

FULLY ADAPTIVE MULTIREOLUTION SCHEMES FOR STRONGLY DEGENERATE PARABOLIC EQUATIONS IN ONE SPACE DIMENSION

RAIMUND BÜRGER¹, RICARDO RUIZ¹, KAI SCHNEIDER² AND MAURICIO SEPÚLVEDA¹

Abstract. We present a fully adaptive multiresolution scheme for spatially one-dimensional quasi-linear strongly degenerate parabolic equations with zero-flux and periodic boundary conditions. The numerical scheme is based on a finite volume discretization using the Engquist-Osher numerical flux and explicit time stepping. An adaptive multiresolution scheme based on cell averages is then used to speed up the CPU time and the memory requirements of the underlying finite volume scheme, whose first-order version is known to converge to an entropy solution of the problem. A particular feature of the method is the storage of the multiresolution representation of the solution in a graded tree, whose leaves are the non-uniform finite volumes on which the numerical divergence is eventually evaluated. Moreover using the L^1 contraction of the discrete time evolution operator we derive the optimal choice of the threshold in the adaptive multiresolution method. Numerical examples illustrate the computational efficiency together with the convergence properties.

Mathematics Subject Classification. 35L65, 35R05, 65M06, 76T20.

Received January 31, 2007. Revised September 17, 2007.
Published online May 27, 2008.

1. INTRODUCTION

Adaptive multiresolution schemes were introduced in the 1990s for hyperbolic conservation laws with the aim to accelerate discretization schemes while controlling the error [31]. This approach has been exploited in different directions. Fully adaptive multiresolution schemes for hyperbolic equations are developed in [17]. In addition to CPU time reduction thanks to the reduced number of costly flux evaluations, these schemes also allow a significant reduction of memory requirements by using dynamic data structures. An overview on multiresolution techniques for conservation laws is given by Müller [40], see also Chiavassa *et al.* [16]. Fully adaptive multiresolution schemes for parabolic equations are presented in [44,45]. Adaptive computations of combustion problems in three space dimensions illustrated the efficiency and accuracy of the method [44].

Some approaches to define adaptive space discretizations emerge from *ad hoc* criteria, while others are based on *a posteriori* error estimators using control strategies by solving computationally expensive adjoint

Keywords and phrases. Degenerate parabolic equation, adaptive multiresolution scheme, monotone scheme, upwind difference scheme, boundary conditions, entropy solution.

¹ Departamento de Ingeniería Matemática, Universidad de Concepción, Casilla 160-C, Concepción, Chile.
rburger@ing-mat.udec.cl; r Ruiz@ing-mat.udec.cl; mauricio@ing-mat.udec.cl

² Centre de Mathématiques et d'Informatique, Université de Provence, 39 rue Joliot-Curie, 13453 Marseille Cedex 13, France.
kschneid@cmi.univ-mrs.fr

problems [1,49]. Adaptive mesh refinement methods introduced by Berger and Oliger [4] are now widely used for many applications using structured or unstructured grids, see *e.g.* [2,3].

It is the purpose of this paper to present an extension of the fully adaptive multiresolution scheme for parabolic PDEs [45] to spatially one-dimensional degenerate parabolic PDEs of the type

$$u_t + b(u)_x = A(u)_{xx}, \quad (x, t) \in Q_T := I \times \mathcal{T}, \quad I \subset \mathbb{R}, \quad \mathcal{T} := (0, T), \quad (1.1)$$

where the function $b(u)$ is assumed to be Lipschitz continuous and piecewise smooth with $b(u) > 0$ for $u \in (0, u_{\max})$ and $b(u) = 0$ otherwise, where u_{\max} is a given maximum solution value, and

$$A(u) = \int_0^u a(s) ds, \quad (1.2)$$

where $a(u)$ is a bounded, non negative integrable function. (In the sequel, whenever we refer to (1.1), it is understood that $b(u)$ and $A(u)$ satisfy these assumptions.) Wherever $a(u)$ vanishes, (1.1) degenerates into a first-order hyperbolic conservation law. Since degeneracy is allowed to occur not only at isolated solution values, but on u -intervals of positive length, (1.1) is called *strongly* degenerate parabolic.

Solutions of strongly degenerate parabolic equations, which include scalar conservation laws as a special case, are in general discontinuous, and need to be defined as weak solutions along with an entropy condition to select the physically relevant weak solution. This property excludes the application of standard numerical schemes for uniformly parabolic equations having smooth solutions; rather, appropriate schemes are based on finite volume (FV) schemes for hyperbolic conservation laws. In fact, our discretization is based on a simple FV scheme with explicit time integration and uses the Engquist-Osher [23] numerical flux [11,12,25].

The analysis of numerical schemes for (1.1) exhibits several difficulties; most notably, due to the involved nonlinear structure of (1.1), it is usually difficult to construct an exact solution, the convergence rate is not known, and numerical experimentation is necessary to identify the best suited parameters for a numerical scheme, for example the threshold parameters in a multiresolution scheme [13].

First applications of multiresolution schemes to (1.1) were presented in [13,46]. In [13], the multiresolution method combines the switch between central interpolation or exact computation of numerical flux with a thresholded wavelet transform applied to cell averages of the solution to control the switch. The multiresolution method used in [13] closely follows the work of Harten [31]. Within that version, the differential operator is always evaluated on the finest grid, but computational effort is saved by replacing, wherever the solution is sufficiently smooth, exact flux evaluations by approximate flux values that have been obtained more cheaply by interpolation from coarser grids. Though the version of the multiresolution method of [13] is effective for (1.1), it does not provide memory savings. In contrast to [13], the method presented herein does provide significant memory savings, since the multiresolution representation of the solution is stored in a graded tree [17,22,40,41,45], whose leaves are the finite volumes for which the numerical divergence is computed. This means that the numerical flux is actually evaluated on the borders of these finite volumes. Since the flux is computed only at these positions, but not on all positions of the fine grid (as in [13]), we refer to our method as *fully* adaptive. On the other hand, the properties of the underlying discretization allow to derive an optimal choice of the threshold parameter for the adaptive multiresolution computations, as suggested in [17]. This choice guarantees that the perturbation error of the adaptive multiresolution scheme is of the same order than the discretization error of the finite volume scheme.

To further put this work into the proper perspective, we first mention that the main applications of (1.1) include a theory of sedimentation-consolidation processes [5], a diffusively corrected kinematic traffic model [7,42], and two-phase flow in porous media [24]. The initial-boundary value problem for (1.1) that models batch sedimentation is analyzed in [8], where the existence of BV entropy solutions is shown *via* the vanishing viscosity method, while their uniqueness follows by a technique introduced by Carrillo [15]. On the other hand, Evje and Karlsen [25] show that explicit monotone finite difference schemes, which were introduced by Harten *et al.* [32] and Crandall and Majda [18] for conservation laws, converge to BV entropy solutions for initial-value

problems of strongly degenerate parabolic equations. These results are extended to several space dimensions in [33]. Further analyses of finite volume schemes for degenerate parabolic equations include the works by Eymard *et al.* [26] and Michel and Vovelle [39].

Variants of the techniques of [25,33] suitable for (1.1) with flux-type boundary conditions are analyzed in [6,11,12]; in particular, in [11,12] it is proved that these schemes converge to an entropy solution. Convergence of monotone schemes to an entropy solution has also been proved for conservation laws and strongly degenerate convection-diffusion equations with discontinuous flux [9,10,34,35,50,51]. Such equations arise, for example, if the sedimentation-consolidation model is extended to so-called clarifier-thickener units. They can also be effectively discretized by the multiresolution technique presented herein [14].

The remainder of this paper is organized as follows. In Section 2, we specify the initial and boundary conditions for two initial-boundary value problems for (1.1): Problem A with zero-flux boundary conditions, and Problem B with periodic boundary conditions. For Problem A, the regularity assumptions on the initial datum and the definition of an entropy solution are given. An existence and uniqueness result for Problem A is recalled; a similar analysis can be conducted for Problem B. In Section 3, the numerical discretization for Problem A is described (the modifications for Problem B are minor). To this end, we introduce in Section 3.1 the basic finite volume scheme. The first-order version of this scheme is a monotone upwind scheme involving the Engquist-Osher flux [23], which is analyzed in [11,12]; we here utilize a spatially second-order MUSCL-type version [6]. The basic time discretization is described in Section 3.2. It is based on a particular embedded Runge-Kutta method, the RK3(2) Runge-Kutta-Fehlberg (RKF) method, which allows to adaptively control the time step. In Section 4, the conservative adaptive multiresolution method is outlined; in particular, in Section 4.1, the construction of the graded tree structure is described. The final multiresolution algorithm, which combines the ingredients of Sections 3 and 4, is outlined in Section 4.2. Further details on the numerical method and on its implementation using dynamical data structures can be found in [45]. An error analysis of the adaptive multiresolution scheme is outlined in Section 5. Based on the analyses by Cohen *et al.* [17] and Roussel *et al.* [45] we show that, if the convergence rate of the FV scheme is known, then the threshold parameter of the multiresolution scheme can be chosen optimally in the sense that the multiresolution scheme has the same order of convergence (with respect to the discretization of the finest grids) than the finite volume scheme. Calculations presented in the Appendix show that within a finite choice of values ranging over several orders of magnitude, the reference tolerance has been chosen optimally.

In Section 6 we present numerical results for a sedimentation-consolidation model, which gives rise to Problem A, and for a model of traffic flow on a circular road, which leads to Problem B (Examples 1 and 2, respectively). To this end, we recall in Section 6.1 the sedimentation-consolidation model, and present in Section 6.2 numerical simulations for Example 1. In light of the findings of Section 5, we first perform test calculations to estimate the order of convergence α of the FV scheme for Problem A, resulting in $\alpha \approx 0.6$, and to analyze the behavior of the RKF time step control compared to a simulation with a fixed time step. Numerical experiments with the multiresolution scheme, in part combined with the RKF device, then demonstrate that the error control outlined in Section 5 is indeed effective here. In Section 6.3, we outline a traffic model, which leads to Problem B. The numerical experiments for traffic flow on a circular road (Example 2) presented in Section 6.4 are similar to those of Section 6.2, and equally illustrate the effectiveness of the method. Some conclusion that can be drawn from the paper are given in Section 7.

2. STATEMENT OF THE PROBLEM AND PRELIMINARIES

For the interval $I := (x_a, x_b)$, the zero-flux initial-boundary value problem, Problem A, is defined by (1.1) and the initial and boundary conditions

$$u(x, 0) = u_0(x), \quad x \in I, \tag{2.1}$$

$$(b(u) - A(u)_x)(x_\ell, t) = 0, \quad t \in \mathcal{T}, x_\ell \in \{x_a, x_b\}. \tag{2.2}$$

Problem B arises from replacing the zero-flux boundary condition (2.2) by the periodicity condition

$$u(x_a, t) = u(x_b, t), \quad t \in \mathcal{T}. \tag{2.3}$$

Problems A and B admit an existence and uniqueness analysis if the functions $b(u)$ and $A(u)$ satisfy the assumptions stated in Section 1. In addition, we assume that the initial function u_0 satisfies a regularity condition [11], which we here state in terms of a spatial discretization that will also be used for the FV scheme. To this end, let $J \in \mathbb{N}$ denote the number of space steps, $\Delta x := (x_b - x_a)/J$, $x_{j+1/2} := x_a + (j + 1/2)\Delta x$ for $j = 0, \dots, J - 1$, $I_j := [x_{j-1/2}, x_{j+1/2})$ and

$$u_j^0 := \frac{1}{\Delta x} \int_{I_j} u_0(x) \, dx, \quad j = 0, \dots, J. \tag{2.4}$$

For Problem A, we assume that $u_0 \in BV(I)$, $0 \leq u_0(x) \leq u_{\max}$ for all $x \in \bar{I}$, and that there exists a constant $M > 0$ such that

$$\sum_{m=1}^{J-1} |A(u_{m+1}^0) - 2A(u_m^0) + A(u_{m-1}^0)| \leq M\Delta x \quad \text{uniformly in } \Delta x; \tag{2.5}$$

note that this requests, in particular, that M be independent of J . As detailed in [11], (2.5) is imposed to ensure uniform L^1 Lipschitz continuity in time for the numerical solution when the discretization parameters tend to zero. The continuous version of (2.5) states that the total variation of $(A_\varepsilon(u_0))_x$ on the interval I must be uniformly bounded with respect to ε , where $\varepsilon > 0$ is a regularization parameter of a smooth regularization $A_\varepsilon(\cdot)$ of $A(\cdot)$. In the continuous case, this condition was introduced in [8] to achieve a uniform estimate of the spatial variation of the time derivative of a viscous regularization of Problem A. Both (2.5) and the continuous condition are satisfied, for example, if u_0 is constant.

For the purpose of illustration, we recall here the definition of an entropy solution of Problem A [8,11].

Definition 2.1. A function $u \in L^\infty(Q_T) \cap BV(Q_T)$ is an *entropy solution* of the initial-boundary value problem (1.1), (2.1), (2.2) (Problem A) if the following conditions are satisfied:

- (1) The integrated diffusion coefficient has the regularity $A(u)_x \in L^2(Q_T)$.
- (2) The boundary condition (2.2) holds in the following sense:

$$\gamma(x_\ell, t)(b(u) - A(u)_x) = 0, \quad t \in \mathcal{T}, \quad x_\ell \in \{x_a, x_b\},$$

where γ is the trace operator.

- (3) The initial condition (2.1) holds in the following sense:

$$\lim_{t \downarrow 0} u(x, t) = u_0(x) \quad \text{for almost all } x \in I.$$

- (4) The following entropy inequality holds for all test functions $\varphi \in C_0^\infty(Q_T)$, $\varphi \geq 0$, and all $k \in \mathbb{R}$:

$$\iint_{Q_T} \left\{ |u - k| \varphi_t + \text{sgn}(u - k) [b(u) - b(k) - A(u)_x] \varphi_x \right\} dt \, dx \geq 0.$$

The proof of existence of an entropy solution by the method of vanishing viscosity is outlined in [8]. That paper also presents a sketch of the uniqueness proof, which relies on results by Carrillo [15] that permit applying Kruřkov’s “doubling of the variables” [36] technique to strongly degenerate parabolic equations. These results allow us to state the following theorem; a similar analysis can be conducted for Problem B.

