

Adaptive moving mesh upwind scheme for the two-species chemotaxis model

Alina Chertock^a, Alexander Kurganov^{b,c,*}, Mario Ricchiuto^d, Tong Wu^c

^a Department of Mathematics, North Carolina State University, Raleigh, NC 27695, USA

^b Department of Mathematics, Southern University of Science and Technology, Shenzhen, 518055, China

^c Mathematics Department, Tulane University, New Orleans, LA 70118, USA

^d Team CARDAMOM, INRIA Bordeaux Sud-Ouest, 33405 Talence, France

ARTICLE INFO

Article history:

Received 5 June 2018

Received in revised form 3 November 2018

Accepted 27 January 2019

Available online 11 March 2019

Keywords:

Two-species chemotaxis system

Adaptive moving mesh (AMM) method

Finite-volume upwind method

Singular (spiky) solutions

ABSTRACT

Chemotaxis systems are used to model the propagation, aggregation and pattern formation of bacteria/cells in response to an external stimulus, usually a chemical one. A common property of all chemotaxis systems is their ability to model a concentration phenomenon—rapid growth of the cell density in small neighborhoods of concentration points/curves. More precisely, the solution may develop singular, spiky structures, or even blow up in finite time. Therefore, the development of accurate and computationally efficient numerical methods for the chemotaxis models is a challenging task.

We study the two-species Patlak–Keller–Segel type chemotaxis system, in which the two species do not compete, but have different chemotactic sensitivities, which may lead to a significantly difference in cell density growth rates. This phenomenon was numerically investigated in Kurganov and Lukáčová-Medvidová (2014) and Chertock et al. (2018), where second- and higher-order methods on uniform Cartesian grids were developed. However, in order to achieve high resolution of the density spikes developed by the species with a lower chemotactic sensitivity, a very fine mesh had to be utilized and thus the efficiency of the numerical method was affected.

In this work, we consider an alternative approach relying on mesh adaptation, which helps to improve the approximation of the singular structures evolved by chemotaxis models. We develop, in particular, an adaptive moving mesh (AMM) finite-volume semi-discrete upwind method for the two-species chemotaxis system. The proposed AMM technique allows one to increase the density of mesh nodes at the blowup regions. This helps to substantially improve the resolution while using a relatively small number of finite-volume cells.

© 2019 Elsevier Ltd. All rights reserved.

1. Introduction

We develop an adaptive moving mesh (AMM) finite-volume semi-discrete upwind method for the two-species Patlak–Keller–Segel type chemotaxis system:

$$\begin{cases} (\rho_1)_t + \chi_1 \nabla \cdot (\rho_1 \nabla c) = \nu_1 \Delta \rho_1, \\ (\rho_2)_t + \chi_2 \nabla \cdot (\rho_2 \nabla c) = \nu_2 \Delta \rho_2, \\ \tau c_t = \nu \Delta c + \gamma_1 \rho_1 + \gamma_2 \rho_2 - \zeta c, \end{cases} \quad (x, y) \in \Omega \subset \mathbb{R}^2, \quad t > 0, \quad (1.1)$$

* Corresponding author at: Department of Mathematics, Southern University of Science and Technology, Shenzhen, 518055, China.

E-mail addresses: chertock@math.ncsu.edu (A. Chertock), kurganov@math.tulane.edu (A. Kurganov), mario.ricchiuto@inria.fr (M. Ricchiuto), twu2@tulane.edu (T. Wu).

where functions $\rho_1(x, y, t)$ and $\rho_2(x, y, t)$ are the cell densities of two non-competing species, $c(x, y, t)$ is the concentration of the chemoattractant, $\chi_2 > \chi_1 > 0$ are the chemotactic sensitivity parameters, and $v_1, v_2, v, \gamma_1, \gamma_2$ and ζ are positive constants which represent the diffusion, production rates and consumption rate, respectively. The parameter τ is equal to either 1 or 0, which correspond to the parabolic–parabolic or reduced parabolic–elliptic coupling, respectively. We assume that homogeneous Neumann (zero flux) boundary conditions are imposed along the entire boundary $\partial\Omega$ of Ω .

The two-species chemotaxis model (1.1) was proposed in [1] and then further analytically studied in [2–8]. Depending on the initial data and parameter values, the solution may either converge to a constant steady state, or develop singular structures. In principle, one may expect that in the blowup scenario, different blowup time scales are exhibited by the two variables and, for example, ρ_2 may blow up faster than ρ_1 . As proved in [3,5], this is not possible in the parabolic–elliptic ($\tau = 0$) case and only simultaneous blowup occurs albeit with scalings which may differ for the two species. Indeed, as it was first observed in [9], when $\chi_2 \gg \chi_1$, ρ_1 and ρ_2 blow up in a different manner: while ρ_2 clearly develops a δ -type singularity, ρ_1 blows up algebraically. This property was numerically discovered in [9] using the second-order hybrid finite-volume–finite-difference method developed in [10]. We note that the authors had to carry out a very careful mesh refinement study in order to demonstrate the simultaneous blowup in [9]. Their study allowed to underline that even with a mild mesh refinement one could have obtained very misleading results due to lack of resolution. In [10], a higher-order (fourth-order) hybrid finite-volume–finite-difference method was developed in an attempt to go beyond this limitation. While this goal was achieved, a very fine mesh was still required to make a blowup conjecture based on the numerical results.

