{d\Phi}{dt} = C(\Phi) + D(\Phi), \quad (4.15)$$

where we define

$$C(\Phi)_i := -\frac{1}{\Delta x}(\mathbf{f}_{i+1/2} - \mathbf{f}_{i-1/2})(\Phi), \quad D(\Phi) := \frac{1}{\Delta x^2} \mathcal{B}(\Phi) \Phi.$$

For general pairs of RK schemes, the computations of a nonlinearly implicit IMEX-RK scheme necessary to advance an Φ^n from time t^n to $t^{n+1} = t^n + \Delta t$ are given in Algorithm 4.3.1 [3, 61]:

Algorithm 4.3.1 (Nonlinearly implicit IMEX-RK scheme [21])

Input: approximate solution vector Φ^n for $t = t_n$

do $i = 1, \dots, s$

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

$$\Phi^{(i)} = \Phi^n + \Delta t \left(\sum_{j=1}^{i-1} a_{ij} K_j + a_{ii} D(\Phi^{(i)}) + \sum_{j=1}^{i-1} \tilde{a}_{ij} \tilde{K}_j \right)$$

$$K_i \leftarrow D(\Phi^{(i)})$$

$$\tilde{K}_i \leftarrow C(\Phi^{(i)})$$

enddo

$$\Phi^{n+1} \leftarrow \Phi^n + \Delta t \sum_{j=1}^s b_j K_j + \Delta t \sum_{j=1}^s \tilde{b}_j \tilde{K}_j$$

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

Algorithm 4.3.1 requires solving for the vector $\mathbf{u} = \Phi^{(i)} \in \mathbb{R}^{MN}$ a nonlinear system of NM scalar equations of the following form

$$\mathbf{u} - a_{ii} \Delta t D(\mathbf{u}) - \mathbf{r}_i = \mathbf{0}, \quad i = 1, \dots, s, \quad (4.16)$$

where $\mathbf{r}_i \in \mathbb{R}^{MN}$ is given by

$$\mathbf{r}_i = \Phi^n + \Delta t \left(\sum_{j=1}^{i-1} a_{ij} K_j + \sum_{j=1}^{i-1} \tilde{a}_{ij} \tilde{K}_j \right).$$

We use Newton-Raphson's method, together with suitable convergence globalization techniques.

Linearly implicit IMEX-RK methods

The nonlinearly implicit IMEX-RK schemes proposed in [21] require solving a nonlinear system of NM scalar equations, as is detailed in Algorithm 4.3.1. To overcome this excessive numerical work for the solution of the nonlinear system (4.16), an essential gain is obtained by the following approach. We rewrite the semidiscrete formulation (4.10) in the form

$$\frac{d\Phi}{dt} = \mathcal{C}(\Phi) + \mathcal{D}(\Phi, \Phi) \quad (4.17)$$

with

$$\mathcal{C}(\Phi) := -\frac{1}{\Delta x}(\Delta^- \mathbf{f})(\Phi), \quad \mathcal{D}(\Phi^*, \Phi) := \frac{1}{\Delta x^2} \mathcal{B}(\Phi^*) \Phi.$$

Here the idea is to distinguish in the system (4.17) between stiff and nonstiff dependence on the variable Φ . More precisely, by (4.17) we consider

$$\frac{d\Phi}{dt} = \mathcal{C}(\Phi^*) + \mathcal{D}(\Phi^*, \Phi) =: \mathcal{K}(\Phi^*, \Phi),$$

where Φ^* is treated explicitly as argument of $\mathbf{f}$ and $\mathcal{B}$, while Φ is implicit in the term to which $\mathcal{B}$ is applied.

Algorithm 4.3.2 (Linearly implicit IMEX-RK scheme)

Input: approximate solution vector Φ^n for $t = t^n$

do $i = 1, \dots, s$

compute the stage values:

$$\Phi^{*(i)} \leftarrow \Phi^n + \Delta t \sum_{j=1}^{i-1} \tilde{a}_{ij} K_j$$

$$\hat{\Phi}^{(i)} \leftarrow \Phi^n + \Delta t \sum_{j=1}^{i-1} a_{ij} K_j$$

solve for K_i the linear equation

$$K_i = \mathcal{C}(\Phi^{*(i)}) + \mathcal{D}(\Phi^{*(i)}, \hat{\Phi}^{(i)} + \Delta t a_{ii} K_i),$$

where

$$\mathcal{D}(\Phi^{*(i)}, \hat{\Phi}^{(i)} + \Delta t a_{ii} K_i) = \frac{1}{\Delta x^2} \mathcal{B}(\Phi^{*(i)}) (\hat{\Phi}^{(i)} + \Delta t a_{ii} K_i)$$

enddo

$$\Phi^{n+1} \leftarrow \Phi^n + \Delta t \sum_{j=1}^s b_j K_j$$

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

Then the step from t^n to $t^{n+1} = t^n + \Delta t$ of the new linearly implicit IMEX-RK scheme is given by the Algorithm 4.3.2. We observe that for the final numerical solution, we require, in particular, $\Phi^{*,n+1} = \Phi^{n+1}$, which is guaranteed by imposing the condition

$$b_i = \tilde{b}_i \quad \text{for } i = 1, \dots, s, \quad (4.18)$$

and no duplication of variables is needed in the computation of the numerical solution (for more details see [6–8]). As an example, we propose a classical second-order IMEX-RK scheme that satisfies (4.18), namely the following scheme IMEX-SSP2(3,3,2) introduced in [61]:

$$\begin{array}{c|ccc} \tilde{\mathbf{c}} & \tilde{\mathbf{A}} & & \\ \hline & & & \\ \tilde{\mathbf{b}}^T & & & \end{array} = \begin{array}{c|ccc} & 0 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 \\ 1 & 1/2 & 1/2 & 0 \\ \hline & 1/3 & 1/3 & 1/3 \end{array}, \quad \begin{array}{c|ccc} \mathbf{c} & \mathbf{A} & & \\ \hline & & & \\ \mathbf{b}^T & & & \end{array} = \begin{array}{c|ccc} & 1/4 & 1/4 & 0 & 0 \\ 1/4 & 0 & 1/4 & 0 \\ 1 & 1/3 & 1/3 & 1/3 \\ \hline & 1/3 & 1/3 & 1/3 \end{array}. \quad (4.19)$$

4.4 Numerical results

4.4.1 Preliminaries

In the following examples, we solve (4.1)–(4.4) numerically for $0 \leq t \leq T$ and $0 \leq x \leq L$, under various choices of the initial datum Φ_0 in (4.4) and the diffusion function $\beta(\phi)$. Numerical results are obtained by NI-IMEX and LI-IMEX techniques for the convection-diffusion problem involving the PVM4 method to handle the convection part. Numerical results and efficiency are compared to those obtained by replacing the PVM4 solver by the component-wise global Lax-Friedrichs WENO method (COMP-GLF), and using either the NI-IMEX or the LI-IMEX technique to handle the diffusive part. Thus, we compare four methods: PVM4-NI-IMEX, PVM4-LI-IMEX, COMPGLF-NI-IMEX and COMPGLF-LI-IMEX. The reference solution is obtained by the COMPGLF-NI-IMEX method on a very fine grid. The x -interval $[0, L]$ is subdivided into M subintervals of length $\Delta x = L/M$. We denote by Δt the time step used to advance the numerical solution from $t = t^n$ to $t^{n+1} = t^n + \Delta t$ and by Φ_i^n the vector of numerical solutions associated with cell $[i\Delta x, (i+1)\Delta x]$, $i = 0, \dots, M-1$, at time t^n . For each iteration, Δt is determined by the following formula (CFL condition):

$$\frac{\Delta t}{\Delta x} \max_{1 \leq i \leq M} \max\{|S_{L,i+1/2}|, |S_{R,i+1/2}|\} = C_{\text{cfl}},$$

where $S_{L,i+1/2}$ and $S_{R,i+1/2}$ are the bounds on the local eigenvalues defined in (4.12). In the numerical examples we choose $C_{\text{cfl}} = 0.6$.

We compute approximate L^1 errors at different times for each scheme as follows. We denote by $(\phi_{i,j}^M(t))_{i=0}^{M-1}$ and $(\phi_{l,j}^{\text{ref}}(t))_{l=0}^{M_{\text{ref}}-1}$ the numerical solution for the i -th component at time t calculated with M and $M_{\text{ref}} = 12800$ cells, respectively. We compute $\tilde{\phi}_{i,j}^{\text{ref}}(t)$ for $i = 0, \dots, M-1$ by (3.25). The total approximate L^1 error of the numerical solution $(\phi_{i,j}^M(t))_{i=0}^{M-1}$ at time t is then given by (3.26). Based on the approximate errors defined by (3.26), we may calculate a numerical order of convergence from pairs of total approximate L^1 errors $e_{M/2}^{\text{tot}}(t)$ and $e_M^{\text{tot}}(t)$ by (3.27)

4.4.2 Examples 4.1 and 4.2.

In Examples 4.1 and 4.2 we consider the test case of the settling of dispersions of glycerol droplets and colloidal particles into biodiesel in a column of depth L and the diffusion constant defined in (4.8) $D_0 = 10^{-7} \text{m}^2/\text{s}$. These examples correspond to

4.4. Numerical results

i	1	2	3	4	5	6	7	8
20% glycerol d_i [μm]	201.430	140.200	99.751	68.986	48.391	34.185	23.810	6.101
ϕ_i^0 [%]	0.0859	0.6410	4.4309	7.9280	4.7065	1.5712	0.5720	0.1758
50% glycerol d_i [μm]	417.819	291.590	202.854	143.384	100.118	68.629	48.259	33.886
ϕ_i^0 [%]	0.329	11.380	25.010	9.921	2.305	0.821	0.502	0.183

Table 4.1: Examples 4.1 and 4.2: particle diameters d_i and initial concentrations ϕ_i^0 .

$N = 8$ species with density of biodiesel $\rho_c = 880 \text{ kg/m}^3$, while the density of glycerol is $\rho_d = 1090 \text{ kg/m}^3$ and the depth is $L = 20 \text{ mm}$.

The diameters d_i and the initial condition ϕ_i^0 are given in Table 4.1. The hindered settling factor $V(\phi)$ is chosen according to (4.5) with the exponent $n_{\text{RZ}} = 4.65$. The remaining parameters are the gravity acceleration $g = 9.81 \text{ m/s}^2$ and $\mu_c = 6.5 \text{ mPa}\cdot\text{s}$. Numerical results are obtained by EX, LI-IMEX and NI-IMEX techniques for the convection-diffusion problem involving the PVM4 method to handle the convection part.

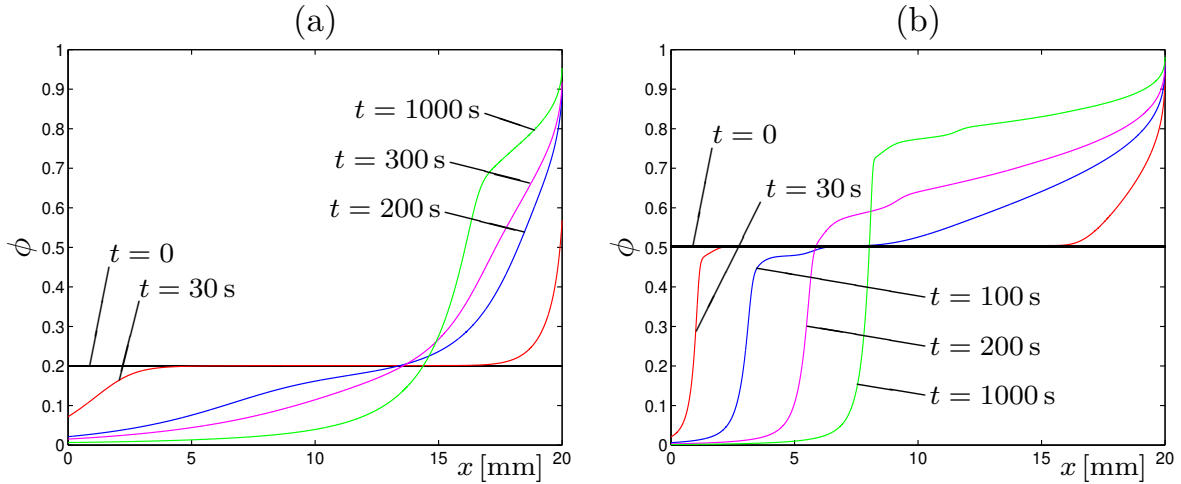


Figure 4.1: Examples 4.1 and 4.2: numerical solution for ϕ at the indicated times for (a) Example 4.1, (b) Example 4.2.

The qualitative and quantitative results can be found in Figures 4.1–4.4 and Tables 4.2–4.5. The numerical results displayed in the figures largely agree with the corresponding numerical results obtained in [1].

4.4. Numerical results

		PVM4-EX			PVM4-LIRKIMEX			PVM4-NIRKIMEX		
	M	e_M^{tot}	θ_M	cpu [s]	e_M^{tot}	θ_M	cpu [s]	e_M^{tot}	θ_M	cpu [s]
Example 4.1 $N = 8$ $t = 30$ s	100	228.798	—	0.49	228.844	—	0.51	228.845	—	1.41
	200	66.407	1.785	3.46	66.417	1.785	2.15	66.430	1.784	5.89
	400	18.421	1.850	24.62	18.423	1.850	8.74	18.424	1.850	23.9
	800	4.811	1.937	188	4.812	1.937	35.86	4.812	1.937	84.07
	1600	1.218	1.982	1480	1.224	1.975	151.2	1.218	1.983	315.3
	M	e_M^{tot}	θ_M	cpu [s]	e_M^{tot}	θ_M	cpu [s]	e_M^{tot}	θ_M	cpu [s]
Example 4.1 $N = 8$ $t = 200$ s	100	681.345	—	3.71	681.362	—	4.95	681.407	—	12.59
	200	228.802	1.574	25.54	228.807	1.574	20.36	228.819	1.574	38.48
	400	69.681	1.715	175.3	69.683	1.715	82.82	69.686	1.715	138.9
	800	19.333	1.850	1324	19.334	1.850	335.9	19.334	1.850	518.3
	1600	5.003	1.950	10330	5.004	1.950	1393	5.003	1.950	2108

Table 4.2: Approximate L^1 errors (e_M^{tot} , figures to be multiplied by 10^6), convergence rates (θ_M) and CPU times (cpu). We use PVM4 with $B = 0.3873$ for to approximate the convective term and Finite Difference for diffusive term.

		PVM4-EX			PVM4-LIRKIMEX			PVM4-NIRKIMEX		
	M	e_M^{tot}	θ_M	cpu [s]	e_M^{tot}	θ_M	cpu [s]	e_M^{tot}	θ_M	cpu [s]
Example 4.1 $N = 8$ $t = 300$ s	100	770.185	—	5.48	770.187	—	7.68	770.228	—	16.81
	200	256.760	1.585	37.36	256.760	1.585	32.14	256.772	1.585	56.27
	400	78.187	1.715	255.6	78.188	1.715	129.5	78.190	1.715	209.1
	800	21.837	1.840	1924	21.838	1.840	519.3	21.838	1.840	794.3
	1600	5.689	1.941	14880	5.690	1.940	2151	5.689	1.941	3332
	M	e_M^{tot}	θ_M	cpu [s]	e_M^{tot}	θ_M	cpu [s]	e_M^{tot}	θ_M	cpu [s]
Example 4.1 $N = 8$ $t = 1000$ s	100	1462.713	—	19.47	1462.719	—	28.45	1462.724	—	47.64
	200	445.103	1.716	131	445.104	1.716	116.7	445.107	1.716	181.9
	400	125.112	1.831	892.1	125.112	1.831	472	125.113	1.831	712
	800	34.716	1.850	6660	34.716	1.850	1915	34.716	1.850	2832
	1600	9.097	1.932	51290	9.098	1.932	7911	9.097	1.932	11930

Table 4.3: Approximate L^1 errors (e_M^{tot} , figures to be multiplied by 10^6), convergence rates (θ_M) and CPU times (cpu). We use PVM4 with $B = 0.3873$ to approximate the convective term.

4.4. Numerical results

		PVM4-EX			PVM4-LIRKIMEX			PVM4-NIRKIMEX		
	M	e_M^{tot}	θ_M	cpu [s]	e_M^{tot}	θ_M	cpu [s]	e_M^{tot}	θ_M	cpu [s]
Example 4.2 $N = 8$ $t = 30$ s	100	228.620	—	0.93	228.144	—	2.33	229.180	—	6.41
	200	78.191	1.548	5.71	77.928	1.550	10.8	78.500	1.546	31.92
	400	24.780	1.658	33.85	24.661	1.660	47.26	24.882	1.658	129.6
	800	7.425	1.739	224.3	7.391	1.738	198.2	7.445	1.741	362.3
	1600	2.134	1.799	1603	2.129	1.796	805.1	2.136	1.801	1287
	M	e_M^{tot}	θ_M	cpu [s]	e_M^{tot}	θ_M	cpu [s]	e_M^{tot}	θ_M	cpu [s]
Example 4.2 $N = 8$ $t = 100$ s	100	300.662	—	3.98	300.204	—	11.28	301.232	—	29.16
	200	93.579	1.684	22.55	93.335	1.685	46.28	93.889	1.682	125.3
	400	27.601	1.761	121	27.489	1.764	188.2	27.701	1.761	395.6
	800	8.046	1.778	780.7	80.141	1.778	767.6	8.065	1.780	1228
	1600	2.281	1.819	5474	22.753	1.816	3207	2.283	1.821	4914

Table 4.4: Example 4.2: Approximate L^1 errors (e_M^{tot} , figures to be multiplied by 10^{-5}), convergence rates (θ_M) and CPU times (cpu). We use PVM4 with $B = 0.3873$ to approximate the convective term.