Theorem 2.1. *The initial-boundary value problem (1.1), (2.1), (2.2) (Problem A) has a unique entropy solution.*

3. NUMERICAL DISCRETIZATION

3.1. Finite volume scheme

Problem A is discretized in space with classical finite volumes. The computational domain I is decomposed into cells $\{I_j\}_{j=1}^{2^L}$. The initial-boundary value problem (IBVP) is then integrated over each cell I_j , where the volume of each cell is denoted by $|I_j|$. Hence we get

$$\frac{du_j}{dt} = \bar{\mathcal{D}}_j(U(t)), \quad j = 1, \dots, 2^L, \tag{3.1}$$

where $U(t) = (u_i(t))_{i=1, \dots, 2^L}$ contains the cell averages of the numerical solution at time t , such that

$$u_j(t) \approx \frac{1}{|I_j|} \int_{I_j} u(x, t) \, dx, \quad j = 1, \dots, 2^L,$$

and $\bar{\mathcal{D}}_j(U(t))$ is the numerical divergence of cell I_j at time t . In the one-dimensional case, I_j is an interval (the cell $[x_{j-1/2}, x_{j+1/2}]$) with step size $\Delta x_j := x_{j+1/2} - x_{j-1/2}$, and we may simply write

$$\bar{\mathcal{D}}_j(U(t)) \approx -\frac{1}{|I_j|} \left((b(u) - A(u)_x)|_{x_{j+1/2}} - (b(u) - A(u)_x)|_{x_{j-1/2}} \right), \quad j = 1, \dots, 2^L. \tag{3.2}$$

If $\bar{F}_{j+1/2}$ and $\bar{F}_{j-1/2}$ are the numerical fluxes associated with the left and right cell boundaries, respectively, then we may write

$$\bar{\mathcal{D}}_j = -\frac{1}{\Delta x_j} (\bar{F}_{j+1/2} - \bar{F}_{j-1/2}), \quad j = 1, \dots, 2^L. \tag{3.3}$$

Available discretization methods differ by the definition of the numerical flux that approximates $\bar{F}_{j\pm 1/2}$.

In addition to the notation introduced previously, we let $N \in \mathbb{N}$ be the number of time steps, $\Delta t := T/N$, and $I^n := [t_n, t_{n+1})$, where $t_n = n\Delta t$ for $n = 0, \dots, N$. We denote by u_j^n the numerical solution at (x_j, t_n) , assume that the values for $n = 0$ are given by (2.4), and approximate the entropy solution of Problem A by a three-point finite difference scheme, which is defined by an ‘‘interior’’ formula for u_1^n, \dots, u_{J-1}^n and two ‘‘boundary’’ formulas for u_0^n and u_J^n , respectively. Defining $\nu := \Delta t/\Delta x^2$ and $\lambda := \Delta t/\Delta x$, we employ the following discretization, where $h_{j+1/2}^n := h(u_j^n, u_{j+1}^n)$ and $d_{j+1/2}^n := A(u_{j+1}^n) - A(u_j^n)$ for $j = 0, \dots, J - 1$:

$$u_0^{n+1} = u_0^n - \lambda h_{1/2}^n + \nu d_{1/2}^n, \tag{3.4a}$$

$$u_j^{n+1} = u_j^n - \lambda(h_{j+1/2}^n - h_{j-1/2}^n) + \nu(d_{j+1/2}^n - d_{j-1/2}^n), \quad j = 1, \dots, J - 1, \tag{3.4b}$$

$$u_J^{n+1} = u_J^n + \lambda h_{J-1/2}^n - \nu d_{J-1/2}^n. \tag{3.4c}$$

We use the Engquist-Osher flux function [23]

$$h(u, v) := b(0) + \int_0^u \max\{b'(s), 0\} \, ds + \int_0^v \min\{b'(s), 0\} \, ds. \tag{3.5}$$

In [12] it is shown that the scheme (3.4), (3.5), which is the first-order version of the basic scheme used herein, converges to the entropy solution of Problem A, provided that the following CFL condition is satisfied:

$$\text{CFL} := \lambda \|b'\|_\infty + \nu \|a\|_\infty \leq 1/2. \tag{3.6}$$

Convergence of a semi-implicit variant of (3.4), (3.5) is shown in [11] under the milder CFL condition $\lambda \|b'\|_\infty \leq 1/2$. In both cases, the convergence proof relies on the monotonicity of the first-order scheme.

In order to upgrade the spatial discretization to second order, we use a MUSCL (variable extrapolation) scheme. We introduce a piecewise linear function $u^n(x)$ defined by

$$u^n(x) := u_j^n + s_j^n(x - x_j), \quad x \in (x_{j-1/2}, x_{j+1/2}),$$

where s_j^n denotes a suitable slope constructed from u^n . For smooth solutions, in regions where s_j^n is an $\mathcal{O}(\Delta x)$ approximation of $u_x(x_j, t_n)$, the reconstruction is linear and the truncation error is $\mathcal{O}(\Delta x^2)$. In regions where $s_j^n = 0$, the reconstruction is piecewise constant and the truncation error is $\mathcal{O}(\Delta x)$. For Cauchy problems, the slopes are limited to enforce the Total Variation Diminishing (TVD) property of the scheme. In our case, we use the θ -limiter [28]

$$s_j^n = \hat{M} \left(\theta \frac{u_j^n - u_{j-1}^n}{\Delta x}, \frac{u_{j+1}^n - u_{j-1}^n}{2\Delta x}, \theta \frac{u_{j+1}^n - u_j^n}{\Delta x} \right), \quad \theta \in [0, 2], \tag{3.7}$$

where we choose $\theta = 0.5$ as in [6], and \hat{M} is the minmod function

$$\hat{M}(a, b, c) := \begin{cases} \min\{a, b, c\} & \text{if } a, b, c > 0, \\ \max\{a, b, c\} & \text{if } a, b, c < 0, \\ 0 & \text{otherwise.} \end{cases}$$

Then we extrapolate the data to the boundaries of each cell and form the corresponding second order scheme. (An alternative ‘‘order upgrading’’ method would be to use a second-order essentially non-oscillatory (ENO) reconstruction [47], as done in [45].) For Problem A (with zero-flux boundary conditions), (3.7) is used for $j = 2, \dots, J - 2$, and we set

$$s_0^n = s_1^n = s_{J-1}^n = s_J^n = 0. \tag{3.8}$$

Experience in previous work [6] shows that the boundary formulas properly approximate the zero-flux boundary conditions (2.2) only if the first-order version (3.4a) and (3.4c) is utilized, that is, if condition (3.8) is imposed. Early numerical experiments showed that dropping this condition and calculating boundary slopes according to (3.7) produces oscillatory solutions.

For Problem B, the basic scheme is defined by formula (3.4b) for $j = 0, \dots, J$, provided that the space index is taken modulo J . In the same sense, (3.7) is used for all j .

3.2. Time integration

The FV scheme described in Section 3.1 is only first order in time, and should be upgraded to at least second order so that the second-order spatial accuracy is effective. This can be achieved if the time integration is done, for example, by a standard second or higher order Runge-Kutta (RK) scheme. However, we herein utilize a particular class of so-called embedded RK schemes [19,27,30], which, apart from providing the necessary accuracy, also allow for an adaptive control of the time step. Specifically, we utilize a simple version of a so-called Runge-Kutta-Fehlberg (RKF) method. RKF schemes are based on the observation that by varying the vector b of weights for the stages calculated in the course of a Runge-Kutta step, pairs of schemes with different orders of accuracy can be generated. This allows to estimate the approximation error in time, and the time step can be automatically adjusted to control the error in time. The additional computational effort is moderate, since both schemes utilize the same nodes and interior weights of the quadrature formula.

In our case, we use the RK3(2) method, which is a RKF method defined for a system of ordinary differential equations $du/dt = \mathcal{L}(t, u)$ that combines a Runge-Kutta scheme

$$u^{m+1} = u^m + \hat{b}_1 \kappa_1 + \hat{b}_2 \kappa_2 + \hat{b}_3 \kappa_3 \tag{3.9}$$

of order $p = 3$ with another scheme

$$u^{m+1} = u^m + \check{b}_1 \kappa_1 + \check{b}_2 \kappa_2 + \check{b}_3 \kappa_3 \tag{3.10}$$

of order $p - 1 = 2$, where the stages

$$\begin{aligned} \kappa_1 &= \Delta t \mathcal{L}(t_m + c_1 \Delta t, u^m), \\ \kappa_2 &= \Delta t \mathcal{L}(t_m + c_2 \Delta t, u^m + a_{21} \kappa_1), \\ \kappa_3 &= \Delta t \mathcal{L}(t_m + c_3 \Delta t, \bar{u}^m + a_{31} \kappa_1 + a_{32} \kappa_2) \end{aligned} \tag{3.11}$$

are the same for both methods. The coefficients for the resulting RK3(2) method are given in the following Butcher tableau:

$$\begin{array}{c|ccc} c_1 = 0 & & & \\ c_2 = 1 & a_{21} = 1 & & \\ c_3 = 1/2 & a_{31} = 1/4 & a_{32} = 1/4 & \\ \hline & \hat{b}_1 = 1/6 & \hat{b}_2 = 1/6 & \hat{b}_3 = 2/3 \\ \hline & \check{b}_1 = 1/2 & \check{b}_2 = 1/2 & \check{b}_3 = 0 \end{array} \tag{3.12}$$

These specific coefficients define an optimal pair of embedded TVD-RK methods of orders two and three [47]. If we denote by \hat{u}^{m+1} and \check{u}^{m+1} the candidate values of u^{m+1} determined by (3.9) and (3.10), respectively, then truncation error between these two approximations is estimated by

$$\delta_{\text{old}} := \hat{u}^{m+1} - \check{u}^{m+1} = \sum_{i=1}^p (\hat{b}_i - \check{b}_i) \kappa_i \sim (\Delta t)^p \quad (\text{with } p = 3 \text{ in our case}). \tag{3.13}$$

We could adjust Δt to maintain prescribed accuracy δ_{desired} in time by using $\Delta t_{\text{new}} = \Delta t_{\text{old}} |\delta_{\text{desired}} / \delta_{\text{old}}|^{1/p}$ with $p = 3$, but to avoid excessively large time steps, we apply a time step limiter function \mathcal{S} defined as

$$\mathcal{S}(t, \Delta t) := (\mathcal{S}_0 - \mathcal{S}_{\text{min}}) \exp(-t/\Delta t) + \mathcal{S}_{\text{min}},$$

where we choose $\mathcal{S}_0 = 0.1$ and $\mathcal{S}_{\text{min}} = 0.01$. This implies that we initially allow 10% of variation in the time step, and after a few iterations, we allow only 1%. The new time step Δt_{new} is now defined as

$$\Delta t_{\text{new}} = \begin{cases} \Delta t_{\text{old}} |\delta_{\text{desired}} / \delta_{\text{old}}|^{1/p} & \text{if } |(\Delta t_{\text{new}} - \Delta t_{\text{old}}) / \Delta t_{\text{old}}| \leq \frac{1}{2} \mathcal{S}(t, \Delta t_{\text{old}}), \\ \frac{1}{2} \mathcal{S}(t, \Delta t_{\text{old}}) \Delta t_{\text{old}} + \Delta t_{\text{old}} & \text{otherwise.} \end{cases} \tag{3.14}$$

We emphasize that Δt_{new} is the time step for computing \bar{u}^{m+2} . More details on the numerical scheme and its implementation can be found in [21]. The nomenclature of ‘‘RK3 method’’ for the embedded Runge-Kutta scheme used herein is widespread in the literature (see *e.g.* [48]). However, this scheme does *not* utilize what has become known as the ‘‘Fehlberg trick’’ [19] (*i.e.*, κ_p equals κ_1 of the next time step).

Both the second- and third-order TVD-RK schemes presented herein are optimal in the sense that if the first-order explicit Euler time discretization $u^{m+1} = u^m + \Delta t \mathcal{L}(t_m, u^m)$ is stable in a certain norm, *i.e.*, $\|u^{m+1}\| \leq \|u^m\|$ under a condition $\Delta t \leq \Delta t_1$, then both schemes are stable under the same condition $\Delta t \leq \Delta t_1$, see [47] for details. This means that even though these TVD-RK schemes are of second-order or third-order accuracy, the CFL condition for the resulting FV scheme is still the condition (3.6) imposed on the first-order version of the scheme, which limits $\Delta t / \Delta x^2$.

4. CONSERVATIVE ADAPTIVE MULTIREOLUTION DISCRETIZATION

To accelerate a given finite volume scheme on a uniform grid without losing accuracy, the solution on a fine grid is represented by values on a coarser grid plus a series of differences at different levels of nested dyadic grids. These differences contain information on the solution when going from a coarse to a finer grid; specifically, these coefficients are small in regions where the solution is smooth. Removing small coefficients by thresholding defines a locally refined adaptive grid. A suitable choice of this threshold guarantees that the discretization error of the reference scheme is balanced with the accumulated thresholding error introduced in each time step. This allows to reduce memory and CPU requirements without losing accuracy.

4.1. Graded tree data structure

Mainly for sake of data and time compression, differences and the solution at different levels are organized in a dynamic tree. We denote by *root* the basis of the tree, and by *node* any element of the tree. In one-dimensional space, a parent node has two sons, and the sons of the same parent are called *brothers*. A given node has nearest neighbors in each direction, called *nearest cousins*. A node without children is called a *leaf*. For the computation of the fluxes of a leaf, we need $s' = 2$ nearest cousins in each direction, if these do not exist, we create them as *virtual leaves*. This graded tree structure has been illustrated in previous work, see e.g. [14,45], and is comprehensively described in [40].

We follow the ideas of Harten's cell average multiresolution framework [31]. We denote by Λ the set of indices of existing nodes, by $\mathcal{L}(\Lambda)$ the restriction of Λ to the leaves, and by Λ_l the restriction of Λ to a multiresolution level l , $0 \leq l \leq L$. To obtain the cell averages of a level l from those of the finer level $l+1$, we use the *projection operator* $P_{l+1 \rightarrow l}$, which in our one-dimensional case it is defined by

$$u_{l,j} = (P_{l+1 \rightarrow l} U_{l+1})_j := \frac{1}{2}(u_{l+1,2j} + u_{l+1,2j-1}).$$

To estimate the cell averages of a level $l+1$ from those of level l , we use the *prediction operator* $P_{l \rightarrow l+1}$. This operator should be local in the sense that the interpolation for a son is made from the cell averages of its parent and the s nearest cousins of its parent; and should be consistent with the projection in the sense that it is conservative with respect to the coarse grid cell averages: $P_{l+1 \rightarrow l} \circ P_{l \rightarrow l+1} = \text{Id}$. For a regular grid structure in one space dimension, we use a polynomial interpolation:

$$\hat{u}_{l+1,2j} = u_{l,j} + \sum_{m=1}^s \gamma_m (u_{l,j+m} - u_{l,j-m}), \quad \hat{u}_{l+1,2j-1} = u_{l,j} - \sum_{m=1}^s \gamma_m (u_{l,j+m} - u_{l,j-m}), \quad j = 1, \dots, N(l). \quad (4.1)$$

For all of our cases, the order of accuracy is $r = 3$, i.e., $s = 1$, which corresponds to quadratic polynomial interpolation with $\gamma_1 = -1/8$. Nevertheless, one may choose an arbitrarily higher order of accuracy.

The detail $\bar{d}_{l,j}$ is the difference between the exact and the predicted value, $\bar{d}_{l,j} := u_{l,j} - \hat{u}_{l,j}$. Given that a parent has two sons, only one detail is independent. Then, knowledge of the cell average values of the two sons is equivalent to the knowledge of the cell average value of the father node and the independent detail. Repeating this operation recursively on L levels, we get the *multiresolution transform* on the cell averages $\bar{\mathbf{M}} : \bar{U}_L \mapsto (\bar{D}_L, \dots, \bar{D}_1, \bar{U}_0)$. Due to the data structure, the decoding and encoding procedures are achieved with complexity of $O(N_L \log N_L)$ (see [45]), and this improves if the solution is highly compressed itself.

