Applied Numerical Mathematics 59 (2009) 1668–1692

Contents lists available at ScienceDirect

Applied Numerical Mathematics

www.elsevier.com/locate/apnum

Adaptive multiresolution schemes with local time stepping for two-dimensional degenerate reaction–diffusion systems

Mostafa Bendahmane[a], Raimund Bürger[b,*], Ricardo Ruiz-Baier[c], Kai Schneider[d]

[a] Department of Mathematics, Al-Imam University, Saudi Arabia
[b] CI^2 MA and Departamento de Ingeniería Matemática, Facultad de Ciencias Físicas y Matemáticas, Universidad de Concepción, Casilla 160-C, Concepción, Chile
[c] Departamento de Ingeniería Matemática, Facultad de Ciencias Físicas y Matemáticas, Universidad de Concepción, Casilla 160-C, Concepción, Chile
[d] Centre de Mathématiques et d'Informatique, Université de Provence, 39, rue Joliot-Curie, 13453 Marseille cedex 13, France

ARTICLE INFO

Article history:
Received 29 July 2008
Received in revised form 27 November 2008
Accepted 2 December 2008
Available online 7 December 2008

MSC:
35K65
35L65
35R05
65M06
76T20
92C17

Keywords:
Degenerate parabolic equation
Adaptive multiresolution scheme
Pattern formation
Finite volume schemes
Chemotaxis
Keller–Segel systems
Flame balls interaction
Locally varying time stepping

ABSTRACT

Spatially two-dimensional, possibly degenerate reaction–diffusion systems, with a focus on models of combustion, pattern formation and chemotaxis, are solved by a fully adaptive multiresolution scheme. Solutions of these equations exhibit steep gradients, and in the degenerate case, sharp fronts and discontinuities. This calls for a concentration of computational effort on zones of strong variation.

The multiresolution scheme is based on finite volume discretizations with explicit time stepping. The multiresolution representation of the solution is stored in a graded tree ("quadtree"), whose leaves are the non-uniform finite volumes on whose borders the numerical divergence is evaluated. By a thresholding procedure, namely the elimination of leaves with solution values that are smaller than a threshold value, substantial data compression and CPU time reduction is attained. The threshold value is chosen such that the total error of the adaptive scheme is of the same order as that of the reference finite volume scheme.

Since chemical reactions involve a large range of temporal scales, but are spatially well localized (especially in the combustion model), a locally varying adaptive time stepping strategy is applied. For scalar equations, this strategy has the advantage that consistence with a CFL condition is always enforced. Numerical experiments with five different scenarios, in part with local time stepping, illustrate the effectiveness of the adaptive multiresolution method. It turns out that local time stepping accelerates the adaptive multiresolution method by a factor of two, while the error remains controlled.

1. Introduction

Multiresolution techniques were first introduced by Harten [18] to improve the performance of schemes for one-dimensional conservation laws. Later on, these original ideas were extended to several related problems [2,5,9], leading finally to the concept of fully adaptive multiresolution schemes [12,13,28,37]. Overviews on multiresolution methods for conservation laws are given by Chiavassa, Donat, and Müller [10] and Müller [28]. The basic aim of this approach is to accelerate a given finite volume scheme on a uniform grid at the cost of an at most controllable loss of accuracy, that is, the accelerated scheme should be of the same order as the original one. The principle of the multiresolution analysis is to

* Corresponding author.
E-mail addresses: mostafab@ing-mat.udec.cl (M. Bendahmane), rburger@ing-mat.udec.cl (R. Bürger), rruiz@ing-mat.udec.cl (R. Ruiz-Baier), kschneid@cmi.univ-mrs.fr (K. Schneider).

0168-9274/$30.00 © 2008 IMACS. Published by Elsevier B.V. All rights reserved.
doi:10.1016/j.apnum.2008.12.001

represent a set of data given on a fine grid as values on a coarser grid plus a series of differences, called *details*, at different levels of nested dyadic grids. These differences contain information on the local regularity of the solution, and are small in regions where the solution is smooth. By thresholding small details (cells whose coefficients are smaller than a prescribed tolerance are removed), a locally refined adaptive grid is defined. This threshold should be chosen such that the discretization error of the reference scheme (i.e., the finite volume scheme on a fine uniform grid) is balanced with the accumulated thresholding error introduced in each time step. Significant speed-up of the computation and data compression is achieved for long-time evolution problems, large systems, multidimensional domains, and solutions with sharp fronts.

The present paper serves two purposes. On one hand, the adaptive multiresolution scheme for parabolic PDEs [37] and strongly degenerate parabolic PDEs in one space dimension [6,7] is extended to two-dimensional systems of (possibly degenerate) parabolic PDEs. These equations produce solutions that vary smoothly wherever the solution causes the PDE to be parabolic, but produce sharp fronts, or even discontinuities, close to solution values at which the equation degenerates, so adaptive multiresolution methods are a proper device to efficiently capture these fronts. Similar features (that is, solutions with steep gradients) also appear in a combustion model of reaction–diffusion type. The analysis made in [36] is extended here to the study of two interacting flame balls. In this case, an adaptive strategy is equally very useful, especially when the flame front is well localized in space, since fine grids are only needed in small subregions of the computational domain. We utilize here an adaptive multiresolution scheme applied to a reference finite volume discretization with explicit time integration.

On the other hand, chemical reactions are known to involve a large range of temporal scales, especially in long-time evolutions. Then an adaptive time stepping strategy is recommendable. Earlier efforts in this direction, which include [7,11,14,17] and the references therein, were based on using the same time step to advance the solution on all parts of the computational domain, and controlling the time step through an embedded pair of Runge–Kutta schemes (known as Runge–Kutta–Fehlberg (RKF) schemes). In these procedures, one compares the numerical solution after each time step with an (approximate) reference solution, and adjusts the time step if the discrepancy is unacceptable. In contrast to this approach, we here adapt the locally varying time stepping strategy recently introduced for multiresolution schemes for conservation laws and multidimensional systems by Lamby, Müller, and Stiriba [23], Müller and Stiriba [29] and Domingues et al. [15]. This strategy is not precisely (time-)adaptive for scalar equations, since the time step for each level remains the same for all times. However, for non-linear systems, the coupling of components entering the CFL condition makes it necessary to compute the time step after each iteration, according the evolving CFL condition, and this produces a scheme adaptive in time. Our results in terms of CPU time savings are encouraging and the strategy is consistent with a CFL condition, in contrast to the approach based on the RKF device. We mention that Müller and Stiriba [29] also combine local time stepping and multiresolution for implicit schemes, and that more details are also given in the germinal papers of Berger and Oliger [1], Osher and Sanders [33] and the references therein.

The remainder of this paper is organized as follows. In Section 2 the reaction–diffusion systems studied herein are briefly described. In Section 3 we present the reference finite volume methods to which we apply the multiresolution device, and in Section 4 the adaptive multiresolution strategy, as well as the required graded tree data structure, are outlined. In Section 5 we recall the error analysis for adaptive multiresolution schemes applied to one-dimensional scalar conservation laws by Cohen et al. [12] and to strictly parabolic equations by Roussel et al. [37], who for these respective cases derive an optimal choice of the threshold, in the sense that the total error of the multiresolution scheme is of the same order as that of the reference finite volume scheme. This choice is rigorously based on the L^1 contractivity of the numerical solution operator and a known order of convergence. These results have not yet been established for the equations under study herein, but we argue that the same threshold based on a convergence rate determined from numerical experiments may be employed; as it turns out later, this choice also provides an effective error control.

In Section 6 we address the local time stepping applied to the multiresolution strategy, and in Section 7, we outline the overall multiresolution procedure. Finally, in Section 8 the method is applied to different scenarios. Example 1 corresponds to a single-species reaction–diffusion equation, examples 2 and 3 deal with the thermo-diffusive model for the interaction between flame balls, examples 4 and 5 show Turing-type pattern formation produced by a reaction–diffusion system, and example 6 arises from a model of chemotaxis with growth. Conclusions of our study are collected in Section 9. All numerical results clearly reveal high resolution and improvement in terms of compression of memory and savings in computational effort.

2. A class of reaction–diffusion systems

2.1. A single-species reaction–diffusion model

Model 1 is the following initial-boundary value problem for a scalar reaction–diffusion equation, where $\mathbf{x} = (x, y)$ and $(x, y, t) \in Q_T := \Omega \times [0, T]$, $\Omega \subset \mathbb{R}^2$:

$$u_t = f(u, \mathbf{x}) + \Delta A(u), \tag{2.1a}$$

$$u(\mathbf{x}, 0) = u_0(\mathbf{x}) \quad \text{on } \Omega, \tag{2.1b}$$

$$\nabla A(u) \cdot \mathbf{n} = 0 \quad \text{on } \Sigma_T := \partial\Omega \times [0, T]. \tag{2.1c}$$

This problem may serve as a scalar prototype degenerate reaction–diffusion model. Here, the zero-flux boundary condition (2.1c) implies that the reaction–diffusion domain is isolated from the external environment. For $f(u,\mathbf{x}) = f(u)$, (2.1a) appears in [31] in an ecological setting, where u denotes the population density of a species, and $f(u)$ is its dynamics, where it is assumed that $f(0) = 0$ and $f'(0) \neq 0$. For example, $f(u) = u(1-u) - u^2/(1+u^2)$ corresponds to the population dynamics of the spruce band-worm [31], and models the growth of the population by a logistic expression and the rate of mortality due to predation by other animals. We modify this expression by a radial spatial factor, and use

$$\begin{aligned} &f(u,\mathbf{x}) := 10\left(\exp(-5r)u(1-u) + \left(\exp(-5r)-1\right)\frac{u^2}{1+u^2}\right), \\ &r := \sqrt{(x-0.5)^2 + (y-0.5)^2}, \end{aligned} \tag{2.2}$$

which means that the birth of individuals is concentrated near the center $(0.5, 0.5)$, and mortality increases with increasing distance from the center. On the other hand, most standard spatial models of population dynamics simply assume that $A(u) = Du$, where the constant diffusion coefficient $D > 0$ measures the dispersal efficiency of the species under consideration. Motivated by Witelski [41], who advanced degenerate diffusion in the context of population dynamics, we utilize herein the strongly degenerate diffusion coefficient

$$A(u) = \begin{cases} 0 & \text{for } u \leqslant u_{\mathrm{c}}, \\ D(u-u_{\mathrm{c}}) & \text{otherwise}, \end{cases} \tag{2.3}$$

where $u_{\mathrm{c}} > 0$ is an assumed critical (threshold) value of u beyond which diffusion will take place. Model 1 gives rise to example 1 of Section 8.

The difficulty in the well-posedness analysis of the problem (2.1) lies in the boundary condition (2.1c) when A is strongly degenerate. It is quite difficult to give a correct formulation of the zero flux boundary conditions. For the case of non-homogeneous Dirichlet boundary conditions, however, Mascia, Porretta, and Terracina [24] demonstrated existence and uniqueness of L^∞ entropy solutions. In the special case where the function A is strictly increasing, the classical framework of variational solutions of parabolic equations is sufficient to satisfy this wish.

2.2. A two-species reaction–diffusion model

Model 2 is given by the following initial-boundary value problem for a reaction–diffusion system on Q_T:

$$u_t = \gamma f(u,v) + \Delta A(u) \quad \text{on } Q_T, \tag{2.4a}$$

$$v_t = \gamma g(u,v) + d\Delta B(v) \quad \text{on } Q_T, \tag{2.4b}$$

$$u(\mathbf{x},0) = u_0(\mathbf{x}), \quad v(\mathbf{x},0) = v_0(\mathbf{x}) \quad \text{for } \mathbf{x} \in \Omega, \tag{2.4c}$$

$$\nabla A(u) \cdot \mathbf{n} = \nabla B(u) \cdot \mathbf{n} = 0 \quad \text{on } \Sigma_T. \tag{2.4d}$$

We study this system in the context of two applications, namely as a model of combustion and as a two-species model of mathematical biology.

For $A(u) = B(u) = u$, $d = 1/\mathrm{Le}$ and $\gamma = 1$, (2.4) represents a reduced dimensionless thermo-diffusive model describing a combustion process, where Le is the Lewis number. We restrict ourselves to a simple chemical reaction with only two reactants and one product, the first reactant and the product being highly diluted in the second reactant, and we neglect gravity. Since the chemical reaction takes place in a lean premixed gas, we focus on the limiting reactant, and denote by v its normalized partial mass, while u represents normalized temperature. The reaction rates are given by an Arrhenius law:

$$f(u,v) := \frac{\beta^2}{2} v \exp\left(\frac{\beta(1-u)}{\alpha(1-u)-1}\right), \qquad g(u,v) := -f(u,v), \tag{2.5}$$

where α and β are the temperature rate and the dimensionless activation energy, called Zeldovich number, respectively. In example 2 of Section 8, this model is employed to simulate the interaction between two flame balls, as an extension of the applications of the same model that were considered in [35,36]. Here, a *flame ball* denotes a slowly propagating spherical flame structure in a premixed gaseous mixture.

If radiation effects are taken into account, (2.4a) is replaced by

$$u_t = \gamma f(u,v) + S(u) + \Delta A(u) \quad \text{on } Q_T, \tag{2.6}$$

where the dimensionless heat loss due to radiation S follows the Stefan–Boltzmann law

$$S(u) = \rho\left[\left(u + \alpha^{-1} - 1\right)^4 - \left(\alpha^{-1} - 1\right)^4\right], \tag{2.7}$$

where α is the same parameter as in (2.5) and the dimensionless coefficient ρ controls the radiation level. Conditions (2.4d) imply that the process takes place inside a box with adiabatic walls. See [36] for details and a discussion of the case with one flame ball. The interaction of two flame balls including radiation is simulated in example 3 of Section 8.

On the other hand, (2.4) also arises in mathematical biology as a well-known reaction–diffusion system modelling the interaction between two chemical species with respective concentrations u and v. Under certain conditions, it produces stationary solutions with Turing-type spatial patterns [31,40]. To simulate the formation of such a pattern, we here select the kinetics between each species due to Schnakenberg [38]:

$$f(u,v)=a-u+u^2v, \qquad g(u,v)=b-u^2v. \tag{2.8}$$

Alternative choices of f and g that lead to Turing-type patterns are discussed in [30,31]. For

$$A(u)=B(u)=u, \tag{2.9}$$

this system has a uniform positive steady state (u^0,v^0) given by $u^0=a+b$ and $v^0=b/(a+b)^2$, where $b>0$ and $a+b>0$, and under certain conditions, (2.4) has a unique solution. See for instance [3] for the proof of existence and uniqueness.

We recall from [31, Section 2.3] some results on the conditions under which (2.4) produces a diffusion-driven instability giving rise to Turing-type pattern in the non-degenerate case. A necessary condition is that the inequalities $f_u+g_v<0$, $f_ug_v-f_vg_u>0$, $df_u+g_v>0$ and $(df_u+g_v)^2-4d(f_ug_v-f_vg_u)>0$ are satisfied at $(u,v)=(u^0,v^0)$. Evaluating these inequalities for the system (2.4a), (2.4b) and the particular kinetics (2.8) at $(u^0,v^0)=(a+b,b/(a+b)^2)$ yields the inequalities

$$\begin{aligned}&0<b-a<(a+b)^3, \quad (a+b)^2>0, \quad d(b-a)>(a+b)^3,\\ &\bigl(d(b-a)-(a+b)^3\bigr)^2>4d(a+b)^4.\end{aligned} \tag{2.10}$$

To characterize the stationary pattern that arises from a choice of (a,b) that satisfies (2.10), we define

$$L^{\pm}:=\frac{d(b-a)-(a+b)^3\pm\sqrt{[d(b-a)-(a+b)^3]^2-4d(a+b)^4}}{2d(a+b)}.$$

The analysis of general rectangular domains [31] implies that in the non-degenerate case, the unstable patterned solution of the initial-boundary value problem (2.4) is given by

$$\mathbf{w}(x,y,t)=\sum_{m,n}\mathbf{c}_{nm}\exp\bigl(\lambda(k^2)t\bigr)\cos(n\pi x)\cos(m\pi y),$$

where $\mathbf{w}=(u,v)^{\mathrm{T}}$, the constants $\mathbf{c}_{nm}$ depend on a Fourier series of the initial condition for $\mathbf{w}$, the summation takes place over all numbers n and m that satisfy $\gamma L^-<k^2=\pi^2(n^2+m^2)<\gamma L^+$, and $\lambda(k^2)$ is the positive solution of the following equation, where f_u, f_v, g_u and g_v are evaluated at (u^0,v^0):

$$\lambda^2+\lambda\bigl(k^2(1+d)-\gamma(f_u+g_v)\bigr)+dk^4-\gamma(df_u+g_v)k^2+\gamma^2(f_ug_v-f_vg_u)=0.$$

Example 4 of Section 8 presents a numerical solution of (2.4) with the kinetics (2.8) and the diffusion terms (2.9), where parameters are chosen according to the preceding discussion such that indeed a Turing-type pattern is produced. On the other hand, in example 5, we present a simulation where (2.9) is replaced by the degenerate diffusion functions

$$A(u)=\begin{cases}0 & \text{for } u\leqslant u_{\mathrm{c}},\\ u-u_{\mathrm{c}} & \text{otherwise},\end{cases}\qquad B(u)=\begin{cases}0 & \text{for } u\leqslant v_{\mathrm{c}},\\ u-v_{\mathrm{c}} & \text{otherwise},\end{cases}\qquad u_{\mathrm{c}},v_{\mathrm{c}}\geqslant 0. \tag{2.11}$$

It turns out that even if the stability analysis does *not* apply to the degenerate case, our numerical experiments (example 5) lead to the formation of a pattern.