		PVM4-EX			PVM4-LIRKIMEX			PVM4-NIRKIMEX		
	M	e_M^{tot}	θ_M	cpu [s]	e_M^{tot}	θ_M	cpu [s]	e_M^{tot}	θ_M	cpu [s]
Example 4.2 $N = 8$ $t = 200$ s	100	350.338	—	8.6	349.912	—	23.57	350.595	—	62.87
	200	103.661	1.757	45.03	103.364	1.759	97.74	103.871	1.755	234.7
	400	29.579	1.809	246.8	29.400	1.814	398.3	29.679	1.807	705.4
	800	8.444	1.809	1573	8.375	1.812	1628	8.471	1.809	2481
	1600	2.368	1.834	11010	2.351	1.833	6716	2.373	1.836	10250
	M	e_M^{tot}	θ_M	cpu [s]	e_M^{tot}	θ_M	cpu [s]	e_M^{tot}	θ_M	cpu [s]
Example 4.2 $N = 8$ $t = 1000$ s	100	749.603	—	46.2	749.612	—	132.2	749.721	—	223.1
	200	239.764	1.645	247.8	239.747	1.645	530.9	239.840	1.644	867.1
	400	71.757	1.740	1303	71.739	1.741	2109	71.786	1.740	3236
	800	20.375	1.816	8158	20.366	1.817	8453	20.382	1.816	12680
	1600	5.517	1.885	56500	5.514	1.885	34930	5.519	1.885	53100

Table 4.5: Example 4.2: Approximate L^1 errors (e_M^{tot} , figures to be multiplied by 10^{-5}), convergence rates (θ_M) and CPU times (cpu). We use PVM4 with $B = 0.3873$ to approximate the convective term.

4.4. Numerical results

We can deduce from these tables that the errors roughly behave as having second order, for in this case the solution has a high degree of smoothness. We notice that the EX, LI-IMEX and NI-IMEX time-stepping methods yield roughly the same errors at the same resolution.

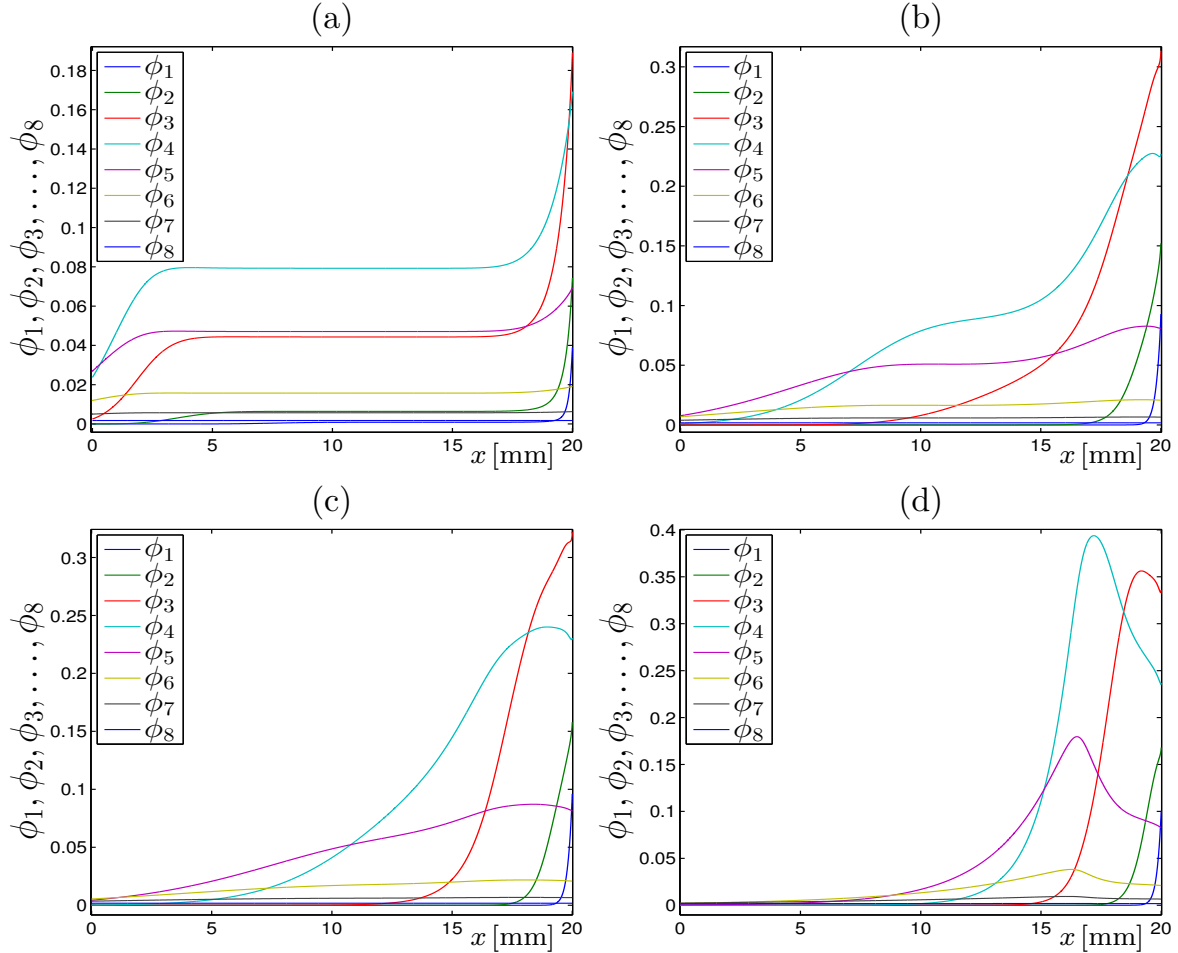


Figure 4.2: Example 4.1: numerical solution for $\phi_1, \dots, \phi_8$ at (a) $t = 30$ s, (b) $t = 200$ s, (c) $t = 300$ s, (d) $t = 1000$ s computed by PVM4-LIMEX with $M_{\text{ref}}=12800$.

4.5 Conclusions of Chapter 4

We have presented some techniques for the efficient approximate solution of systems of convection diffusion partial differential equations modelling the sedimentation of droplets of different sizes in a viscous fluids. These techniques are based on Polynomial Viscosity Matrix convective numerical fluxes and implicit treatment of the nonlinear diffusion terms. We can conclude that in the majority of cases the most efficient technique is the LI-IMEX time discretization. As a topic for future research we remark

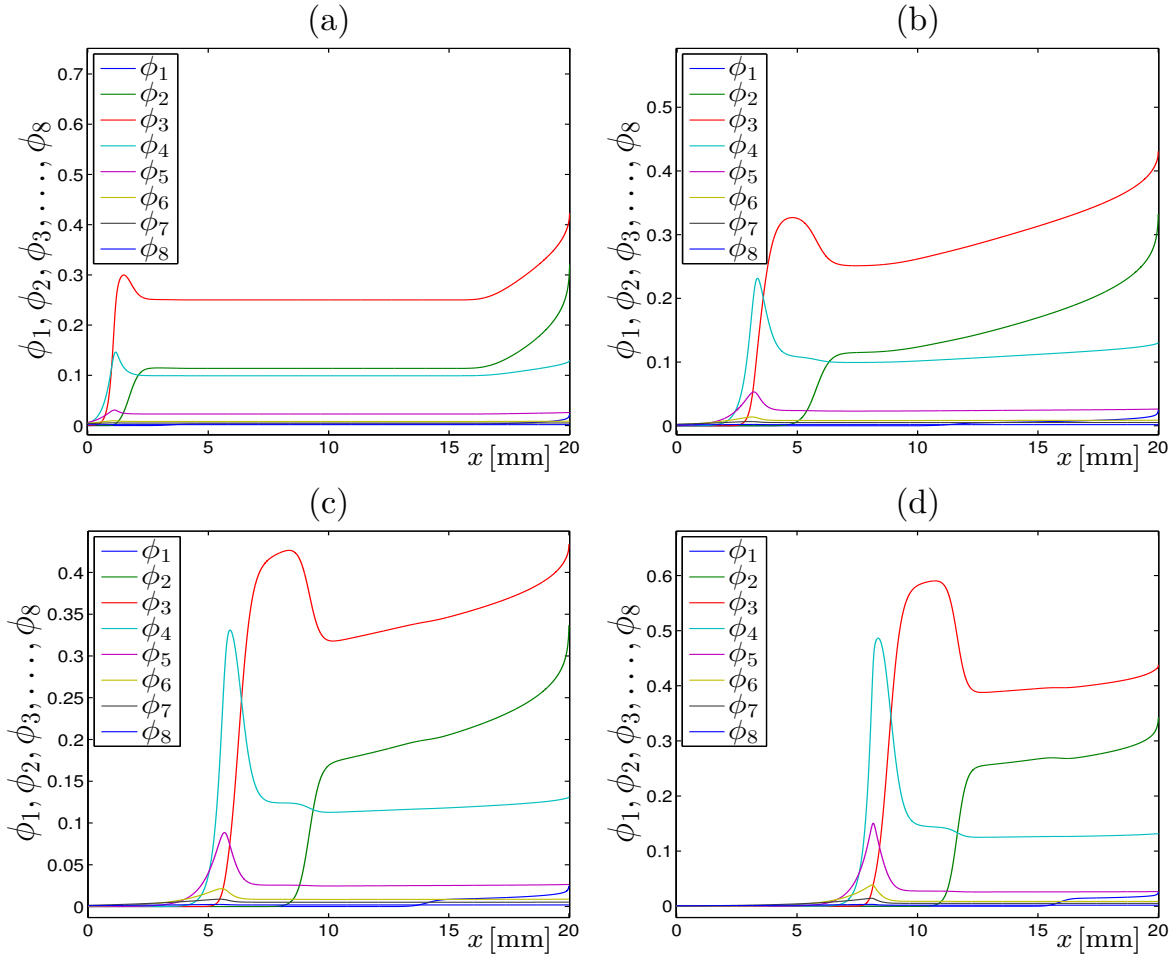


Figure 4.3: Example 4.2: numerical solution for $\phi_1, \dots, \phi_8$ at (a) $t = 30$ s, (b) $t = 100$ s, (c) $t = 200$ s, (d) $t = 1000$ s computed by PVM4-LIMEX with $M_{\text{ref}}=12800$.

4.5. Conclusions of Chapter 4

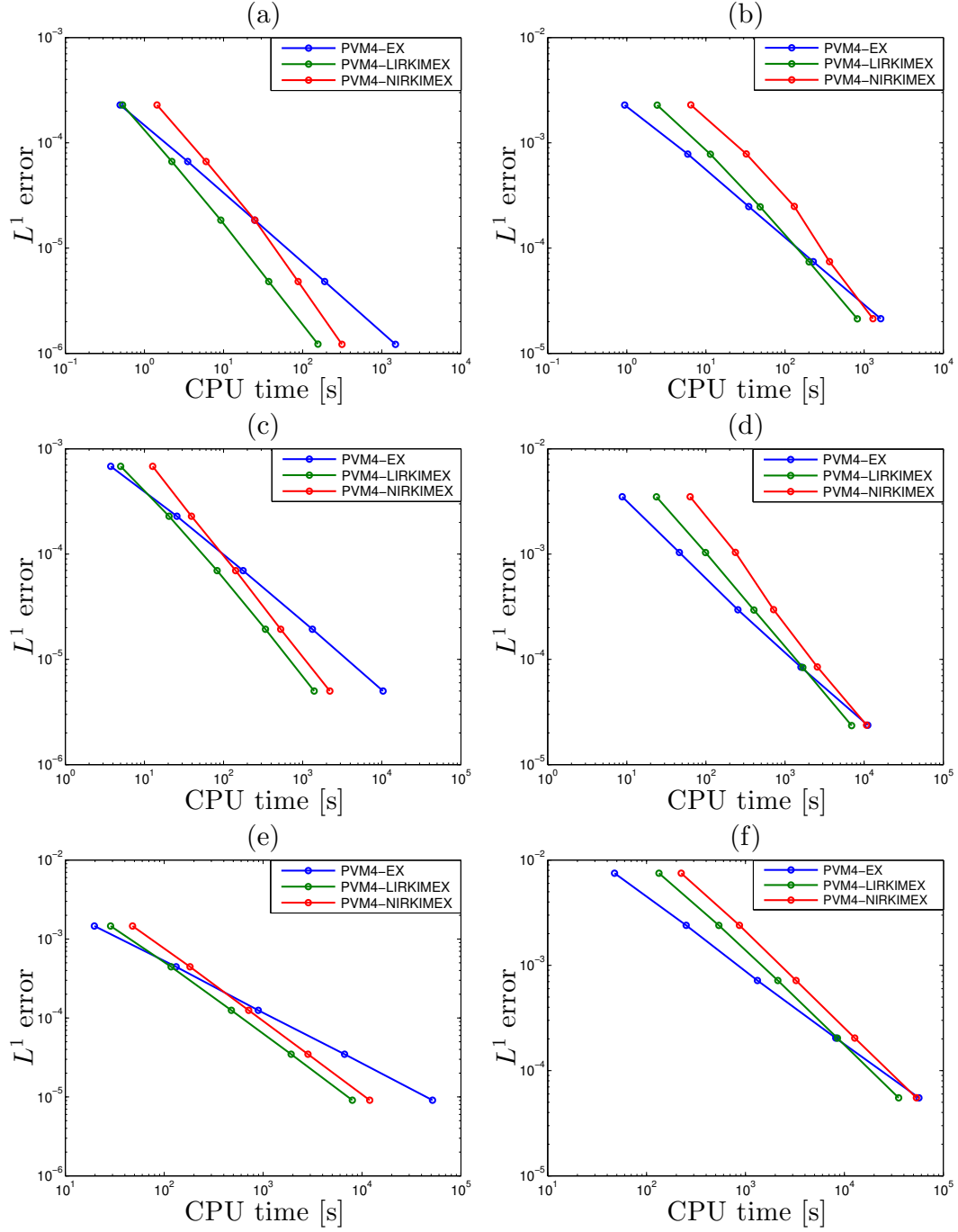


Figure 4.4: Examples 4.1 and 4.2: efficiency plots (approximate L^1 error versus CPU time) for (a, c, e) Example 4.1, (b, d, f) Example 4.2 at (a, b) $t = 30$ s, (c, d) $t = 200$ s, (e, f) $t = 1000$ s.

that the authors of [28] discuss the Péclet number Pe of the system, which indicates the importance of convection over diffusion and may here be given by $Pe = v_1 L / D_0$, where the non-diffusive limit, $Pe \rightarrow \infty$, corresponds in the monodisperse case to the well-known kinematic sedimentation model by Kynch [51]. However, there is evidence in [28] that diffusion is negligible at some dilute concentration regime, so that D_0 should assume a smaller value for ϕ belonging to some range of dilute concentrations, so that diffusion is strongly degenerate.

Chapter 5

Linearly implicit IMEX schemes for the equilibrium dispersive model of chromatography

5.1 Introduction

5.1.1 Scope

Chromatography is used to separate complex fluid mixtures, when a high purity of the product is demanded. In liquid batch chromatography, a pulse of fluid mixture, the solute, is injected at one end of a long cylindrical column filled with a porous medium (the stationary phase), followed by a continuous flow of liquid, the mobile phase, along the column. The solute interacts with the porous medium and its components begin to separate according to the strength of their affinity with the stationary phase. If the column is long enough, band profiles of single components move through it, so making it possible to collect pure fractions of components at its end.

The Equilibrium Dispersive (ED) model [42, 50, 72] is applicable when the mass transfer kinetics between the mobile phase and the stationary phase is fast, and when all band-broadening effects can be lumped into an apparent dispersion coefficient D_a . Within the ED model, chromatographic processes can be modeled by first-order non-linear convection-dominated conservation laws [42, 58, 62], coupled with some algebraic relations between the concentrations of the components of the mixture in the mobile and solid phases. Since analytic solutions can seldom be obtained, it is crucial to design

numerical schemes for performing simulations with these models, and thereby to help practitioners to reduce the need for costly empirical experimentation.

There are other approaches that take into account the kinetics between the mobile phase and the stationary phase assuming that the equilibrium is not instantaneous, obtaining systems of equations with relaxation terms [47, 48] (see also [42] for a more physical description). Both models (ED and relaxation) are similar when this relaxation parameter tends to zero.

Nonlinear convection terms cause sharp moving transitions between concentrations of different solute components and numerical methods should be able to cope with this situation, i.e., be conservative. Several works of simulation in chromatography propose conservative numerical schemes in which if D_a is null, then the roles of time t and of position z can be interchanged, being then the amounts conserved given by concentration in the mobile phase, and the flux given by the total solute concentrations (see [42, 67]). These schemes are efficient and even used to solve certain problems of identification of parameters in chromatography [46] since they do not require the inversion of the non-linear function that algebraically connects these two concentration vectors (total concentration and mobile phase concentration). The problem with these schemes is that, on the one hand, they do not conserve the original quantities, i.e., the total concentrations, and on the other hand, they can not apply if $D_a > 0$. In this sense it is proposed in [50] to suitably rewrite the model including the diffusion term and to solve it numerically by a non-conservative, linearized scheme in order to obtain an efficient method. However, Donat et al. [33] show that the non-conservative scheme proposed by Javeed et al. [50] for the simulation of the ED model can yield simulations for which the chromatographic fronts, that correspond to shocks when diffusion is neglected, move at a wrong speed and individual solute concentrations are not conserved when they should.

The main difficulty in the design of conservative numerical schemes in this formulation is that the conversion from conserved variables to primitive variables (concentration of solute and mobile phase) can only be achieved through an implicit function whose properties can be deduced from the mathematical structure of the adsorption isotherm. Nevertheless this implicit function can be approximated numerically by efficient root finders.