4.2. Algorithm implementation

Now we give a brief description of the multiresolution procedure used to solve the test problems. A more detailed description can be found in [14].

- (1) Initialize the physical and numerical parameters.
- (2) *Creation of the initial graded tree structure:*
 - Create the root and compute its cell average value.

- Split the cell, compute the cell average values in the sons and compute the corresponding details.
- Recursively apply the thresholding strategy until all sons have details below the tolerance ε_l .

DO $n = 1 : total_time_steps$.

- (3) Determine the set of leaves and virtual leaves.
- (4) *Time evolution with fixed time step*: Compute the discretized divergence operator $\bar{\mathcal{D}}$ for all the leaves, compute the RK steps for the leaves, and compute $\bar{\mathcal{D}}$ for the intermediate RK steps.
- (5) *Update of the tree structure*:
 - Recalculate the values on the nodes and the virtual nodes by projection from the leaves.
 - Compute the details in the whole tree. If the detail in a node is smaller than the tolerance, then the node and its brothers are *deletable*. If some node and its sons are deletable, and the sons are leaves without virtual sons, then delete sons. If some node has no son and it is not deletable and it is not at level $l = L$, then create sons for this node.
 - Update the values in the new sons of the former leaves by prediction from the former leaves.

END DO n .

- (6) *Output*: Save mesh, leaves and cell averages.

Step (4) is slightly modified when we use the RKF adaptive time step. In that case, the new step is

- (4) *Time evolution with adaptive time step*:
 - Compute the discretized divergence operator for all the leaves as in (3.11).
 - Compute the difference between the two solutions obtained as in (3.13).
 - Apply the limiter for the time step variation and compute the new time step by (3.14).

The effectiveness of the multiresolution method is measured by the *data compression rate* μ and the *speed-up* V between the CPU time of the FV solution and that of the MR solution. These quantities are defined by

$$\mu := \frac{N_L}{N_L/2^L + |\mathcal{L}(\Lambda)|}, \quad V := \frac{(\text{CPU time})_{\text{VF}}}{(\text{CPU time})_{\text{MR}}}, \quad (4.2)$$

where N_L is the number of points of the finest grid, and $|\mathcal{L}(\Lambda)|$ is the number of leaves.

5. ERROR ANALYSIS OF THE ADAPTIVE MULTIREOLUTION SCHEME

The main properties of the basic FV scheme with Runge-Kutta time discretization, *i.e.* the contraction property of the scheme in the L^1 norm, the CFL stability condition and the order of approximation in space, allow to derive the optimal choice of the threshold parameter ε for the adaptive multiresolution scheme. Following the ideas introduced by Cohen *et al.* [17] and thereafter used by Roussel *et al.* [45], we decompose the global error between the cell average values of the exact solution at the level L , denoted by \bar{u}_{ex}^L , and those of the multiresolution computation with a maximal level L , denoted by \bar{u}_{MR}^L , into two errors:

$$\|\bar{u}_{\text{ex}}^L - \bar{u}_{\text{MR}}^L\| \leq \|\bar{u}_{\text{ex}}^L - \bar{u}_{\text{FV}}^L\| + \|\bar{u}_{\text{FV}}^L - \bar{u}_{\text{MR}}^L\|. \quad (5.1)$$

The first error on the right-hand side, called *discretization error* is the one of the FV scheme on the finest grid of level L . It can be bounded by

$$\|\bar{u}_{\text{ex}}^L - \bar{u}_{\text{FV}}^L\| \leq C_1 2^{-\alpha L}, \quad C_1 > 0, \quad (5.2)$$

where α is the convergence order of the FV scheme. The classical approach of Kuznetsov [37] yields $\alpha = 1/2$ for a hyperbolic scalar equation. Excepting the discontinuity due to the degeneracy, we can anticipate that $\alpha = 0.5$ is a pessimistic estimate of the convergence rate for our case. Unfortunately to our knowledge, there is not yet any theoretical result of convergence rate for numerical schemes of strongly degenerated parabolic equations. Numerical tests presented in Section 6 give $\alpha \approx 0.6$, slightly over our chosen value.

For the second error, called *perturbation error*, Cohen *et al.* [17] assume that the details on a level l are deleted when smaller than a prescribed tolerance ε_l . Under this assumption, they show that if the numerical scheme, *i.e.* the discrete time evolution operator, is contractive in the chosen norm, and if the tolerance ε_l at the level l is set to $\varepsilon_l = 2^{l-L}\varepsilon$, then the difference between FV solution on the fine grid and the solution obtained by multiresolution accumulates in time and satisfies

$$\|\bar{u}_{\text{FV}}^L - \bar{u}_{\text{MR}}^L\| \leq C_2 n \varepsilon, \quad C_2 > 0, \quad (5.3)$$

where n denotes the number of time steps. At a fixed time $T = n\Delta t$, this leads to

$$\|\bar{u}_{\text{FV}}^L - \bar{u}_{\text{MR}}^L\| \leq C_2 \frac{T}{\Delta t} \varepsilon, \quad C_2 > 0.$$

According to the CFL condition (3.6), the time step Δt must satisfy

$$\Delta t \leq \frac{\Delta x^2}{2(\Delta x \|b'\|_\infty + \|a\|_\infty)}.$$

Denoting $|I|$ the size of the domain and Δx the smallest space step, we have $\Delta x = |I|^{-1} 2^{-L}$, from which we deduce that

$$\Delta t = C_3 \frac{|I|^2 2^{-2L}}{|I| 2^{-L} \|b'\|_\infty + \|a\|_\infty}, \quad 0 < C_3 \leq \frac{1}{2}.$$

If we want the perturbation error to be of the same order as the discretization error, we need that $\frac{\varepsilon}{\Delta t} \propto 2^{-\alpha L}$ or $\varepsilon 2^{2L} (|I| 2^{-L} \|b'\|_\infty + \|a\|_\infty) \propto 2^{-\alpha L}$. This can be rewritten as

$$\varepsilon \propto \frac{2^{-(\alpha+1)L}}{|I| \|b'\|_\infty + 2^L \|a\|_\infty}. \quad (5.4)$$

For the hyperbolic case, *i.e.* $\|a\|_\infty = 0$, (5.4) is equivalent to the result $\varepsilon \propto 2^{-(\alpha+1)L}$ obtained by Cohen *et al.* [17]. Then, in the computations in Section 6, the so-called *reference tolerance* will be set to

$$\varepsilon_{\text{R}} = C \frac{2^{-(\alpha+1)L}}{|I| \|b'\|_\infty + 2^L \|a\|_\infty}. \quad (5.5)$$

Note that $\|b'\|_\infty$ and $\|a\|_\infty$ are computed numerically in the next section. To choose an acceptable value for the factor C (which, of course, depends on T , C_1 , C_2 , C_3 and $|I|$) in our numerical examples, a series of computations with different tolerances are necessary, as explained in the Appendix.

6. NUMERICAL RESULTS

6.1. Sedimentation-consolidation processes

The analysis of strongly degenerate parabolic equations has in part been motivated by a theory of sedimentation-consolidation processes of flocculated suspensions outlined in [5]. We here limit ourselves here to one-dimensional batch settling of a suspension of initial concentration $u_0 = u_0(x)$ in a column of depth H , where $u_0(x) \in [0, u_{\text{max}}]$ and u_{max} is a maximum solids volume fraction. The relevant initial-boundary value problem is Problem A, (1.1)–(2.2), with $x_a = 0$ and $x_b = H$. The unknown is the solid concentration u as a function of time t and depth x . The suspension is characterized by the hindered settling function $b(u)$ and the integrated diffusion coefficient $A(u)$, which models the sediment compressibility. The function $b(u)$ is assumed to be

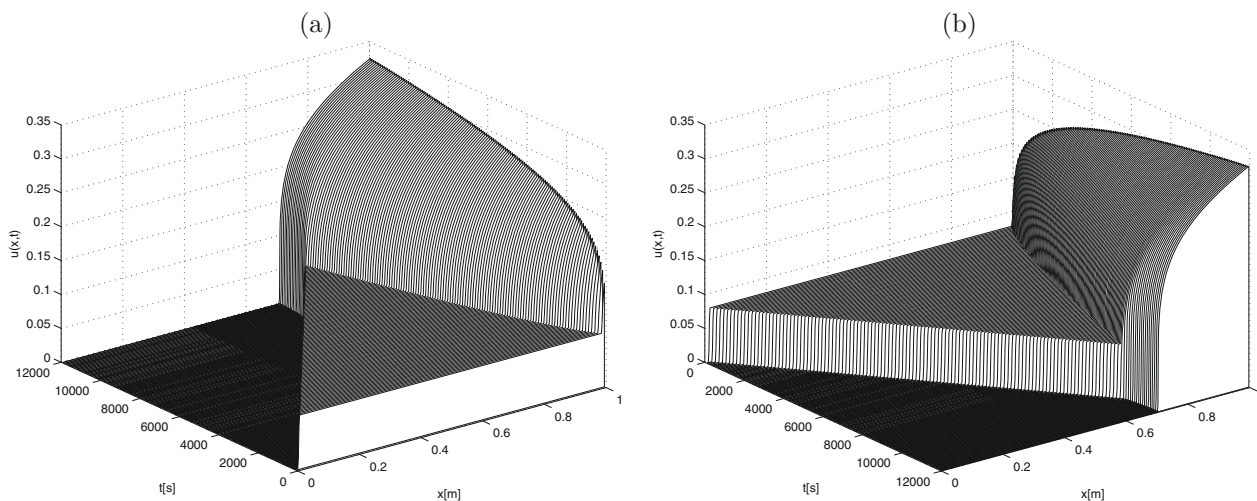


FIGURE 1. Example 1 (batch sedimentation): time-space representation of the solution.

continuous and piecewise smooth with $b(u) > 0$ for $u \in (0, u_{\max})$ and $b(u) = 0$ for $u \leq 0$ and $u \geq u_{\max}$. A typical example is

$$b(u) = \begin{cases} v_{\infty} u(u_{\max} - u)^K & \text{for } u \in (0, u_{\max}), \\ 0 & \text{otherwise,} \end{cases} \quad v_{\infty} > 0, \quad K > 0, \quad (6.1)$$

where $v_{\infty} > 0$ is the settling velocity of a single particle in unbounded fluid. Moreover, in the framework of the sedimentation-consolidation model we have that $a(u) := b(u)\sigma'_e(u)/(\Delta_{\varrho}gu)$, where $\Delta_{\varrho} > 0$ is the solid-fluid density difference, g is the acceleration of gravity, and $\sigma'_e(u)$ is the derivative of the material specific effective solid stress function $\sigma_e(u)$. We assume that the solid particles touch each other at a critical concentration value (or gel point) $0 \leq u_c \leq u_{\max}$, and that $\sigma_e(u), \sigma'_e(u) = 0$ for $u \leq u_c$ and $\sigma_e(u), \sigma'_e(u) > 0$ for $u > u_c$. This implies that $a(u) = 0$ for $u \leq u_c$, such that this application motivates an equation of type (1.1) that is indeed strongly degenerate parabolic. A typical function is

$$\sigma_e(u) = \begin{cases} 0 & \text{for } u \leq u_c, \\ \sigma_0[(u/u_c)^{\beta} - 1] & \text{for } u > u_c, \end{cases} \quad \sigma_0 > 0, \quad \beta > 1. \quad (6.2)$$

6.2. Numerical results for batch sedimentation (Example 1)

As in [10,46], we consider a suspension characterized by the functions (6.1) with $v_{\infty} = 1.0 \times 10^{-4} \text{ m}\cdot\text{s}^{-1}$, $K = 5$ and $u_{\max} = 1.0$, and (6.2) with $\sigma_0 = 1.0 \text{ Pa}$, $u_c = 0.1$ and $\beta = 6$, respectively. The corresponding functions $b(u)$ and $A(u)$ are plotted in Figure 2. The remaining parameters are $\Delta_{\varrho} = 1660 \text{ kg}\cdot\text{m}^{-3}$ and $g = 9.81 \text{ m}\cdot\text{s}^{-2}$. Note that for (6.1) and $\beta \in \mathbb{N}$, the function $A(u)$ has a closed algebraic form [6]. In Example 1, we consider an initially homogeneous suspension with $u_0 \equiv 0.08$ in a column of depth $H = 1 \text{ m}$.

Before discussing in detail the performance of the scheme for this case, we display in Figure 1 the complete numerical solution until $t = 20\,000 \text{ s}$ obtained by the fully adaptive multiresolution scheme in order to illustrate the physics of the model. Figure 1 illustrates that the suspension-clear liquid interface propagates as a sharp shock and the transition between the region of initial concentration and the sediment rising from below is sharp. We can also note the formation of a stationary sediment. In Figure 1 the visual grid used to represent the numerical solution is much coarser than the computational.

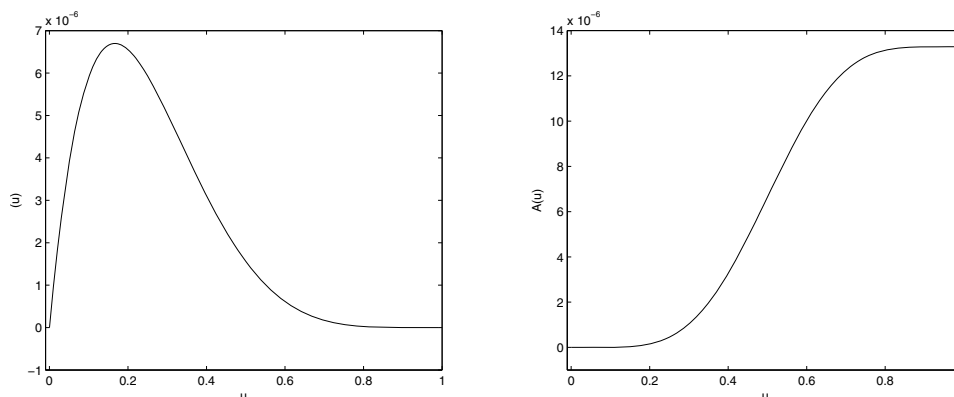


FIGURE 2. Example 1 (batch sedimentation): model functions $b(u)$ (left) and $A(u)$ (right).

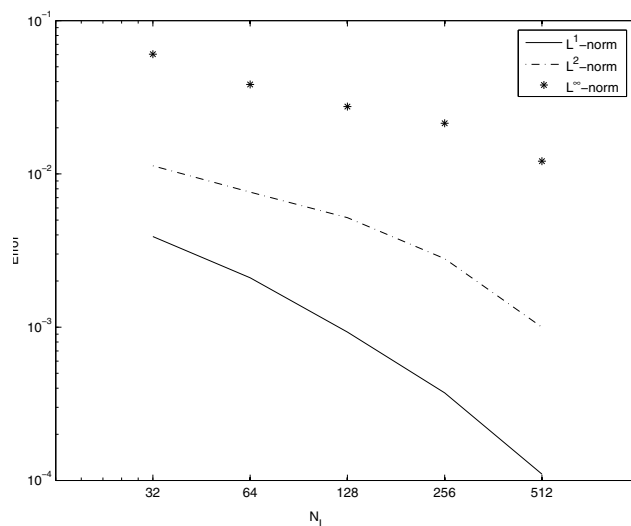


FIGURE 3. Example 1 (batch sedimentation): errors for different degrees of freedom.