The mathematical analysis of the system (2.4) is still an open problem due to the boundary condition (2.4d). A successful technique for proving uniqueness of (entropy weak) solutions to degenerate parabolic equations with Dirichlet boundary condition is based on Kružkov's method [22]. Related to this approach we mention that Holden, Karlsen, and Risebro [19] prove existence and uniqueness of entropy solutions of weakly coupled systems of degenerate parabolic equations in an unbounded domain. The system (2.4a), (2.4b) is an example of the degenerate reaction–diffusion system analyzed in [19], but is equipped here with the boundary condition (2.4d).

2.3. A chemotaxis-growth system

We assume that $\Omega\subset\mathbb{R}^2$ is convex, bounded and open. Model 3 is the following generalization of the Keller–Segel model [20,21] for chemotactical movement:

$$u_t=\nabla\cdot\bigl(\sigma\nabla u-u\nabla\chi(v)\bigr)+g(u) \quad \text{on } Q_T, \tag{2.12a}$$

$$v_t=h(u,v)+d\Delta v \quad \text{on } Q_T, \tag{2.12b}$$

$$u(\mathbf{x},0)=u_0(\mathbf{x}), \quad v(\mathbf{x},0)=v_0(\mathbf{x}) \quad \text{on } \Omega, \tag{2.12c}$$

$$\nabla u\cdot\mathbf{n}=\nabla v\cdot\mathbf{n}=0 \quad \text{on } \Sigma_T. \tag{2.12d}$$

The system (2.12) describes the aggregation of slime molds caused by their chemotactical features. Cell migration appears in numerous biological phenomena. In the case of chemotaxis, cells (or an organism) move in response to a chemical gradient. Specifically, (2.12) corresponds to the model proposed by Mimura and Tsujikawa [25] for the spatio-temporal aggregation patterns shown by the bacteria *Escherichia coli*. This model incorporates four effects: diffusion, chemotaxis, production of chemical substance, and population growth. In the absence of growth, $u = u(\mathbf{x}, t)$ usually represents the density of the cell population of the amoebae *Dictyostelium discoideum*, $v = v(\mathbf{x}, t)$ is the concentration of the chemo-attractant (*cAMP*: *cyclic Adenosine Monophosphate*), and χ denotes the chemotactical sensitivity function, which may be given by

$$\chi(v) = \nu v, \quad \nu > 0, \tag{2.13}$$

where ν is a chemotactical parameter. The function g takes into account the growth rate of the population, and can be given by

$$g(u) = u^2(1-u). \tag{2.14}$$

Moreover, $\sigma > 0$ and $d > 0$ are constant diffusion rates for both components. The function h describes the rates of production and degradation of the chemo-attractant; here, we choose

$$h(u, v) = \alpha u - \beta v, \quad \alpha, \beta \geqslant 0. \tag{2.15}$$

Osaki et al. [32] showed that if $0 \leqslant u_0 \in L^2(\Omega)$, $0 \leqslant v_0 \in H^{1+r}(\Omega)$, and $\partial\Omega$ is smooth enough, (2.12) possesses a unique global solution, and Perthame [34] proved that if u_0 and v_0 are radial and $\|u_0\|_{L^1}$ is sufficiently large, then $\|u(t)\|_{L^2}$ blows up in finite time. On the other hand, Efendiev, Kläre, and Lasser [16] analyzed the fractal dimension of the exponential attractor in dependence of ν. Our example 6 of Section 8 is based on examples presented in [16], and presents numerical solutions of (2.12) for various values of ν.

3. Finite volume schemes

We discretize a reaction–diffusion equation by a standard finite volume scheme, which is described here for a uniform grid. The rectangular spatial domain $\Omega \subset \mathbb{R}^2$ is partitioned into control volumes $(\Omega_{ij})_{(i,j)\in\Lambda}$, where Λ is the index set

$$\Lambda := \{(i, j)\colon\ i = 1, \ldots, N_{Lx},\ \ j = 1, \ldots, N_{Ly}\},$$

where $N_{Lx} \cdot N_{Ly}$ is the number of cells of the grid,

$$\Omega_{ij} := [x_{i-1/2}, x_{i+1/2}] \times [y_{j-1/2}, y_{j+1/2}],$$

$\Delta x := x_{i+1/2} - x_{i-1/2}$, $\Delta y := y_{j+1/2} - y_{j-1/2}$, for all $(i, j) \in \Lambda$, and $\widetilde{\Delta x} := \min\{\Delta x, \Delta y\}$. The cell average of a quantity q at time t is defined by

$$\bar{q}_{ij}(t) = \frac{1}{|\Omega_{ij}|} \iint\limits_{\Omega_{ij}} q(\mathbf{x}, t)\, d\mathbf{x}. \tag{3.1}$$

3.1. Discretization of models 1 and 2

The finite volume scheme is described here for (2.1) and as it applies to the first equation of (2.4); for the second equation of (2.4), we replace u by v, $f(u, v)$ by $g(u, v)$, and $A(u)$ by $dB(v)$. Integrating the respective equation and averaging over Ω_{ij} yields

$$\frac{1}{|\Omega_{ij}|} \iint\limits_{\Omega_{ij}} u_t(\mathbf{x}, t)\, d\mathbf{x} = \frac{1}{|\Omega_{ij}|} \iint\limits_{\Omega_{ij}} \mathcal{D}\big(u(\mathbf{x}, t), \nabla A\big(u(\mathbf{x}, t)\big)\big)\, d\mathbf{x} + \frac{1}{|\Omega_{ij}|} \iint\limits_{\Omega_{ij}} f\big(u(\mathbf{x}, t)\big)\, d\mathbf{x}, \tag{3.2}$$

where $\mathcal{D}$ denotes the right-hand side of the PDE under consideration except for the reaction term. For the two-dimensional case and on a Cartesian grid, $\mathcal{D}$ is discretized via

$$\begin{aligned}
\bar{\mathcal{D}}_{ij} &:= -\frac{1}{\Delta x}(\bar{F}_{i+1/2,j} - \bar{F}_{i-1/2,j}) - \frac{1}{\Delta y}(\bar{F}_{i,j+1/2} - \bar{F}_{i,j-1/2}),\\
\bar{F}_{i+1/2,j} &:= -\frac{1}{\Delta x}\big(A(\bar{u}_{i+1,j}) - A(\bar{u}_{ij})\big), \qquad \bar{F}_{i,j+1/2} := -\frac{1}{\Delta y}\big(A(\bar{u}_{i,j+1}) - A(\bar{u}_{ij})\big).
\end{aligned}$$

The reaction term is approximated by $\bar{f}_{ij} \approx f(\bar{u}_{ij}, \bar{v}_{ij})$. If we incorporate a first-order Euler time integration for both components, then the corresponding interior marching formula for model 2 is

$$\begin{aligned}
\bar{u}_{ij}^{n+1} &= \bar{u}_{ij}^{n} + \Delta t \gamma \bar{f}_{ij} + \Delta t \bar{\mathcal{D}}_{ij}\big(\mathcal{S}\big(\bar{u}_{ij}^{n}\big), \widetilde{\Delta x}\big),\\
\bar{v}_{ij}^{n+1} &= \bar{v}_{ij}^{n} + \Delta t \gamma \bar{g}_{ij} + d \Delta t \bar{\mathcal{D}}_{ij}\big(\mathcal{S}\big(\bar{v}_{ij}^{n}\big), \widetilde{\Delta x}\big),
\end{aligned} \tag{3.3}$$

where $\mathcal{S}(\cdot)$ denotes the stencil utilized for computing $\bar{\mathcal{D}}_{ij}$. According to [8,19], this scheme is stable under the following CFL condition, where $\lambda := \Delta t/\widetilde{\Delta x}$ and $\mu := \Delta t/\widetilde{\Delta x}^2$:

$$\lambda \gamma \big(\|f_u\|_\infty + \|f_v\|_\infty + \|g_u\|_\infty + \|g_v\|_\infty\big) + 4\mu\big(\|A'\|_\infty + d\|B'\|_\infty\big) \leqslant 1. \tag{3.4}$$

3.2. Discretization of model 3

We define the difference operators $\delta_x^\pm V_{ij} := \pm(V_{i\pm1,j} - V_{ij})$ and $\delta_y^\pm V_{ij} := \pm(V_{i,j\pm1} - V_{ij})$. Then a suitable second order difference operator for a general term $\nabla \cdot (Q\, \nabla u)$ is

$$\nabla \cdot (Q\, \nabla u) \approx \frac{1}{\Delta x^2} \delta_x^+ (Q_{i+1/2,j} \delta_x^- u_{ij}) + \frac{1}{\Delta y^2} \delta_y^+ (Q_{i,j+1/2} \delta_y^- u_{ij}).$$

Integrating the corresponding equations, averaging over Ω_{ij} and discretizing yields the following interior marching formula:

$$\begin{aligned}
\bar{u}_{ij}^{n+1} &= \bar{u}_{ij}^{n} + \frac{\sigma \Delta t}{\Delta x^2} \delta_x^+ \delta_x^- \bar{u}_{ij}^{n} + \frac{\sigma \Delta t}{\Delta y^2} \delta_y^+ \delta_y^- \bar{u}_{ij}^{n} + \frac{\Delta t}{\Delta x^2}\big(\delta_x^+\big(Q_{i-1/2,j}^{n} \delta_x^- \bar{v}_{ij}^{n}\big)\big) + \frac{\Delta t}{\Delta y^2}\big(\delta_y^+\big(Q_{i,j-1/2}^{n} \delta_y^- \bar{v}_{ij}^{n}\big)\big) + \Delta t g\big(\bar{u}_{ij}^{n}\big),\\
\bar{v}_{ij}^{n+1} &= \bar{v}_{ij}^{n} + \Delta t h\big(\bar{u}_{ij}^{n}, \bar{v}_{ij}^{n}\big) + \frac{d \Delta t}{\Delta x^2} \delta_x^+ \delta_x^- \bar{v}_{ij}^{n} + \frac{d \Delta t}{\Delta y^2} \delta_y^+ \delta_y^- \bar{v}_{ij}^{n},\\
Q_{i,j+1/2}^{n} &:= \frac{1}{2}\big(\chi'\big(\bar{v}_{ij}^{n}\big)\bar{u}_{ij}^{n} + \chi'\big(\bar{v}_{i,j+1}^{n}\big)\bar{u}_{i,j+1}^{n}\big),\\
Q_{i+1/2,j}^{n} &:= \frac{1}{2}\big(\chi'\big(\bar{v}_{ij}^{n}\big)\bar{u}_{ij}^{n} + \chi'\big(\bar{v}_{i+1,j}^{n}\big)\bar{u}_{i+1,j}^{n}\big).
\end{aligned} \tag{3.5}$$

Analogously to (3.4), the scheme (3.5) is stable under the corresponding CFL condition

$$\lambda\big(\|h_u\|_\infty + \|h_v\|_\infty + \|g'\|_\infty\big) + 4\mu\big(\sigma + \|\chi'\|_\infty + d\big) \leqslant 1. \tag{3.6}$$

The left-hand sides of (3.4) and (3.6) obviously evolve in time, so in practice, at each time step we obtain Δt from these conditions, and λ and μ are not constants; rather, they are adjusted in each time step.

4. Conservative adaptive multiresolution discretization

In this section we recall some basic properties of the multiresolution discretization and the data structure. For a more detailed description, we refer to [6,37]. We organize the numerical solution and corresponding differences at different levels, in a *dynamic graded tree structure*: whenever a *node* is included in the tree, all other nodes corresponding to the same spatial region on coarser levels of resolution are also included. The tree structure is mainly needed for ease of navigation, which contributes to accelerating the scheme; data compression would also be possible by other techniques. The adaptive grid corresponds to a set of nested dyadic grids generated by refining recursively a given cell depending on the local regularity of the solution.

We denote by *root* the basis of the tree. In two space dimensions, a parent node has four sons, and the sons of the same parent are called *brothers*. A node without sons is called a *leaf*. A given node has $s' = 2$ *nearest neighbors* in each direction needed for the computation of the fluxes of leaves; if these nearest neighbors do not exist, we create them as *virtual leaves*. Fig. 1 illustrates the graded tree structure. The leaves of the tree are the control volumes from which we form the adaptive mesh. Here, the property of the tree of being *graded* means that grid refinement and coarsening is governed by the rule that two neighboring control volumes cannot differ by more than one level in the tree. This is equivalent to the so-called "one-irregular rule" (see e.g., [26,27]) in the finite element community. 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, $0 \leqslant l \leqslant L$. We denote

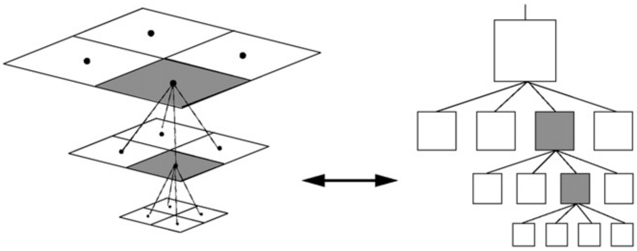

Fig. 1. Graded tree data structure ("quadtree"), after [28].

by $\bar{\mathbf{w}}_{i,j,l} = (\bar{u}_{i,j,l}, \bar{v}_{i,j,l})$ the vector of cell averages for both components of the solution (and the obvious simplification $\bar{\mathbf{w}}_{i,j,l} = \bar{u}_{i,j,l}$ for a single-species problem) located at spatial position (i, j) at level l, and by $\bar{\mathbf{W}}_l$ the set of cell averages for all nodes at level l. To estimate the cell averages of a level l from those of the next finer level $l+1$, we use the projection operator $P_{l+1\to l}$, which is exact, unique, and in our case is defined by

$$\bar{\mathbf{w}}_{i,j,l} = \frac{1}{4}\sum_{e_1,e_2\in\{0,1\}} \bar{\mathbf{w}}_{2i+e_1,2j+e_2,l+1}. \tag{4.1}$$

To estimate the cell averages of a level $l+1$ from those of level l, we employ the prediction operator $P_{l\to l+1}$, which provides an approximation $\hat{\mathbf{w}}$ by interpolation of $\bar{\mathbf{W}}_l$ at level $l+1$. This operator is local in the sense that the interpolation for a son is made from the cell averages of its parent and the s nearest neighbors of its parent; and it is consistent with the projection in the sense that it is conservative with respect to the coarse grid cell averages or equivalently, $P_{l+1\to l} \circ P_{l\to l+1} = \mathrm{Id}$. For a regular grid structure in two dimensions, we use a polynomial interpolation introduced in [2]:

$$\hat{\mathbf{w}}_{2i+e_1,2j+e_2,l+1} = \bar{\mathbf{w}}_{i,j,l} - (-1)^{e_1} Q_x - (-1)^{e_2} Q_y + (-1)^{e_1e_2} Q_{xy}, \quad e_1, e_2 \in \{0,1\}, \tag{4.2}$$

where

$$\begin{aligned} Q_x &:= \sum_{n=1}^{s} \tilde{\gamma}_n (\bar{\mathbf{w}}_{i+n,j,l} - \bar{\mathbf{w}}_{i-n,j,l}), \qquad Q_y := \sum_{p=1}^{s} \tilde{\gamma}_p (\bar{\mathbf{w}}_{i,j+p,l} - \bar{\mathbf{w}}_{i,j-p,l}), \\ Q_{xy} &:= \sum_{n=1}^{s} \tilde{\gamma}_n \sum_{p=1}^{s} \tilde{\gamma}_p (\bar{\mathbf{w}}_{i+n,j+p,l} - \bar{\mathbf{w}}_{i+n,j-p,l} - \bar{\mathbf{w}}_{i-n,j+p,l} + \bar{\mathbf{w}}_{i-n,j-p,l}). \end{aligned} \tag{4.3}$$

The chosen accuracy order of the multiresolution method for our cases is $r = s + 1 = 3$, where s is the number of required nearest uncles for each spatial direction. The corresponding coefficients are $\tilde{\gamma}_1 = -\frac{22}{128}$ and $\tilde{\gamma}_2 = \frac{3}{128}$. Nevertheless, one may select here an arbitrarily higher order of accuracy.