The particular structure of the ED model [33], summarized in the next section, provides the theoretical background to implement conservative spatial semi-discretizations of the ED model (5.2), in a method of lines strategy. It is the purpose of this chapter

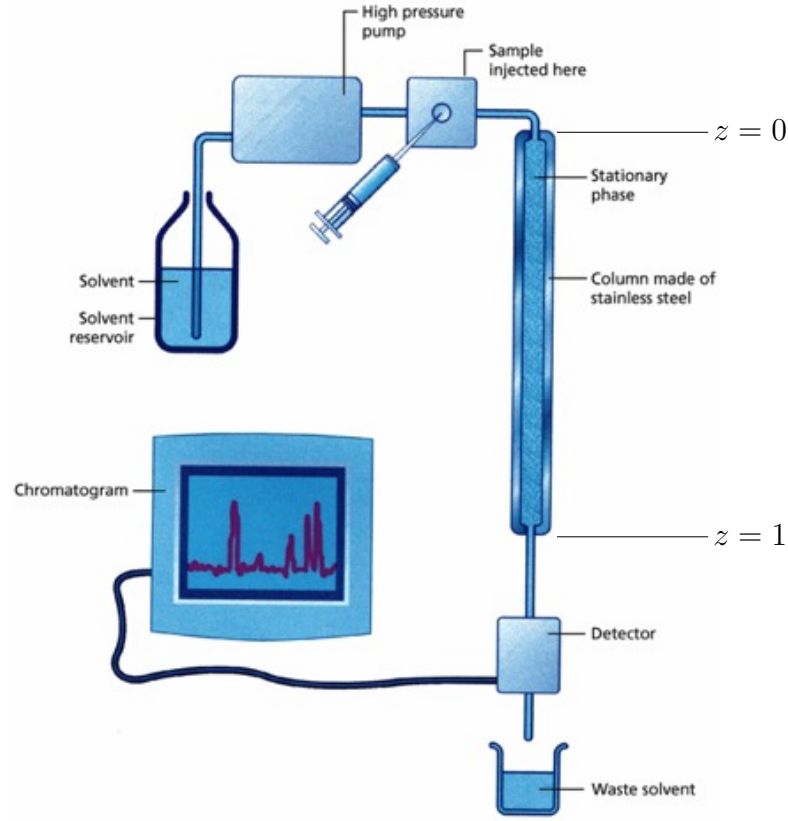


Figure 5.1: Physical Process of Liquid Chromatography. Source: www.pharmaguideline.com.

to advance fully discrete conservative numerical schemes that are obtained by applying suitable time integrators to the spatial semi-discretization. Explicit schemes applied to diffusion problems can strongly restrict the time step due to stability constraints. Therefore we aim to treat diffusion implicitly and convection explicitly. To avoid the necessity to solve nonlinear systems appearing in the implicit treatment of the nonlinear diffusion [21], we propose second-order linearly implicit IMEX Runge-Kutta schemes recently introduced in [8].

5.1.2 The Equilibrium Dispersive (ED) model of chromatography

We denote time by t and let z be the axial coordinate along the column that is normalized to have unit height, so that the top is at $z = 0$ and the bottom at $z = 1$. We assume that ε is the constant total porosity of the solid phase, i.e., the proportion of void space that can be occupied by fluid and u is the (constant) velocity of the mobile phase.

We denote by c_i the concentrations of the i -th liquid phase and by q_i the concentration of solid phase adsorbent permeated by the i -th phase. Thus, the total amount of liquid/solid material occupied by the i -th phase is $\varepsilon c_i + (1 - \varepsilon)q_i$. The flux for the i -th phase is postulated as $\varepsilon(uc_i - D_a \partial c_i / \partial z)$, so that the continuity equations of the ED model can be written as

$$\partial_t (\varepsilon c_i + (1 - \varepsilon)q_i) + \partial_z (\varepsilon(uc_i - D_a \partial c_i / \partial z)) = 0, \quad i = 1, \dots, N. \quad (5.1)$$

We assume that the mobile phase corresponds to the last index N . With the notation $\mathbf{c} := (c_1, \dots, c_N)^T$ and $\mathbf{q} := (q_1, \dots, q_N)^T$ and dividing (5.1) by ε , we obtain the system of continuity equations in the form

$$\partial_t \left(\mathbf{c} + \frac{1 - \varepsilon}{\varepsilon} \mathbf{q} \right) + u \partial_z \mathbf{c} = D_a \partial_{zz} \mathbf{c}. \quad (5.2)$$

Appropriate boundary conditions for this model are proposed in [42], namely

$$u\mathbf{c} - D_a \partial_z \mathbf{c}|_{z=0} = u\mathbf{c}_{\text{inj}}(t), \quad \partial_z \mathbf{c}|_{z=1} = \mathbf{0}, \quad (5.3)$$

for a known function $\mathbf{c}_{\text{inj}}(t)$ that models the continuous injection of the liquid phases (components 1 to $N - 1$) and the “displacer” (component N) through the top of the column. Within the ED model, the equilibrium relationship between the solid phase and liquid phase concentrations is given by the adsorption isotherm $\mathbf{q} = \mathbf{q}(\mathbf{c})$, which is usually a non-linear function [42]. In this chapter we consider multi-component mixtures for which the adsorption isotherms are of Langmuir type, that is

$$q_i = \frac{\alpha_i c_i}{1 + \boldsymbol{\beta}^T \mathbf{c}}, \quad i = 1, \dots, N, \quad (5.4)$$

where $\boldsymbol{\alpha} := (\alpha_1, \dots, \alpha_N)^T$, $\boldsymbol{\beta} := (\beta_1, \dots, \beta_N)^T$, and the constants $\alpha_i, \beta_i > 0$ quantify the nonlinearity of the isotherm. This Langmuir isotherm [53] has the particularity of being

a monotonous and concave function in each component, and is based on thermodynamic statistical models of multicomponent phase equilibrium. If we add terms that consider internal energies by grouping the particles in each phase, the model generalizes to isotherms not necessarily concave [42, 49] and considered in models of chromatography for example in [10].

In [33] the theoretical results about the well-posedness of the ED model (hyperbolicity when $D_a = 0$ and parabolicity when $D_a > 0$) obtained in [58] for $N = 1$ are generalized to $N > 1$, see Section 5.2.

The ED model (5.2) can be written as the system of conservation laws

$$\partial_t \mathbf{w} + \partial_z(u\mathbf{c}) = D_a \partial_{zz} \mathbf{c}, \quad \text{where } \mathbf{w} := \mathbf{W}(\mathbf{c}) := \mathbf{c} + \frac{1 - \varepsilon}{\varepsilon} \mathbf{q}(\mathbf{c}). \quad (5.5)$$

This system can be written in standard form as long as there is a one-to-one correspondence between the variables $\mathbf{w}$ and the concentrations $\mathbf{c}$. Then, a conservative discretization of the terms with spatial derivatives guarantees mass conservation for the conserved variables $\mathbf{w}$, and, as a consequence, the shock-capturing property, i.e. shocks (for $D_a = 0$) or steep profiles (for $D_a > 0$) in the numerical solution propagate at the correct speed.

In [33] it is shown that there is indeed a globally well-defined, one-to-one correspondence between $\mathbf{c}$ and $\mathbf{w}$, so that (5.5) can be rewritten as

$$\partial_t \mathbf{w} + \partial_z \mathbf{f}(\mathbf{w}) = D_a \partial_{zz} \mathbf{C}(\mathbf{w}), \quad \mathbf{f}(\mathbf{w}) = u \mathbf{C}(\mathbf{w}), \quad (5.6)$$

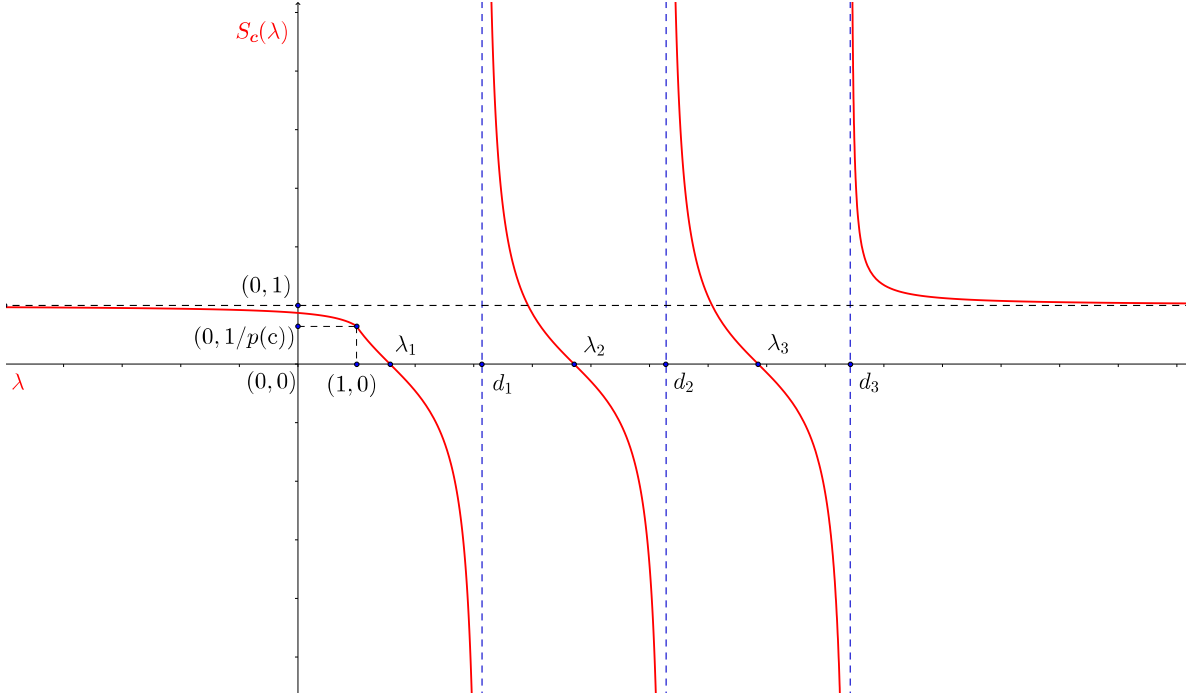
where $\mathbf{C}(\mathbf{w})$ is a continuously differentiable function that satisfies $\mathbf{C} = \mathbf{W}^{-1}$. Furthermore, although there is no explicit expression for the function $\mathbf{C}(\mathbf{w})$ for $N > 1$, the value of $\mathbf{C}(\mathbf{w})$ for any $\mathbf{w} \neq 0$, $w_i \geq 0$ can be determined by computing the only positive root of a particular rational function [33].

5.2 Mathematical structure of the ED model

In what follows we shall assume that the components of the mixture are ordered so that $0 < \alpha_1 < \alpha_2 < \dots < \alpha_N$, see (5.4). We quote here the main results from [33, 35] that will be needed in the sequel.

Theorem 5.2.1 *For any $\mathbf{c} \in \mathbb{R}_+^N$, the Jacobian matrix $\mathbf{W}'(\mathbf{c})$ is diagonalizable, with real, strictly positive, pairwise distinct eigenvalues $\lambda_1, \dots, \lambda_N$ satisfying*

$$1 < \lambda_1 < d_1 < \lambda_2 < \dots < d_{N-1} < \lambda_N < d_N, \quad d_i := 1 + \frac{\eta_i}{1 + \boldsymbol{\beta}^T \mathbf{c}},$$


 Figure 5.2: Eigenvalues of the Jacobian matrix $\mathbf{W}$.

where $\eta_i := (1 - \varepsilon)\alpha_i/\varepsilon$ for $i = 1, \dots, N$.

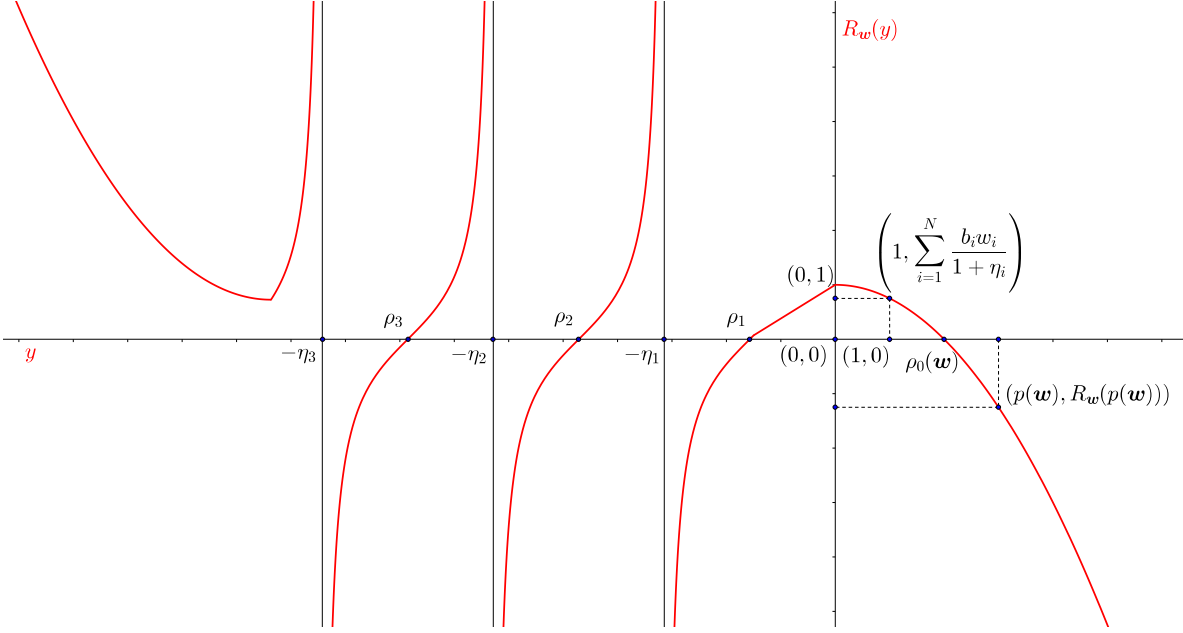
Lemma 5.2.1 *For any fixed $\mathbf{w} \in \mathbb{R}_+^N$, the rational function*

$$\mathbb{R} \ni y \mapsto R_{\mathbf{w}}(y) := 1 - y + \sum_{i=1}^N \frac{y}{y + \eta_i} \beta_i w_i \in \mathbb{R}$$

has only one positive root, denoted by $\rho_0(\mathbf{w})$. In fact, $1 \leq \rho_0(\mathbf{w}) \leq 1 + \boldsymbol{\beta}^T \mathbf{w}$.

Theorem 5.2.2 *The function $\mathbf{W} : \mathbb{R}_+^N \rightarrow \mathbb{R}_+^N$ given in (5.5) is invertible. The inverse function $\mathbf{C} := \mathbf{W}^{-1} : \mathbb{R}_+^N \rightarrow \mathbb{R}_+^N$ is continuously differentiable in $\mathbb{R}_+^N$ and is given by $\mathbf{C} = (C_1(\mathbf{w}), \dots, C_N(\mathbf{w}))^T$, where*

$$C_i(\mathbf{w}) := \frac{w_i}{1 + \eta_i/\rho_0(\mathbf{w})}, \quad i = 1, \dots, N.$$


 Figure 5.3: Positive roots of the rational function $R_{\mathbf{w}}(y)$.

Corollary 5.2.1 *The ED model (5.6) is well posed in the following sense. For $D_a = 0$ the system of conservation laws*

$$\partial_t \mathbf{w} + \partial_z \mathbf{f}(\mathbf{w}) = \mathbf{0}$$

is strictly hyperbolic, and for any $\mathbf{w}$ such that $w_i > 0$, all the eigenvalues μ_j of its Jacobian matrix $u\mathbf{C}'(\mathbf{w})$ are positive, pairwise distinct, and bounded above by u . These eigenvalues $\mu_j = u/\lambda_j$ satisfy

$$u > \mu_1 > u/d_1 > \mu_2 > \dots > u/d_{N-1} > \mu_N > u/d_N > 0$$

(cf. Theorem 5.2.1). For $D_a > 0$, the system (5.6) is parabolic in the sense of Petrovskii (cf., e.g., [38]), i.e., the eigenvalues of the matrix $D_a\mathbf{C}'(\mathbf{w})$ are bounded below by some positive constant for any $\mathbf{w} \in \mathbb{R}_+^N$.

5.3 Linearly implicit IMEX schemes

5.3.1 Method of lines approach

In order to apply this technique to the ED model, we rewrite (5.6) as

$$\partial_t \mathbf{w} + \partial_z \left(\mathbf{f}(\mathbf{w}) - \mathbf{g} \left(\mathbf{w}, \frac{\partial \mathbf{w}}{\partial z} \right) \right) = \mathbf{0}, \quad \mathbf{g} \left(\mathbf{w}, \frac{\partial \mathbf{w}}{\partial z} \right) := D_a \mathbf{C}'(\mathbf{w}) \frac{\partial \mathbf{w}}{\partial z}. \quad (5.7)$$

We consider a uniform mesh with grid points $z_j = (j - 1/2)\Delta z$, $j = 1, \dots, M$, where $\Delta z = 1/M$, and write the resulting semi-discrete scheme as

$$\mathbf{w}'(t) = \mathcal{L}(\mathbf{w}(t)) + \mathcal{D}(\mathbf{w}(t)),$$

where $\mathbf{w}(t)$ is an $M \times N$ matrix whose j -th column, $\mathbf{w}_j(t)$, is an approximation of $w(z_j, t) \in \mathbb{R}^N$, $j = 1, \dots, M$, $\mathcal{L}$ represents the spatial discretization of the convective term $-\partial \mathbf{f}(\mathbf{w})/\partial z$ and $\mathcal{D}$ the spatial discretization of the diffusion term $\partial \mathbf{g}(\mathbf{w}, \partial \mathbf{w}/\partial z)/\partial z$ in (5.7). The schemes we propose compute numerical approximations to the point-values of the conserved variables, $\mathbf{w}_j(t) \approx \mathbf{w}(x_j, t)$, and are characterized by a conservative discretization of the convective and diffusive terms of the form

$$\mathcal{L}_j = -\frac{1}{\Delta z}(\hat{\mathbf{f}}_{j+1/2} - \hat{\mathbf{f}}_{j-1/2}), \quad \mathcal{D}_j = \frac{1}{\Delta z}(\hat{\mathbf{g}}_{j+1/2} - \hat{\mathbf{g}}_{j-1/2})$$

(dropping the dependencies for simplicity), using convective and diffusive numerical fluxes $\hat{\mathbf{f}}_{j+1/2}$ and $\hat{\mathbf{g}}_{j+1/2}$, respectively, that approximate the respective exact fluxes at the corresponding cell interface $z_{j+1/2} = z_j + \Delta z/2$. The convective numerical flux

$$\hat{\mathbf{f}}_{j+1/2} = \hat{\mathbf{f}}(\mathbf{w}_{j-1}, \mathbf{w}_j, \mathbf{w}_{j+1}) \quad (5.8)$$

is computed by MUSCL reconstructions [74] as follows, where we take into account that all characteristic velocities are positive (see Corollary 5.2.1):

$$\begin{aligned} \hat{\mathbf{f}}_{j+1/2} &= u \mathbf{C}(\mathbf{w}_{j+1/2}^L), \\ \mathbf{w}_{i,j+1/2}^L &:= \mathbf{w}_{i,j} + \frac{1}{2} \min \text{mod}(\mathbf{w}_{i,j} - \mathbf{w}_{i,j-1}, \mathbf{w}_{i,j+1} - \mathbf{w}_{i,j}), \end{aligned} \quad (5.9)$$