6.2.1. Computations on uniform fine meshes

To estimate the convergence rate of the FV scheme detailed in Section 3.1, we performed numerical experiments using a fixed time step (with the third-order, three-step RK scheme defined by (3.9), (3.12) instead of an adaptive RKF time discretization) for all computations. We calculated a reference solution on a fine mesh with $J = 2048$ and performed computations using $J = 128, 256, 512$ and 1024 . We compared errors at three different instants: before the front and back waves meet ($t = 4000$ s), near the point at which they meet ($t = 9000$ s), and after the solution has reached a steady state ($t = 12000$ s). In all cases, the errors between the FV computations with 2^l , $l = 7, \dots, 10$ control volumes are plotted in Figure 3 versus the corresponding degrees of freedom in a double logarithmic graphic, where the slope corresponding to the L^1 norm of the error is $\alpha \approx 0.6$. We have also used different initial functions $u_0(x)$ (smooth functions and piecewise constants with several discontinuities). Of course, the observed convergence rate depends on the number and strength of discontinuities of the initial datum; to obtain a conservative estimate, we choose the following fairly rough initial condition, on which

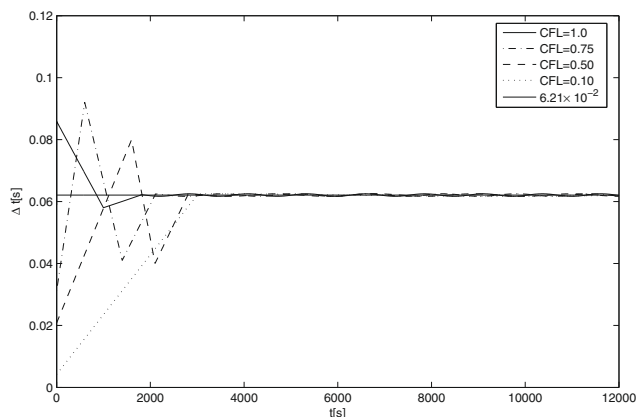


FIGURE 4. Example 1 (batch sedimentation): evolution of the time step depending on the initial CFL value for the FV scheme with global time adaptivity using the RK3(2) method.

TABLE 1. Example 1 (batch sedimentation): initial CFL, speed-up factor V , initial and final time steps, and corresponding final errors. $N_L = 128$, $\delta_{\text{desired}} = 0.0005$, $\mathcal{S}_0 = 0.1$, $\mathcal{S}_{\text{min}} = 0.01$. (*): Fixed time step RK3.

Initial CFL	V	Initial Δt [s]	Final Δt [s]	L^∞ -error	L^2 -error	L^1 -error
(*) 0.50	1	7.81×10^{-3}	7.81×10^{-3}	6.99×10^{-3}	5.96×10^{-3}	5.78×10^{-3}
1.00	27.22	5.09×10^{-2}	6.18×10^{-2}	3.16×10^{-3}	3.04×10^{-3}	2.17×10^{-3}
0.75	16.49	3.11×10^{-2}	6.15×10^{-2}	3.56×10^{-3}	2.79×10^{-3}	2.03×10^{-3}
0.50	10.34	7.81×10^{-3}	6.13×10^{-2}	3.21×10^{-3}	2.71×10^{-3}	2.05×10^{-3}
0.10	6.20	4.21×10^{-3}	6.12×10^{-2}	3.01×10^{-3}	2.38×10^{-3}	1.64×10^{-3}

the results of Figure 3 are based:

$$u_0(x) = \begin{cases} 0 & \text{for } x \in [0, \frac{1}{8}] \cup [\frac{2}{8}, \frac{1}{4}] \cup [\frac{5}{8}, \frac{3}{4}] \cup [\frac{7}{8}, 1], \\ 0.1 & \text{otherwise,} \end{cases} \tag{6.3}$$

In Figure 4 we show the evolution of the time step depending on the initial CFL value, using the FV scheme on a uniform fine mesh with global adaptivity, and the RK3(2) method introduced in Section 3.2. The parameters are $N_L = 128$, $\delta_{\text{desired}} = 0.0005$ and $\mathcal{S}_0 = 0.01$. For all choices of the initial CFL number, the time step converges to the value 0.0621 s, and Table 1 indicates that the adaptation of the step size reduces the computational cost. Here, the gain is substantially larger for initial CFL values above the maximum CFL number allowed by (3.6). We also note that the final error with respect to the reference solution is reduced for the time adaptive schemes compared to the FV scheme with fixed time step, even when these configurations violate (3.6).

6.2.2. Multiresolution examples

For this example, we take an initial dynamic graded tree, allowing $L = 11$ multiresolution levels. We use a fixed time step determined by $\lambda = 20 \text{ s} \cdot \text{m}^{-1}$, so that $\Delta t = \lambda h_L$. The prescribed tolerance ε_R is obtained from (5.5), where the constant C for this example corresponds to a factor $C = 500$ (see the Appendix for details), so $\varepsilon_R = 5.16 \times 10^{-5}$ and the threshold strategy is $\varepsilon_l = 2^{l-L} \varepsilon_R$. The constants in (5.5) are $\|b'\|_\infty = 9.0296 \times 10^{-5} \text{ m} \cdot \text{s}^{-1}$ and $\|a\|_\infty = 3.5981 \times 10^{-5} \text{ m}^2 \cdot \text{s}^{-1}$.

We simulate the process until the phenomenon enters in a steady state ($t_{\text{final}} = 12000 \text{ s}$). Figures 5a, 5c and 5e show the solution before the front and back waves meet ($t = 2000 \text{ s}$), near the point when these waves

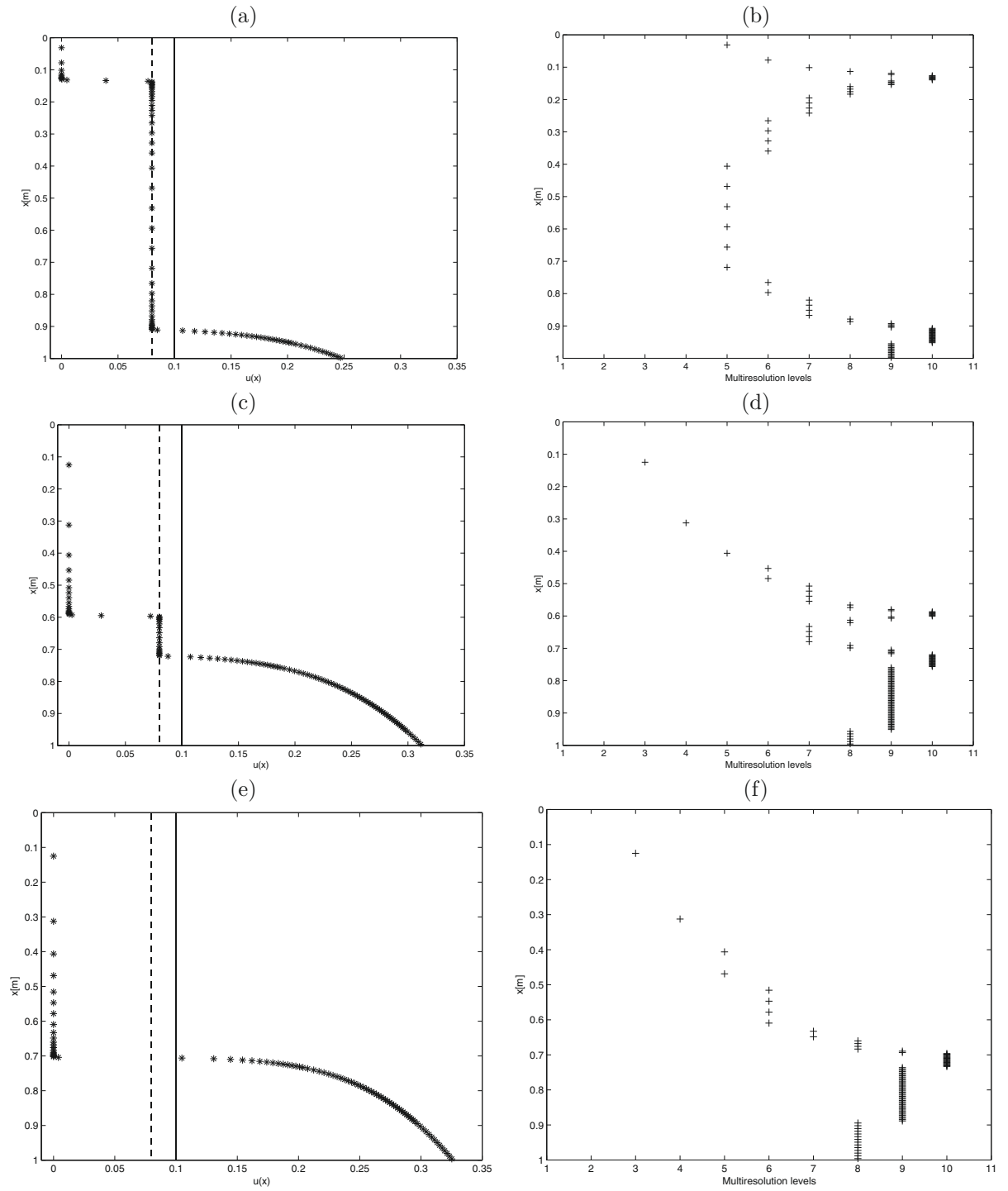


FIGURE 5. Example 1 (batch sedimentation): (a, c, e) numerical solution and (b, d, f) positions of the leaves at time instants $t = 2000$ s, $t = 9000$ s and $t = 12000$ s. The dashed vertical line is the initial datum $u_0 = 0.08$; the solid vertical line marks u_c .

TABLE 2. Example 1 (batch sedimentation): corresponding simulated time t , speed-up factor V , data compression rate μ , and normalized errors for different methods at different times. $L = 11$ multiresolution levels.

Method	V	μ	L^1 -error	L^2 -error	L^∞ -error
$t = 2000$ s					
FV	1	1	5.28×10^{-6}	1.41×10^{-6}	3.76×10^{-6}
FV + RKF	14.32	1	1.79×10^{-5}	8.91×10^{-6}	1.66×10^{-5}
MR	8.21	15.93	2.23×10^{-5}	2.35×10^{-5}	3.83×10^{-5}
MR + RKF	27.42	15.92	1.28×10^{-4}	4.57×10^{-5}	6.36×10^{-5}
$t = 9000$ s					
FV	1	1	2.11×10^{-6}	1.28×10^{-6}	8.46×10^{-6}
FV + RKF	14.38	1	3.45×10^{-5}	8.95×10^{-6}	3.29×10^{-5}
MR	9.32	15.64	3.32×10^{-5}	3.84×10^{-5}	4.72×10^{-5}
MR + RKF	31.20	15.65	2.49×10^{-4}	4.08×10^{-5}	7.65×10^{-5}
$t = 12000$ s					
FV	1	1	6.38×10^{-7}	4.84×10^{-7}	5.03×10^{-7}
FV + RKF	14.49	1	5.61×10^{-6}	1.27×10^{-6}	4.31×10^{-6}
MR	9.87	14.52	5.41×10^{-6}	6.79×10^{-5}	1.43×10^{-6}
MR + RKF	33.05	14.52	8.74×10^{-5}	1.82×10^{-5}	4.29×10^{-5}

meet ($t = 9000$ s), and after the solution has reached a steady state ($t = 12000$ s), respectively. Figures 5b, 5d and 5f display the corresponding positions of the leaves of the tree (where a “+” marks the center of each leaf). The plotted positions indicate that the adaptation of the mesh is adapted correctly, in the sense that the multiresolution method automatically detects steep gradient regions and adds finer scales.

From Table 2 we infer that the multiresolution computation is always cheaper in CPU time and in memory requirements than the FV method on the finest grid. The CPU time used with adaptive time stepping is roughly a third of the CPU time required with a fixed time step. The fixed time step for the MR calculation (with CFL = 1/2) is $\Delta t_{\text{fixed}} = 8.24 \times 10^{-3}$ s and the adaptive time step for the MR (+RKF) calculation apparently converges to $\Delta t = 2.34 \times 10^{-2}$ s $\approx 3\Delta t_{\text{fixed}}$.

It is worth pointing out that the time adaptivity is global in the sense that time stepping is dominated by the time step on the finest resolution level $l = L$.

The L^1 , L^2 , and L^∞ norms of the error between the numerical solution obtained by multiresolution for different multiresolution levels L , and a finite volume approximation on a uniform fine grid, are depicted in Figure 6. In fact, we compute in practice the error between the numerical solution obtained by multiresolution and the projection of the numerical solution by FV approximation. Comparisons of speed-up factor and data compression for different maximal multiresolution levels L are displayed in Figure 7.

Table 3 summarizes the speed-up factor V , the compression rate μ and errors for the FV and multiresolution schemes using different maximal multiresolution levels. The simulated time is $t = 2000$ s. Figure 8 shows the corresponding L^1 -errors, compared with a reference solution obtained by finite volume approximation on a uniform fine grid with 2^{13} control volumes. Note that the slope for all methods is practically the same.

6.3. Traffic flow with driver reaction

The classical Lighthill-Whitham-Richards (LWR) kinematic wave model [38,43] for unidirectional traffic flow on a single-lane highway starts from the principle of “conservation of cars” $u_t + (uv)_x = 0$ for $x \in \mathbb{R}$ and $t > 0$, where u is the density of cars as a function of distance x and time t and $v = v(x, t)$ is the velocity

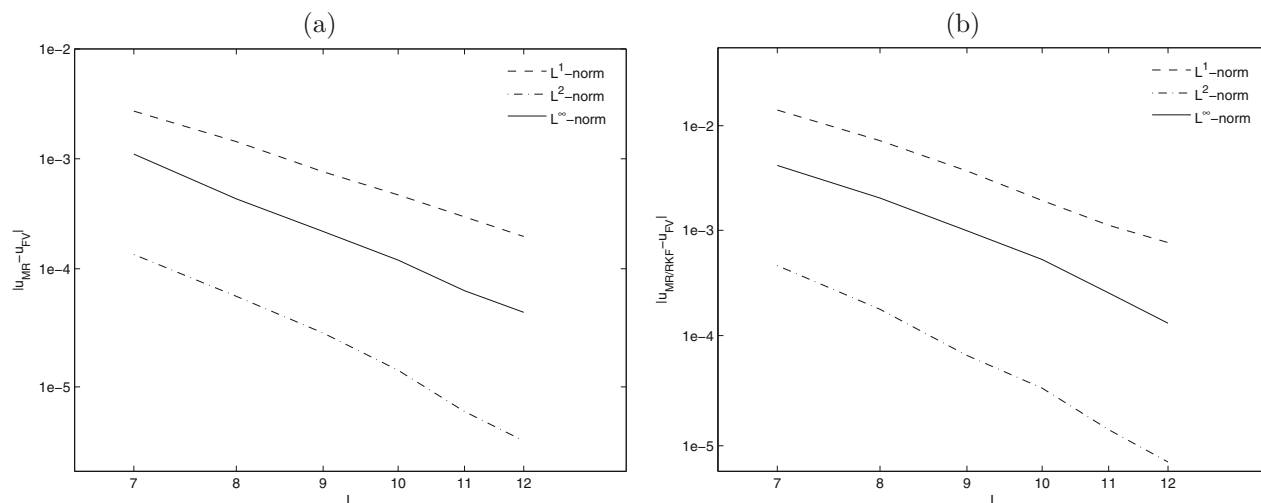


FIGURE 6. Example 1 (batch sedimentation): (a) errors $\|\bar{u}_{\text{MR}} - \bar{u}_{\text{FV}}\|_1$, $\|\bar{u}_{\text{MR}} - \bar{u}_{\text{FV}}\|_2$ and $\|\bar{u}_{\text{MR}} - \bar{u}_{\text{FV}}\|_\infty$ and (b) errors $\|\bar{u}_{\text{MR/RKF}} - \bar{u}_{\text{FV}}\|_1$, $\|\bar{u}_{\text{MR/RKF}} - \bar{u}_{\text{FV}}\|_2$ and $\|\bar{u}_{\text{MR/RKF}} - \bar{u}_{\text{FV}}\|_\infty$ for different scales L . The simulated time is $t = 2000$ s.