As stated before, the adaptive grid consists in the set of leaves $\mathcal{L}(\Lambda)$, which forms a partition of the computational domain Ω.

A *detail* is the difference between the exact and the predicted value:

$$\bar{d}^u_{i,j,l} := \bar{u}_{i,j,l} - \hat{u}_{i,j,l}, \qquad \bar{d}^v_{i,j,l} := \bar{v}_{i,j,l} - \hat{v}_{i,j,l}.$$

For *multicomponent solutions*, there are several possible definitions of a scalar detail $\bar{d}_{i,j,l}$ that is calculated from the details of the components (in our case, $\bar{d}^u_{i,j,l}$ and $\bar{d}^v_{i,j,l}$), depending mainly on the nature of the problem. Roussel and Schneider [35] utilize the Euclidean norm, $\bar{d}_{i,j,l} = ((\bar{d}^u_{i,j,l})^2 + (\bar{d}^v_{i,j,l})^2)^{1/2}$. In our case, given the nature of our problems, we could simply select one component $\bar{d}_{i,j,l} = \bar{d}^u_{i,j,l}$, as was done by Sjögreen [39] for the compressible Euler equations, but to guarantee that the computations of the refinement and coarsening procedures are always on the safe side, in the sense that we always prefer to keep a node with a detail pair containing at least one value above the threshold

$$\varepsilon_l = 2^{2(l-L)}\varepsilon, \tag{4.4}$$

we use

$$\bar{d}_{i,j,l} = \min\{|\bar{d}^u_{i,j,l}|, |\bar{d}^v_{i,j,l}|\}$$

for the refinement procedure and

$$\bar{d}_{i,j,l} = \max\{|\bar{d}^u_{i,j,l}|, |\bar{d}^v_{i,j,l}|\}$$

for the coarsening procedure (see Algorithm 7.1), similar to Harten's choice [18] for the Euler equations of gas dynamics.

Since a parent has four sons, the consistency property of the prediction operator implies that knowledge of the cell average values of the four sons is equivalent to that of the cell average value of the parent node and three independent details. Repeating this operation recursively on L levels, we get the *multiresolution transform* on the cell average values $\bar{\mathbf{M}} : \bar{\mathbf{W}}_L \mapsto (\bar{D}_L, \ldots, \bar{D}_1, \bar{\mathbf{W}}_0)$, where $\bar{D}_l := (\bar{d}^u_{i,j,l}, \bar{d}^v_{i,j,l})_{(i,j)}$. This means that knowledge of the cell averages of all leaves is equivalent to that of the cell average value of the root and of the details of all nodes of the tree.

After thresholding, i.e., removing nodes where the detail is below the prescribed tolerance $\bar{d}_{i,j,l} < \varepsilon_l$, where ε_l is given by (4.4); a *safety zone* is added to the tree, which means that one finer level is added to the tree for all active nodes without violating the graded tree data structure. This is done by splitting each leaf into four new leaves, such that the new tree remains graded. This device, which was proposed e.g. in [18,37], ensures that the graded tree will represent adequately the solution in the next time step, and depends strongly on the assumption of finite propagation speed of sharp fronts.

Let us now suppose that the tree has only two levels l and $l+1$. (The reasoning based on this assumption is straightforwardly extensible to an arbitrarily larger tree.) To ensure conservativity of the scheme, we compute only the fluxes at level

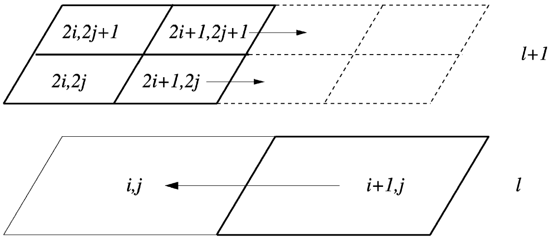

Fig. 2. Conservative flux computation for coarser levels.

$l+1$ and we set the ingoing flux on the leaf at level l equal to the sum of the outgoing fluxes on the leaves of level $l+1$ (see Fig. 2)

$$F_{(i+1,j,l)\to(i,j,l)} = F_{(2i+1,2j,l+1)\to(2i+2,2j,l+1)} + F_{(2i+1,2j+1,l+1)\to(2i+2,2j+1,l+1)}. \tag{4.5}$$

This choice decreases the number of costly flux evaluations without loosing the conservativity in the flux computation, and this presents a real advantage when using a graded tree structure, see e.g. [37]. This advantage is lost for a non-graded tree structure, in which case these data (fluxes for leaves on an immediately finer level) are not always available.

For systems of reaction–diffusion equations, where species tend to attract each other, we manage the multiresolution framework and the data structure as one unified mesh with two components per control volume. This means that we construct only one graded tree and apply only one thresholding strategy for both species; however, there are other cases where it is preferable to organize the different species in separate adaptive meshes, for example when the species segregate spatially, as in systems of conservation laws modelling traffic flow and polydisperse sedimentation [4].

5. Choice of the threshold parameter

5.1. Error analysis for conservation laws and strictly parabolic equations

We briefly recall here the error analysis conducted first in [12] and then in [37] for scalar, one-dimensional conservation laws and strictly parabolic equations, respectively. Based on the main properties of the respective reference finite volume scheme on a uniform grid at the finest level L, such as the contraction property in L^1 norm, the CFL stability condition and order of approximation in space, the authors of both papers decompose the global error between the cell average values of the exact solution vector at the level L, denoted here by u^L_{ex}, and those of the multiresolution computation with a maximal level L, denoted by $\bar{u}^L_{\mathrm{MR}}$, into two errors

$$\left\|\bar{u}^L_{\mathrm{ex}} - \bar{u}^L_{\mathrm{MR}}\right\| \leqslant \left\|\bar{u}^L_{\mathrm{ex}} - \bar{u}^L_{\mathrm{FV}}\right\| + \left\|\bar{u}^L_{\mathrm{FV}} - \bar{u}^L_{\mathrm{MR}}\right\|. \tag{5.1}$$

The first error on the right-hand side, called *discretization error*, is that of the reference finite volume scheme on a uniform grid at the finest level L. For both a scalar, one-dimensional conservation law [12] and a strictly parabolic equation [37], the order of convergence, denoted $\tilde{\alpha}$, of the corresponding reference finite volume scheme is known ($\tilde{\alpha} = 1/2$ and $\tilde{\alpha} = 2$, respectively), which permits us to bound the discretization error by

$$\left\|\bar{u}^L_{\mathrm{ex}} - \bar{u}^L_{\mathrm{FV}}\right\| \leqslant C_1 2^{-\tilde{\alpha} L} \tag{5.2}$$

for a solution-dependent constant $C_1 > 0$. For the second error, called *perturbation error*, Cohen et al. [12] assume that the details on a level l are deleted when they are smaller than a prescribed tolerance ε_l. Under this assumption, they show that if the discrete time evolution operator is contractive in the chosen norm, and if ε_l is given by (4.4), then the perturbation error accumulates in time and satisfies $\|\bar{u}^L_{\mathrm{FV}} - \bar{u}^L_{\mathrm{MR}}\| \leqslant C_2 n\varepsilon$, where $C_2 > 0$ and n denotes the number of time steps. At a fixed time $T = n\Delta t$, this gives

$$\left\|\bar{u}^L_{\mathrm{FV}} - \bar{u}^L_{\mathrm{MR}}\right\| \leqslant C_2 \frac{T}{\Delta t}\varepsilon, \quad C_2 > 0. \tag{5.3}$$

For the equations considered in [12,37], the reference (explicit) finite volume scheme is convergent provided that a CFL-type condition

$$\Delta t \leqslant \frac{\Delta x^2}{a\Delta x + b} \tag{5.4}$$

is satisfied, where the constants a and b depend on the coefficients of the equation under consideration, and Δx is the meshwidth of the finest grid, i.e. $\Delta x = \tilde{C}(\Omega)2^{-L}$, where $\tilde{C}$ depends on the dimension and shape of the computational domain. Consequently, if Δt denotes the largest time step possible, then this quantity can be expressed in terms of 2^{-L} if we consider equality in (5.4), i.e.,

$$\Delta t = \frac{[\tilde{C}(\Omega)]^2 2^{-2L}}{\tilde{C}(\Omega)2^{-L}a + b}. \tag{5.5}$$

The main idea of the adaptive multiresolution scheme is to perturb the solution given by a finite volume scheme on a uniform discretization (reference mesh) in such a way that the total error, i.e., the error between the exact solution and the adaptive solution that is projected to the reference fine mesh, is of the same order as the discretization error. For this purpose, one has to balance the discretization error and the perturbation error, which means that the discretization and perturbation errors should be of the same order as Δx, or equivalently, L varies. To derive an expression for ε from this requirement, we observe that the right-hand sides of (5.2) and (5.3) must be proportional, or equivalently,

$$\varepsilon \propto 2^{-\tilde{\alpha}L} \Delta t. \tag{5.6}$$

In light of (5.4), this yields a proportionality of the type

$$\varepsilon = \hat{C} 2^{-\tilde{\alpha}L} \frac{[\tilde{C}(\Omega)]^2 2^{-2L}}{\tilde{C}(\Omega)2^{-L}a + b}, \tag{5.7}$$

from which one may deduce a value of the reference tolerance $\varepsilon_{\mathrm{R}} = \varepsilon$, provided that the factor of proportionality $\hat{C}$ can be determined, for example from suitable experiments, as is done in [37].

5.2. Reference tolerance for degenerate reaction–diffusion systems

The derivation of the reference tolerance by the preceding method is supported by a rigorous analysis only in those cases where the order of convergence $\tilde{\alpha}$ is known, and the discrete time evolution operator is contractive in the chosen norm. These properties hold for a scalar, one-dimensional first-order conservation law and a second-order parabolic equation studied in [12] and [37], respectively. For finite volume discretizations of strongly degenerate parabolic equations, and in particular of the degenerate reaction–diffusion systems studied herein, an exact rate of convergence has not yet been derived. However, in [6,7] it is demonstrated that also for these equations, an equation of the type (5.7) may be employed to determine the reference tolerance ε_{R} if $\tilde{\alpha}$ is replaced by a rate of convergence determined by numerical experiments on a sequence of uniform grids, and the corresponding coefficient $\hat{C}$ is selected after a series of preliminary computations, with the final result that the total error of the multiresolution scheme is of the same order as that of the reference finite volume scheme.

In light of these results, we apply the same methodology to determine a reference tolerance for the problem at hand. Based on preliminary numerical experiments (obtained in a similar fashion as in [7]), for our examples we obtain the approximate value $\tilde{\alpha} = 2.18$. The global CFL condition (3.4) of the reference finite volume scheme defined for the discretization of model 2 implies that

$$\Delta t \leqslant \frac{\widetilde{\Delta x}^2}{\|f_u\|_\infty + \|f_v\|_\infty + \|g_u\|_\infty + \|g_v\|_\infty + \widetilde{\Delta x} 4(\|A'\|_\infty + d\|B'\|_\infty)}.$$

Now, if we write $\widetilde{\Delta x} = \sqrt{|\Omega|}2^{-L}$ (i.e., $\tilde{C}(\Omega) = \sqrt{|\Omega|}$), we obtain the following analogue of (5.5):

$$\Delta t = C_3 \frac{|\Omega| 2^{-2L}}{\|f_u\|_\infty + \|f_v\|_\infty + \|g_u\|_\infty + \|g_v\|_\infty + \sqrt{|\Omega|}2^{-L} 4(\|A'\|_\infty + d\|B'\|_\infty)}, \quad 0 < C_3 \leqslant 1.$$

If we require that the proportionality (5.6) holds, we arrive at the following expression for the reference tolerance ε_{R} for the numerical computations of model 2, which is a version of (5.7):

$$\varepsilon_{\mathrm{R}} = C \frac{2^{-(\tilde{\alpha}+2)L}}{|\Omega|(\|f_u\|_\infty + \|f_v\|_\infty + \|g_u\|_\infty + \|g_v\|_\infty) + |\Omega|^{3/2} 2^L 4(\|A'\|_\infty + d\|B'\|_\infty)}. \tag{5.8}$$

Analogously, for model 3, the reference tolerance may be set to

$$\varepsilon_{\mathrm{R}} = C \frac{2^{-(\tilde{\alpha}_1+2)L}}{|\Omega|(\|h_u\|_\infty + \|h_v\|_\infty + \|g'\|_\infty) + |\Omega|^{3/2} 2^L 4(\sigma + \|\chi'\|_\infty + d)}, \tag{5.9}$$

where $\tilde{\alpha}_1$ is a value of the convergence rate for model 3.

Note that all the L^∞ norms in (5.8) and (5.9) are computed numerically. To determine an acceptable value for the factor C (which, of course, depends on T, C_1, C_2 and C_3), a series of computations with different tolerances are needed in each case, prior to final computations. Essentially, we select the largest available candidate value for C such that the same order of accuracy (same slopes for the error computation) as that of the reference finite volume scheme is maintained. This procedure basically generalizes the treatment in [6] of spatially one-dimensional strongly degenerate parabolic equations. The optimal choice of C by test computations on a fine grid causes an extra cost which can be kept minor in many circumstances where the test computations need to be performed over a small fraction of the total simulated time interval only. However, our general experience is that the value of C needs to be determined to within an order of magnitude only, represents a structural property of the reaction–diffusion system under consideration only rather than the effect of particular initial and boundary data, and therefore may be employed for given equations under a variety of initial and boundary data. The additional effort in determining C will therefore be well justified in practical circumstances.

In [14] the authors prove for scalar, one-dimensional non-linear conservation laws, that the threshold error is stable in the sense that the constant C is uniformly bounded and, in particular, does not depend on the threshold value ε_{R}, the number of refinement levels L and the number of time steps n. In our case, even when a rigorous proof is still missing for the systems considered in the present work, from the previous deduction we see a similar behavior for C.

We also mention that as in previous works [6,7,12], here the reference tolerance remains fixed for all times, though it is certainly possible to recompute the reference tolerance at each time step.

6. Local time stepping

We utilize a version of the locally varying time stepping strategy advanced by Müller and Stiriba [29], and summarize here its principles. The basic idea is to enforce a local CFL condition by using the same CFL number for all levels, and the strategy consists in evolving all leaves on level l using the local time step

$$\Delta t_l = 2^{L-l}\Delta t, \quad l = L-1,\dots,0, \tag{6.1}$$

where $\Delta t = \Delta t_L$ corresponds to the time step on the finest level L. It is assumed that the fine grid is only used locally, so this strategy allows us to increase the time step for the major part of the adaptive mesh without violating the CFL stability condition.