The diffusive numerical fluxes are computed by second-order centered finite differences

$$\begin{aligned} \hat{\mathbf{g}}_{j+1/2} &= \frac{1}{\Delta z} \mathbf{B}_{j+1/2}(\mathbf{w})(\mathbf{w}_{j+1} - \mathbf{w}_j), \\ \mathbf{B}_{j+1/2}(\mathbf{w}) &:= \mathbf{B}(\mathbf{w}_j, \mathbf{w}_{j+1}) = \frac{D_a}{2} (\mathbf{C}'(\mathbf{w}_{j+1}) + \mathbf{C}'(\mathbf{w}_j)). \end{aligned} \quad (5.10)$$

The boundary conditions (5.3) at $z = 0 = z_{1/2}$ are discretized by prescribing the sum of convective and diffusive numerical fluxes as follows:

$$\hat{\mathbf{f}}_{1/2} - \hat{\mathbf{g}}_{1/2} = u \mathbf{c}_{\text{inj}}(t) = \mathbf{f}(\mathbf{w}) - \mathbf{g}\left(\mathbf{w}, \frac{\partial \mathbf{w}}{\partial z}\right) \Big|_{z=0}.$$

The term $\mathcal{L}_1 + \mathcal{D}_1$ is modified accordingly:

$$\begin{aligned} \mathcal{L}_1 + \mathcal{D}_1 &= \frac{1}{\Delta z} \left((-\hat{\mathbf{f}}_{3/2} + \hat{\mathbf{g}}_{3/2}) - (-\hat{\mathbf{f}}_{1/2} + \hat{\mathbf{g}}_{1/2}) \right) \\ &= \frac{1}{\Delta z} (-\hat{\mathbf{f}}_{3/2} + \hat{\mathbf{g}}_{3/2} + u \mathbf{c}_{\text{inj}}(t)). \end{aligned}$$

The boundary conditions (5.3) at $z = 1 = z_{M+1/2}$ are discretized by taking

$$\hat{\mathbf{g}}_{M+1/2} = \mathbf{0} = \mathbf{g}\left(\mathbf{w}, \frac{\partial \mathbf{w}}{\partial z}\right) \Big|_{z=1},$$

so the term $\mathcal{L}_M + \mathcal{D}_M$ is modified accordingly:

$$\begin{aligned} \mathcal{L}_M + \mathcal{D}_M &= \frac{1}{\Delta z} \left((-\hat{\mathbf{f}}_{M+1/2} + \hat{\mathbf{g}}_{M+1/2}) - (-\hat{\mathbf{f}}_{M-1/2} + \hat{\mathbf{g}}_{M-1/2}) \right) \\ &= \frac{1}{\Delta z} (\hat{\mathbf{f}}_{M-1/2} - \hat{\mathbf{f}}_{M+1/2} - \hat{\mathbf{g}}_{M-1/2}). \end{aligned}$$

Furthermore, the computation of the convective fluxes as in (5.8), for $j = 2, M$ requires values at the corresponding ghost cells z_k , whose indices are $k = 0$ for the first case and $k = M + 1$ for the second. We obtain the values at those ghost cells by using extrapolation with a linear polynomial that satisfies the boundary condition and that interpolates the data for the internal point which is symmetric with respect to the boundary. For $z = 0$ and $k = 1 - j$, taking into account (5.3), this extrapolation yields the value

$$\mathbf{c}_{1-j}(t) = -\mathbf{c}_j(t) + \frac{2(j-1/2)h}{(j-1/2)h + D_a/u} \mathbf{c}_{\text{inj}}(t).$$

For $z = 1$ and $k = M + 1$, in view of (5.3) we obtain by extrapolation

$$\mathbf{c}_{M+1}(t) = \mathbf{c}_M(t).$$

With all these comments and the notation in (5.10) we can write the resulting spatial semi-discretization as

$$\frac{d}{dt} \mathbf{U}(t) = \mathcal{L}(\mathbf{U}, t) + \mathcal{D}(\mathbf{U}), \quad (5.11)$$

where

$$\mathbf{U}(t) = \begin{pmatrix} \mathbf{w}_1(t) \\ \mathbf{w}_2(t) \\ \vdots \\ \mathbf{w}_M(t) \end{pmatrix}, \quad \mathcal{L}(\mathbf{U}, t) = \begin{pmatrix} \mathcal{L}_1(\mathbf{U}, t) \\ \mathcal{L}_2(\mathbf{U}, t) \\ \vdots \\ \mathcal{L}_M(\mathbf{U}, t) \end{pmatrix}, \quad \mathcal{D}(\mathbf{U}) = \frac{1}{\Delta z^2} \mathcal{B}(\tilde{\mathbf{U}}) \mathbf{U},$$

where the sub-vectors $\mathcal{L}_1, \dots, \mathcal{L}_M$ are given by

$$\mathcal{L}_j(\mathbf{U}, t) = \frac{1}{\Delta z} \begin{cases} -\hat{\mathbf{f}}_{j+1/2}(\mathbf{U}) + \hat{\mathbf{f}}_{j-1/2}(\mathbf{U}) & \text{for } 1 < j < M, \\ -\hat{\mathbf{f}}_{3/2}(\mathbf{U}) + u \mathbf{c}_{\text{inj}}(t) & \text{for } j = 1, \\ \hat{\mathbf{f}}_{M-1/2}(\mathbf{U}) & \text{for } j = M, \end{cases}$$

and $\mathcal{B}$ is an $\mathbb{R}^{(NM) \times (NM)}$ block tridiagonal matrix function formed of blocks $\mathbf{B}_{j+1/2} \in \mathbb{R}^{N \times N}$ such that

$$(\mathcal{B}(\tilde{\mathbf{U}}) \mathbf{U})_j = \begin{cases} \mathbf{B}_{j+1/2}(\tilde{\mathbf{U}})(\mathbf{w}_{j+1} - \mathbf{w}_j) \\ \quad - \mathbf{B}_{j-1/2}(\tilde{\mathbf{U}})(\mathbf{w}_j - \mathbf{w}_{j-1}) & \text{for } 1 < j < M, \\ \mathbf{B}_{3/2}(\tilde{\mathbf{U}})(\mathbf{w}_2 - \mathbf{w}_1) & \text{for } j = 1, \\ -\mathbf{B}_{M-1/2}(\tilde{\mathbf{U}})(\mathbf{w}_M - \mathbf{w}_{M-1}) & \text{for } j = M. \end{cases}$$

we observe that the only stiff term is the linear term $\mathbf{U}$ that multiplies $\mathcal{B}(\tilde{\mathbf{U}})$. Therefore we treat $\tilde{\mathbf{U}}$ explicitly as an argument of $\mathbf{f}$ and $\mathcal{B}$, while $\mathbf{U}$ is implicit in the term to which $\mathcal{B}$ is applied.

The approximate solution of (5.11) can be obtained by the application of Runge-Kutta ODE solvers. Strong Stability Preserving (SSP) explicit Runge-Kutta methods are a popular class of time integrators whose use leads to the following stability constraint (see [34] for details):

$$\Delta t \left(\frac{u}{\Delta z} + \frac{2D_a}{\Delta z^2} \right) \max_{\mathbf{U}} \varrho(\mathbf{C}'(\mathbf{U})) \leq C_0 \leq 1 \quad \text{for some constant } C_0 > 0,$$

where $\varrho(\mathbf{C}'(\mathbf{U}))$ is the spectral radius of $\mathbf{C}'(\mathbf{U})$. Since $\max_{\mathbf{U}} \varrho(\mathbf{C}'(\mathbf{U})) < 1$, a practical bound would be

$$\Delta t \left(\frac{u}{\Delta z} + \frac{2D_a}{\Delta z^2} \right) \leq C_0 \tag{5.12}$$

This restriction on Δt can be very stringent if D_a is not very small or Δz is very small. This restriction originates from the explicit treatment of the diffusion term.

An alternative to explicit Runge-Kutta methods are implicit-explicit Runge-Kutta (IMEX-RK) methods (see [8] and references therein), for which only the diffusion term is treated implicitly. It is hoped (and can be proven in some cases) that the stability restrictions on Δt are of the form

$$\Delta t \frac{u}{\Delta z} \max_w \varrho(\mathbf{C}'(\mathbf{w})) \leq C_1 \leq 1,$$

or, from a practical point of view,

$$\Delta t \frac{u}{\Delta z} \leq C_1. \quad (5.13)$$

An IMEX-RK method is specified by the pair of Butcher arrays 1.7 (see Section 1.2). To overcome the excessive numerical work for the solution of nonlinear systems, an essential gain is obtained by the approach proposed in [8]. We rewrite the semidiscrete formulation (5.11) in the form

$$\begin{aligned} \frac{d}{dt} \tilde{\mathbf{U}} &= \mathbf{\kappa}(\tilde{\mathbf{U}}, \mathbf{U}, t), \\ \frac{d}{dt} \mathbf{U} &= \mathbf{\kappa}(\tilde{\mathbf{U}}, \mathbf{U}, t), \end{aligned}$$

with initial condition $\tilde{\mathbf{w}}_j(0) = \mathbf{w}_j(0) = \mathbf{w}(x_j, 0)$, $j = 1, \dots, M$, where we use the notation

$$\mathbf{\kappa}(\tilde{\mathbf{U}}, \mathbf{U}, t) := \mathbf{\mathcal{L}}(\tilde{\mathbf{U}}, t) + \frac{1}{\Delta z^2} \mathbf{\mathcal{B}}(\tilde{\mathbf{U}}) \mathbf{U},$$

with the aim of applying a partitioned Runge-Kutta scheme, consisting in the application of the explicit part to the first block of equations and the implicit part to the second block. If both Butcher arrays satisfy $\tilde{\mathbf{b}} = \mathbf{b}$, then the step from t^n to $t^{n+1} = t^n + \Delta t$ of the linearly implicit IMEX-RK scheme is given by the Algorithm 4.3.2.

The linear equation for $\mathbf{K}_i$ is solved by using standard and efficient block tridiagonal solvers (see [40]).

In our Examples we use the classical second-order IMEX-RK scheme given by the following Butcher arrays:

$$\begin{array}{c|c} \tilde{\mathbf{c}} & \tilde{\mathbf{A}} \\ \hline & \tilde{\mathbf{b}}^T \end{array} = \begin{array}{c|cc} 0 & 0 & 0 \\ 1 & 1 & 0 \\ \hline & 1/2 & 1/2 \end{array}, \quad \begin{array}{c|c} \mathbf{c} & \mathbf{A} \\ \hline & \mathbf{b}^T \end{array} = \begin{array}{c|cc} 0 & 0 & 0 \\ 1 & 1/2 & 1/2 \\ \hline & 1/2 & 1/2 \end{array},$$

denoted by H-CN(2,2,2) in [8], since it is a natural choice when dealing with convection-diffusion problems, since Heun's method is an SSP explicit RK one [39], and the Crank-Nicolson method is A -stable and widely used for diffusion problems.

5.4 Numerical Examples

5.4.1 Preliminaries

In this section we perform several numerical Examples to illustrate the behavior of the proposed IMEX-WENO scheme. Specifically, we consider the simulation of a three-component mixture proposed in [50].

Displacement chromatography (see, e.g., [54, 62]) relies on the idea that one component (the displacer) has a stronger affinity to the solid phase than any of the other components in the sample mixture, hence it has the capability to displace the other components of the mixture from the stationary phase. For a sufficiently long column and appropriate adsorption isotherms, the concentrations of the components form rectangular regions of high concentration of one component in the mixture. The series of such zones are the so-called isotachic train [25]. We consider the case of a mixture of two components and one displacer proposed in [50, Sect. 4.3]. The values of the parameters are: $\alpha_1 = 4, \alpha_2 = 5, \alpha_3 = 6, \beta_1 = 4, \beta_2 = 5, \beta_3 = 1$. In addition $\varepsilon = 0.5$ and $u = 0.2$.

For each iteration, Δt is determined by (5.12), (5.13) with the appropriate parameters $C_0 = C_1 = 1$. Moreover, we compute approximate L^1 errors at different times for each scheme as follows. We denote by $(c_{i,j}^M(t))_{j=1}^M$ and $(c_{i,l}^{\text{ref}}(t))_{l=1}^{M_{\text{ref}}}$ the numerical solution for the i -th component at time t calculated with M and $M_{\text{ref}} = 12800$ cells, respectively. We compute $\tilde{c}_{i,j}^{\text{ref}}(t)$ for $j = 1, \dots, M$ by

$$\tilde{c}_{i,j}^{\text{ref}}(t) = \frac{1}{R} \sum_{k=1}^R c_{i,R(j-1)+k}^{\text{ref}}(t), \quad R = M_{\text{ref}}/M.$$

The total approximate L^1 error $(c_{i,j}^M(t))_{j=1}^M$ at time t is then given by

$$e_M^{\text{tot}}(t) := \frac{1}{M} \sum_{i=1}^N \sum_{j=1}^M |\tilde{c}_{i,j}^{\text{ref}}(t) - c_{i,j}^M(t)|.$$

Based on these approximate total errors, we may calculate a numerical order of convergence from pairs $e_{M/2}^{\text{tot}}(t)$ and $e_M^{\text{tot}}(t)$ by

$$\theta_M(t) := \log_2(e_{M/2}^{\text{tot}}(t)/e_M^{\text{tot}}(t)).$$

5.4.2 Three-component displacement tests

Example 5.1

We assume that components 1 and 2 are injected between $t = 0$ and $t = 0.1$ with $c_1 = c_2 = 1$ at $z = 0$. Component 3, the displacer, is injected from $t = 0.1$ with $c_3 = 1$, that is,

$$\mathbf{c}_{\text{inj}}(t) = \begin{cases} (1, 1, 0)^T & \text{for } 0 \leq t \leq 0.1, \\ (0, 0, 1)^T & \text{for } t > 0.1. \end{cases}$$

Figures 5.4 (a) and (b) show the results for Example 5.1 for $D_a = 10^{-4}$ and $D_a = 10^{-3}$, respectively, obtained by the IMEX-RK scheme for the reference resolution $M = M_{\text{ref}} = 12800$. The formation of the displacement train can be clearly appreciated in both simulations. This fact (see [50]) prevents the formation of a rectangular pulse for component 1. This makes that only the second component can form a rectangular pulse. We see that the numerical solution behaves as expected.

Figures 5.5 and 5.6 display the numerical solution at times $T = 4$ and $T = 12$, respectively, obtained by both schemes (the explicit and the linearly implicit IMEX-RK scheme, denoted henceforth by “IMEX-RK2”) at an intermediate spatial discretization $M = 400$ and with $D_a = 10^{-4}$, while Table 5.1 shows the corresponding approximate errors and CPU times for both $D_a = 10^{-4}$ and $D_a = 10^{-3}$. Both figures, as well as Table 5.1, indicate that the explicit and IMEX-RK2 schemes produce numerical solutions of nearly the same quality; however, both numerical solutions are significantly more “smeared out” than the reference solution, which becomes especially appreciable near transitions between zones of slow variation and steep gradients.

Since $\mathbf{c}_{\text{inj}}(t)$ is discontinuous at $t = 0.1$ here and in Examples 5.2 and 5.3, we avoid a loss of order of accuracy in the time integration by ensuring that $t_n = 0.1$ for some intermediate n , by shortening Δt accordingly.

Example 5.2

Next, we assume that the concentration injected for the displacer is further reduced, so that

$$\mathbf{c}_{\text{inj}}(t) = \begin{cases} (1, 1, 0)^T & \text{for } 0 \leq t \leq 0.1, \\ (0, 0, 0.5)^T & \text{for } t > 0.1. \end{cases}$$

5.4. Numerical Examples

The values of the rest of the parameters are the same as in Example 5.1. In this case, none of the isotherms is intersected by the operating line and both components fail to form equilibrated rectangular pulses. The results of the reference solution are shown in Figures 5.4 (c) and (d), Figures 5.7 and 5.8 display numerical solutions for $M = 400$, and Table 5.2 provides the corresponding error and CPU time information.

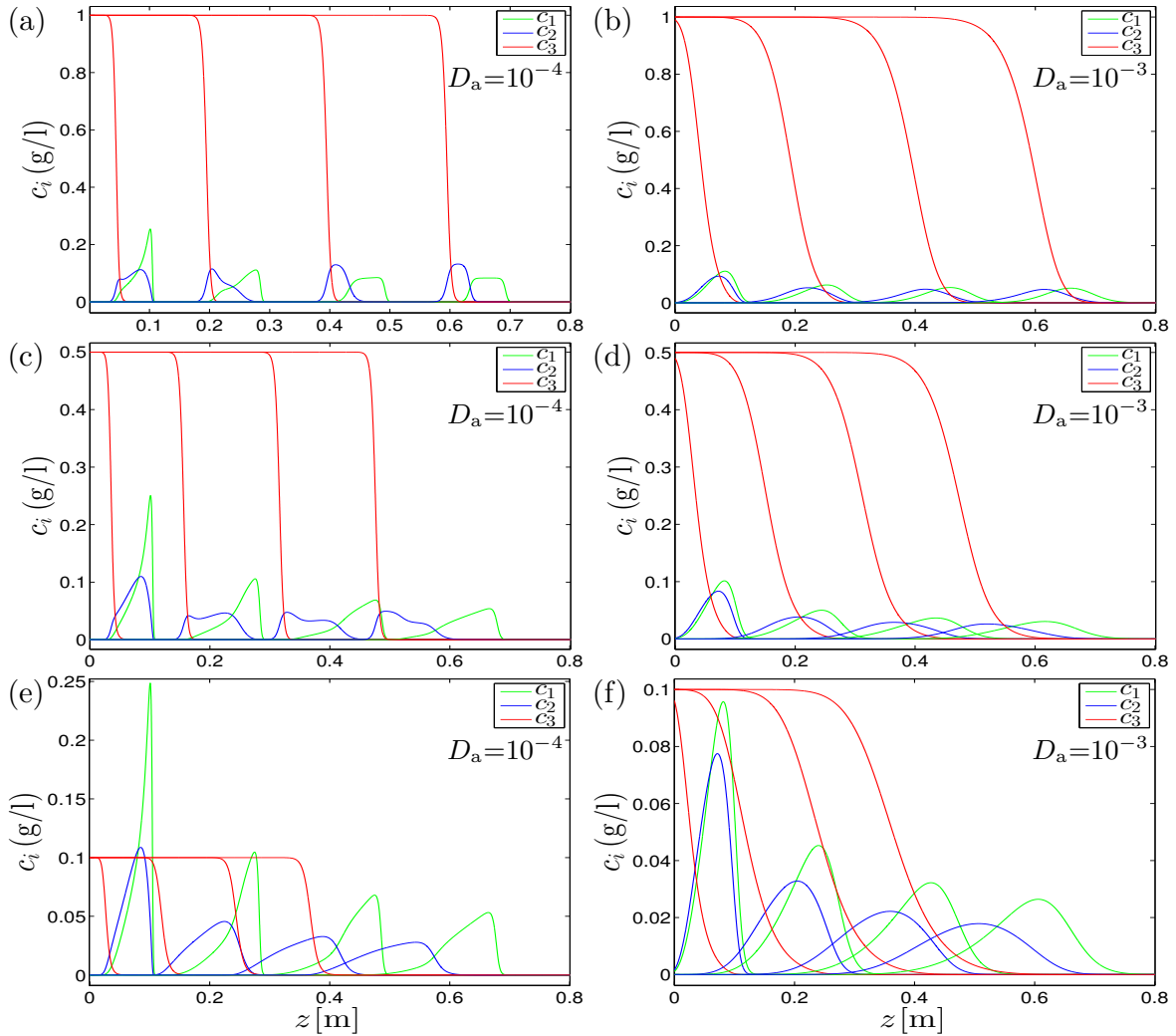


Figure 5.4: Reference numerical solution obtained by IMEX-RK2 at final times $T = 1, 4, 8, 12$ for (a, b) Example 5.1, (c, d) Example 5.2 and (e, f) Example 5.3.