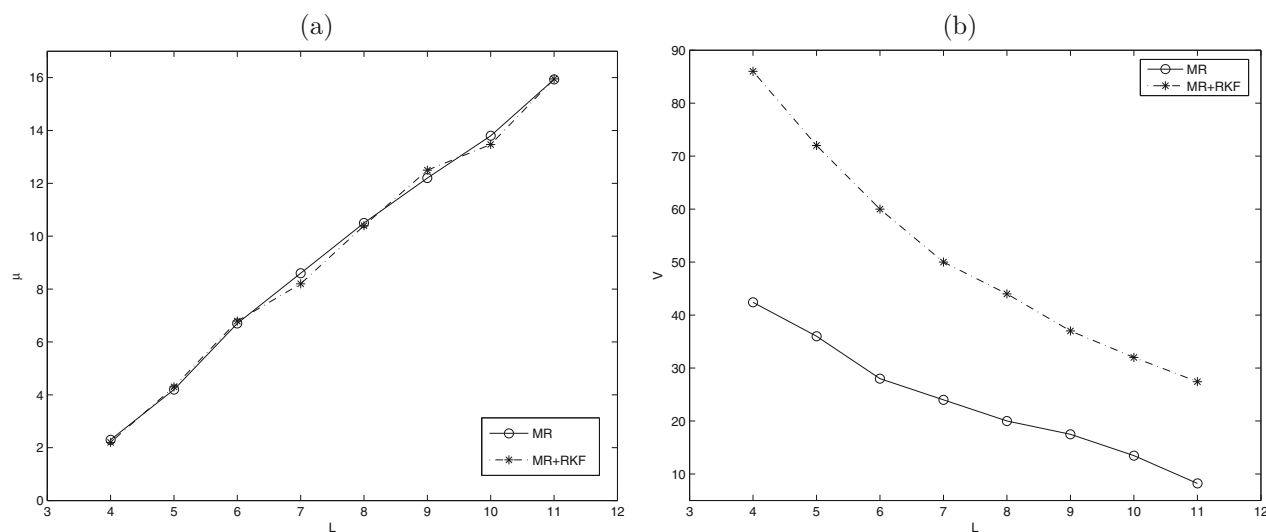


FIGURE 7. Example 1 (batch sedimentation): (a) data compression rate μ and (b) speed-up factor V for different scales L . Comparison between multiresolution with fixed and adaptive time stepping. The simulated time is $t = 2000$ s.

of the car located at position x at time t . The basic assumption of the LWR model is that v is a function of u only, $v = v(u)$, *i.e.*, each driver instantaneously adjusts his velocity to the local car density. A common choice is $v(u) = v_{\max}V(u)$, where v_{\max} is a maximum velocity a driver assumes on a free highway, and $V(u)$ is a hindrance function taking into account the presence of other cars that urges each driver to adjust his speed. Thus, the flux is

$$b(u) := uv(u) = v_{\max}uV(u) \text{ for } 0 \leq u \leq u_{\max}, \quad b(u) = 0 \text{ otherwise,} \quad (6.4)$$

TABLE 3. Example 1 (batch sedimentation): corresponding simulated time, speed-up factor V , data compression rate μ , and errors for different methods at different maximal scales.

Method	V	μ	L^1 -error	L^2 -error	L^∞ -error
$L = 10$					
FV	1	1	1.35×10^{-5}	8.70×10^{-4}	9.35×10^{-4}
MR	8.78	12.76	4.09×10^{-4}	1.35×10^{-4}	7.49×10^{-4}
MR+RKF	32.15	12.76	6.35×10^{-4}	2.74×10^{-4}	8.88×10^{-4}
$L = 11$					
FV	1	1	4.37×10^{-6}	5.68×10^{-7}	1.47×10^{-5}
MR	10.46	15.93	1.23×10^{-5}	2.35×10^{-5}	3.83×10^{-5}
MR+RKF	39.19	15.95	5.34×10^{-5}	4.39×10^{-5}	6.01×10^{-5}
$L = 12$					
FV	1	1	1.29×10^{-9}	6.37×10^{-9}	8.65×10^{-10}
MR	12.70	16.34	4.16×10^{-6}	1.01×10^{-7}	6.72×10^{-7}
MR+RKF	43.21	16.35	5.47×10^{-6}	6.44×10^{-7}	1.38×10^{-6}

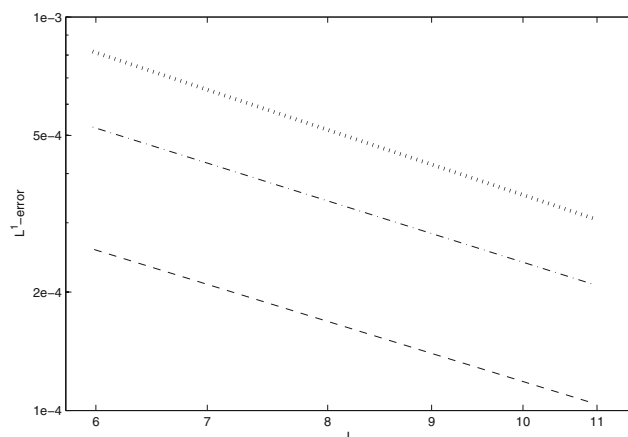


FIGURE 8. Example 1 (batch sedimentation): L^1 -errors for the MR method (dash-dotted), MR+RKF method (dotted) and FV method (dashed).

where u_{\max} is the maximum “bumper-to-bumper” car density. The simplest choice is the linear interpolation $V(u) = V_1(u) := 1 - u/u_{\max}$; but we may also consider the alternative Dick-Greenberg model [20,29]

$$V(u) = V_2(u) := \min\{1, \Theta \ln(u_{\max}/u)\}, \quad \Theta > 0. \tag{6.5}$$

The diffusively corrected kinematic wave model (DCKWM) [7,42] extends the LWR model by a strongly degenerating diffusion term. This model incorporates a reaction time τ , representing drivers’ delay in their response to events, and an anticipating distance $L_{\tilde{a}}$, which means that drivers adjust their velocity to the density seen the distance $L_{\tilde{a}}$ ahead. We use the equation [42] $L_{\tilde{a}} = \max\{(v(u))^2/(2\tilde{a}), L_{\min}\}$, where $(v(u))^2/(2\tilde{a})$ is the distance required to decelerate to full stop from speed $v(u)$ at deceleration \tilde{a} , and L_{\min} is a minimal anticipation distance, regardless of how small the velocity is. If one assumes that reaction time and anticipation distance are only effective when the local car density exceeds a critical value u_c , then the governing equation

(replacing $u_t + b(u)_x = 0$) of the DCKWM is the strongly degenerate parabolic equation

$$u_t + b(u)_x = A(u)_{xx}, \quad x \in \mathbb{R}, \quad t > 0, \tag{6.6}$$

where for simplicity we suppress dependence on the parameters τ and $L_{\tilde{a}}$, and

$$A(u) := \int_0^u a(s) ds, \quad a(u) := \begin{cases} 0 & \text{if } u \leq u_c, \\ -uv_{\max}V'(u)(L_{\tilde{a}}(u) + \tau v_{\max}uV'(u)) & \text{if } u > u_c. \end{cases} \tag{6.7}$$

There are at least two motivations for a critical density u_c . One of them is based on the Dick-Greenberg model (6.5); obviously, $V'_2(u) = 0$ for $u \leq u_c := u_{\max} \exp(-1/\Theta)$, so that (6.7) is satisfied for $V(u) = V_2(u)$. A more general viewpoint is that u_c is a threshold value in the sense that the drivers' reaction can be considered instantaneous in relatively free traffic flow, *i.e.* when $u \leq u_c$, and otherwise is modeled by the diffusion term. Both motivations give rise to the same behavior of the diffusion coefficient. Moreover, we assume that $V(u)$ is chosen such that $a(u) > 0$ for $u_c < u < u_{\max}$. Consequently, the right-hand side of (6.6) vanishes on the interval $[0, u_c]$, and possibly at the maximum density u_{\max} . Thus, the governing equation of the DCKWM model (6.6) is strongly degenerate parabolic.

6.4. Traffic flow on a circular road (Example 2)

We consider the traffic model outlined in Section 6.3, and choose the velocity function and model parameters according to [7,42]. The velocity function is given by (6.5) with $u_{\max} = 220 \text{ cars} \cdot \text{mi}^{-1}$, $C = e/7 = 0.38833$ and $v_{\max} = 70 \text{ mph}$, so that

$$b(u) = \begin{cases} v_{\max}u & \text{for } 0 \leq u \leq u_c = \exp(-1/\Theta)u_{\max} = 16.7512 \text{ cars} \cdot \text{mi}^{-1}, \\ v_{\max}(e/7)u \ln(u_{\max}/u) & \text{for } u_c < u \leq u_{\max}, \\ 0 & \text{otherwise.} \end{cases} \tag{6.8}$$

For Example 2, we choose the parameters $\tilde{a} = 0.1g$, where g is the acceleration of gravity in $\text{mi} \cdot \text{h}^{-2}$, $\tau = 2 \text{ s} = 0.0005 \text{ h}$ and $L_{\min} = 0.05 \text{ mi}$. This yields the following integrated diffusion coefficient $A(u)$, measured in $\text{cars} \cdot \text{mph}$, see [7] for details on its algebraic derivation:

$$A(u) = \begin{cases} 0 & \text{for } u \leq u_c, \\ 1.27099 \cdot (41.878u - 12.787u \ln u + u(\ln u)^2) & \text{for } u_c < u \leq u^* := 78.2198 \text{ cars} \cdot \text{mi}^{-1}, \\ -0.4105u - 286.54 & \text{for } u_c < u \leq u^*, \\ 117.9003 + 0.94864u & \text{for } u > u^*. \end{cases} \tag{6.9}$$

The functions $b(u)$ and $A(u)$ for this example are shown in Figure 10.

We assume a circular road of length $H = 10 \text{ mi}$, so that the periodic boundary condition

$$u(H, t) = u(0, t), \quad t \in (0, T] \tag{6.10}$$

applies. We assume the smooth initial density distribution $u_0(x) = 50(1 + \sin(0.4\pi x)) \text{ cars} \cdot \text{mi}^{-1}$. In addition, at $x = 5 \text{ mi}$, a traffic light S is placed with

$$S(t) = \begin{cases} 0 \text{ (red)} & \text{for } t \in [(k + 0.125) \text{ h}, (k + 0.375) \text{ h}], \quad k = 0, 1, 2, \dots, \\ 1 \text{ (green)} & \text{otherwise.} \end{cases} \tag{6.11}$$

Note that this example involves periodicity both in space (due to the periodic boundary condition (6.10)), in time (due to the behavior of the traffic light), and is provided with periodic initial condition.

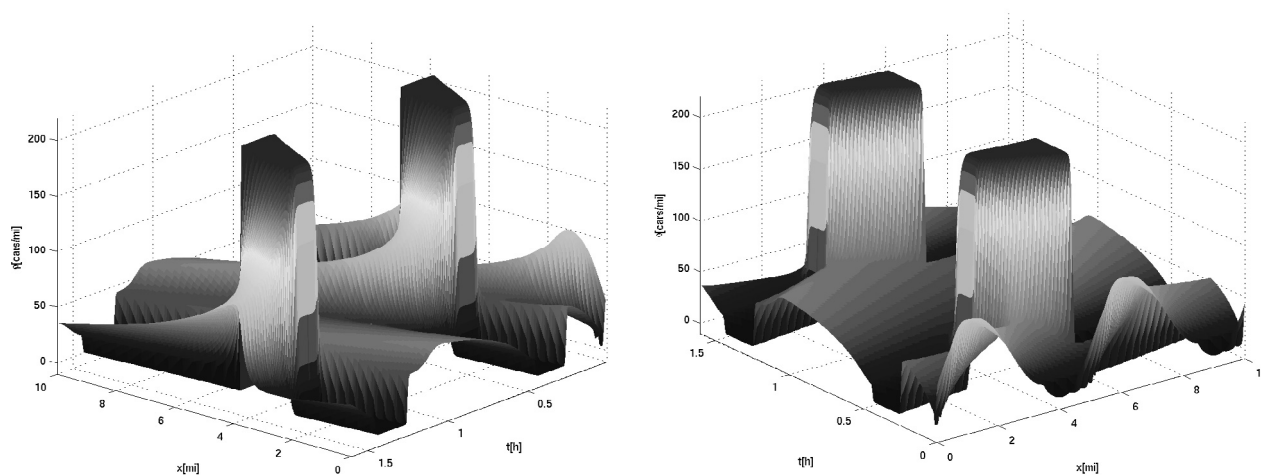


FIGURE 9. Example 2 (traffic model): time-space representation of the solution.

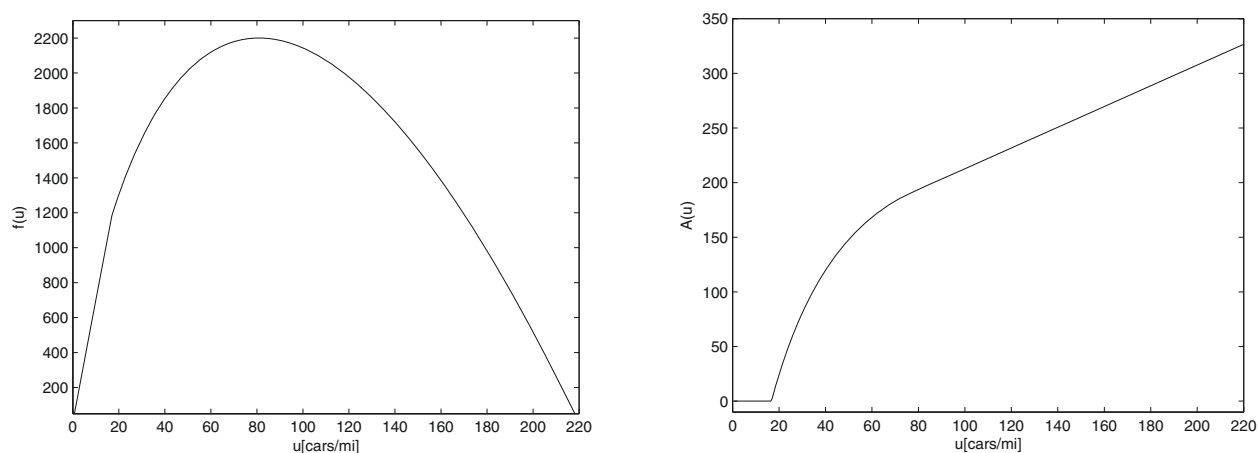


FIGURE 10. Example 2 (traffic on a circular road): model functions $b(u)$ (left) and $A(u)$ (right).