Clearly, portions of the solution lying on different resolution levels need to be synchronized to obtain a conservative scheme. This will be achieved after 2^l time steps using Δt_l: all leaves forming the adaptive mesh are synchronized in time, so one time step with Δt_0 is equivalent to 2^L intermediate time steps with Δt_L. In order to additionally save computational effort, we update the tree structure (refinement and coarsening) only each odd intermediate time step $1,3,\dots,2^L-1$ (as suggested in [1]), and furthermore, the projection and prediction operators are performed only on the levels occupied by the leaves of the current tree, i.e., we do not update the tree structure by prediction from the root cell, but from the coarsest leaves (we refer to this as *partial grid adaptation*). For the rest of the intermediate time steps, we use the current tree structure. Notice that the updating of the tree is still done in each *global* time step. For the sake of synchronization and conservativity of the flux computation, for coarse levels (levels without leaves), we employ *the same* fluxes ($\bar{\mathcal{D}}_{ij,l}$ and $\bar{f}_{ij,l}$) computed in the previous intermediate time step, because the cell averages on these levels are the same as in the previous intermediate time step. Only for finer levels (levels containing leaves), we compute fluxes, and do so in the following way: if there is a leaf at the corresponding cell edge and at the same resolution level l, we simply perform a flux computation using the brother leaves, and the virtual leaves at the same level if necessary; and if there is a leaf at the corresponding cell edge but on a finer resolution level $l+1$ (in this case we refer to this edge as an *interface edge*), the flux will be determined as in (4.5), that is, we compute the fluxes at a level $l+1$ on the same edge, and we set the ingoing flux on the corresponding edge at level l equal to the sum of the outgoing fluxes of the son cells of level $l+1$ (for the same edge). We recall that the graded tree structure ensures that two neighboring control volumes of the adaptive mesh do not differ by more than one resolution level, which is equivalent to the satisfaction of the one-irregular rule. In order to always have at hand the computed fluxes as in (4.5), we need to perform the locally varying time stepping recursively from fine to coarse levels. If at any instance of the procedure there is a missing value, we can project the value from the sons nodes or we can predict this value from the parent nodes. For illustrative purposes, we give an example of an interior first-order flux calculation for model 2, to complete a full macro time step, by the following algorithm (we show only the flux calculation for $(i,j,l)\to(i+1,j,l)$ and $(i,j,l)\to(i,j+1,l)$ since the other fluxes are obtained analogously):

Algorithm 6.1 *(Locally varying intermediate time stepping).*

```
(1) Grid adaptation (provided the former sets of leaves and virtual leaves).
(2) do $k = 1,\dots,2^L$ (and therefore the intermediate time steps are $n+2^{-L}, n+2\cdot 2^{-L}, n+3\cdot 2^{-L},\dots,n+1$)
    (a) Synchronization:
        do $l = L,\dots,1$
            do $i = 1,\dots,|\tilde{\Lambda}|_x(l),\ j = 1,\dots,|\tilde{\Lambda}|_y(l)$
                if $1 \leqslant l \leqslant \tilde{l}_{k-1}$ then
                    if $(i,j,l)$ is a virtual leaf then
                        $\bar{F}^{n+k2^{-L}}_{(i,j,l)\to(i+1,j,l)} \leftarrow \bar{F}^{n+(k-1)2^{-L}}_{(i,j,l)\to(i+1,j,l)}$
                        $\bar{f}^{n+k2^{-L}}_{ij,l} \leftarrow \bar{f}^{n+(k-1)2^{-L}}_{ij,l}, \qquad \bar{g}^{n+k2^{-L}}_{ij,l} \leftarrow \bar{g}^{n+(k-1)2^{-L}}_{ij,l}$
                    endif
                else
                    if $(i,j,l)$ is a leaf then
                    $\bar{f}_{ij,l}^{n+k2^{-L}} \leftarrow f(\bar{u}_{ij,l}^{n+k2^{-L}}, \bar{v}_{ij,l}^{n+k2^{-L}})$,
                    $\bar{g}_{ij,l}^{n+k2^{-L}} \leftarrow g(\bar{u}_{ij,l}^{n+k2^{-L}}, \bar{v}_{ij,l}^{n+k2^{-L}})$
                    if $(i+1, j, l)$ is a leaf or $(i, j+1, l)$ is a leaf then compute fluxes by
                        $\bar{F}_{(i,j,l)\to(i+1,j,l)} \leftarrow -\frac{1}{h(l)}(A(\bar{u}_{i+1,j,l}) - A(\bar{u}_{i,j,l}))$

                        $\bar{F}_{(i,j,l)\to(i,j+1,l)} \leftarrow -\frac{1}{h(l)}(A(\bar{u}_{i,j+1,l}) - A(\bar{u}_{i,j,l}))$
                    endif
                    if $(2i+2, 2j, l+1)$, $(2i+2, 2j+1, l+1)$ are leaves (interface edges) then compute fluxes by
                        $\bar{F}_{(i,j,l)\to(i+1,j,l)} \leftarrow \bar{F}_{(2i+2,2j,l+1)\to(2i+1,2j,l+1)} + \bar{F}_{(2i+2,2j+1,l+1)\to(2i+1,2j+1,l+1)}$

                        $\bar{F}_{(i,j,l)\to(i,j+1,l)} \leftarrow \bar{F}_{(2i,2j+2,l+1)\to(2i,2j+1,l+1)} + \bar{F}_{(2i+1,2j+2,l+1)\to(2i+1,2j+1,l+1)}$
                    endif
                endif
            endif
        enddo
    enddo
(b) Time evolution:
    do $l = 1, \dots, L$, $i = 1, \dots, |\tilde{\Lambda}|_x(l)$, $j = 1, \dots, |\tilde{\Lambda}|_y(l)$
        if $1 \leqslant l \leqslant \tilde{l}_{k-1}$ then there is no evolution:
          $\bar{u}_{ij,l}^{n+(k+1)2^{-L}} \leftarrow \bar{u}_{ij,l}^{n+k2^{-L}}$,   $\bar{v}_{ij,l}^{n+(k+1)2^{-L}} \leftarrow \bar{v}_{ij,l}^{n+k2^{-L}}$
        else
            Interior marching formula only for the leaves $(i, j, l)$:
          $\bar{u}_{ij,l}^{n+(k+1)2^{-L}} \leftarrow \bar{u}_{ij,l}^{n+k2^{-L}} + \gamma \Delta t_l \bar{f}_{ij,l}^{n+k2^{-L}} + \Delta t_l \bar{\mathcal{D}}_{ij,l}(\mathcal{S}(\bar{u}_{ij,l}^{n+k2^{-L}}), h(l))$
          $\bar{v}_{ij,l}^{n+(k+1)2^{-L}} \leftarrow \bar{v}_{ij,l}^{n+k2^{-L}} + \gamma \Delta t_l \bar{g}_{ij,l}^{n+k2^{-L}} + d\Delta t_l \bar{\mathcal{D}}_{ij,l}(\mathcal{S}(\bar{v}_{ij,l}^{n+k2^{-L}}), h(l))$
        endif
    enddo
(c) Partial grid adaptation each odd intermediate time step:
    do $l = L, \dots, \tilde{l}_k + 1$
        Projection from the leaves.
    enddo
    do $l = \tilde{l}_k, \dots, L$
        Thresholding, prediction, and addition of the safety zone.
    enddo
enddo
```

Here, $\tilde{l}_k$ denotes the coarsest level containing leaves in the intermediate step k (as introduced in [29]), $h(l)$ is the mesh size on level l, and $|\tilde{\Lambda}|_z(l)$ is the size of the set formed by leaves and virtual leaves per resolution level l in the direction z. The interior marching formula is the modified marching formula (3.3) for model 2, for the intermediate time steps $k = 1, \dots, 2^L$, for the leaf in the position (i, j) at level l.

Our scheme is formulated for a first-order, explicit Euler time discretization. Generalizations for higher order schemes are given in [29] for Crank–Nicolson schemes and in [15] for Runge–Kutta schemes, respectively.

7. Algorithm implementation

Now we give a brief description of the multiresolution procedure used to solve the test problems.

Algorithm 7.1 *(Multiresolution procedure).*

(1) Initialization of parameters.
(2) Creation of the initial graded tree structure:
   (a) Create the root of the tree and compute its cell average value.
   (b) Split the cell, compute the cell average values in the sons and compute the corresponding details.
   (c) Apply the thresholding strategy for the splitting of the new sons: If $d_{i,j,l} > \varepsilon_l$ then split the son (here we use $d_l = \min\{d_l^u, d_l^v\}$).

```
    (d) Repeat this until all sons have details below the required tolerance $\varepsilon_l$.
(3) do $n = 1, \ldots, total\_time\_steps$
    (a) Determination of the leaves and virtual leaves sets.
    (b) Time evolution with global time step: Compute the discretized divergence operator for all the leaves.
    (c) Updating the tree structure:
        • Recalculate the values on the nodes and the virtual nodes by projection from the leaves. Compute the details for
          all positions $(\cdot, \cdot, l)$ for $l \geqslant \tilde{l}_k$.
          if $|\bar{d}_{i,j,l}| < \varepsilon_l$ (here we use $d_l = \max(d_l^u, d_l^v)$) in a node and in its brothers then
             mark the cell and its brothers as deletable
          endif
        • if some node and all its sons are deletable and the sons are leaves without virtual sons then
             delete sons (coarsening)
             if this node has no sons and it is not deletable and it is not at level $l = L$, then create sons (refinement).
             endif
          endif
        • Update the values in the new sons by prediction operator from the former leaves.
    enddo
(4) Output: Save meshes, leaves and cell averages.
```

Here *total_time_steps* stands for the total number of time steps needed to reach T_{final} using Δt as the maximum time step allowed by the CFL condition using the finest space step.

When using a locally varying time stepping, replace step (3)(b) by the new step

```
(3) do $n = 1, \ldots, total\_time\_steps$
    (a) Determination of the leaves and virtual leaves sets.
    (b) Time evolution with local time stepping: Compute the discretized divergence operator for all the leaves and virtual
        leaves
    (c) do $k = 1, \ldots, 2^L$ ($k$ counts intermediate time steps)
        • Compute the intermediate time steps depending on the position of the corresponding leaf as explained in Sec-
          tion 6.
        • if $k$ is odd then update 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 $|\bar{d}_{i,j,l}| < \varepsilon_l$ (here we use $d_l = \max(d_l^u, d_l^v)$) in a node and in its brothers then
               mark the cell and its brothers as deletable
            endif
          ◦ if some node and all its sons are deletable and the sons are leaves without virtual sons then
               delete sons (coarsening)
               if this node has no sons and it is not deletable and it is not at level $l = L$, then create sons (refinement).
               endif
            endif
          ◦ Update the values in the new sons by prediction operator from the former leaves.
          endif
        enddo
        (Now, after $2^L$ intermediate steps, all nodes are synchronized.)
    enddo
```

Here *total_time_steps* for the new step stands for the total time steps needed to reach T_{final}, with Δt_0 as the maximum time step allowed by the CFL condition using the coarsest space step.

Notice that with such a process, we obtain high-order approximation in the smooth regions and mesh refinement near discontinuities as a consequence of the polynomial accuracy in the multiresolution prediction operator, even if the reference finite volume scheme is low-order accurate.

Bihari and Harten [2] use the following quantity, which we call *data compression rate*,

$$\eta := \frac{N_{Lx} N_{Ly}}{N_{Lx} N_{Ly}/2^{2L} + |\mathcal{L}(\Lambda)|}, \tag{7.1}$$

to measure the possible improvement in data compression, whose feasibility in turn strongly depends on a smart implementation to navigate inside the tree, see for example [37]. Here, $N_{Lx} N_{Ly}$ is the number of cells in the finest grid, and $|\mathcal{L}(\Lambda)|$ is the size of the set of leaves. We measure the *speed-up* between the CPU time of the numerical solution obtained by the FV method on the finest uniform grid and the CPU time of the numerical solution obtained by the multiresolution method: $V = (\text{CPU time})_{\text{FV}}/(\text{CPU time})_{\text{MR}}$.

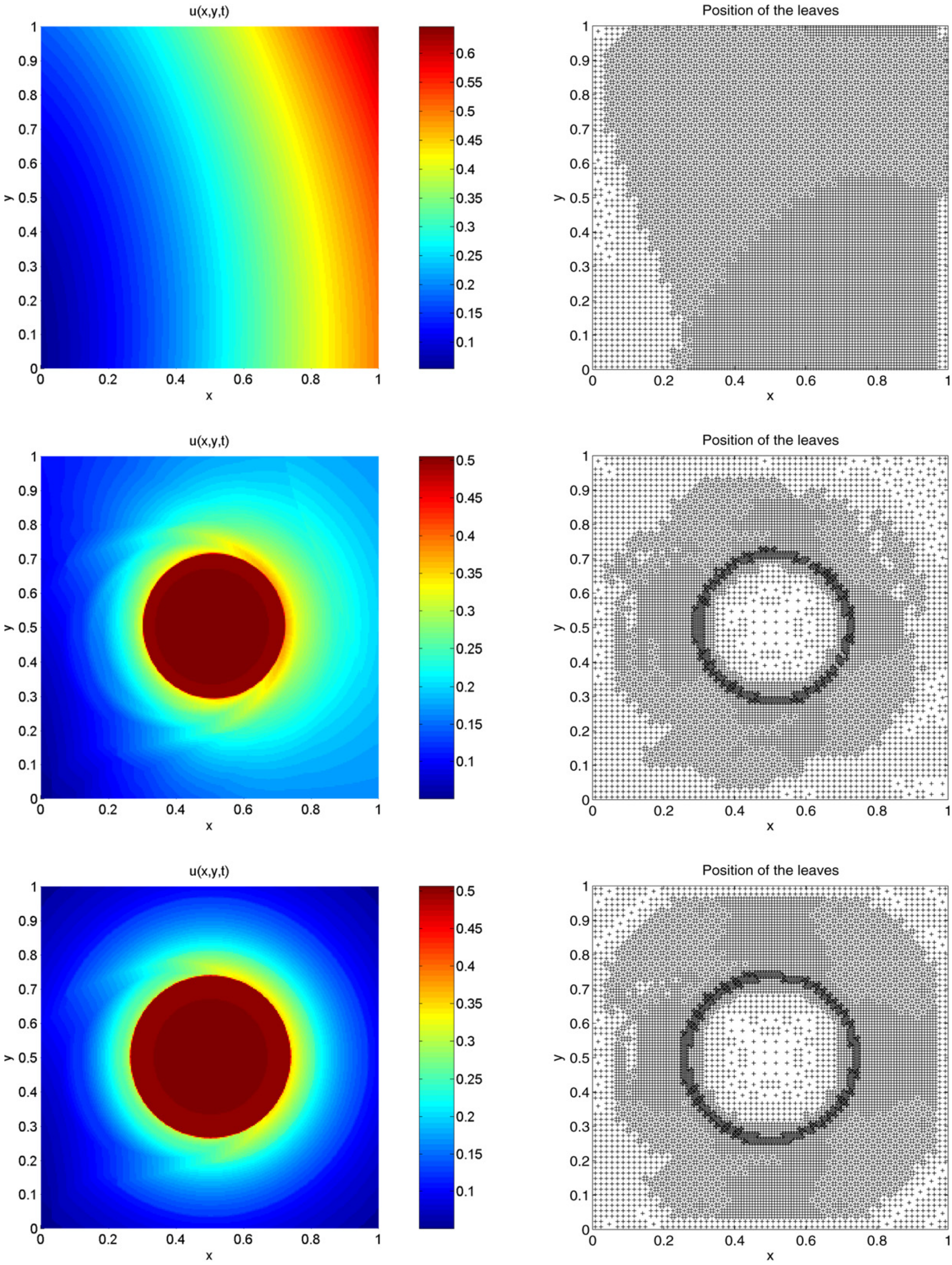

Fig. 3. Example 1 (single-species model): numerical solution (left) and leaves of the tree data structure (right) at times $t = 0$ (top), $t = 0.5$ (middle) and $t = 3$ (bottom).