Example 5.3

The concentration injected for the displacer is even further reduced, so that

$$\mathbf{c}_{\text{inj}}(t) = \begin{cases} (1, 1, 0)^T & \text{for } 0 \leq t \leq 0.1, \\ (0, 0, 0.1)^T & \text{for } t > 0.1. \end{cases}$$

The values of the rest of the parameters are the same as in Examples 5.1 and 5.2. In this case, none of the isotherms is intersected by the operating line. According to this, both components fail to form equilibrated rectangular pulses. The results of the reference solution are shown in Figure 5.4 (e) and (f), Figures 5.9 and 5.10 display numerical solutions for $M = 400$, and Table 5.3 provides the corresponding error and CPU time information.

5.4.3 Numerical error and efficiency

To assess the numerical error and efficiency of the linearly implicit IMEX-RK scheme we utilize the information of Tables 5.1, 5.2 and 5.3 for the final times $T = 4$ and $T = 12$ and the alternative diffusion coefficients $D_a = 10^{-4}$ and $D_a = 10^{-3}$. First of all, this information corroborates what can be inferred from Figures 5.5 to 5.10, namely that the explicit and IMEX-RK2 schemes produce almost the same errors, and moreover the observed rates of convergence θ_M are consistent with the fact that both schemes are formally second order in space and time (where we recall that the MUSCL extrapolation (5.9) is inactive near local extrema of the numerical solution, where the scheme locally degenerates to first-order accuracy). Furthermore, consistently with the respective CFL conditions (5.12) and (5.13) for the explicit and IMEX-RK2 schemes we observe that although each time step is computationally more involved for the IMEX-RK2 than for the explicit scheme, the IMEX-RK2 becomes faster than the explicit one for sufficiently fine discretizations. This gain is more significant, and becomes visible already for coarser discretizations, for the larger of the diffusion coefficients chosen for inspection. The same observation can be deduced from Figures 5.11 and 5.12, where we present a number of efficiency plots, that is, of approximate L^1 error versus CPU time, deduced from the error and CPU time information of Tables 5.1 to 5.3. Here we understand by efficiency the amount of computational effort necessary to achieve a reduction of numerical error.

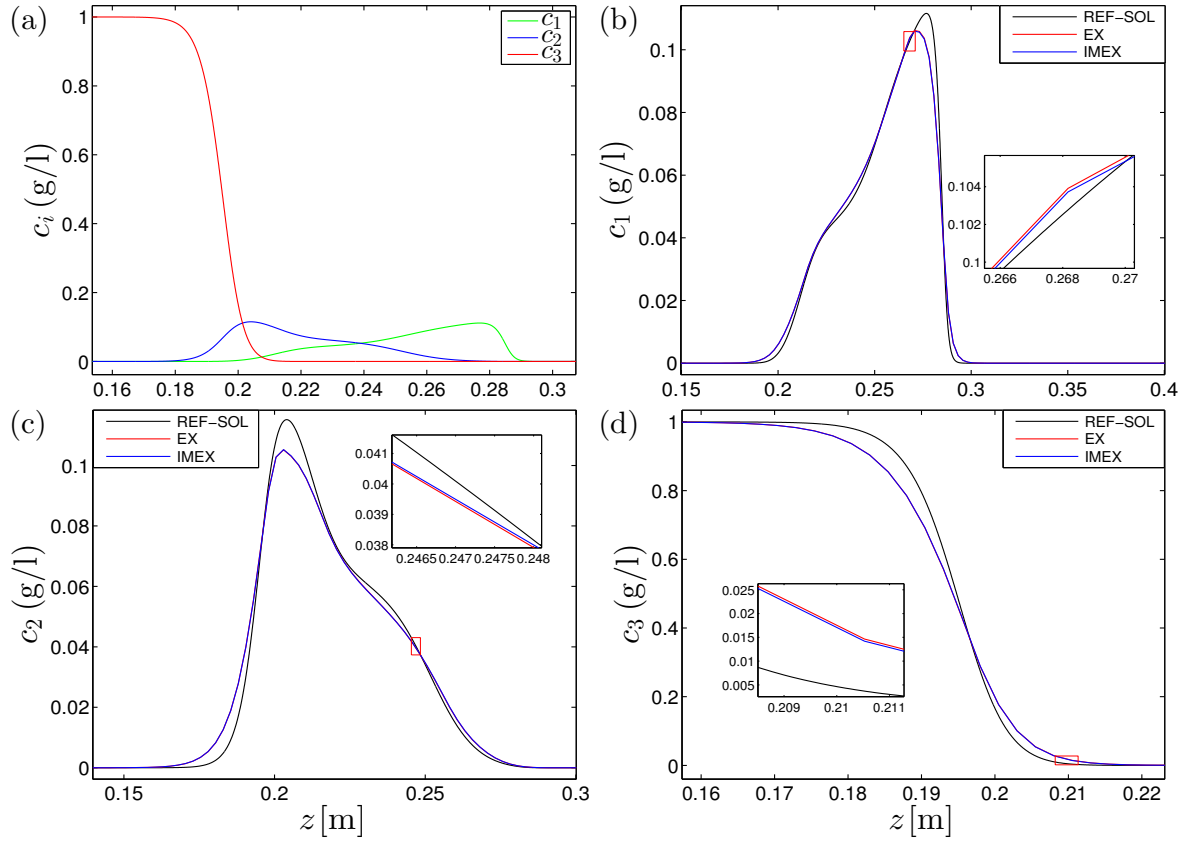


Figure 5.5: Example 5.1: numerical solutions obtained by IMEX-RK2 and explicit methods with $D_a = 10^{-4}$ at final time $T = 4$. The spatial resolution corresponds to $M = 400$, results are labeled by “IMEX” and “EX”, and the reference solution (“REF-SOL”) is defined by $M_{\text{ref}} = 12800$.

5.4. Numerical Examples

		Explicit			IMEX-RK2		
	M	e_M^{tot}	θ_M	cpu [s]	e_M^{tot}	θ_M	cpu [s]
Example 5.1 $T = 1$	50	141.043	—	0.00	118.530	—	0.00
	100	68.429	1.04	0.00	61.557	0.95	0.01
	200	36.594	0.90	0.01	36.460	0.76	0.08
	400	15.548	1.23	0.10	15.633	1.22	0.29
	800	5.273	1.56	0.69	5.432	1.53	1.15
	1600	1.485	1.83	3.54	1.567	1.79	3.28
	M	e_M^{tot}	θ_M	cpu [s]	e_M^{tot}	θ_M	cpu [s]
Example 5.1 $T = 4$	50	160.577	—	0.00	155.641	—	0.01
	100	85.933	0.90	0.01	84.925	0.87	0.03
	200	42.821	1.00	0.09	42.667	0.99	0.23
	400	16.657	1.36	0.66	16.824	1.34	1.12
	800	5.159	1.69	3.69	5.260	1.68	5.06
	1600	1.341	1.94	22.77	1.408	1.90	22.28
	M	e_M^{tot}	θ_M	cpu [s]	e_M^{tot}	θ_M	cpu [s]
Example 5.1 $T = 8$	50	198.165	—	0.01	197.344	—	0.02
	100	107.076	0.89	0.03	107.476	0.88	0.07
	200	49.010	1.13	0.18	49.117	1.13	0.36
	400	17.742	1.47	1.13	17.928	1.45	1.82
	800	5.307	1.74	7.30	5.400	1.73	9.90
	1600	1.367	1.96	50.26	1.412	1.94	47.24
	M	e_M^{tot}	θ_M	cpu [s]	e_M^{tot}	θ_M	cpu [s]
Example 5.1 $T = 12$	50	207.584	—	0.01	207.958	—	0.03
	100	115.315	0.85	0.05	115.895	0.84	0.11
	200	52.878	1.12	0.29	53.194	1.12	0.50
	400	19.811	1.42	1.52	19.981	1.41	2.45
	800	6.017	1.72	9.82	6.089	1.71	13.67
	1600	1.563	1.94	68.56	1.589	1.94	62.19

Table 5.1: Approximate L^1 errors (e_M^{tot} , figures to be multiplied by 10^{-3}), convergence rates (θ_M) and CPU times (cpu). We use Upwind Method to approximate the convective term and Finite Difference for diffusive term. The diffusion coefficient used was $D_a = 10^{-4}$.

5.4. Numerical Examples

		Explicit			IMEX-RK2		
	M	e_M^{tot}	θ_M	cpu [s]	e_M^{tot}	θ_M	cpu [s]
Example 5.2 $T = 1$	50	138.123	—	0.00	110.223	—	0.00
	100	62.363	1.15	0.00	54.662	1.01	0.01
	200	31.554	0.98	0.06	28.875	0.92	0.16
	400	13.003	1.28	0.13	12.491	1.21	0.48
	800	4.150	1.65	1.27	4.366	1.52	1.42
	1600	1.147	1.86	7.67	1.249	1.81	3.93
	M	e_M^{tot}	θ_M	cpu [s]	e_M^{tot}	θ_M	cpu [s]
Example 5.2 $T = 4$	50	134.469	—	0.00	125.243	—	0.01
	100	68.048	0.98	0.02	66.023	0.92	0.04
	200	33.060	1.04	0.24	32.170	1.04	0.39
	400	12.374	1.42	0.86	12.467	1.37	1.55
	800	3.784	1.71	5.40	3.881	1.68	5.42
	1600	0.990	1.93	23.11	1.039	1.90	17.06
	M	e_M^{tot}	θ_M	cpu [s]	e_M^{tot}	θ_M	cpu [s]
Example 5.2 $T = 8$	50	141.953	—	0.01	137.863	—	0.02
	100	67.383	1.07	0.06	66.577	1.05	0.09
	200	31.907	1.08	0.34	31.664	1.07	0.55
	400	11.916	1.42	1.41	12.058	1.39	2.43
	800	3.534	1.75	9.54	3.610	1.74	10.09
	1600	0.909	1.96	45.83	0.951	1.92	34.76
	M	e_M^{tot}	θ_M	cpu [s]	e_M^{tot}	θ_M	cpu [s]
Example 5.2 $T = 12$	50	144.528	—	0.02	142.593	—	0.04
	100	66.308	1.12	0.09	66.392	1.10	0.13
	200	30.631	1.11	0.43	30.615	1.12	0.71
	400	11.162	1.46	1.89	11.238	1.45	3.17
	800	3.294	1.76	12.63	3.400	1.72	3.67
	1600	0.842	1.97	63.33	0.897	1.92	47.57

Table 5.2: Approximate L^1 errors (e_M^{tot} , figures to be multiplied by 10^{-3}), convergence rates (θ_M) and CPU times (cpu). We use Upwind Method to approximate the convective term and Finite Difference for diffusive term. The diffusion coefficient used was $D_a = 10^{-4}$.

5.4. Numerical Examples

		Explicit			IMEX-RK2		
	M	e_M^{tot}	θ_M	cpu [s]	e_M^{tot}	θ_M	cpu [s]
Example 5.3 $T = 1$	50	101.976	—	0.00	93.112	—	0.00
	100	57.804	0.82	0.00	51.422	0.86	0.01
	200	29.373	0.98	0.03	25.839	0.99	0.10
	400	11.113	1.40	0.12	10.201	1.34	0.34
	800	3.410	1.70	0.86	3.606	1.50	1.12
	1600	0.873	1.97	4.33	1.008	1.84	3.87
	M	e_M^{tot}	θ_M	cpu [s]	e_M^{tot}	θ_M	cpu [s]
Example 5.3 $T = 4$	50	103.436	—	0.00	99.208	—	0.01
	100	56.514	0.87	0.02	53.569	0.89	0.04
	200	25.372	1.16	0.13	23.806	1.17	0.25
	400	8.878	1.51	0.77	8.547	1.48	1.29
	800	2.496	1.83	5.32	2.603	1.71	4.73
	1600	0.612	2.03	19.74	0.682	1.93	16.92
	M	e_M^{tot}	θ_M	cpu [s]	e_M^{tot}	θ_M	cpu [s]
Example 5.3 $T = 8$	50	107.073	—	0.01	104.789	—	0.02
	100	53.895	0.99	0.04	51.716	1.02	0.08
	200	21.924	1.30	0.24	20.982	1.30	0.39
	400	7.603	1.53	1.36	7.422	1.50	2.08
	800	2.138	1.83	10.72	2.229	1.74	9.18
	1600	0.518	2.04	41.63	0.581	1.94	35.63
	M	e_M^{tot}	θ_M	cpu [s]	e_M^{tot}	θ_M	cpu [s]
Example 5.3 $T = 12$	50	107.299	—	0.02	105.659	—	0.03
	100	51.060	1.07	0.06	49.203	1.10	0.12
	200	20.082	1.35	0.34	19.321	1.35	0.55
	400	6.966	1.53	1.91	6.844	1.50	2.75
	800	1.926	1.85	14.60	2.038	1.75	12.36
	1600	0.466	2.05	59.07	0.530	1.94	49.78

Table 5.3: Approximate L^1 errors (e_M^{tot} , figures to be multiplied by 10^{-3}), convergence rates (θ_M) and CPU times (cpu). We use Upwind Method to approximate the convective term and Finite Difference for diffusive term. The diffusion coefficient used was $D_a = 10^{-4}$.

5.4. Numerical Examples

		Explicit			IMEX-RK2		
	M	e_M^{tot}	θ_M	cpu [s]	e_M^{tot}	θ_M	cpu [s]
Example 5.1 $T = 1$	50	63.344	—	0.00	58.860	—	0.00
	100	21.213	1.58	0.00	22.924	1.36	0.00
	200	6.911	1.62	0.05	7.648	1.58	0.04
	400	1.756	1.98	0.51	2.068	1.89	0.28
	800	0.423	2.06	4.20	0.544	1.93	1.41
	1600	0.103	2.04	40.63	0.159	1.78	6.02
	M	e_M^{tot}	θ_M	cpu [s]	e_M^{tot}	θ_M	cpu [s]
Example 5.1 $T = 4$	50	65.175	—	0.00	63.798	—	0.01
	100	21.373	1.61	0.03	21.287	1.58	0.03
	200	5.999	1.83	0.23	5.973	1.83	0.14
	400	1.503	2.00	1.60	1.498	1.99	0.82
	800	0.367	2.03	17.93	0.369	2.02	5.40
	1600	0.089	2.04	119.73	0.092	2.01	18.62
	M	e_M^{tot}	θ_M	cpu [s]	e_M^{tot}	θ_M	cpu [s]
Example 5.1 $T = 8$	50	69.264	—	0.01	68.589	—	0.02
	100	21.559	1.68	0.06	21.491	1.67	0.06
	200	5.783	1.90	0.47	5.754	1.90	0.28
	400	1.448	2.00	2.96	1.434	2.00	1.55
	800	0.357	2.02	37.25	0.353	2.02	8.43
	1600	0.088	2.03	205.27	0.086	2.04	31.11
	M	e_M^{tot}	θ_M	cpu [s]	e_M^{tot}	θ_M	cpu [s]
Example 5.1 $T = 12$	50	71.154	—	0.02	70.996	—	0.03
	100	21.958	1.70	0.09	21.999	1.69	0.10
	200	5.795	1.92	0.73	5.817	1.92	0.43
	400	1.452	2.00	4.43	1.453	2.00	2.36
	800	0.360	2.01	51.05	0.360	2.02	11.72
	1600	0.088	2.02	295.10	0.088	2.04	44.16

Table 5.4: Approximate L^1 errors (e_M^{tot} , figures to be multiplied by 10^{-3}), convergence rates (θ_M) and CPU times (cpu). We use Upwind Method to approximate the convective term and Finite Difference for diffusive term. The diffusion coefficient used was $D_a = 10^{-3}$.

5.4. Numerical Examples

		Explicit			IMEX-RK2		
	M	e_M^{tot}	θ_M	cpu [s]	e_M^{tot}	θ_M	cpu [s]
Example 5.2 $T = 1$	50	51.746	—	0.00	48.642	—	0.00
	100	17.708	1.55	0.00	19.341	1.33	0.01
	200	5.644	1.65	0.07	6.693	1.53	0.05
	400	1.529	1.88	0.68	1.902	1.81	0.32
	800	0.378	2.01	5.23	0.526	1.85	1.44
	1600	0.092	2.04	33.73	0.169	1.63	5.11
	M	e_M^{tot}	θ_M	cpu [s]	e_M^{tot}	θ_M	cpu [s]
Example 5.2 $T = 4$	50	49.780	—	0.01	48.121	—	0.01
	100	16.137	1.63	0.04	16.597	1.54	0.03
	200	4.651	1.79	0.28	4.852	1.77	0.17
	400	1.182	1.98	2.22	1.258	1.95	0.86
	800	0.290	2.03	17.82	0.319	1.98	4.09
	1600	0.071	2.04	123.10	0.087	1.88	16.77
	M	e_M^{tot}	θ_M	cpu [s]	e_M^{tot}	θ_M	cpu [s]
Example 5.2 $T = 8$	50	49.747	—	0.02	49.116	—	0.02
	100	15.689	1.66	0.08	15.829	1.63	0.08
	200	4.282	1.87	0.58	4.324	1.87	0.32
	400	1.071	2.00	3.88	1.089	1.99	1.55
	800	0.263	2.03	32.10	0.270	2.01	7.11
	1600	0.064	2.03	214.56	0.069	1.97	29.28
	M	e_M^{tot}	θ_M	cpu [s]	e_M^{tot}	θ_M	cpu [s]
Example 5.2 $T = 12$	50	49.937	—	0.03	49.250	—	0.04
	100	15.331	1.70	0.13	15.390	1.68	0.13
	200	4.078	1.91	0.89	4.094	1.91	0.49
	400	1.017	2.00	5.60	1.023	2.00	2.28
	800	0.250	2.02	48.94	0.253	2.02	10.24
	1600	0.061	2.03	306.46	0.063	2.00	42.23

Table 5.5: Approximate L^1 errors (e_M^{tot} , figures to be multiplied by 10^{-3}), convergence rates (θ_M) and CPU times (cpu). We use Upwind Method to approximate the convective term and Finite Difference for diffusive term. The diffusion coefficient used was $D_a = 10^{-3}$.