Before discussing the performance of the fully adaptive multiresolution scheme, we present in Figure 9 three-dimensional plots of the numerical solution obtained for by this scheme. It illustrates that a nearly periodic solution quickly evolves. Moreover, the depletion of the zone behind the traffic light and the queue of cars waiting in front of it periodically produces traveling type-change interfaces, *i.e.* jumps between the solution value zero and sub-critical values. In Figure 9, the visual grid used to represent the numerical solution is again much coarser than the computational one.

6.4.1. Finest grid computations

In Figure 11 we show the time evolution of the time step depending on the initial value of CFL, using the FV scheme on a uniform fine mesh with global adaptivity and the RK3(2) method described in Section 3.2. In all cases, the time step apparently converges to the value 1.217×10^{-4} s. Here, the adaptivity of the step size reduces the computational cost, see Table 4.

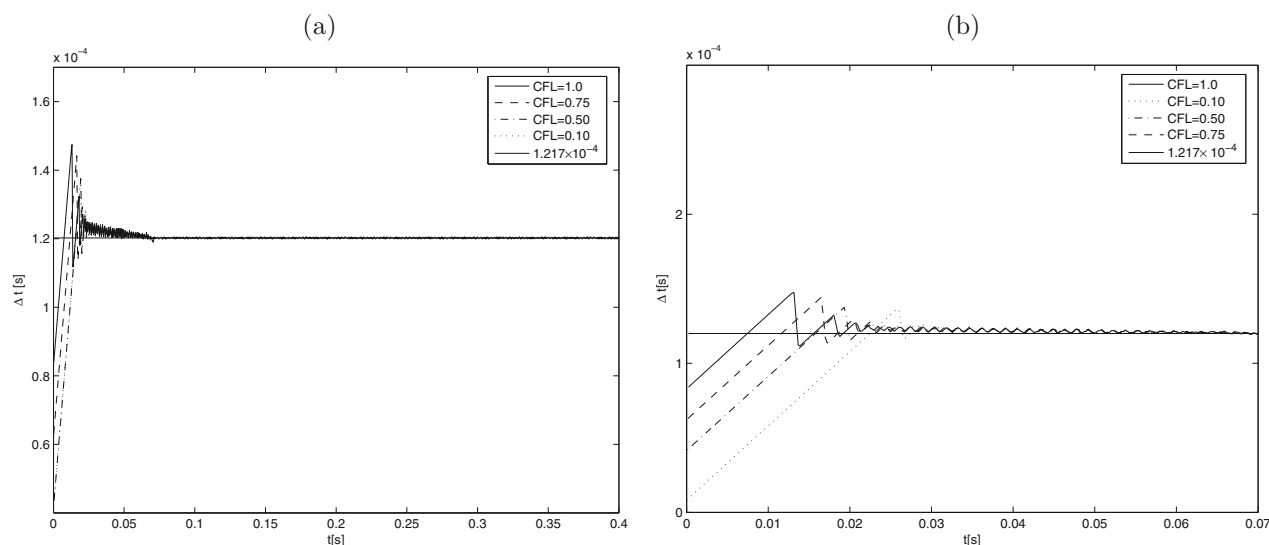


FIGURE 11. Example 2 (traffic model): (a) evolution of the time step depending on the initial value of CFL for the FV scheme with global time adaptivity using the RK3(2) method, (b) zoom until $t = 0.07$ h. $N_L = 256$, $\delta_{\text{desired}} = 0.001$ and $\mathcal{S}_0 = 0.01$.

TABLE 4. Example 2 (traffic model): initial CFL, speed-up factor V , initial and final time steps, and corresponding normalized errors. $N_L = 256$, $\delta_{\text{desired}} = 0.001$, $\mathcal{S}_0 = 0.1$, $\mathcal{S}_{\text{min}} = 0.01$. (*): Fixed time step RK3.

Initial CFL	V	Initial Δt [h]	Final Δt [h]	L^∞ -error	L^2 -error	L^1 -error
(*) 0.50	1	4.10×10^{-5}	4.10×10^{-5}	5.62×10^{-4}	1.47×10^{-4}	8.32×10^{-4}
1.00	16.32	8.22×10^{-5}	1.29×10^{-4}	8.53×10^{-3}	5.41×10^{-3}	5.06×10^{-3}
0.75	11.34	6.16×10^{-5}	1.17×10^{-4}	1.38×10^{-3}	2.77×10^{-3}	3.46×10^{-3}
0.10	8.75	8.62×10^{-5}	1.22×10^{-4}	7.51×10^{-4}	2.01×10^{-3}	8.86×10^{-4}

6.4.2. Multiresolution examples

For this example, we take an initial dynamic graded tree, allowing $L = 10$ multiresolution levels. First we use a multiresolution procedure with fixed time step, taking $\text{CFL} = 1/2$, and we simulate the process until $t_{\text{final}} = 1.6$ h. Figure 12 shows the solution before the traffic light is red and during the red phase. Figure 13 shows the solution right after the *green* phase and part of the second cycle of the traffic light. For this example, the factor C in (5.5) is set to $C = 1 \times 10^6$ (see details in the Appendix), yielding a reference tolerance of $\varepsilon_R = 1.33 \times 10^{-5}$ for $L = 10$. Here, the Lipschitz constants in (5.5) are $\|b'\|_\infty = 70 \text{ mi} \cdot \text{h}^{-1}$ and $\|a\|_\infty = 7.9177 \text{ mi}^2 \cdot \text{h}^{-1}$.

The errors in L^1 , L^2 , and L^∞ norms between the numerical solution obtained by multiresolution for different multiresolution levels L , and the solution obtained by finite volume approximation in a uniform fine grid, are depicted in Figure 14. In fact, we compute in practice the error between the numerical solution obtained by multiresolution and the projection of the numerical solution by FV approximation. Furthermore, comparisons of speed-up factor and data compression for different maximal multiresolution levels L , are depicted in Figure 15. All these calculations corresponds to the “red” phase, $t = 0.2$ h.

Tables 5 and 6 summarize the CPU time, compression rate and errors for the FV method and the multiresolution scheme. Table 5 displays these quantities for several different times, while in Table 6 we use different

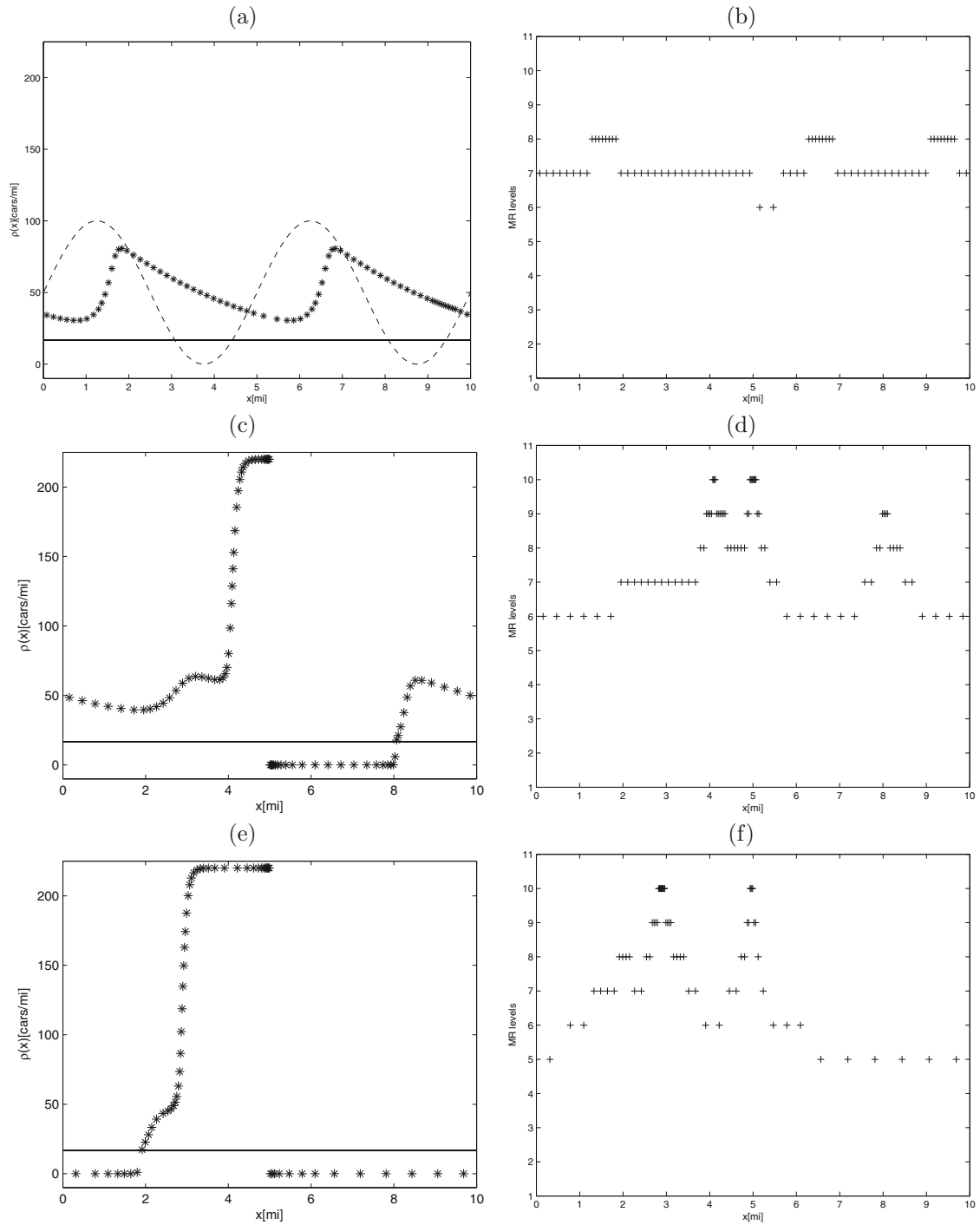


FIGURE 12. Example 2 (traffic model): (a, c, e) numerical solution and (b, d, f) positions of the leaves. The simulated times are $t = 0.1$ h, $t = 0.2$ h and $t = 0.3$ h. The dashed curve in plot (a) is the initial datum $u_0(x)$. The solid horizontal line marks u_c .

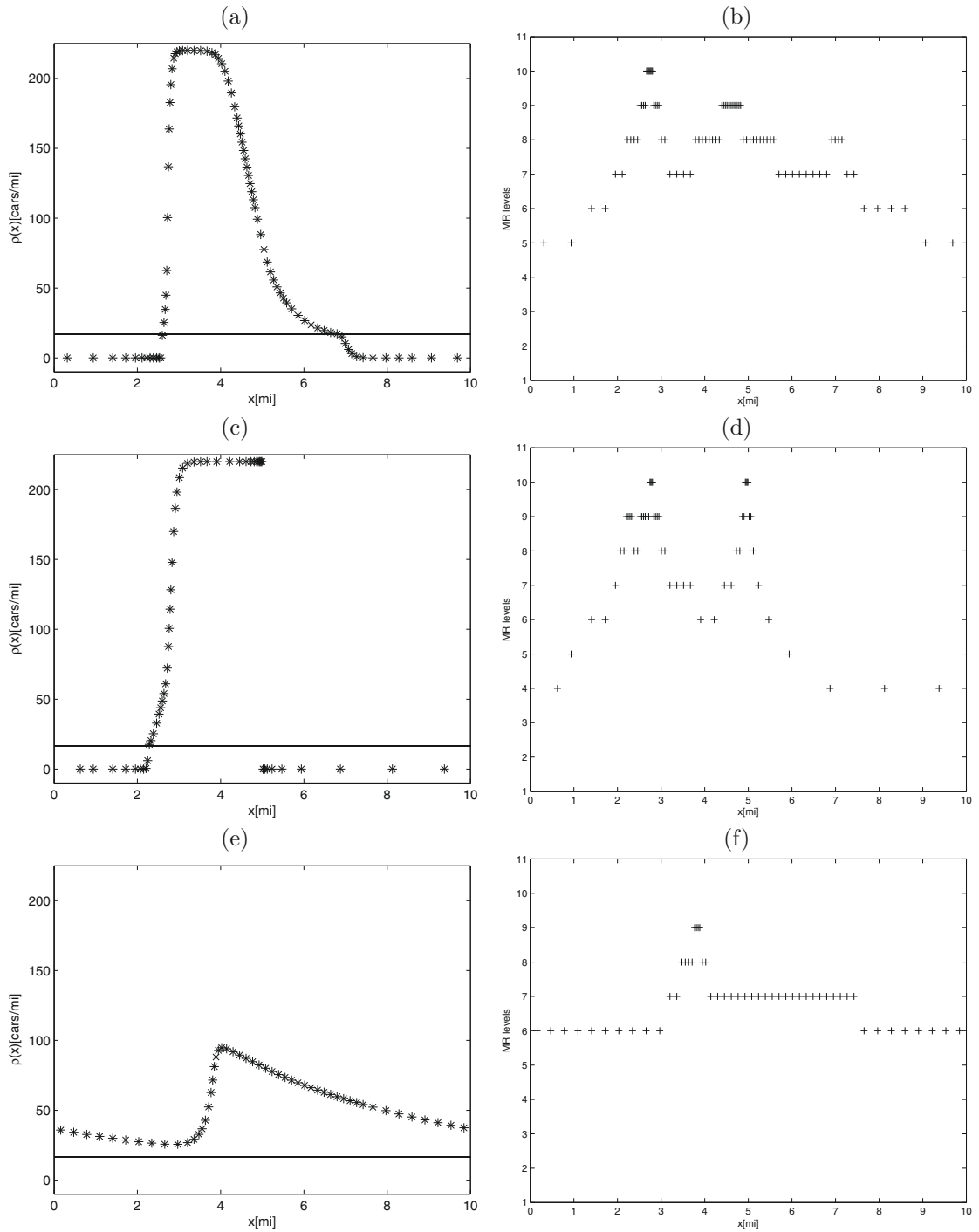


FIGURE 13. Example 2 (traffic model): numerical solution (a, c, e) and positions of the leaves (b, d, f). The simulated times are $t = 0.4$ h, $t = 1.3$ h and $t = 1.6$ h. The solid horizontal line marks u_c .