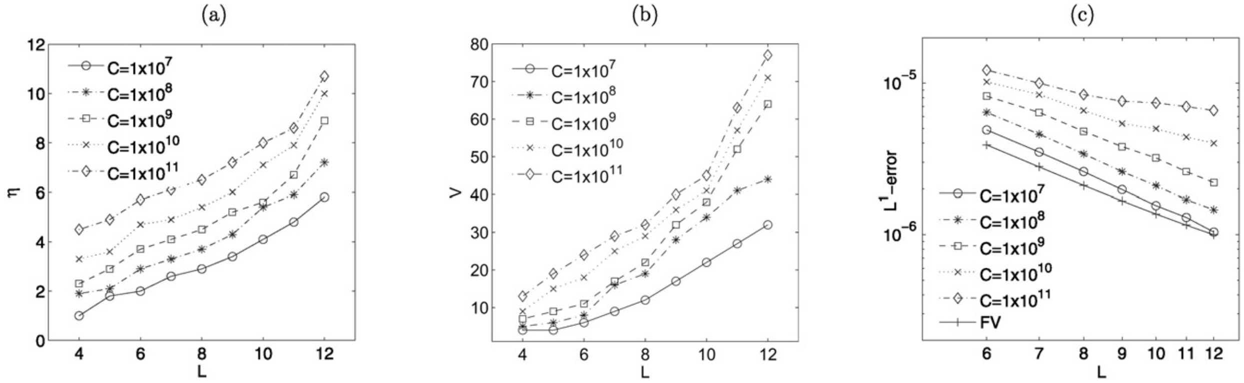

Fig. 4. Example 1 (single-species model): (a) data compression rate η, (b) speed-up factor V and (c) L^1 errors for different levels L and values of C at simulated time $t = 10$.

To measure errors between a reference solution u computed with the finite volume scheme on the finest regular grid and an approximate solution u_{MR} obtained using multiresolution, we will use L^p-errors: $e_p = \|u^n - u^n_{\mathrm{MR}}\|_p,\ p = 1, 2, \infty$, where

$$e_\infty = \max_{1\leqslant i\leqslant N_{Lx},1\leqslant j\leqslant N_{Ly}} \left|u^n_{i,j} - u^n_{\mathrm{MR}i,j,L}\right|,$$

$$e_p = \left(\frac{1}{N_{Lx}N_{Ly}} \sum_{i=1}^{N_{Lx}} \sum_{j=1}^{N_{Ly}} \left|u^n_{i,j} - u^n_{\mathrm{MR}i,j,L}\right|^p\right)^{1/p}, \quad p = 1, 2.$$

Here $u^n_{\mathrm{MR}i,j,L}$ is the value on the finest level L obtained by prediction from the corresponding leaf.

8. Numerical results

8.1. Example 1: single-species model

For this example, we consider (2.1) with a strongly degenerate diffusion term (2.3), where we choose $D := 1$ and $u_{\mathrm{c}} := 0.5$, a square domain $\Omega = [0, 1]^2$, and the function $f(u, \mathbf{x})$ given by (2.2). Fig. 3 displays the numerical solution starting from the smooth initial function

$$u_0(x, y) = 0.5\bigl(1 + \sin\bigl(1.1\bigl(x - \cos(0.7y)\bigr)\bigr)\bigr)\cos\bigl(0.5\bigl(y - \sin(1.3x)\bigr)\bigr).$$

We choose a maximal resolution level of $N_L = 256^2 = 65\,536$ control volumes on the finest grid. Fig. 4 illustrates how the factor C in (5.8) is selected in our case as the optimal value from a finite selection of test values (each provides a different value for the reference tolerance ε_{R}, see (5.8)). Fig. 4 (a) and (b) indicate that for all displayed levels, the multiresolution procedure is in every case (for different values of C) cheaper (in terms of both CPU time acceleration and memory savings) than the corresponding reference FV computation on the finest grid. The graph of Fig. 4(b) can be interpreted as follows: the better the resolution becomes, the better is the gain in CPU time. This happens independently of how far the grid is from being uniform, since the behavior is similar for different values of C. Fig. 4(c) indicates that the computations obtained using $C = 1.0 \times 10^9$ (and hence $\varepsilon_{\mathrm{R}} = 9.43 \times 10^{-4}$) are sufficiently accurate, in the sense that with these choices, we keep the same slope for the L^1-error as the FV calculations while increasing V and η. We remark that here, as in other works that use similar methods (see e.g. [12]), there actually exists a range of threshold parameters that preserve the same slope for errors with respect to reference solution, for which $C = 1.0 \times 10^9$ is an average value. Since an exact solution of the underlying problem is currently not available, we compute errors using a reference solution obtained by the finite volume scheme on a fine grid with $N_L = 2048^2 = 4\,194\,304$ control volumes.

8.2. Examples 2 and 3: interaction between two flame balls

We study (2.4) as a dimensionless model for the interaction between two flame balls of different sizes. We consider a square domain $\Omega = [-30, 30]^2$ and that the walls are sufficiently far from the flame balls so that their influence is negligible. Physical parameters characterizing the gaseous mixture and the combustion process are chosen as in [35,36]. We use the parameters $\alpha = 0.64$ and $\beta = 10$. The initial data is given by $u(x, y, 0) = u_0(r_1, r_2),\ v(x, y, 0) = v_0(r_1, r_2)$ with $r_1^2 = (x - x_1)^2 + y^2$, $r_2^2 = (x - x_2)^2 + y^2$, where

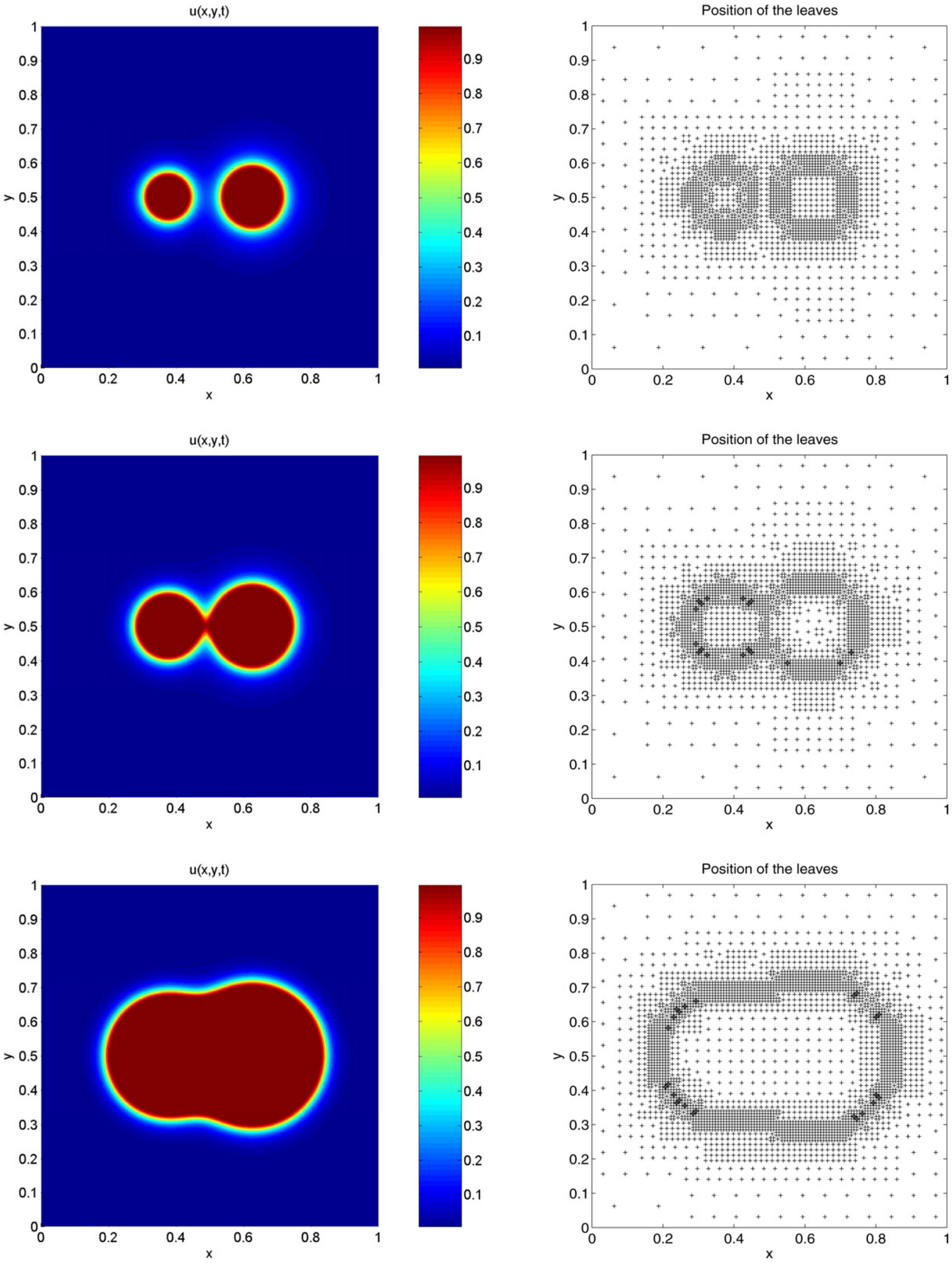

Fig. 5. Example 2 (interaction of two flame balls without radiation): numerical solution for species u (left) and leaves of the tree data structure (right) at times $t = 2$ (top), $t = 4$ (middle) and $t = 10$ (bottom).

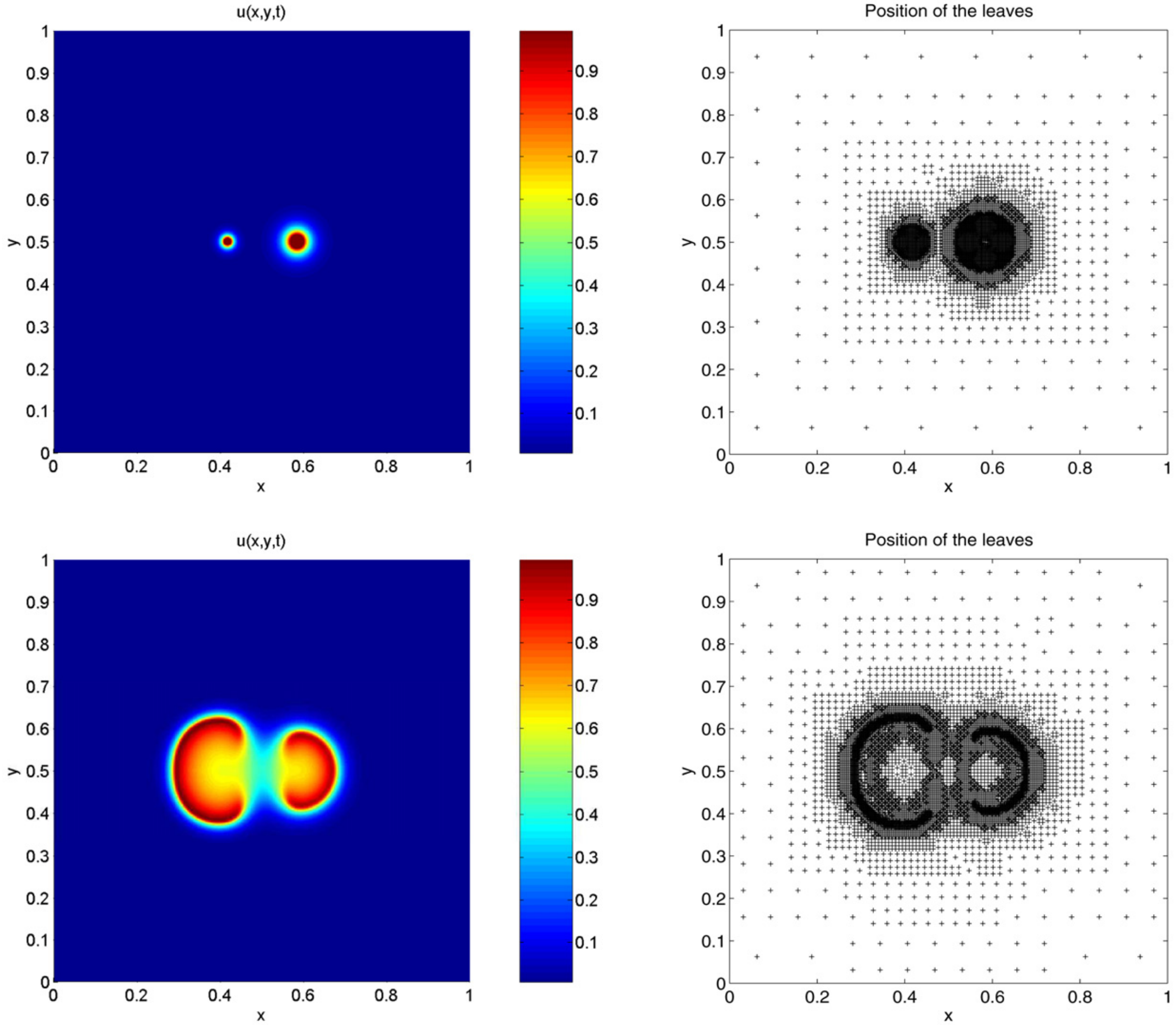

Fig. 6. Example 3 (interaction of two flame balls with radiation): numerical solution for species u (left) and leaves of the tree (right) at times $t = 0$ (top) and $t = 5$ (bottom).

$$\begin{aligned} u_0(r_1, r_2) &:= \begin{cases} 1 & \text{if } r_1 < a \text{ or } r_2 < b, \\ \max\{\exp(1 - r_1/a), \exp(1 - r_2/b)\} & \text{otherwise}, \end{cases} \\ v_0(r_1, r_2) &:= 1 - u_0(r_1, r_2). \end{aligned} \tag{8.1}$$

In example 2, we simulate the process without radiation, i.e., $\rho = 0$ and hence $S(u) = 0$. We set the Lewis number to $Le = 1$. Here $x_1 = -7.5$, $x_2 = 7.5$ and $a = 1.8$, $b = 2.5$ are the respective x-position and initial radii of the two flame balls. This choice ensures that there is no interaction between the two flame balls at $t = 0$ and that there is no extinction of the flame balls. We simulate the process until $t = 10$, and Fig. 5 shows from left to right the temperature and reaction rate configuration obtained using the fully adaptive multiresolution scheme, and the positions of the dynamic graded tree leaves, which form the corresponding adaptive mesh. The different times correspond from top to bottom to: before ($t = 2$), during ($t = 4$) and after ($t = 10$) direct interaction between the two flame balls, when the balls tend to create a new circular flame structure. We choose the following multiresolution parameters: the maximal resolution level $L = 9$ corresponding to $N_L = 512^2 = 262\,144$ control volumes in the finest grid, and the reference tolerance $\varepsilon_R = 4.94 \times 10^{-3}$.

For comparison purposes, we introduce the global chemical reaction rate

$$R(t) := \iint\limits_{\Omega} f(u, v)\, dx\, dy.$$

Errors in different norms, reaction rates, information on data compression and speed-up rate for different methods at three different times are depicted in Table 1. Due to the particular shape of solutions, which is nearly constant away from the combustion front, by using multiresolution, one can obtain very high rates of data compression, speed-up and low errors.