5.4. Numerical Examples

		Explicit			IMEX-RK2		
	M	e_M^{tot}	θ_M	cpu [s]	e_M^{tot}	θ_M	cpu [s]
Example 5.3 $T = 1$	50	41.151	—	0.00	40.734	—	0.00
	100	14.208	1.53	0.00	15.384	1.40	0.01
	200	4.189	1.76	0.04	5.392	1.51	0.05
	400	1.171	1.84	0.59	1.673	1.69	0.31
	800	0.307	1.93	5.28	0.497	1.75	1.49
	1600	0.077	1.99	34.46	0.169	1.56	5.18
	M	e_M^{tot}	θ_M	cpu [s]	e_M^{tot}	θ_M	cpu [s]
Example 5.3 $T = 4$	50	33.902	—	0.01	31.962	—	0.01
	100	10.045	1.75	0.03	10.854	1.56	0.03
	200	2.900	1.79	0.24	3.358	1.69	0.18
	400	0.784	1.89	1.92	0.951	1.82	0.85
	800	0.200	1.97	18.06	0.262	1.86	4.27
	1600	0.050	2.01	120.61	0.081	1.70	17.13
	M	e_M^{tot}	θ_M	cpu [s]	e_M^{tot}	θ_M	cpu [s]
Example 5.3 $T = 8$	50	30.668	—	0.02	29.654	—	0.02
	100	9.020	1.77	0.07	9.524	1.64	0.07
	200	2.487	1.86	0.50	2.732	1.80	0.36
	400	0.649	1.94	3.52	0.740	1.88	1.51
	800	0.164	1.99	32.17	0.198	1.90	7.33
	1600	0.040	2.01	205.80	0.059	1.74	29.78
	M	e_M^{tot}	θ_M	cpu [s]	e_M^{tot}	θ_M	cpu [s]
Example 5.3 $T = 12$	50	28.628	—	0.03	28.158	—	0.03
	100	8.418	1.77	0.11	8.786	1.68	0.11
	200	2.271	1.89	0.73	2.441	1.85	0.54
	400	0.584	1.96	5.17	0.647	1.92	2.21
	800	0.146	2.00	46.85	0.170	1.93	10.40
	1600	0.036	2.02	296.05	0.050	1.77	42.45

Table 5.6: Approximate L^1 errors (e_M^{tot} , figures to be multiplied by 10^{-3}), convergence rates (θ_M) and CPU times (cpu). We use Upwind Method to approximate the convective term and Finite Difference for diffusive term. The diffusion coefficient used was $D_a = 10^{-3}$.

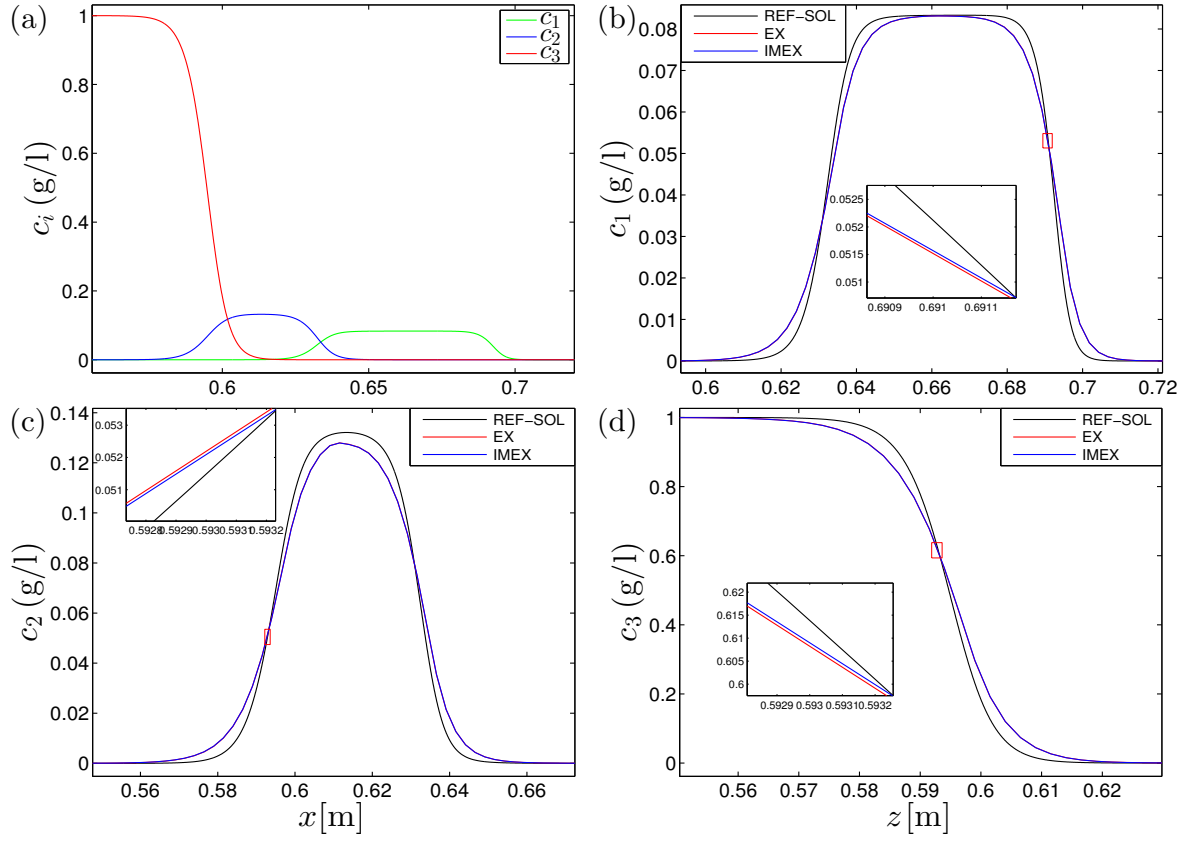


Figure 5.6: Example 5.1: numerical solutions for $D_a = 10^{-4}$ and final time $T = 12$. The spatial resolution corresponds to $M = 400$, results are labeled by “IMEX” and “EX”, and the reference solution (“REF-SOL”) is defined by $M_{\text{ref}} = 12800$.

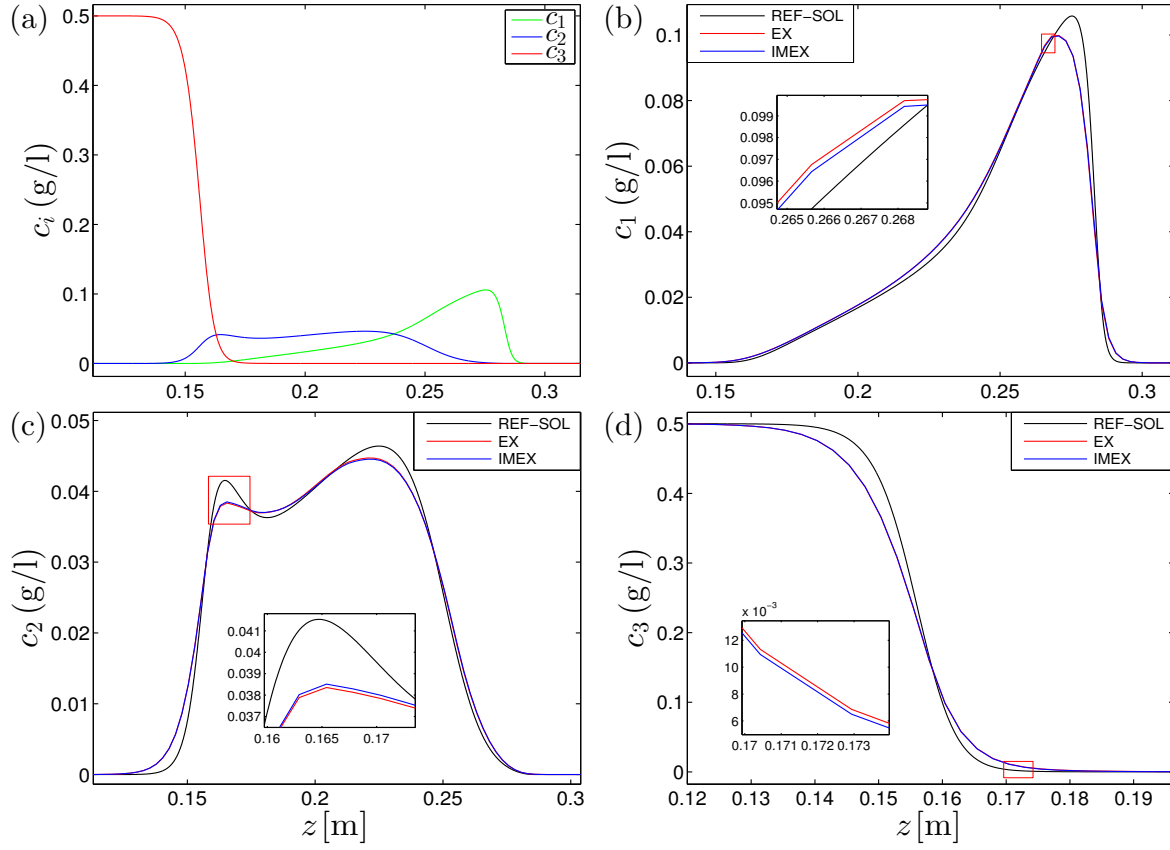


Figure 5.7: Example 5.2: numerical solutions for $D_a = 10^{-4}$ and final time $T = 4$. The spatial resolution corresponds to $M = 400$, results are labeled by “IMEX” and “EX”, and the reference solution (“REF-SOL”) is defined by $M_{\text{ref}} = 12800$.

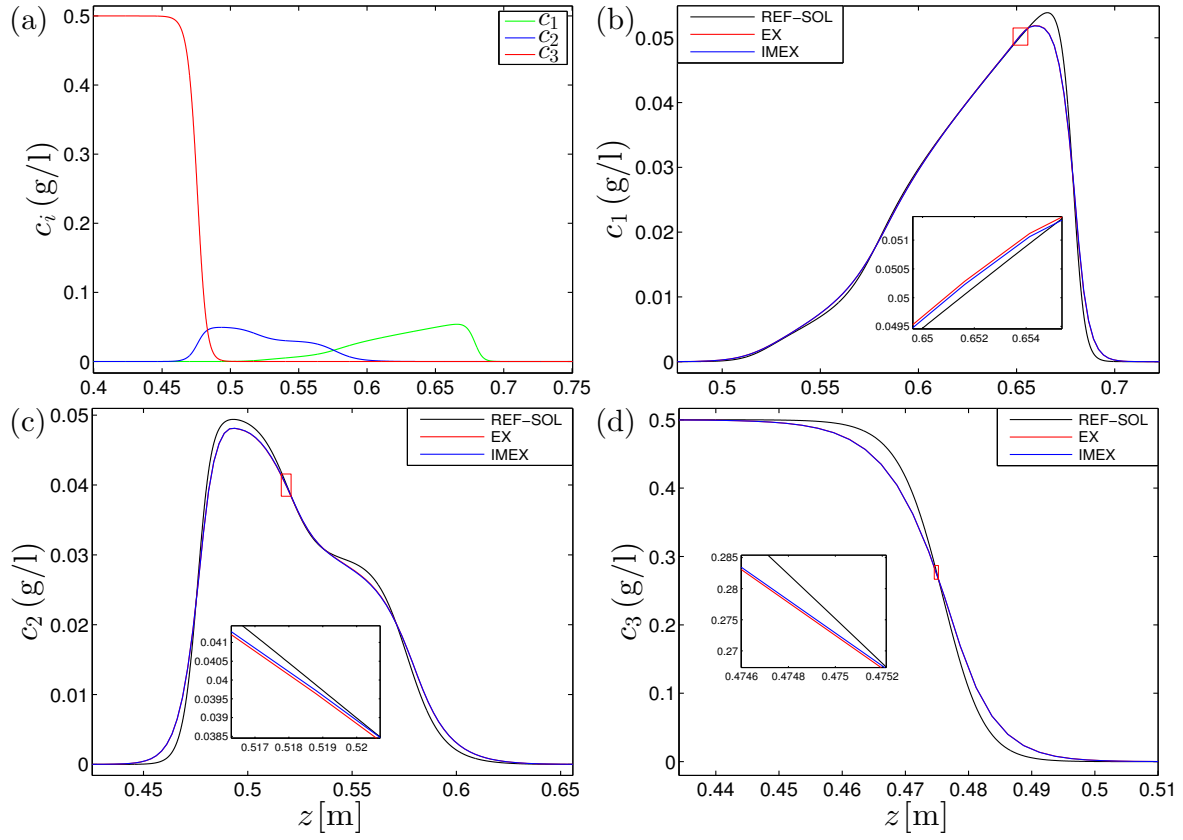


Figure 5.8: Example 5.2: numerical solutions for $D_a = 10^{-4}$ and final time $T = 12$. The spatial resolution corresponds to $M = 400$, results are labeled by “IMEX” and “EX”, and the reference solution (“REF-SOL”) is defined by $M_{\text{ref}} = 12800$.

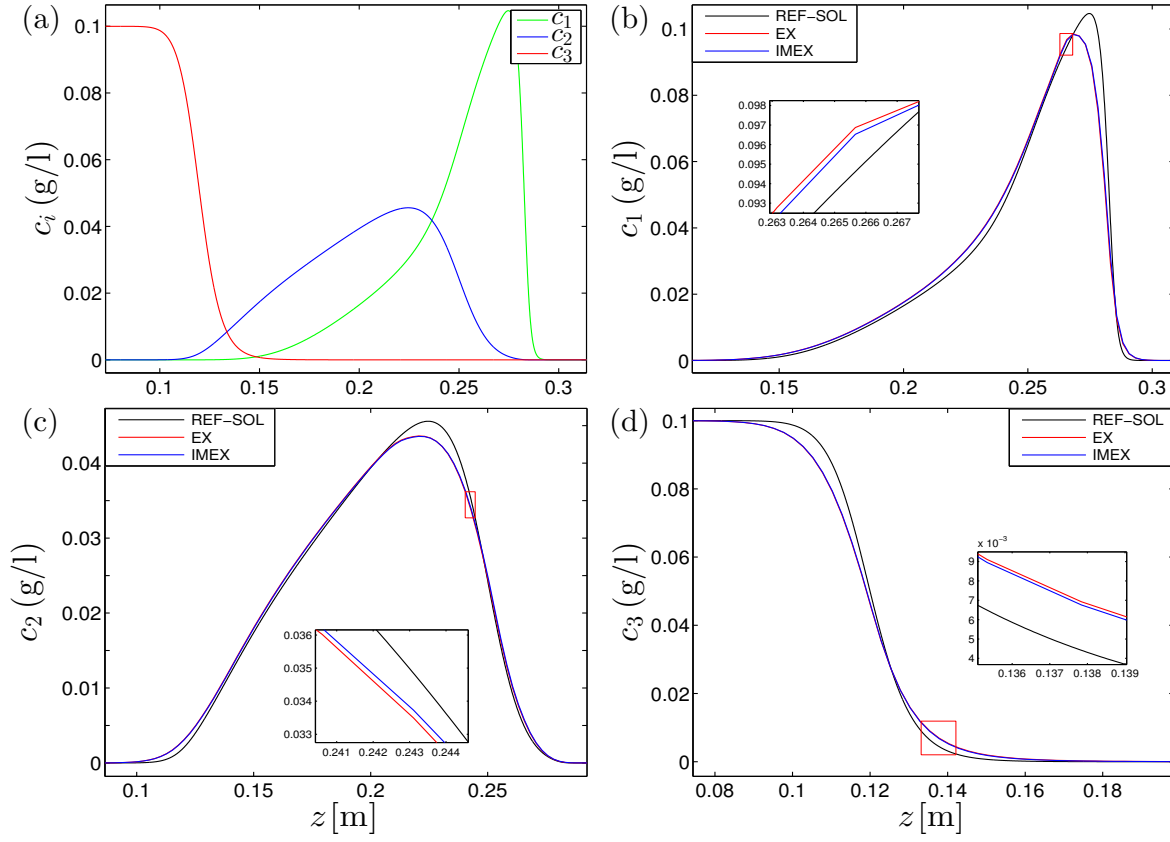


Figure 5.9: Example 5.3: numerical solutions for $D_a = 10^{-4}$ and final time $T = 4$. The spatial resolution corresponds to $M = 400$, results are labeled by “IMEX” and “EX”, and the reference solution (“REF-SOL”) is defined by $M_{\text{ref}} = 12800$.