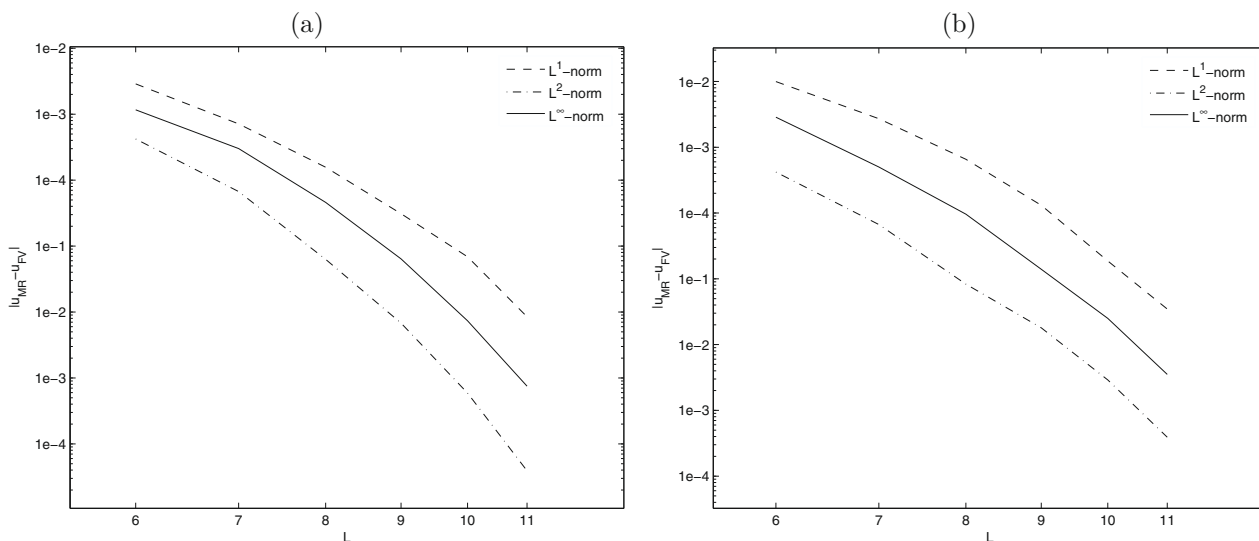


FIGURE 14. Example 2 (traffic model): (a) errors $\|\bar{u}_{MR} - \bar{u}_{FV}\|_1$, $\|\bar{u}_{MR} - \bar{u}_{FV}\|_2$ and $\|\bar{u}_{MR} - \bar{u}_{FV}\|_\infty$ and (b) errors $\|\bar{u}_{MR/RKF} - \bar{u}_{FV}\|_1$, $\|\bar{u}_{MR/RKF} - \bar{u}_{FV}\|_2$ and $\|\bar{u}_{MR/RKF} - \bar{u}_{FV}\|_\infty$ for different scales L . The simulated time is $t = 0.2$ h.

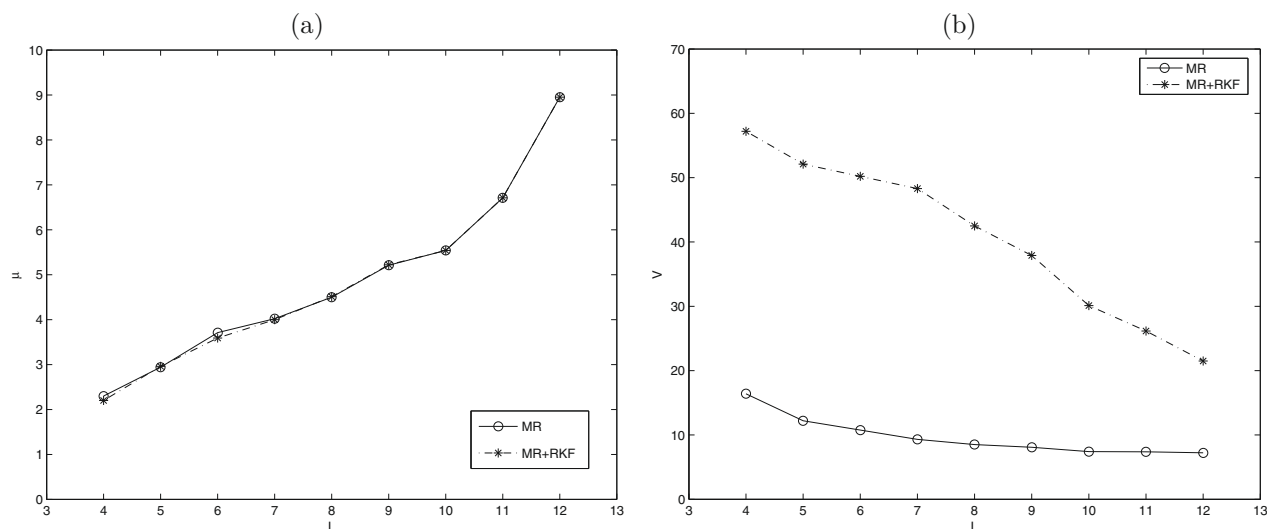


FIGURE 15. Example 2 (traffic on a circular road): (a) data compression rate μ and (b) speed-up factor V for different scales L . The simulated time is $t = 0.2$ h.

maximal multiresolution levels and consider the simulated time $t = 0.4$ h. Here we can see that the multiresolution computation is always cheaper in CPU time and in memory requirements than the finite volume method on the finest grid, and we can also notice that the gain in speed-up factor using an adaptive time stepping is about four times the speed-up factor attained by fixed time stepping, this comes clear if we also note that the fixed time step for the MR calculation, with $CFL = 1/2$ is $\Delta t = 2.217 \times 10^{-5}$ h and the adaptive time step for

TABLE 5. Example 2 (traffic model): corresponding simulated time, speed-up factor V , data compression rate μ , and errors for different methods. $L = 10$ multiresolution levels.

Method	V	μ	L^1 -error	L^2 -error	L^∞ -error
$t = 0.1\text{h}$					
MR	5.12	7.93	2.31×10^{-4}	4.79×10^{-4}	5.83×10^{-5}
MR+RKF	12.47	7.93	6.18×10^{-4}	1.07×10^{-3}	6.66×10^{-5}
$t = 0.2\text{h}$					
MR	5.21	5.99	1.76×10^{-4}	7.68×10^{-5}	8.39×10^{-6}
MR+RKF	16.34	5.99	3.67×10^{-4}	2.73×10^{-4}	3.00×10^{-5}
$t = 0.3\text{h}$					
MR	5.23	7.58	2.24×10^{-4}	7.89×10^{-5}	9.46×10^{-6}
MR+RKF	17.34	7.58	7.47×10^{-4}	9.21×10^{-5}	3.67×10^{-5}
$t = 0.4\text{h}$					
MR	7.42	5.54	1.83×10^{-4}	3.32×10^{-5}	4.36×10^{-6}
MR+RKF	19.18	5.54	4.62×10^{-4}	5.29×10^{-5}	1.08×10^{-3}
$t = 1.3\text{h}$					
MR	7.78	7.36	1.55×10^{-3}	5.64×10^{-5}	2.61×10^{-5}
MR+RKF	21.62	7.37	2.31×10^{-3}	8.06×10^{-5}	3.17×10^{-5}
$t = 1.6\text{h}$					
MR	7.59	9.75	6.17×10^{-3}	1.01×10^{-4}	2.94×10^{-4}
MR+RKF	19.63	9.75	8.40×10^{-3}	5.16×10^{-4}	4.55×10^{-4}

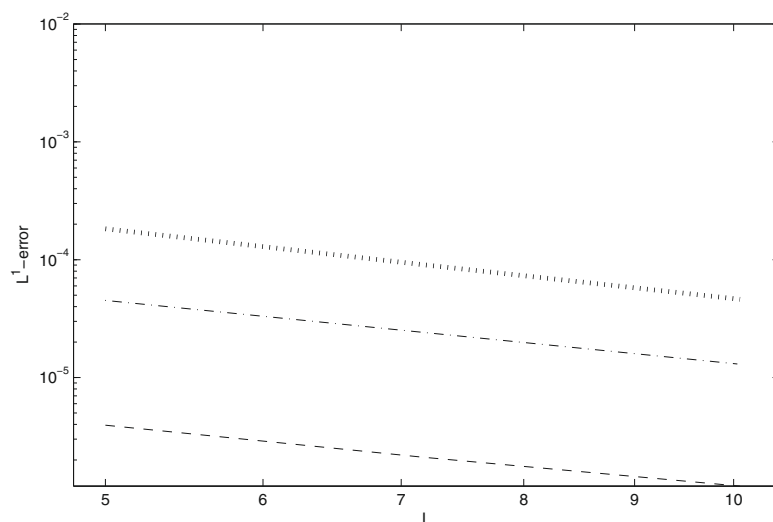
the MR (+RKF) calculation tends to converge to the value $\Delta t = 9.86 \times 10^{-5}$ h, which is roughly four times the above value. Figure 16 shows the corresponding L^1 -errors, compared with a reference solution obtained on a uniform fine grid with 2^{13} control volumes.

7. CONCLUSIONS

In the present paper we have presented numerical results using a fully space-time adaptive finite volume-multiresolution scheme developed to be used for strongly degenerate parabolic equations. The strong degeneracy of the diffusive term leads to solutions that are discontinuous in general, and in particular exhibit sharp interfaces where the equation changes between parabolic and hyperbolic type. Multiresolution schemes are natural candidates of adaptive schemes to capture these interfaces along with the classical shocks appearing in hyperbolic regions. The numerical results shown herein confirm that the proposed scheme is indeed well adapted to this kind of equations. We have also examined the advantage of this approach compared with standard FV schemes. In our experiments, we have obtained a considerable speed-up on the computations and also a substantial memory compression, while keeping the same accuracy order as in the reference schemes. The approximation order of the FV scheme is maintained due to a suitable choice of the threshold, which is a considerable improvement with respect to other multiresolution schemes where the tolerance is chosen in practice heuristically. Time step control is then introduced using an embedded RKF method which permits to adjust the time step automatically and to control the error. Different examples in one space dimension for sedimentation processes and traffic flow show the validity and efficiency of the new scheme. The interplay of the key ingredients mentioned above could represent an asset specially in the case of large number of mesh elements, systems, higher space dimensions

TABLE 6. Example 2 (traffic model): speed-up factor V , data compression rate μ , and normalized errors for different methods at different maximal scales for $t = 0.4$ h.

Method	V	μ	L^1 -error	L^2 -error	L^∞ -error
$L = 10$					
FV	2.21	1	3.94×10^{-6}	4.77×10^{-6}	2.53×10^{-6}
MR	7.42	5.54	1.83×10^{-4}	3.32×10^{-5}	4.36×10^{-4}
MR+RKF	32.11	5.54	4.62×10^{-4}	5.29×10^{-5}	1.08×10^{-3}
$L = 11$					
FV	1.54	1	2.11×10^{-6}	1.28×10^{-7}	8.46×10^{-7}
MR	7.38	6.31	9.07×10^{-5}	4.68×10^{-5}	5.71×10^{-5}
MR+RKF	26.16	6.31	3.27×10^{-4}	7.21×10^{-5}	2.15×10^{-6}
$L = 12$					
FV	1	1	1.14×10^{-8}	4.29×10^{-8}	8.65×10^{-9}
MR	7.22	8.95	4.61×10^{-5}	1.21×10^{-6}	4.72×10^{-6}
MR+RKF	21.49	8.95	1.26×10^{-4}	6.89×10^{-6}	2.02×10^{-5}

FIGURE 16. Example 2 (traffic model): L^1 -errors for the MR method (dash-dotted), MR+RKF method (dotted) and FV method (dashed).

and when the reference scheme is computationally expensive. Although we deal only with scalar and one space dimension applications, it is known that the main features of our method (*i.e.*, Runge-Kutta Fehlberg time stepping, general multiresolution and FV schemes) are applicable for the mentioned extensions.

The success of the improvements to the reference scheme depends on the proper tuning of the parameters, especially for the adaptivity of the time stepping. Also it is known that the efficiency of the multiresolution strategy is problem dependent. These are very delicate issues, and they are subject of future work. Work currently in preparation includes other applications of interest that maintain the same essential structure: strong degeneracy of the diffusive terms, but for systems and higher space dimensions (by means of tensor product strategies). Other possible extensions include the use of penalisation methods for complex domains and schemes with local time adaptivity [22,41].

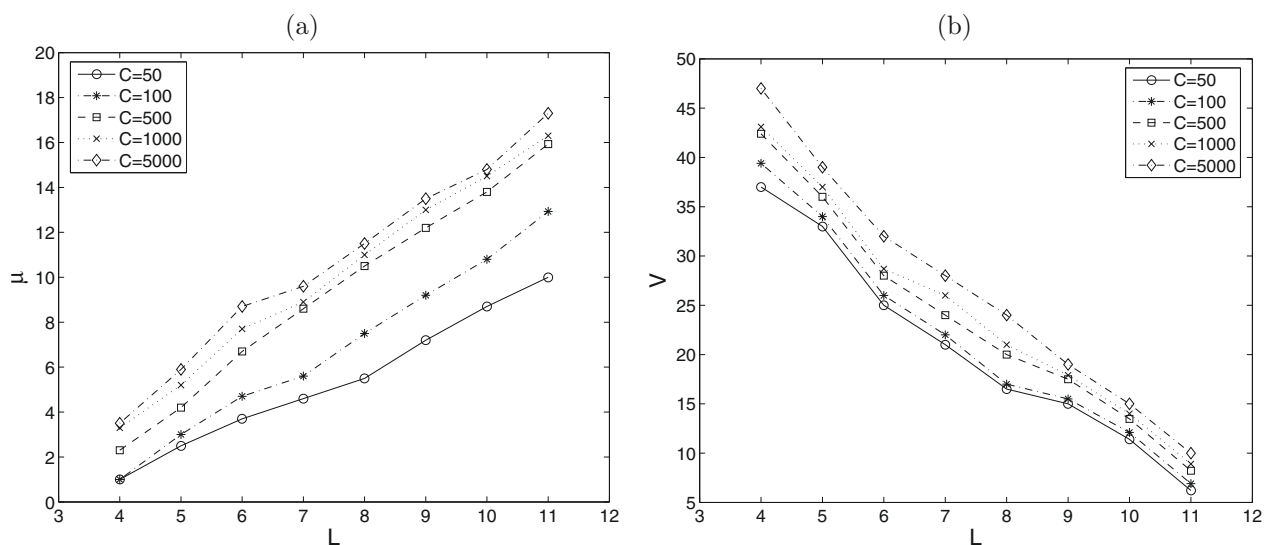


FIGURE 17. Example 1 (batch sedimentation): data compression rate μ and speed-up factor V for different scales L and values of C . The simulated time is $t = 2000$ s.

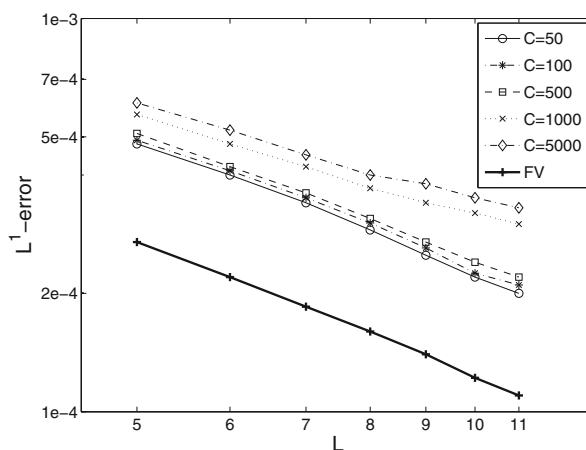


FIGURE 18. Example 1 (batch sedimentation): L^1 -errors for different scales L and values of C . The simulated time is $t = 2000$ s.