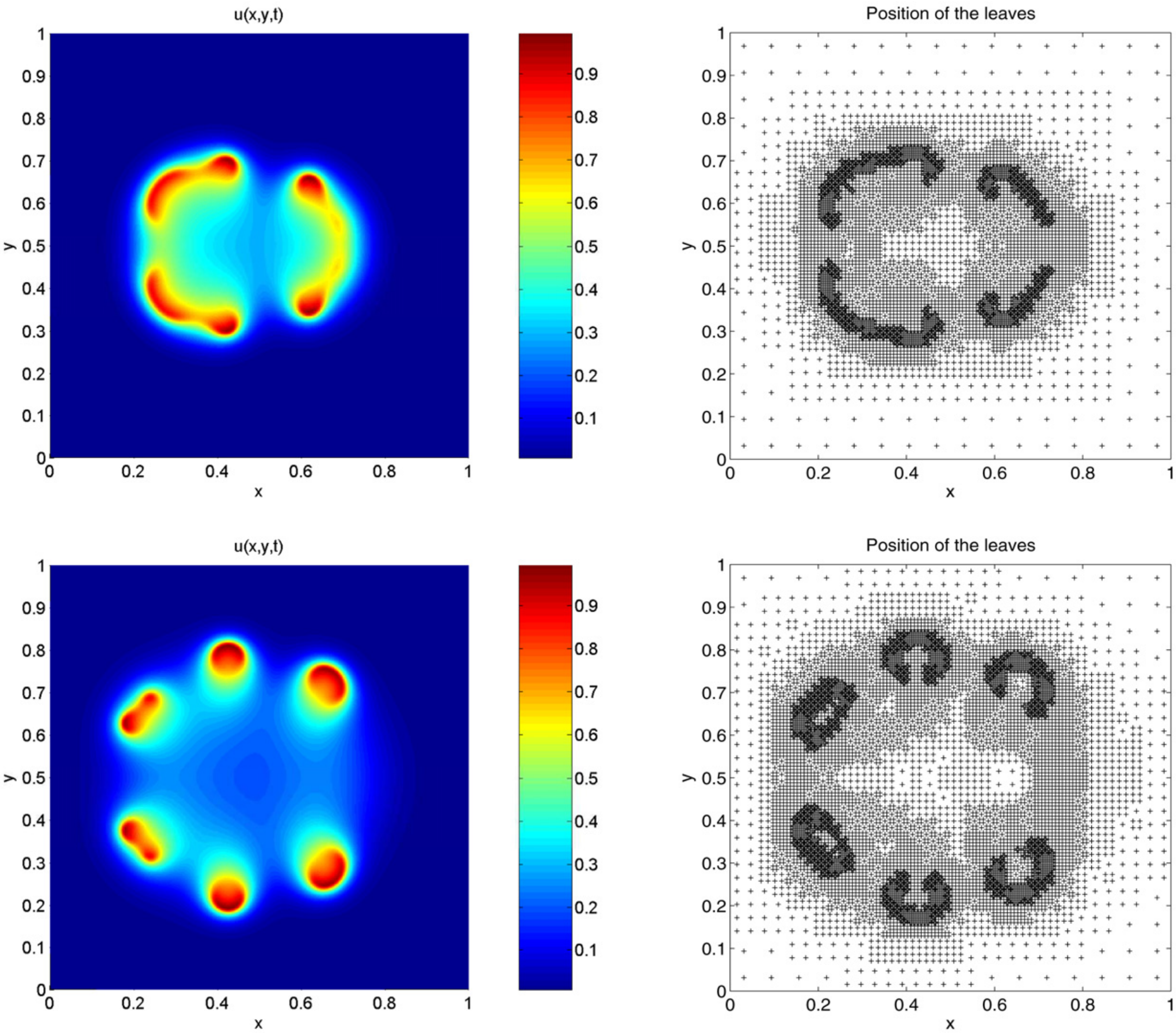

Fig. 7. Example 3 (interaction of two flame balls with radiation): numerical solution for species u and leaves of the tree (right) at times $t = 12$ (top) and $t = 20$ (bottom).

Table 1
Example 2 (interaction of two flame balls without radiation): corresponding simulated time, speed-up rate V, compression rate η, errors for species u, and total reaction rate $R(t)$.

Time	V	η	L^1-error	L^2-error	L^∞-error	Method	$R(t)$
$t = 2.0$	12.47	138.2613	5.41×10^{-3}	5.77×10^{-3}	2.46×10^{-2}	MR	56.7230
						FV	56.0078
$t = 4.0$	20.56	113.4331	6.39×10^{-3}	8.42×10^{-4}	3.02×10^{-2}	MR	80.0374
						FV	79.5247
$t = 10.0$	34.42	83.9129	5.20×10^{-3}	4.90×10^{-3}	5.49×10^{-2}	MR	98.9210
						FV	98.7942

In example 3, we simulate the case with radiation, i.e. we use (2.6) and (2.7), where $\rho = 0.05$ and $Le = 0.3$. Now $x_1 = -5$, $x_2 = 5$ and $a = 0.5$, $b = 1$ are the respective x-position and initial radii of the two flame balls. We simulate the process until $t = 10$ and Figs. 6 and 7 show the scenario for this case. First, the balls grow spherically, tend to create a new flame structure, and then their fronts tend to extinguish when they touch each other, while the radiation effect causes the entire flame front to split. Here the maximal resolution level is set to $L = 9$ corresponding to $N_L = 512^2 = 262\,144$ control volumes in the finest grid, and the reference tolerance is $\varepsilon_R = 7.43 \times 10^{-3}$.

Notice that the multiresolution procedure automatically detects the higher gradient regions and uses this information to adaptively represent the solution by the refinement and coarsening of the mesh, i.e., by the adaptive addition and removal of control volumes on these areas.

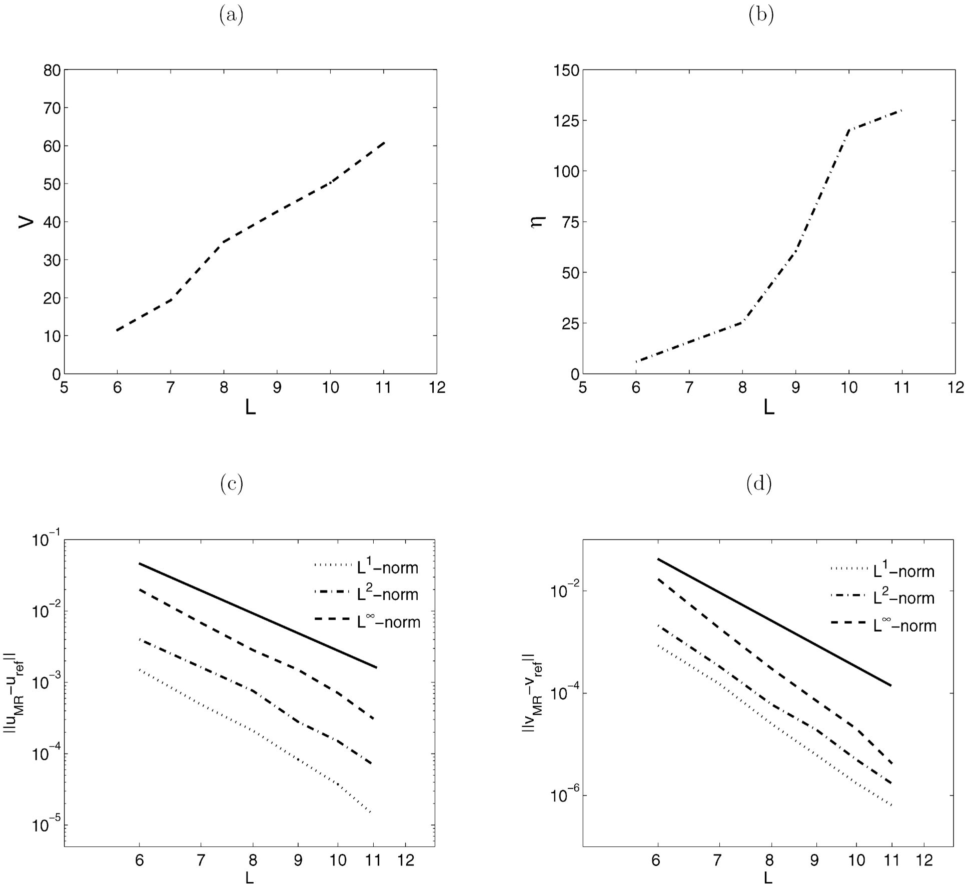

Fig. 8. Example 2 (interaction of two flame balls without radiation): (a) speed-up rate V, (b) data compression rate η, for different levels, at time $t = 4.0$; (c) errors $\|\bar{u}_{\rm MR} - \bar{u}_{\rm ref}\|_1$, $\|\bar{u}_{\rm MR} - \bar{u}_{\rm ref}\|_2$, $\|\bar{u}_{\rm MR} - \bar{u}_{\rm ref}\|_\infty$ and (d) $\|\bar{v}_{\rm MR} - \bar{v}_{\rm ref}\|_1$, $\|\bar{v}_{\rm MR} - \bar{v}_{\rm ref}\|_2$ and $\|\bar{v}_{\rm MR} - \bar{v}_{\rm ref}\|_\infty$ respectively for different levels L, at time $t = 4$. The thick solid line in plots (c) and (d) is the theoretical slope of a convergence rate 2.

The L^1, L^2 and L^∞ errors between the numerical solution obtained by our multiresolution scheme for different multiresolution levels L and the reference solution (obtained by finite volume approximation in a uniform fine grid with $2^{2\times 14} = 2^{28}$ control volumes) for example 2 are depicted in Fig. 8 (c) and (d). The slopes indicate a rate of convergence slightly better than two.

For example 3, we apply the locally varying time stepping (LTS) strategy detailed in Section 6. We choose the maximum CFL number allowed by (3.4), which is $\mathsf{CFL}_0 = 1$ for the coarsest level. For the remaining levels we use $\mathsf{CFL}_l = 2^l \mathsf{CFL}_0$, which means that we perform each macro time step with $\Delta t = \Delta t_0 = 2^L \Delta t_L$ as given by (6.1).

In Fig. 9 we compare speed-up, data compression rate and total reaction rate for the finite volume reference scheme, the multiresolution scheme with global time step, and the multiresolution method with level-dependent time stepping. Notice that with LTS, the speed-up rate is approximately doubled for all times, while the compression rate and the total reaction rate remain of the same order as the multiresolution computation with global time step.

8.3. Examples 4 and 5: a Turing model of pattern formation

We select the parameters $a = -0.5$, $b = 1.9$, $d = 4.8$ and $\gamma = 210$. According to the discussion of Section 2.2, these parameters allow diffusion-driven instabilities to evolve. The initial concentration distribution is a normally distributed random perturbation around the stationary state (u^0, v^0) for the non-degenerate case, with a variance lower than the amplitude of the final patterns, see Fig. 10. For the case of non-degenerate diffusion (example 4), we use $A(u)$ and $B(u)$ as given by (2.9). For these parameters, the steady state is $(u^0 = 1.4, v^0 = 0.96939)$.

In example 4 we choose a maximal resolution level of $N_L = 256^2 = 65\,536$ control volumes in the finest grid and the reference tolerance $\varepsilon_{\rm R} = 2.6 \times 10^{-3}$. The time step is the maximum allowed by the CFL condition (3.4). Table 2 summarizes the speed-up rate, compression rate and errors in different norms between the numerical solution by multiresolution and the fine-mesh finite volume reference solution for different times. We depict errors between our multiresolution scheme and a reference solution obtained by the finite volume scheme with $N_L = 1024^2 = 1\,048\,576$ control volumes in the finest grid, for different multiresolution levels L in Fig. 12 (c) and (d). In this case, the slopes equally indicate a rate of convergence

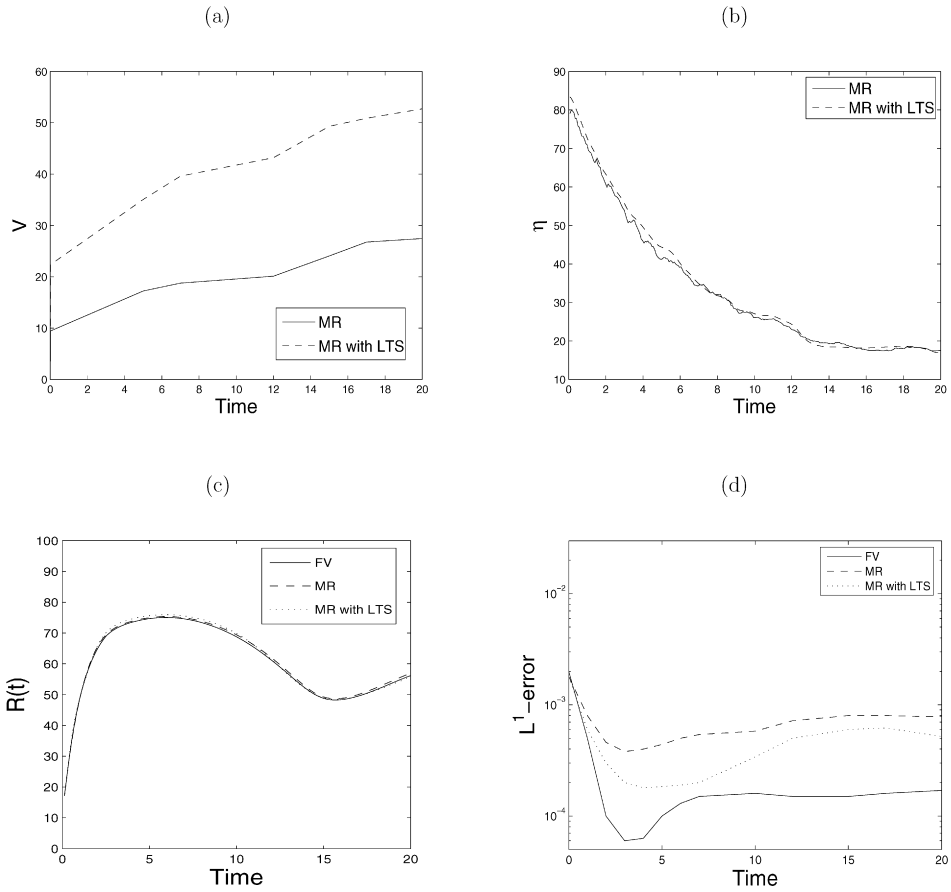

Fig. 9. Example 3 (interaction of two flame balls with radiation): time evolution of speed-up, data compression, and total reaction rates; and L^1-errors for different methods. $L = 9$ multiresolution levels and reference tolerance $\varepsilon_{\mathrm{R}} = 7.43 \times 10^{-3}$.

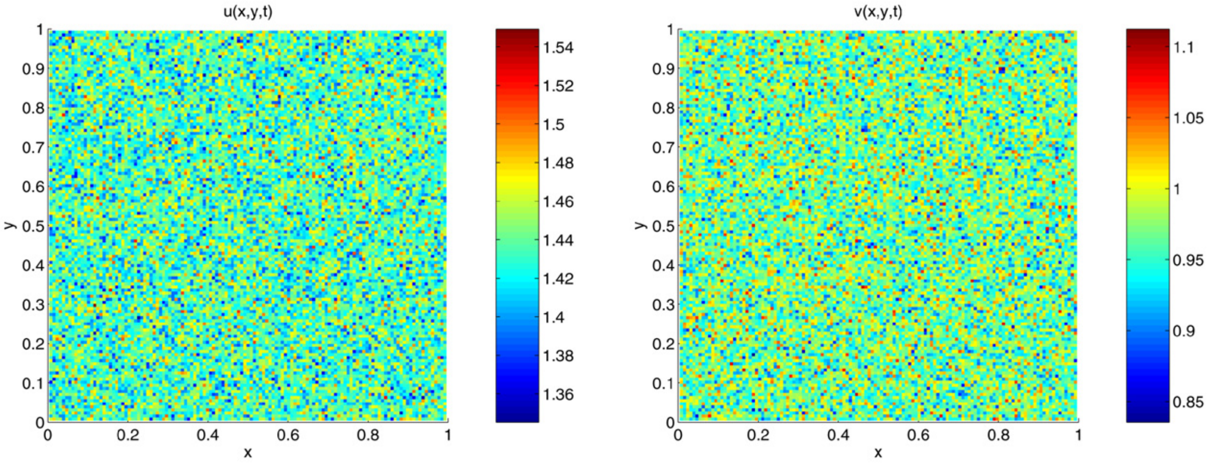

Fig. 10. Examples 4 and 5 (Turing pattern-formation): initial data $u_0(x, y)$ (left) and $v_0(x, y)$ (right).

slightly larger than two. Concerning the computation of errors, in examples 4, 5 and 6 the system is evolved until the “random noise”, which is imposed as an initial condition on the finest grid, has been smoothed sufficiently; then, this solution is projected on coarser levels to obtain auxiliary initial conditions for all the needed levels.