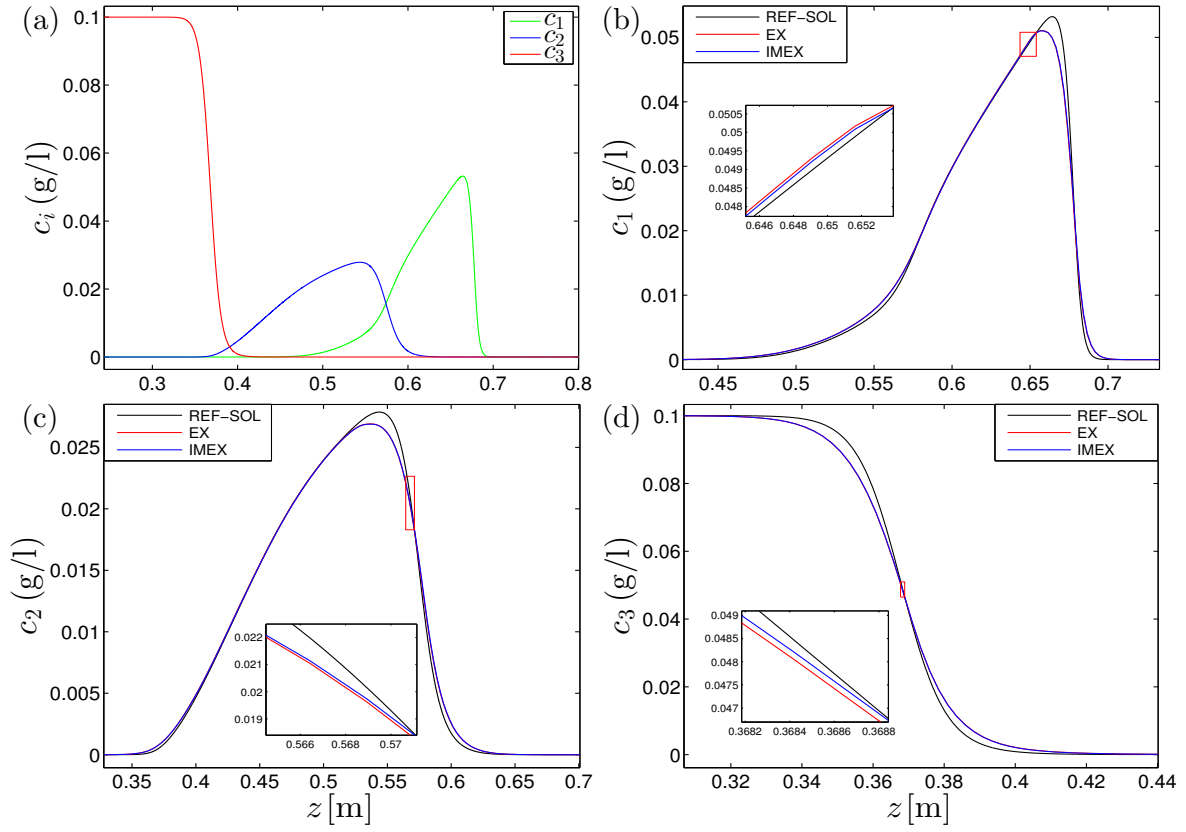


Figure 5.10: Example 5.3: numerical solutions for $D_a = 10^{-4}$ and final time $T = 12$. The spatial resolution corresponds to $M = 400$, results are labeled by “IMEX” and “EX”, and the reference solution (“REF-SOL”) is defined by $M_{\text{ref}} = 12800$.

5.4. Numerical Examples

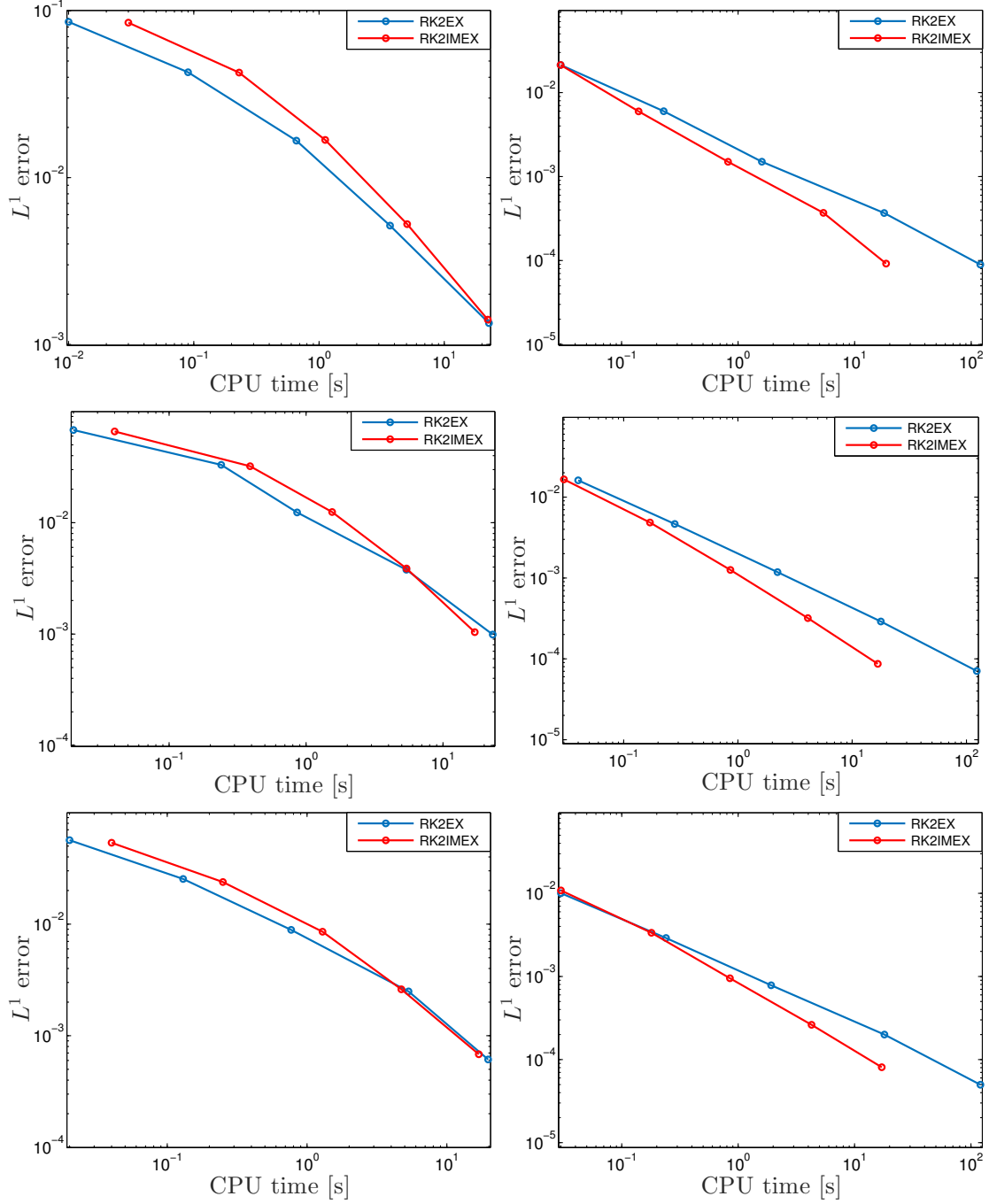


Figure 5.11: Examples 5.1, 5.2, 5.3: Efficiency plots (approximate L^1 error versus CPU time). The final is $T = 4$. Left: $D_a = 10^{-4}$, right: $D_a = 10^{-3}$.

5.4. Numerical Examples

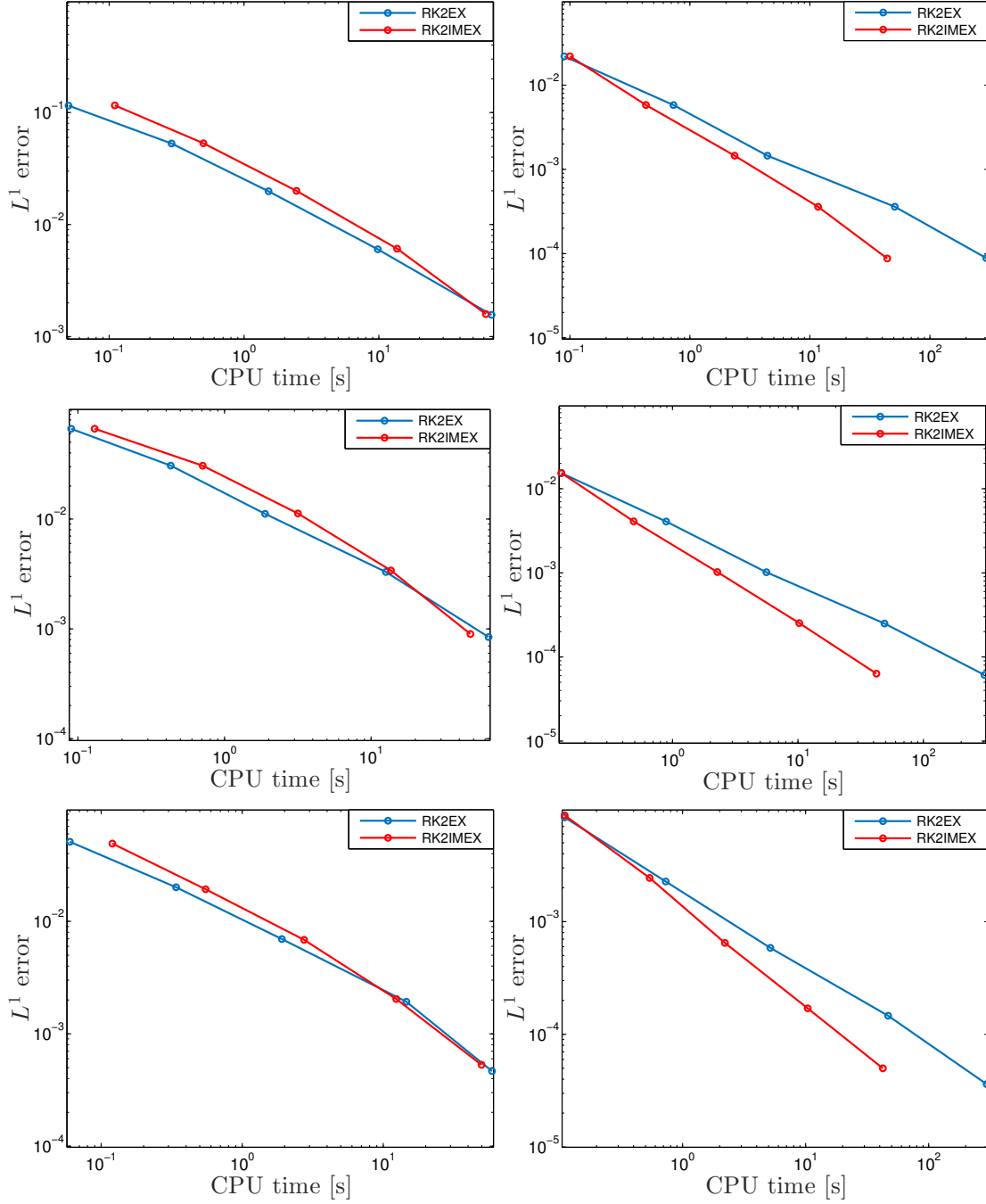


Figure 5.12: Examples 5.1, 5.2, 5.3: Efficiency plots (approximate L^1 error versus CPU time). The final is $T = 12$. Left: $D_a = 10^{-4}$, right: $D_a = 10^{-3}$.

5.5 Conclusions of Chapter 5

Linearly implicit IMEX-RK methods were recently advanced as a tool for the efficient numerical solution of diffusively corrected multi-species kinematic flow models with applications to traffic flow and polydisperse sedimentation [8,9,21]. It has been demonstrated that these methods can also be applied to the ED model of chromatography, and they share the advantage of conservativity with the class of methods introduced in [33]. Moreover it turns out that they are competitive in accuracy with explicit methods of the same formal order of accuracy in space and time, and that they are more efficient than comparable explicit methods for relatively large diffusion coefficients and fine discretizations. Future work will explore the options of further increasing efficiency by alternative choices of underlying Runge-Kutta schemes, and by applying these methods to scenarios with degenerate or discontinuous diffusion coefficients, which were the prime motivation of our above-cited previous treatments.

Chapter 6

Conclusions and outlook

In this thesis we propose a number of techniques for obtaining efficient numerical methods for the approximate solution of some multispecies one-dimensional flow models. For polydisperse sedimentation and vehicular traffic models we propose the use of PVM (Polynomial Viscosity Matrix) methods and MUSCL reconstructions for a second-order spatial discretization, followed by second-order Runge-Kutta integration to obtain second-order fully discrete numerical methods for these models.

The PVM technique is then applied to discretize the convective terms of a system of convection-diffusion equations modelling the settling of droplets dispersed in a viscous fluid. To overcome the severe stability restrictions on the time step associated with explicit time integrators in the presence of diffusion terms, we propose to use implicit-explicit (IMEX) Runge-Kutta (RK) methods. These methods are characterized by the need to solve large systems of algebraic equations involving the diffusion operator, since this operator is treated implicitly. These equations are nonlinear when the diffusion operator is so. This is the case for the models for the settling of droplets and the Equilibrium Dispersive (ED) model for liquid chromatography.

We have satisfactorily tackled several challenges when applying these techniques in the previous contexts. PVM methods are designed to provide a suitable approximation of the viscosity matrix of Roe's method. They require only knowledge of the matrix itself, but not its spectral structure. The main challenge of the application of PVM methods to the kinematic flow models is that the Roe matrix is not available in closed form and has to be approximated by quadrature rules applied to the integral of the Jacobian matrix of the flux on a segment joining the two states the matrix depends on. In this work, Gaussian quadratures with one point (midpoint rule), two and three

points have been tested to arrive to the conclusion that the midpoint rule achieves the most efficient method.

Another challenge is that PVM methods require bounds on the characteristic velocities, i.e., eigenvalues of the Jacobian matrices of the flux. For the kinematic flow models that have been considered in this thesis this information is not readily available in closed form, but the interlacing property of the eigenvalues and velocities of each class can provide with the necessary bounds.

Another contribution is the design of PVM-4 methods with some optimal approximation properties that can avoid some instabilities that appeared with the usual PVM-4 methods when the second eigenvalue of the Jacobian matrix of the flux was very small compared to the largest corresponding eigenvalue.

The last challenge faced in this thesis was the solution of the large systems of nonlinear equations that appear when applying standard IMEX-RK methods. In the context of the settling of droplets or colloidal particles and of the ED model of liquid chromatography, it is shown that linearly implicit implicit-explicit IMEX-RK methods [8, 9] constitute an efficient and simple alternative to the more classical nonlinearly implicit IMEX-RK methods [21].

As future work we propose to explore the options of increasing efficiency by exploring several options of the underlying Runge-Kutta schemes and applying these methods to scenarios with degenerate diffusion or with discontinuous diffusion coefficients, which were the main motivation of our previously mentioned treatments. Such strongly degenerate convection-diffusion systems arise in polydisperse sedimentation [5, 8, 9, 12], traffic flow [22], and can be motivated in other applications. For instance, with respect to the model of settling of dispersions of droplets and colloidal particles, Davis and Russel [28] speculate that the diffusivity in the dilute regime of sedimentation is much smaller (i.e., the corresponding Péclet number is much higher) than in the concentrated one, which could give rise to a function $\beta(\phi)$ with a jump at some critical concentration that marks the transition between both regimes. On the other hand, advanced applications such as continuously operated clarifier-thickener units with the recently formulated extension to handle biological reactions [11] pose the additional difficulty that diffusion terms also depend discontinuously on spatial position, and that at coarse discretizations, the limitation of the time step comes from the discretization of the reaction terms rather than from that of the diffusion terms [13]. That said, we mention that except for the scalar case, the well-posedness analysis of strongly degenerate parabolic systems is an open issue. This is unsurprising since there is no well-posedness theory for systems of

nonlinear conservation laws in general (partial results are either limited to $N \leq 2$ or are based on particular structural features of the system under study) [27], and such systems form a special case of strongly degenerate parabolic systems.

We also propose to apply this type of methods to multidimensional problems that are able to predict some additional effects, such as the direct influence of bulk flows and boundary conditions [29, 36, 43, 75]. This necessarily implies the solution of the Navier-Stokes equations for the flow field of the mixture, increasing the complexity of the mathematical description of a strongly coupled system of conservation laws and also making difficult to solve these equations numerically. While this complication arises naturally with the attempt to extend multispecies kinematic flow models to several space dimensions, it is of equal interest to extend this type of methods to applications that maintain their convection-diffusion character in several dimensions, such as reaction-diffusion systems describing phenomena of pattern formation in mathematical biology [59, 69].

Capítulo 7

Conclusiones y trabajos futuros

En esta tesis proponemos una serie de técnicas para obtener métodos numéricos eficientes para la solución aproximada de algunos modelos unidimensional de flujo multi-especie. Para modelos de sedimentación polidispersa y tráfico vehicular proponemos el uso de métodos PVM (Polynomial Viscosity Matrix) y reconstrucciones MUSCL para reconstrucciones espaciales de segundo orden, seguidas de integración por Runge-Kutta de segundo orden, para obtener métodos numéricos completamente discretos para estos modelos.

La técnica PVM se aplica entonces para discretizar los términos convectivos de un sistema de ecuaciones de convección-difusión que modela el asentamiento de gotitas dispersas en un fluido viscoso. Para superar las severas restricciones para la estabilidad asociadas a los integradores temporales explícitos en presencia de términos difusivos, proponemos el uso de métodos Runge-Kutta (RK) implícitos-explicitos (IMEX). Estos métodos se caracterizan por la necesidad de resolver grandes sistemas de ecuaciones algebraicas que involucran al operador de difusión, dado que este operador es tratado implícitamente. Estas ecuaciones son no-lineales cuando el operador de difusión es de esta manera. Este es el caso de los modelos para el asentamiento de gotitas y el modelo de Equilibrio Dispersivo (ED) para la cromatografía líquida.

Hemos abordado satisfactoriamente varios retos al aplicar estas técnicas en los contextos previos. Los métodos PVM se diseñan para proveer de una aproximación adecuada de la matriz de viscosidad del método de Roe. Solo requieren conocimiento de la matriz, pero no de su estructura espectral. El reto principal de la aplicación de métodos PVM a modelos de flujo cinemático es que la matriz de Roe no está disponible de forma cerrada y ha de ser aproximada por reglas de cuadratura aplicadas a la integral de la

matriz jacobiana del flujo sobre el segmento que une los dos estados de los que depende. En este trabajo se comprueban reglas de cuadratura Gaussianas de 1 punto (regla del punto medio), dos y tres puntos para llegar a la conclusión de que la regla del punto medio lleva al método más eficiente.

Otro reto es que los métodos PVM requieren cotas de las velocidades características, es decir, los valores propios de la matrices jacobianas de los flujos. Para los modelos de flujo cinemático que se consideran en esta tesis esta información no está disponible inmediatamente en forma cerrada, pero la propiedad de interrelación de los autovalores y las velocidades de cada clase puede proporcionar las cotas necesarias.