As an observation of a practical nature for readers who may wish to implement the method presented herein, we mention that in order to obtain memory compression, not only the tree data structure is essential, but also the way of navigating inside the data structure, as proposed in [45]. Each node should be represented by a pair (j, l) corresponding to the spatial position j on the level l . The children nodes need to be connected to its parent node for example by using pointers, so that the maximum number of steps required to get to any node is L because we move recursively between children nodes and its parent. After thresholding we have added a “safety zone” (extra refinement near the leaves of the tree) to ensure the proper representation of the solution in the following time step.

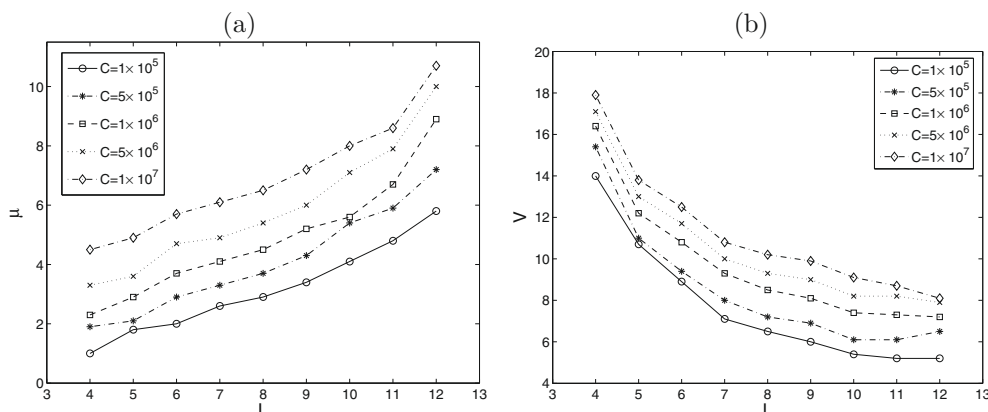


FIGURE 19. Example 2 (traffic model): data compression rate μ and speed-up factor V for different scales L and values of C . The simulated time is $t = 0.2$ h.

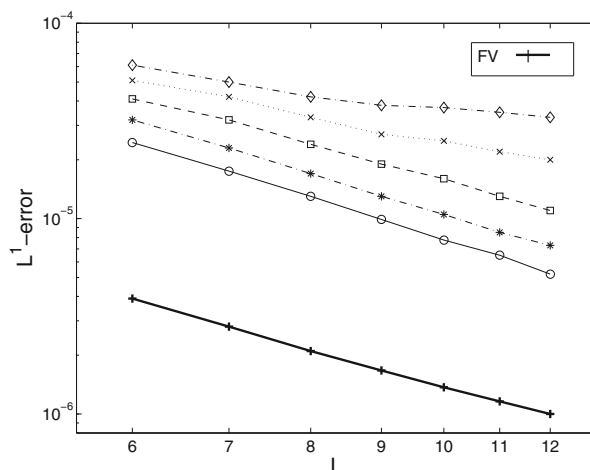


FIGURE 20. Example 2 (traffic model): L^1 -errors for different scales L and values of C ; $t = 0.2$ h. Same linestyle as in Figure 19.

APPENDIX

In this section we deal with the selection of the optimal factor C in (5.5) for the numerical examples. First, notice from Figures 17 and 19 that for all displayed scales, the multiresolution procedure is in every case (for different values of C) cheaper than the reference FV computations on the finest grid (*i.e.*, both a higher compression rate and a higher speed-up factor are attained); but the real advantage is achieved in practice for higher scales, for every tolerance level. We cannot compute a solution to the same order as that of the FV scheme for small values of L and large tolerance levels, as we can deduce from Figures 18 and 20, this loss of accuracy is caused by the perturbation error. The aim of introducing ε_R is to maintain the order of accuracy of the FV computations on the finest grid, but to gain in compression; so looking closely at Figures 18 and 20, we notice that the computations obtained using $C = 500$ for Example 1 and $C = 1 \times 10^6$ for Example 2 (and hence $\varepsilon_R = 5.16 \times 10^{-5}$ for Example 1 and $\varepsilon_R = 1.33 \times 10^{-5}$ for Example 2) are sufficiently accurate, in the sense that with these choices we keep the same order as the FV calculations while increasing V and μ . In particular,

$C = 500$ for Example 1 and $C = 1 \times 10^6$ for Example 2, the best available values of C (among those that have been tested in both cases). Also the difference of magnitude (a factor 2000) between the two constants is reasonable if we consider that C depends on the magnitude of the solution ($u_{\max} = 1$ for Example 1 and $u_{\max} = 220$ for Example 2) and on the size of the domain ($|I| = 1$ for Example 1 and $|I| = 10$ for Example 2).

Acknowledgements. RB and MS acknowledge support by Fondecyt projects 1050728 and 1070694, Fondap in Applied Mathematics, and DAAD/Conicyt Alechile project. KS and MS acknowledge support by Fondecyt of International Cooperation project 7050230. RR acknowledges support by Conicyt Fellowship. KS acknowledges support from the ANR, project M2TFP. Finally, the authors acknowledge the valuable hints and comments received from the anonymous referees.

REFERENCES

- [1] R. Becker and R. Rannacher, An optimal control approach to a posteriori error estimation in finite element methods. *Acta Numer.* **10** (2001) 1–102.
- [2] J. Bell, M.J. Berger, J. Saltzman and M. Welcome, Three-dimensional adaptive mesh refinement for hyperbolic conservation laws. *SIAM J. Sci. Comput.* **15** (1994) 127–138.
- [3] M.J. Berger and R.J. LeVeque, Adaptive mesh refinement using wave-propagation algorithms for hyperbolic systems. *SIAM J. Numer. Anal.* **35** (1998) 2298–2316.
- [4] M.J. Berger and J. Olinger, Adaptive mesh refinement for hyperbolic partial differential equations. *J. Comput. Phys.* **53** (1984) 484–512.
- [5] S. Berres, R. Bürger, K.H. Karlsen and E.M. Tory, Strongly degenerate parabolic-hyperbolic systems modeling polydisperse sedimentation with compression. *SIAM J. Appl. Math.* **64** (2003) 41–80.
- [6] R. Bürger and K.H. Karlsen, On some upwind schemes for the phenomenological sedimentation-consolidation model. *J. Eng. Math.* **41** (2001) 145–166.
- [7] R. Bürger and K.H. Karlsen, On a diffusively corrected kinematic-wave traffic model with changing road surface conditions. *Math. Models Meth. Appl. Sci.* **13** (2003) 1767–1799.
- [8] R. Bürger, S. Evje and K.H. Karlsen, On strongly degenerate convection-diffusion problems modeling sedimentation-consolidation processes. *J. Math. Anal. Appl.* **247** (2000) 517–556.
- [9] R. Bürger, K.H. Karlsen, N.H. Risebro and J.D. Towers, Well-posedness in BV_t and convergence of a difference scheme for continuous sedimentation in ideal clarifier-thickener units. *Numer. Math.* **97** (2004) 25–65.
- [10] R. Bürger, K.H. Karlsen and J.D. Towers, A model of continuous sedimentation of flocculated suspensions in clarifier-thickener units. *SIAM J. Appl. Math.* **65** (2005) 882–940.
- [11] R. Bürger, A. Coronel and M. Sepúlveda, A semi-implicit monotone difference scheme for an initial-boundary value problem of a strongly degenerate parabolic equation modelling sedimentation-consolidation processes. *Math. Comp.* **75** (2006) 91–112.
- [12] R. Bürger, A. Coronel and M. Sepúlveda, On an upwind difference scheme for strongly degenerate parabolic equations modelling the settling of suspensions in centrifuges and non-cylindrical vessels. *Appl. Numer. Math.* **56** (2006) 1397–1417.
- [13] R. Bürger, A. Kozakevicius and M. Sepúlveda, Multiresolution schemes for strongly degenerate parabolic equations in one space dimension. *Numer. Meth. Partial Diff. Equ.* **23** (2007) 706–730.
- [14] R. Bürger, R. Ruiz, K. Schneider and M. Sepúlveda, Fully adaptive multiresolution schemes for strongly degenerate parabolic equations with discontinuous flux. *J. Eng. Math.* **60** (2008) 365–385.
- [15] J. Carrillo, Entropy solutions for nonlinear degenerate problems. *Arch. Rat. Mech. Anal.* **147** (1999) 269–361.
- [16] G. Chiavassa, R. Donat and S. Müller, Multiresolution-based adaptive schemes for hyperbolic conservation laws, in *Adaptive Mesh Refinement-Theory and Applications*, T. Plewa, T. Linde and V.G. Weiss Eds., *Lect. Notes Computat. Sci. Engrg.* **41**, Springer-Verlag, Berlin (2003) 137–159.
- [17] A. Cohen, S. Kaber, S. Müller and M. Postel, Fully adaptive multiresolution finite volume schemes for conservation laws. *Math. Comp.* **72** (2002) 183–225.
- [18] M.G. Crandall and A. Majda, Monotone difference approximations for scalar conservation laws. *Math. Comp.* **34** (1980) 1–21.
- [19] P. Deuffhard and F. Bornemann, *Scientific Computing with Ordinary Differential Equations*. Springer-Verlag, New York (2002).
- [20] A.C. Dick, Speed/flow relationships within an urban area. *Traffic Eng. Control* **8** (1966) 393–396.
- [21] M. Domingues, O. Roussel and K. Schneider, An adaptive multiresolution method for parabolic PDEs with time step control. *ESAIM: Proc.* **16** (2007) 181–194.
- [22] M. Domingues, S. Gomes, O. Roussel and K. Schneider, An adaptive multiresolution scheme with local time-stepping for evolutionary PDEs. *J. Comput. Phys.* **227** (2008) 3758–3780.
- [23] B. Engquist and S. Osher, One-sided difference approximations for nonlinear conservation laws. *Math. Comp.* **36** (1981) 321–351.

- [24] M.S. Espedal and K.H. Karlsen, Numerical solution of reservoir flow models based on large time step operator splitting methods, in *Filtration in Porous Media and Industrial Application*, M.S. Espedal, A. Fasano and A. Mikelić Eds., Springer-Verlag, Berlin (2000) 9–77.
- [25] S. Evje and K.H. Karlsen, Monotone difference approximations of BV solutions to degenerate convection-diffusion equations. *SIAM J. Numer. Anal.* **37** (2000) 1838–1860.
- [26] R. Eymard, T. Gallouët, R. Herbin and A. Michel, Convergence of a finite volume scheme for nonlinear degenerate parabolic equations. *Numer. Math.* **92** (2002) 41–82.
- [27] E. Fehlberg, Low order classical Runge-Kutta formulas with step size control and their application to some heat transfer problems. *Computing* **6** (1970) 61–71.
- [28] E. Godlewski and P.A. Raviart, *Numerical Approximation of Hyperbolic Systems of Conservation Laws*. Springer-Verlag, New York (1996).
- [29] H. Greenberg, An analysis of traffic flow. *Oper. Res.* **7** (1959) 79–85.
- [30] E. Hairer, S.P. Nørsett and G. Wanner, *Solving Ordinary Differential Equations I. Nonstiff Problems*. 2nd Edn., Springer-Verlag, Berlin (1993).
- [31] A. Harten, Multiresolution algorithms for the numerical solution of hyperbolic conservation laws. *Comm. Pure Appl. Math.* **48** (1995) 1305–1342.
- [32] A. Harten, J.M. Hyman and P.D. Lax, On finite-difference approximations and entropy conditions for shocks. *Comm. Pure Appl. Math.* **29** (1976) 297–322.
- [33] K.H. Karlsen and N.H. Risebro, Convergence of finite difference schemes for viscous and inviscid conservation laws with rough coefficients. *ESAIM: M2AN* **35** (2001) 239–269.
- [34] K.H. Karlsen, N.H. Risebro and J.D. Towers, Upwind difference approximations for degenerate parabolic convection-diffusion equations with a discontinuous coefficient. *IMA J. Numer. Anal.* **22** (2002) 623–664.
- [35] K.H. Karlsen, N.H. Risebro and J.D. Towers, L^1 stability for entropy solutions of nonlinear degenerate parabolic convection-diffusion equations with discontinuous coefficients. *Skr. K. Nor. Vid. Selsk.* (2003) 1–49.
- [36] S.N. Kruzkov, First order quasilinear equations in several independent space variables. *Math. USSR Sb.* **10** (1970) 217–243.
- [37] N.N. Kuznetsov, Accuracy of some approximate methods for computing the weak solutions of a first order quasilinear equation. *USSR Comp. Math. Math. Phys.* **16** (1976) 105–119.
- [38] M.J. Lighthill and G.B. Whitham, On kinematic waves. II. A theory of traffic flow on long crowded roads. *Proc. Roy. Soc. London Ser. A* **229** (1955) 317–345.
- [39] A. Michel and J. Vovelle, Entropy formulation for parabolic degenerate equations with general Dirichlet boundary conditions and application to the convergence of FV methods. *SIAM J. Numer. Anal.* **41** (2003) 2262–2293.
- [40] S. Müller, *Adaptive Multiscale Schemes for Conservation Laws*. Springer-Verlag, Berlin (2003).
- [41] S. Müller and Y. Stiriba, Fully adaptive multiscale schemes for conservation laws employing locally varying time stepping. *J. Sci. Comp.* **30** (2007) 493–531.
- [42] P. Nelson, Traveling-wave solutions of the diffusively corrected kinematic-wave model. *Math. Comp. Modelling* **35** (2002) 561–579.
- [43] P.I. Richards, Shock waves on the highway. *Oper. Res.* **4** (1956) 42–51.
- [44] O. Roussel and K. Schneider, An adaptive multiresolution method for combustion problems: Application to flame ball-vortex interaction. *Comput. Fluids* **34** (2005) 817–831.
- [45] O. Roussel, K. Schneider, A. Tsigulin and H. Bockhorn, A conservative fully adaptive multiresolution algorithm for parabolic conservation laws. *J. Comput. Phys.* **188** (2003) 493–523.
- [46] R. Ruiz, *Métodos de Multiresolución y su Aplicación a un Problema de Ingeniería*. Tesis para optar al título de Ingeniero Matemático, Universidad de Concepción, Chile (2005).
- [47] C.-W. Shu, Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws, in *Advanced Numerical Approximation of Nonlinear Hyperbolic Equations*, B. Cockburn, C. Johnson, C.-W. Shu and E. Tadmor, in *Lecture Notes in Mathematics* **1697**, A. Quarteroni Ed., Springer-Verlag, Berlin (1998) 325–432.
- [48] J. Stoer and R. Bulirsch, *Numerische Mathematik 2*. 3rd Edn., Springer-Verlag, Berlin (1990).
- [49] E. Süli and D.F. Mayers, *An Introduction to Numerical Analysis*. Cambridge University Press, Cambridge (2003).
- [50] J.D. Towers, Convergence of a difference scheme for conservation laws with a discontinuous flux. *SIAM J. Numer. Anal.* **38** (2000) 681–698.
- [51] J.D. Towers, A difference scheme for conservation laws with a discontinuous flux: the nonconvex case. *SIAM J. Numer. Anal.* **39** (2001) 1197–1218.