For example 5, we use the degenerate diffusion coefficients (2.11) with $u_{\mathrm{c}} = 1.2$ and $v_{\mathrm{c}} = 0.7$, and employ again the kinetics (2.8), but this time we choose the parameters $a = -0.5$, $b = 1.9$, $d = 4.8$ and $\gamma = 395$. We select a maximal resolution level of $N_L = 256^2 = 65\,536$ control volumes in the finest grid, with a reference tolerance given by $\varepsilon_{\mathrm{R}} = 3.59 \times$

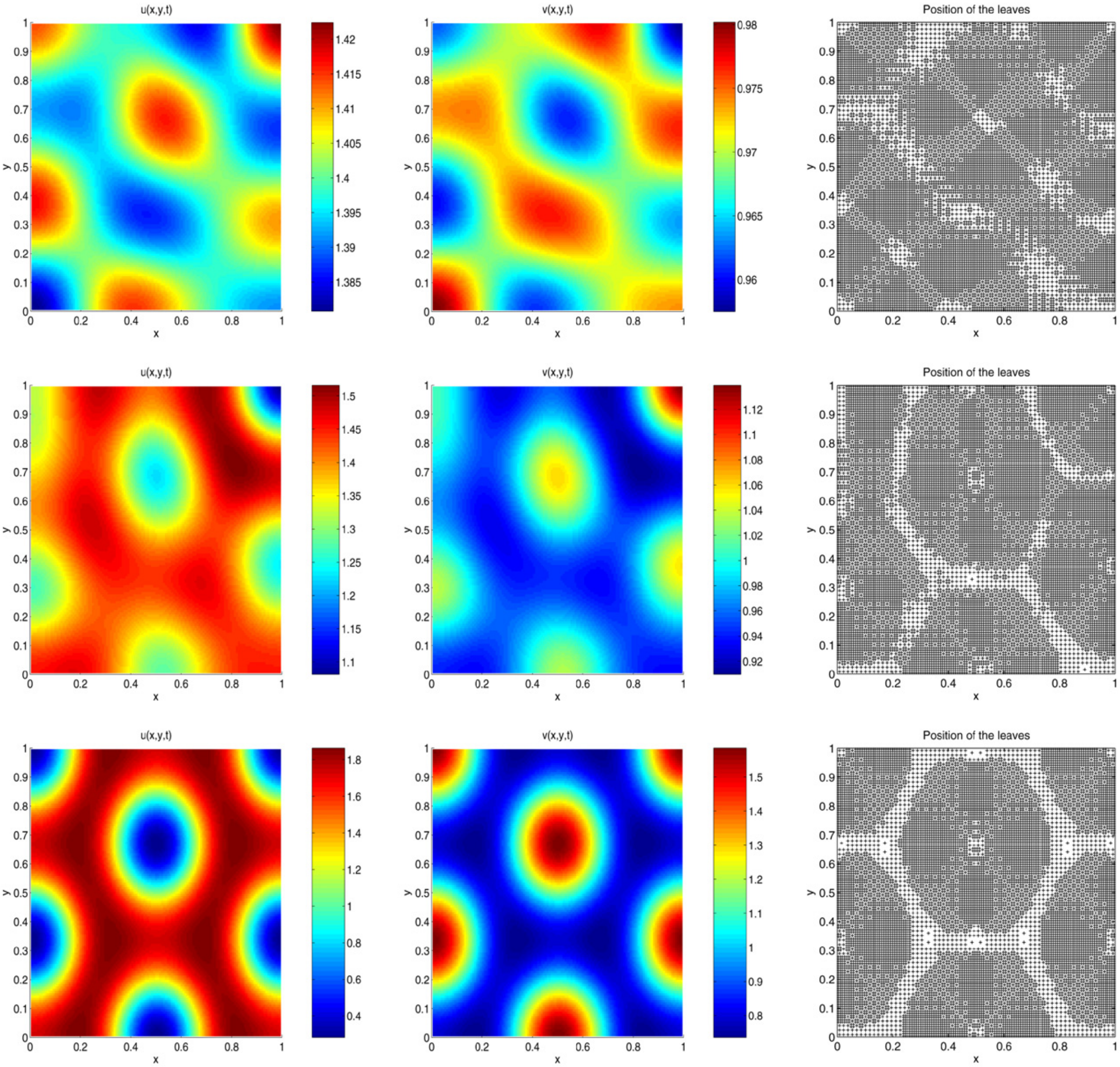

Fig. 11. Example 4 (model 2 with non-degenerate diffusion): numerical solution for species u (left) and v (right), and leaves of the tree data structure at times $t = 0.05$ (top), $t = 0.25$ (middle) and $t = 1.5$ (bottom).

Table 2
Example 4 (model 2 with non-degenerate diffusion): corresponding simulated time, CPU ratio V, compression rate η and componentwise errors.

Time	V	η	Species	L^1-error	L^2-error	L^∞-error
$t = 0.05$	7.16	11.3783	u	6.81×10^{-4}	4.76×10^{-5}	3.46×10^{-3}
			v	4.09×10^{-4}	3.92×10^{-4}	5.38×10^{-4}
$t = 0.25$	9.29	11.9756	u	8.37×10^{-4}	6.94×10^{-5}	9.93×10^{-3}
			v	4.22×10^{-4}	5.43×10^{-4}	8.48×10^{-4}
$t = 1.50$	11.87	14.4739	u	9.26×10^{-4}	2.71×10^{-4}	2.44×10^{-2}
			v	4.30×10^{-4}	9.77×10^{-5}	8.39×10^{-3}

10^{-4}. From Table 3 we see that the multiresolution algorithm allows significant acceleration and data compression rate are significantly increased by the multiresolution algorithm with very good accuracy. Fig. 13 indicates that due to the degeneracy of the diffusion given by (2.11), and in contrast to example 4, species u exhibits patterns with steeper gradients, and especially at $t = 0.25$ and $t = 1.5$, singularities appear.

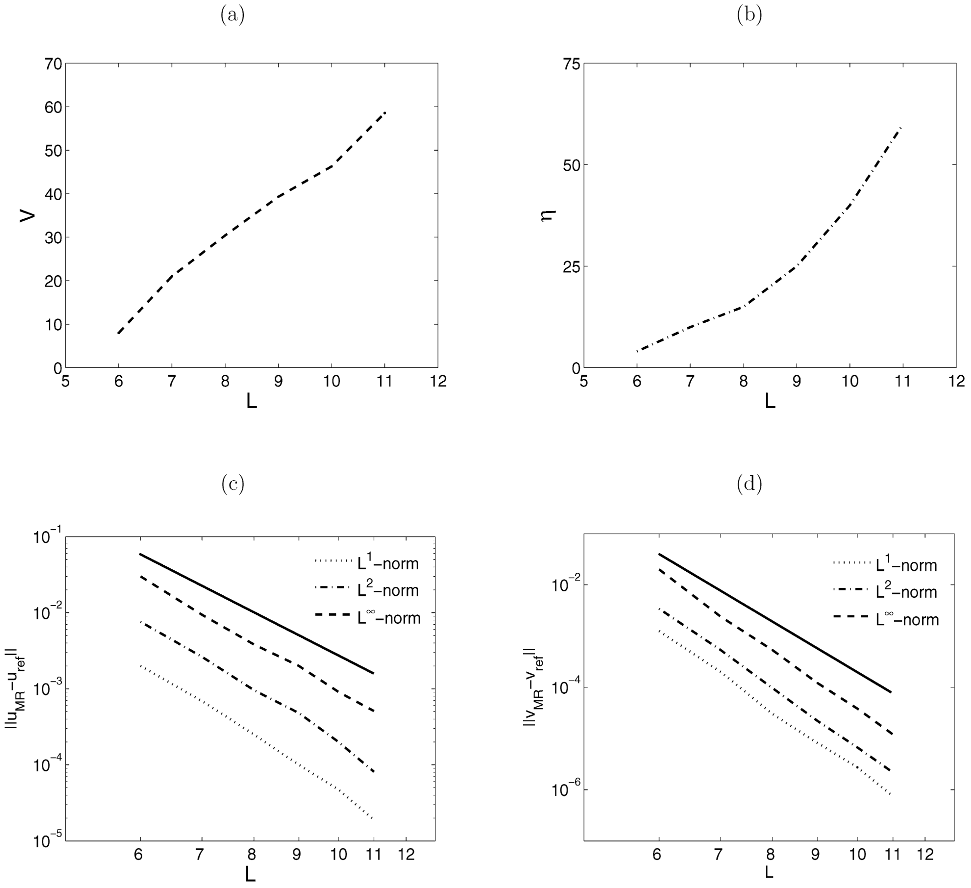

Fig. 12. Example 4 (model 2 with non-degenerate diffusion): (a) CPU acceleration V, (b) data compression rate η, for different levels, at time $t=0.75$ s, (c) errors $\|\bar{u}_{\mathrm{MR}}-\bar{u}_{\mathrm{ref}}\|_1$, $\|\bar{u}_{\mathrm{MR}}-\bar{u}_{\mathrm{ref}}\|_2$, $\|\bar{u}_{\mathrm{MR}}-\bar{u}_{\mathrm{ref}}\|_\infty$ and (d) $\|\bar{v}_{\mathrm{MR}}-\bar{v}_{\mathrm{ref}}\|_1$, $\|\bar{v}_{\mathrm{MR}}-\bar{v}_{\mathrm{ref}}\|_2$ and $\|\bar{v}_{\mathrm{MR}}-\bar{v}_{\mathrm{ref}}\|_\infty$ respectively for different levels L, at time $t=0.75$. The thick solid line in plots (c) and (d) is the theoretical slope of a convergence rate 2.

Table 3
Example 5 (model 2 with degenerate diffusion): corresponding simulated time, speed-up rate V, compression rate η and componentwise errors.

Time	V	η	Species	L^1-error	L^2-error	L^∞-error
$t=0.10$	6.32	12.5438	u	6.31×10^{-4}	5.82×10^{-4}	2.72×10^{-3}
			v	4.98×10^{-4}	5.37×10^{-4}	9.46×10^{-4}
$t=0.25$	9.79	10.3457	u	6.12×10^{-4}	2.46×10^{-5}	3.03×10^{-3}
			v	3.91×10^{-4}	9.22×10^{-4}	9.92×10^{-4}
$t=1.50$	11.60	10.1984	u	3.42×10^{-4}	7.34×10^{-4}	3.40×10^{-3}
			v	2.63×10^{-4}	4.98×10^{-4}	2.81×10^{-3}

8.4. Example 6: chemotaxis-growth system

For example 6, in (2.12) we consider a square domain $\Omega=[0,16]^2$ and fix the parameters $\sigma=0.0625$ and $d=1$. The function $h(u,v)$ is given by (2.15) with $\alpha=1$ and $\beta=32$. The growth function $g(u)$ for the species u is given by (2.14), and the chemotactical sensitivity is given by (2.13). This configuration corresponds to the model of chemotaxis and growth presented in [25], which is further analyzed in [16]. Similarly to [16], the initial datum is $(u_0,v_0)=(1+\varepsilon(\mathbf{x}),1/32)$, where $\varepsilon(\mathbf{x})$ is a particular smooth perturbation which goes to zero near $(8,8)$. We simulate the process until the solution reaches inhomogeneous stationary states, and we present three cases corresponding to different values of ν, which is responsible for the complexity of the spatial patterns. For example, for $\nu=7$ Fig. 14 (middle) shows labyrinth-shaped patterns and for $\nu=10$ (bottom), single filaments and spots. The corresponding adaptive meshes were generated with $N_L=512^2=262\,144$ control volumes in the finest grid, with $\varepsilon_{\mathrm{R}}=8.43\times10^{-4}$. For all these cases we implement locally varying time stepping, so we will choose the maximum CFL number allowed by (3.6), $\mathsf{CFL}_0=1$ for the coarsest level and $\mathsf{CFL}_l=2^l\mathsf{CFL}_0$ for finer levels. From Fig. 15 we can observe that if we incorporate the local time stepping strategy, a substantial gain (a factor slightly lower than 2, which is consistent with the results of [23]) is obtained in speed-up rate when comparing with a multiresolution calculation using global time stepping. The errors are computed from a reference solution in a grid with

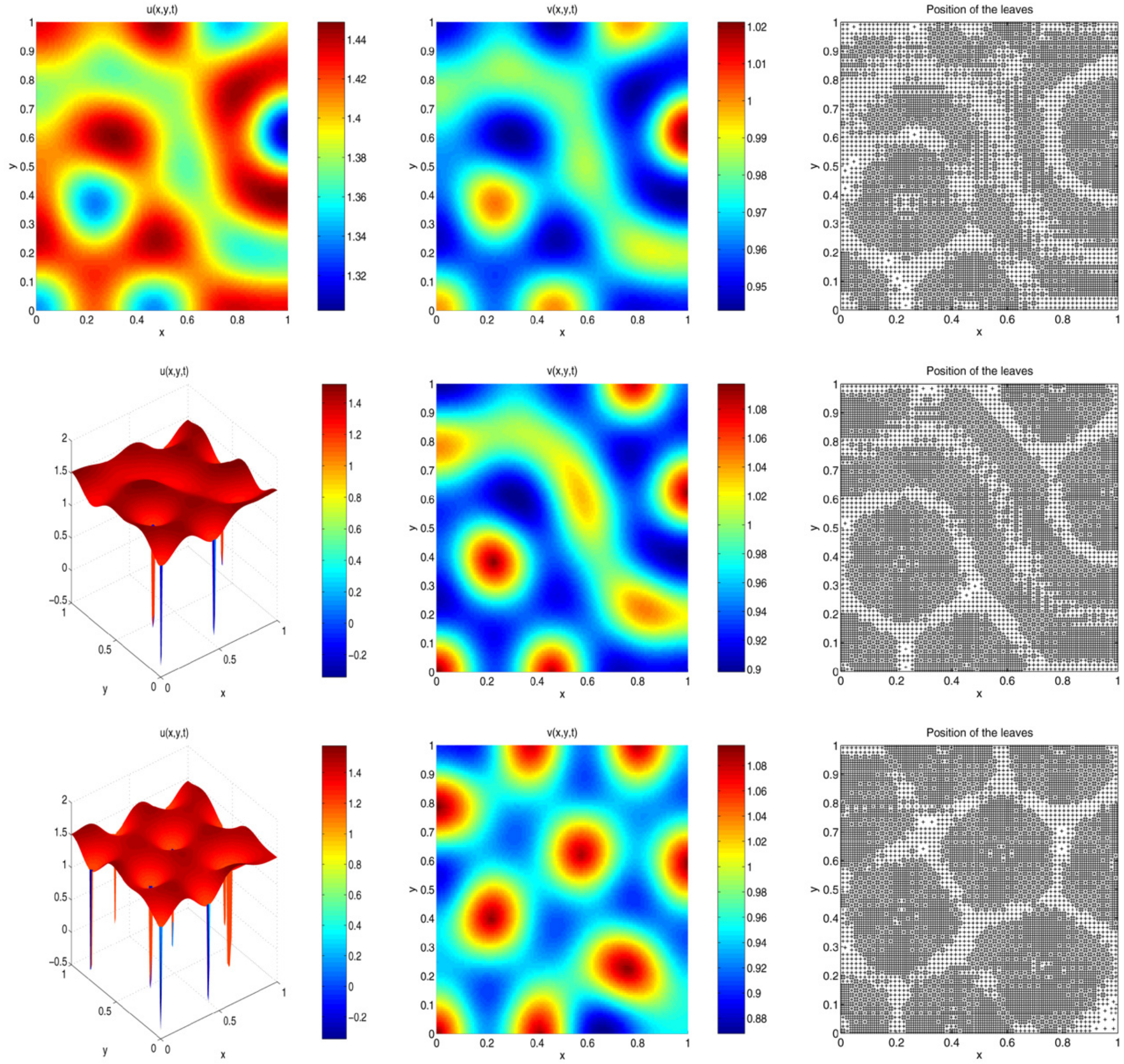

Fig. 13. Example 5 (model 2 with degenerate diffusion): numerical solution for species u (left) and v (middle), and leaves of the tree data structure (right) at times $t = 0.1$ (top), $t = 0.25$ (middle) and $t = 1.5$ (bottom).

$N_L = 2048^2 = 4\,194\,304$ control volumes. We conclude that the errors obtained by using this strategy are nearly the same as those produced by global time stepping.

The compression rate η for both methods is lower than in the previous examples, which could be explained by the complexity and density of the spatial patterns in this particular example.