Otra contribución es el diseño de métodos PVM-4 con algunas propiedades de aproximación óptima que pueden evitar algunas inestabilidades que aparecieron con los métodos PVM-4 usuales cuando el segundo autovalor en tamaño es muy pequeño en relación al mayor.

El último reto encarado en esta tesis es la solución de grandes sistemas de ecuaciones no lineales que aparecen cuando se aplican métodos IMEX-RK estándar. En el contexto del asentamiento de gotitas o partículas coloidales y el del modelo ED para la cromatografía líquida, se muestra que los métodos IMEX-RK linealmente implícitos-explicitos [8, 9] constituyen una alternativa eficiente y simple a los más clásicos no linealmente implícitos métodos IMEX-RK [21].

Como trabajo futuro proponemos explorar las opciones de aumentar la eficiencia explorando distintas opciones de los métodos Runge-Kutta suyacentes y aplicando estos métodos en escenarios con difusión degenerada o con coeficientes de difusión discontinuos, que fueron la principal motivación de los tratamientos mencionados anteriormente. Tales sistemas de convección-difusión fuertemente degenerada en sedimentación polidispersa [5, 8, 9, 12], flujo de tráfico [22], y se pueden motivar en otras aplicaciones. Por ejemplo, con respecto al modelo de asentamiento de dispersiones de gotitas y partículas coloidales, Davis y Russel [28] especulan que la difusividad en el régimen diluido de sedimentación es mucho menor (es decir, el número de Péclet correspondiente es mucho mayor que en el concentrado, lo que podría dar sentido a una función $\beta(\phi)$ con un salto en alguna concentración crítica que marca la transición entre los dos regímenes. Por otra parte, aplicaciones avanzadas tales como unidades clarificadoras-espesadoras con la recientemente formulada extensión para tratar reacciones biológicas [11] impone las dificultades adicionales que los términos difusivos también dependes discontinuamente de la localización espacial y que para discretizaciones groseras la limitación del paso de tiempo proviene de la discretización de los términos reactivas en vez de los términos de

difusión [13]. Dicho esto, mencionamos que, excepto en el caso escalar, un análisis de buen planteamiento de sistemas parabólicos fuertemente degenerados es aún una cuestión abierta. Esto no es sorprendente, ya que no hay teoría para el buen planteamiento de sistemas de leyes de conservación no lineales en general (los resultados parciales se limitan a $N \leq 2$ o están basados en particularidades estructurales del sistema bajo estudio) [27], y tales sistemas forman un caso especial de sistemas parabólicos fuertemente degenerados.

También proponemos aplicar este tipo de métodos a problemas multidimensionales que sean capaces de predecir algunos efectos adicionales, tales como la influencia directa de flujos gruesos y condiciones de frontera [29, 36, 43, 75]. Esto implica necesariamente la solución de las ecuaciones de Navier-Stokes para el campo de flujo de la mezcla, lo cual aumenta la complejidad de la descripción matemática de un sistema fuertemente acoplado de leyes de conservación y también dificultando la solución numérica de dichas ecuaciones.

Mientras que esta complicación aparece naturalmente en el intento de extender los modelos de flujo cinemático multiespecie a varias dimensiones espaciales, es de igual interés extender este tipo de métodos a aplicaciones que mantienen su carácter de convección-difusión en varias dimensiones, tales como los sistemas de ecuaciones de reacción-difusión que describen fenómenos de formación de patrones en biología matemática [59, 69].

Bibliography

- [1] Abeynaïke A, Sederman AJ, Khan Y, Johns ML, Davidson JF, Mackley MR. The experimental measurement and modelling of sedimentation and creaming for glycerol/biodiesel droplet dispersions. *Chemical Engineering Science* 2012; **79**:125–137.
- [2] Anderson J. A secular equation for the eigenvalues of a diagonal matrix perturbation. *Linear Algebra and Applications* 1996; **246**:49–70.
- [3] Ascher U, Ruuth S, Spiteri J. Implicit-explicit Runge-Kutta methods for time dependent partial differential equations. *Applied Numerical Mathematics* 1997; **25**:151–167.
- [4] Benzoni-Gavage S, Colombo RM. An n -populations model for traffic flow. *European Journal of Applied Mathematics* 2003; **14**:587–612.
- [5] Berres S, Bürger R, Karlsen KH, Tory EM. Strongly degenerate parabolic-hyperbolic systems modeling polydisperse sedimentation with compression. *SIAM Journal on Applied Mathematics* 2003; **64**:41–80.
- [6] Boscarino S, LeFloch PG, Russo G. High order asymptotic-preserving methods for fully nonlinear relaxation problems. *SIAM Journal on Scientific Computing* 2014; **36**:A377–A395.
- [7] Boscarino S, Filbet F, Russo G. High order semi-implicit schemes for time dependent partial differential equations. *Journal of Scientific Computing* 2016; **68**:975–1001.
- [8] Boscarino S, Bürger R, Mulet P, Russo G, Villada LM. Linearly implicit IMEX Runge-Kutta methods for a class of degenerate convection-diffusion problems. *SIAM Journal on Scientific Computing* 2015; **37**:B305–B331.

- [9] Boscarino S, Bürger R, Mulet P, Russo G, Villada LM. On linearly implicit IMEX Runge-Kutta Methods for degenerate convection-diffusion problems modelling polydisperse sedimentation, *Bulletin of the Brazilian Mathematical Society, New Series* 2016; **47**:171–185.
- [10] Bourdarias C, Gisclon M, Junca S. Existence of weak entropy solutions for gas chromatography system with one or two active species and non convex isotherms, *Communications in Mathematical Sciences* 2007; **5**:67–84.
- [11] Bürger R, Careaga J, Diehl S, Mejías C, Nopens I, Torfs E, Vanrolleghem PA. Simulations of reactive settling of activated sludge with a reduced biokinetic model. *Computers & Chemical Engineering* 2016; **92**:216–229.
- [12] Bürger R, Diehl S, Martí MC, Mulet P, Nopens I, Torfs E, Vanrolleghem PA. Numerical solution of a multi-class model for batch settling in water resource recovery facilities, *Applied Mathematical Modelling*, to appear.
- [13] Bürger R, Diehl S, Mejías C. A difference scheme for a degenerating convection-diffusion-reaction system modelling continuous sedimentation. Pre-print CI²MA; submitted.
- [14] Bürger R, Donat R, Mulet P, Vega CA. Hyperbolicity analysis of polydisperse sedimentation models via a secular equation for the flux Jacobian. *SIAM Journal on Applied Mathematics* 2010; **70**:2186–2213.
- [15] Bürger R, Donat R, Mulet P, Vega CA. On the implementation of WENO schemes for a class of polydisperse sedimentation models. *Journal of Computational Physics* 2011; **230**:2322–2344.
- [16] Bürger R, Karlsen KH, Tory EM, Wendland WL. Model equations and instability regions for the sedimentation of polydisperse suspensions of spheres. *ZAMM Zeitschrift für Angewandte Mathematik und Mechanik* 2002; **82**:699–722.
- [17] Bürger R, Kozakevicius A. Adaptive multiresolution WENO schemes for multi-species kinematic flow models. *Journal of Computational Physics* 2007; **224**:1190–1222.
- [18] Bürger R, Mulet P, Rubio L. Polynomial viscosity methods for multispecies kinematic flow models. *Numerical Methods for Partial Differential Equations* 2016; **32**:1265–1288.

- [19] Bürger R, Mulet P, Rubio L. Implicit-Explicit methods for the efficient simulation of the settling of dispersions of droplets and colloidal particles. Submitted to *Advances in Applied Mathematics and Mechanics* 2017.
- [20] Bürger R, Mulet P, Rubio L, Sepúlveda M. Linearly implicit IMEX schemes for the equilibrium dispersive model of chromatography. Pre-print CI²MA submitted to *Applied Mathematics and Computation* 2017.
- [21] Bürger R, Mulet P, Villada LM. Regularized nonlinear solvers for IMEX methods applied to diffusively corrected multi-species kinematic flow models. *SIAM Journal on Scientific Computing* 2013; **35**:B751–B777.
- [22] Bürger R, Mulet P, Villada LM. A diffusively corrected multiclass Lighthill-Whitham-Richards traffic model with anticipation lengths and reaction times. *Advances in Applied Mathematics and Mechanics* 2013; **5**:728–758.
- [23] Castro Díaz MJ, Fernández-Nieto E. A class of computationally fast first order finite volume solvers: PVM methods. *SIAM Journal on Scientific Computing* 2012; **34**:A2173–A2196.
- [24] Castro CE, Toro EF. Roe-type Riemann solvers for general hyperbolic systems. *International Journal for Numerical Methods in Fluids* 2014; **75**:467–486.
- [25] Cazes J. Encyclopedia of Chromatography. *Den New Dekker Encyclopedias*. Taylor & Francis, 2001.
- [26] Crouzeix M. Une méthode multipas implicite-explicite pour l’approximation des équations d’évolution paraboliques. *Numerische Mathematik* 1980; **35**:257–276.
- [27] Dafermos CM. Hyperbolic Conservation Laws in Continuum Physics. Fourth Edition. Springer, New York, 2016.
- [28] Davis KE, Russel WB. An asymptotic description of transient settling and ultra-filtration of colloidal dispersions. *Physics of Fluids A* 1989; **1**:82–100.
- [29] Davis RH, Gecol H. Classification of concentrated suspensions using inclined settlers. *International Journal of Multiphase Flow* 1996; **22**:563–574.
- [30] Degond P, Peyrard PF, Russo G, Villedieu P. Polynomial upwind schemes for hyperbolic systems. *Comptes Rendus de l’Académie de Sciences de Paris Série I* 1999; **328**:479–483.

- [31] Donat R, Mulet P. A secular equation for the Jacobian matrix of certain multi-species kinematic flow models. *Numerical Methods for Partial Differential Equations* 2010; **26**:159–175.
- [32] Donat R, Mulet P. Characteristic-based schemes for multi-class Lighthill-Whitham-Richards traffic models. *Journal of Scientific Computing* 2008; **37**:233–250.
- [33] Donat R, Guerrero F, Mulet P. Implicit-Explicit WENO scheme for the equilibrium dispersive model of chromatography. *Preprint submitted to Elsevier*, 2017.
- [34] Donat R, Guerrero F, Mulet P. IMEX WENO schemes for two-phase flow vertical equilibrium processes in a homogeneous porous medium. *Applied Mathematics and Information Sciences* 2013; **7**:1865–1878.
- [35] Donat R, Mulet P. A secular equation for the Jacobian matrix of certain multi-species kinematic flow models, *Numerical Methods for Partial Differential Equations* 2010; **26**:159–175.
- [36] Dorrell RM, Hogg AJ, Pritchard D. Polydisperse suspensions: Erosion, deposition, and flow capacity. *Journal of Geophysical Research: Earth Surface* 2013; **118**:1939–1955.
- [37] Drake JS, Schofer JL, May AD. A statistical analysis of speed-density hypotheses. *Highway Research Record* 1967; **154**:53–87.
- [38] Eidelman SD, Zhitarashu NV. Parabolic boundary value problems, *Operator Theory: Advances and Applications*. **101** Birkhäuser Verlag, Basel, 1998.
- [39] Gottlieb S, Shu C-W, Tadmor E. Strong stability-preserving high-order time discretization methods, *SIAM Review* 2001; **43**:89–112.
- [40] Golub GH, van Loan CF. Matrix Computations, *The Johns Hopkins University Press*; 1996.
- [41] Greenshields BD. A study of traffic capacity. *Highway Research Board Proceedings* 1935; **14**:448–477.
- [42] Guiochon G, Shirazi G, Katti M. Fundamentals of preparative and nonlinear chromatography (2nd ed.). *Elsevier*; 2006.

- [43] Harris TC, Hogg AJ, Huppert HE. Polydisperse particle-driven gravity currents. *Journal of Fluid Mechanics* 2002; **472**:333–371.
- [44] Harten A, Lax PD, van Leer B. On upstream differencing and Godunov type schemes for hyperbolic conservation laws. *SIAM Review* 1983; **25**:35–61.
- [45] Henrick A.K, Aslam T.D, Powers J.M, Mapped weighted essentially non-oscillatory schemes: Achieving optimal order near critical points, *Journal of Computational Physics* **207** (2005) 542–567.
- [46] James F, Sepúlveda M, Charton F, Quiñones I, Guiochon G. Determination of binary competitive equilibrium isotherms from the individual chromatographic band profiles, *Chemical Engineering Science* 1999; **54**:1677–1696.
- [47] James F. Convergence results for some conservation laws with a reflux boundary condition and a relaxation term arising in chemical engineering, *SIAM Journal on Mathematical Analysis* 1998; **29**:1200–1223.
- [48] James F, Postel M, Sepúlveda M. Numerical comparison between relaxation and nonlinear equilibrium models. Application to chemical engineering, *Physica D: Nonlinear Phenomena*; **138** (2000) 316–333.
- [49] James F, Sepúlveda M, Valentin P. Statistical thermodynamics models for multicomponent isothermal diphasic equilibria, *Mathematical Methods in the Applied Sciences* 1997; **7**:1–29.
- [50] Javeed S, Qamar A, Seidel-Morgenstern A, Warnecke G. Efficient and accurate numerical simulation of nonlinear chromatographic processes. *Computers and Chemical Engineering* 2011; **35**:2294–2305.
- [51] Kynch GJ. A theory of sedimentation. *Transactions of the Faraday Society* 1952; **48**:166–176.
- [52] Kurganov A, Tadmor E. New high-resolution central schemes for nonlinear conservation laws and convection-diffusion equations. *Journal of Computational Physics* 2000; **160**:241–282.
- [53] Langmuir I. The adsorption of gases on plane surfaces of glass, mica and platinum, *Journal of the American Chemical Society* 1918; **40**:1361–1403.

- [54] Le Van M.D, Carta G, Yan CM. Adsorption and Ion Exchange, Chapter 16 in Perry RH, Green DW, Maloney JO (Eds.), *Perry's Chemical Engineer's Handbook*, Seventh Edition, McGraw Hill, New York 1998, 16-1–16-66.
- [55] Lighthill MJ, Whitham GB. On kinematic waves: II. A theory of traffic flow on long crowded roads. *Proceedings of the Royal Society A* 1955; **229**:317–345.
- [56] Lockett MJ, Bassoon KS. Sedimentation of binary particle mixtures. *Powder Technology* 1979; **24**:1–7.
- [57] Masliyah JH. Hindered settling in a multiple-species particle system, *Chemical Engineering Science* 1979; **34**:1166–1168.
- [58] Mazzotti M, Rajendran A. Equilibrium theory-based analysis of nonlinear waves in separation processes. *The Annual Review of Chemical and Biomolecular Engineering* 2013; **4**:119–141.
- [59] Murray JD. Mathematical Biology. II: Spatial Models and Biomedical Applications, Third Ed., Springer, New York, 2003.
- [60] Narsimhan G. Analysis of creaming and formation of foam layer in aerated liquid. *Journal of Colloid and Interface Science* 2010; **345**:566–572.
- [61] Pareschi L, Russo G. Implicit-Explicit Runge-Kutta schemes and applications to hyperbolic systems with relaxation. *Journal of Scientific Computing* 2005; **25**:129–155.
- [62] Rhee HK, Aris R, Admundson NR. On the theory of multicomponent chromatography, *Philosophical Transactions of the Royal Society of London* 1970; **A 267**:419–455.
- [63] Richards PI. Shock waves on the highway. *Operations Research* 1956; **4**:42–51.
- [64] Richardson JF, Zaki WN. Sedimentation and fluidization: Part I. *Transactions of the Institution of Chemical Engineers (London)* 1954; **32**:35–53.
- [65] Roe PL. Approximate Riemann solvers, parameter vectors and difference schemes. *Journal of Computational Physics* 1981; **43**:357–371.
- [66] Rosso F, Sona G. Gravity-driven separation of oil-water dispersions. *Advances in Mathematical Sciences and Applications* 2001; **11**:127–151.

- [67] Rouchon P, Schonauer M, Valentin P, Guiochon G. Numerical simulation of band propagation in nonlinear chromatography, *Separation Science and Technology* 1987; **22**:1793–1833.
- [68] Rusanov VV. Calculation of Interaction of Non-Steady Shock Waves with Obstacles, *Computational Mathematics and Mathematical Physics. USSR* 1961; **1**:267–279.
- [69] Ruuth S. Implicit-explicit methods for reaction-diffusion problems in pattern formation. *Journal of Mathematical Biology* 1995; **34**:148–176.
- [70] Schneider W, Anestis G, Schaffinger U. Sediment composition due to settling of particles of different sizes, *International Journal of Multiphase Flow* 1981; **11**:419–423.
- [71] Shannon PT, Stroupe E, Tory EM. Batch and continuous thickening, *Industrial & Engineering Chemistry Fundamentals* 1963; **2**:203–211.
- [72] Staerk D.U, Shitangkoon A, Winchester E, Vigh G, Felinger A, Guiochon G. Use of the equilibrium-dispersive model of nonlinear gas chromatography for the modelling of the elution band profiles in the preparative-scale gas chromatographic separation of enantiomers, *Journal of Chromatography A* 1996; **734**:289–296.
- [73] Toro EF. Riemann Solvers and Numerical Methods for Fluid Dynamics: A Practical Introduction (2nd ed.). *Springer Science & Business Media*; 2013.
- [74] van Leer B. Towards the ultimate conservative finite difference scheme, V. A second order sequel to Godunov’s method. *Journal of Computational Physics* 1979; **32**:101–136.
- [75] Weiland RH, Fessas YP, Ramarao BV. On instabilities arising during sedimentation of two-component mixtures of solids. *Journal of Fluid Mechanics* 1984; **142**:383–389.
- [76] Wong GCK, Wong C. A multi-class traffic flow model: an extension of LWR model with heterogeneous drivers. *Transportation Research Part A* 2002; **36**:827–841.
- [77] Zhang P, Liu RX, Wong SC, Dai SQ. Hyperbolicity and kinematic waves of a class of multipopulation partial differential equations. *European Journal of Applied Mathematics* 2006; **17**:171–200.

- [78] Zhang M, Shu CW, Wong GCK, Wong SC. A weighted essentially non-oscillatory numerical scheme for a multi-class Lighthill-Whitham-Richards traffic flow model. *Journal of Computational Physics* 2003; **191**:639–659.