9. Conclusions

This paper describes an adaptive multiresolution scheme combined with a locally varying time stepping used to approximate solutions of a class of two-dimensional reaction–diffusion systems in Cartesian geometry. Several numerical examples show that the adaptive multiresolution mechanism with a suitable choice of the threshold value represents a gain in CPU time while the errors are kept of the same order as the reference finite volume method. In examples 3 and 6, we also see that the local time stepping strategy is responsible for a gain in CPU time speed-up of a factor of about 2. Also, the errors between the solution using local time stepping and a reference solution are of the same order as the corresponding errors of the solution obtained by the adaptive multiresolution with global time stepping.

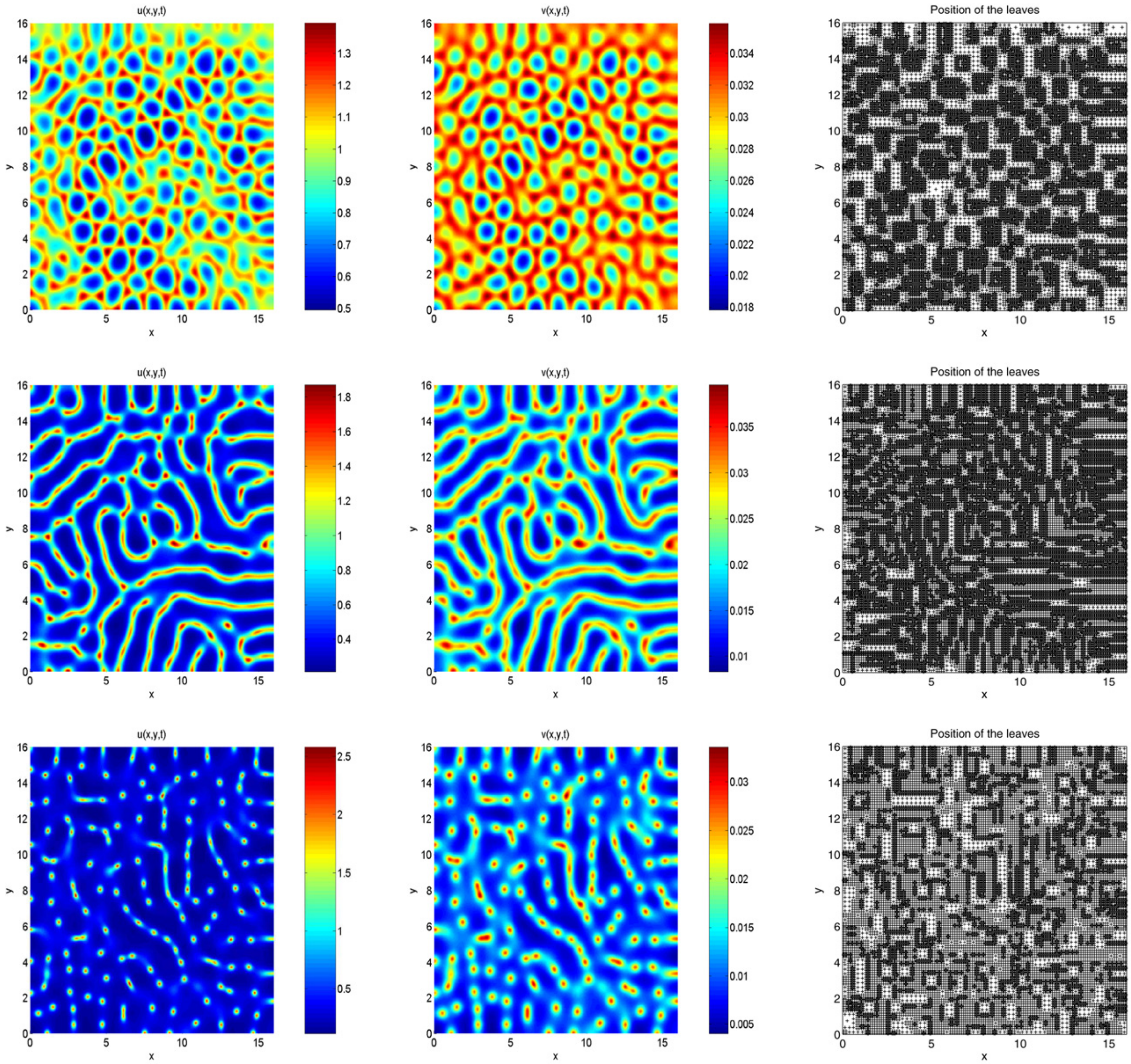

Fig. 14. Example 6 (chemotaxis with growth): numerical solution for species u (left) and v (middle), and leaves of the tree data structure (right) for $\nu = 6.04$ (top), $\nu = 7$ (middle) and $\nu = 10$ (bottom).

The motivation to employ explicit schemes only is the following. Even though implicit methods allow larger time steps, we need to iteratively solve a non-linear system in each time step, using e.g. Newton–Raphson method. The number of iterations is usually controlled by measuring the residual error, and cannot be controlled a priori. Thus, it appears difficult to assess the true benefits of a time-stepping strategy if the basic time discretization is an implicit one.

On the other hand, for the Turing-type pattern formation problem, patterns appear when one eigenvalue goes from negative to positive. At steady state (when the pattern is visible) all the eigenvalues again have negative real part. Thus to converge to steady state once the domain of attraction of the pattern is reached, implicit methods offer significant advantages since they can use larger and larger time steps.

We remark that for hyperbolic problems, the incorporation of an implicit time discretization to the MR-LTS strategy can possibly form a substantial improvement in the speed-up rate, as presented in [29].

Acknowledgements

MB acknowledges support by Fondecyt project 1070682, RB acknowledges support by Fondecyt project 1050728, Fondap in Applied Mathematics, project 15000001, and BASAL project CMM, Universidad de Chile and Centro de Investigación en

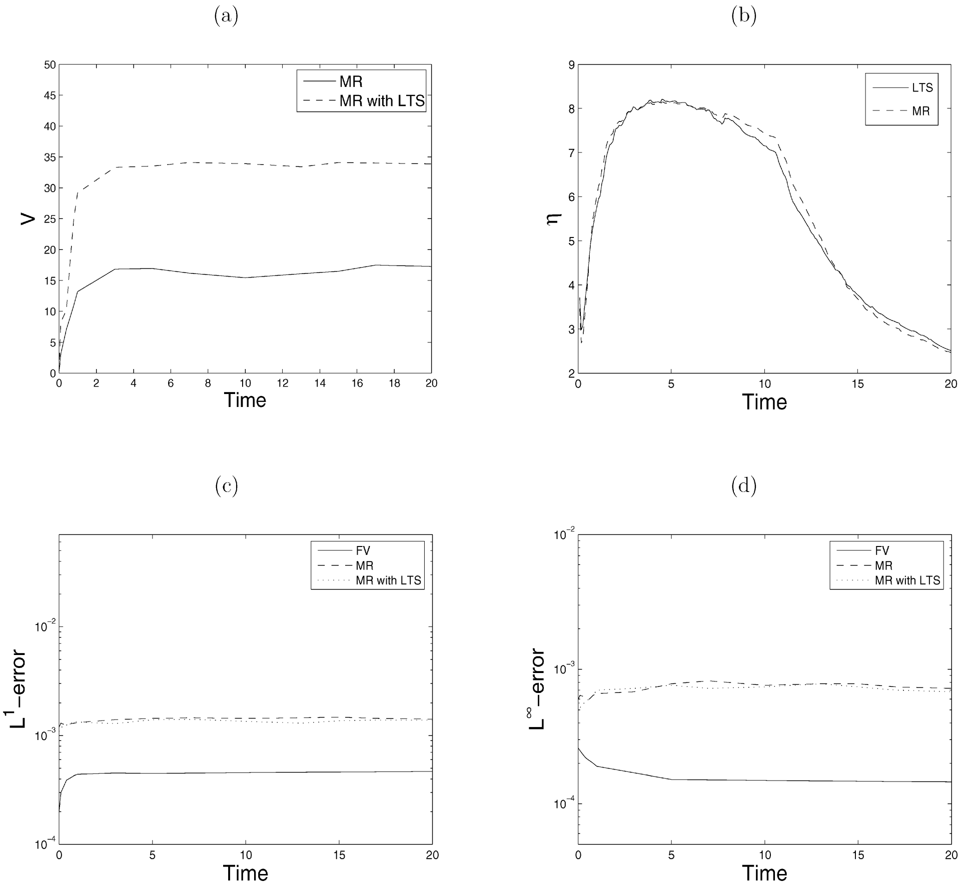

Fig. 15. Example 6 (chemotaxis with growth, $\nu = 7$): time evolution for speed-up rate V, data compression rate η and errors for different methods: multiresolution scheme with global time step, and multiresolution with locally varying time step.

Ingeniería Matemática (CI^{2}MA), Universidad de Concepción; RR acknowledges support by Conicyt Fellowship and Mecesup projects UCO0406 and UCO9907, and KS acknowledges support by the Agence Nationale de la Recherche, project M2TFP. This work was partially done while RR visited the Laboratoire de Modélisation et Simulation Numérique en Mécanique du CNRS and the Centre de Mathématiques et d'Informatique at the Université de Provence in Marseille, France.

References

[1] M.J. Berger, J. Oliger, Adaptive mesh refinement for hyperbolic partial differential equations, J. Comput. Phys. 53 (1984) 482–512.

[2] B.L. Bihari, A. Harten, Multiresolution schemes for the numerical solution of 2-D conservation laws I, SIAM J. Sci. Comput. 18 (1997) 315–354.

[3] N.F. Britton, Reaction–Diffusion Equations and Their Application to Biology, Academic Press, NY, 1986.

[4] R. Bürger, A. Kozakevicius, Adaptive multiresolution WENO schemes for multi-species kinematic flow models, J. Comput. Phys. 224 (2007) 1190–1222.

[5] R. Bürger, A. Kozakevicius, M. Sepúlveda, Multiresolution schemes for degenerate parabolic equations in one space dimension, Numer. Methods Partial Differential Equations 23 (2007) 706–730.

[6] R. Bürger, R. Ruiz, K. Schneider, M. Sepúlveda, Fully adaptive multiresolution schemes for strongly degenerate parabolic equations with discontinuous flux, J. Engrg. Math. 60 (2008) 365–385.

[7] R. Bürger, R. Ruiz, K. Schneider, M. Sepúlveda, Fully adaptive multiresolution schemes for strongly degenerate parabolic equations in one space dimension, M2AN Math. Model. Numer. Anal. 42 (2008) 535–563.

[8] A. Chertock, A. Kurganov, A second-order positivity preserving central-upwind scheme for chemotaxis and haptotaxis models, Numer. Math. 111 (2008) 169–205.

[9] G. Chiavassa, R. Donat, Point value multiresolution for 2D compressible flows, SIAM J. Sci. Comput. 23 (2001) 805–823.

[10] G. Chiavassa, R. Donat, S. Müller, Multiresolution-based adaptive schemes for hyperbolic conservation laws, in: T. Plewa, T. Linde, V.G. Weiss (Eds.), Adaptive Mesh Refinement – Theory and Applications, Springer-Verlag, Berlin, 2003, pp. 137–159.

[11] C. Chiu, J.L. Yu, An optimal adaptive time-stepping scheme for solving reaction–diffusion–chemotaxis systems, Math. Biosci. Eng. 4 (2007) 187–203.

[12] A. Cohen, S. Kaber, S. Müller, M. Postel, Fully adaptive multiresolution finite volume schemes for conservation laws, Math. Comp. 72 (2003) 183–225.

[13] W. Dahmen, B. Gottschlich-Müller, S. Müller, Multiresolution schemes for conservation laws, Numer. Math. 88 (2001) 399–443.

[14] M. Domingues, O. Roussel, K. Schneider, On space-time adaptive schemes for the numerical solution of partial differential equations, ESAIM: Proc. 16 (2007) 181–194.

[15] M. Domingues, S. Gomes, O. Roussel, K. Schneider, An adaptive multiresolution scheme with local time-stepping for evolutionary PDEs, J. Comput. Phys. 227 (2008) 3758–3780.

[16] M. Efendiev, M. Kläre, R. Lasser, Dimension estimate of the exponential attractor for the chemotaxis-growth system, Math. Meth. Appl. Sci. 30 (2007) 579–594.

[17] 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.

[18] A. Harten, Multiresolution algorithms for the numerical solution of hyperbolic conservation laws, Comm. Pure Appl. Math. 48 (1995) 1305–1342.

[19] H. Holden, K.H. Karlsen, N.H. Risebro, On uniqueness and existence of entropy solutions of weakly coupled systems of nonlinear degenerate parabolic equations, Electron. J. Differential Equations 46 (2003) 1–31.

[20] D. Horstmann, From 1970 until present: the Keller–Segel model in chemotaxis and its consequences II, Jahresber. Deutsch. Math.-Verein. 106 (2004) 51–69.

[21] E.F. Keller, L.A. Segel, Model for chemotaxis, J. Theor. Biol. 30 (1971) 225–234.

[22] S.N. Kružkov, First order quasi-linear equations in several independent variables, Math. USSR Sb. 10 (1970) 217–243.

[23] P. Lamby, S. Müller, Y. Stiriba, Solution of shallow water equations using fully adaptive multiscale schemes, Int. J. Numer. Meth. Fluids 49 (2005) 417–437.

[24] C. Mascia, A. Porretta, A. Terracina, Nonhomogeneous Dirichlet problems for degenerate parabolic-hyperbolic equations, Arch. Ration. Mech. Anal. 163 (2002) 87–124.

[25] M. Mimura, T. Tsujikawa, Aggregating pattern dynamics in a chemotaxis model including growth, Physica A 230 (1996) 499–543.

[26] P.K. Moore, An adaptive finite element method for parabolic differential systems: some algorithmic considerations in solving in three space dimensions, SIAM J. Sci. Comput. 21 (2000) 1567–1586.

[27] P.K. Moore, Solving regularly and singularly perturbed reaction–diffusion equations in three space dimensions, J. Comput. Phys. 224 (2007) 601–615.

[28] S. Müller, Adaptive Multiscale Schemes for Conservation Laws, Springer-Verlag, Berlin, 2003.

[29] S. Müller, Y. Stiriba, Fully adaptive multiscale schemes for conservation laws employing locally varying time stepping, J. Sci. Comput. 30 (2007) 493–531.

[30] J.D. Murray, Parameter space for Turing instability in reaction diffusion mechanisms: a comparison of models, J. Theor. Biol. 98 (1982) 143–163.

[31] J.D. Murray, Mathematical Biology II: Spatial Models and Biomedical Applications, third ed., Springer-Verlag, New York, 2003.

[32] K. Osaki, T. Tsujikawa, A. Yagi, M. Mimura, Exponential attractor for a chemotaxis-growth system of equations, Nonlinear Anal. 51 (2002) 119–144.

[33] S. Osher, R. Sanders, Numerical approximations to nonlinear conservation laws with locally varying time and space grids, Math. Comp. 41 (1983) 321–336.

[34] B. Perthame, PDE models for chemotactic movements: parabolic, hyperbolic and kinetic, Appl. Math. 6 (2004) 539–564.

[35] O. Roussel, K. Schneider, An adaptive multiresolution method for combustion problems: application to flame ball-vortex interaction, Computers & Fluids 34 (2005) 817–831.

[36] O. Roussel, K. Schneider, Numerical study of thermodiffusive flame structures interacting with adiabatic walls using an adaptive multiresolution scheme, Combust. Theory Modelling 10 (2006) 273–288.

[37] O. Roussel, K. Schneider, A. Tsigulin, H. Bockhorn, A conservative fully adaptive multiresolution algorithm for parabolic PDEs, J. Comput. Phys. 188 (2003) 493–523.

[38] J. Schnakenberg, Simple chemical reaction systems with limit cycle behavior, J. Theor. Biol. 81 (1979) 389–400.

[39] B. Sjögreen, Numerical experiments with the multiresolution scheme for the compressible Euler equations, J. Comput. Phys. 117 (1995) 251–261.

[40] A.M. Turing, The chemical basis of morphogenesis, Philos. Trans. Roy. Soc. London Ser. B 237 (1952) 37–72.

[41] T.P. Witelski, Segregation and mixing in degenerate diffusion in population dynamics, J. Math. Biol. 35 (1997) 695–712.