Thus, the development of robust numerical methods for (1.1) represents a real challenge. Following the work in [9,10], in this paper we propose an alternative path to improve the capturing of the singular behavior of solutions of (1.1): the use of local mesh adaptation. In particular, we design an AMM method allowing one to enhance the resolution by clustering mesh nodes in the blowup regions; for references on AMM methods, see, e.g., [11–14] and references therein. The resulting method is obtained by combining a second-order finite-volume semi-discrete upwind scheme for (1.1) with the AMM approach from [13]. The latter, which is based on an earlier work [14], was designed in the context of nonlinear hyperbolic PDEs and is reformulated here to account for the behavior of (1.1). We demonstrate that the proposed AMM method allows one to confirm the conjecture of simultaneous blowup using a much smaller number of mesh cells.

The paper is organized as follows. In Section 2, we provide a systematic description of the proposed method. In Section 3, we test the new method on a number of numerical examples. The obtained results clearly demonstrate the advantages of the proposed AMM method.

2. Numerical method

2.1. Main notation

At a given time level t , we assume to have a nonuniform structured quadrilateral tessellation of the computational domain $\Omega = [x_{\min}, x_{\max}] \times [y_{\min}, y_{\max}]$, consisting of non-overlapping quadrilateral cells $\{I_{j,k}(t)\}$. The coordinate vector is denoted by $\mathbf{z} := (x, y)$, and in particular we use $\mathbf{z}_{j,k}(t) := (x_{j,k}(t), y_{j,k}(t))$ to label the geometric center of $I_{j,k}(t)$, while cell vertices are denoted by $\mathbf{z}_{j \pm \frac{1}{2}, k \pm \frac{1}{2}}(t) := (x_{j \pm \frac{1}{2}, k \pm \frac{1}{2}}(t), y_{j \pm \frac{1}{2}, k \pm \frac{1}{2}}(t))$, and cell areas by $|I_{j,k}(t)|$.

For every mesh face, we introduce a specific notation for length, mid-point and unit normal. The face between $I_{j,k}(t)$ and $I_{j+1,k}(t)$ has the length $\ell_{j+\frac{1}{2},k}(t) := |\mathbf{z}_{j+\frac{1}{2},k+\frac{1}{2}}(t) - \mathbf{z}_{j+\frac{1}{2},k-\frac{1}{2}}(t)|$, midpoint $\mathbf{z}_{j+\frac{1}{2},k}(t) := (\mathbf{z}_{j+\frac{1}{2},k+\frac{1}{2}}(t) + \mathbf{z}_{j+\frac{1}{2},k-\frac{1}{2}}(t))/2$ and unit normal vector $\mathbf{n}_{j+\frac{1}{2},k}(t)$ that points into $I_{j+1,k}(t)$. The face between cells $I_{j,k}(t)$ and $I_{j,k+1}(t)$ has the length $\ell_{j,k+\frac{1}{2}}(t) := |\mathbf{z}_{j+\frac{1}{2},k+\frac{1}{2}}(t) - \mathbf{z}_{j-\frac{1}{2},k+\frac{1}{2}}(t)|$, midpoint $\mathbf{z}_{j,k+\frac{1}{2}}(t) := (\mathbf{z}_{j+\frac{1}{2},k+\frac{1}{2}}(t) + \mathbf{z}_{j-\frac{1}{2},k+\frac{1}{2}}(t))/2$ and unit normal vector $\mathbf{n}_{j,k+\frac{1}{2}}(t)$ that points into $I_{j,k+1}(t)$.

At the same time level, the computed solution is represented by means of its cell averages:

$$(\bar{\rho}_i)_{j,k}(t) \approx \frac{1}{|I_{j,k}(t)|} \int_{I_{j,k}(t)} \rho_i(x, y, t) dx dy, \quad i = 1, 2 \quad \text{and} \quad \bar{c}_{j,k}(t) \approx \frac{1}{|I_{j,k}(t)|} \int_{I_{j,k}(t)} c(x, y, t) dx dy.$$

The cell averages of ρ_1 and ρ_2 are used to construct global (in space) piecewise polynomial interpolants of ρ_1 and ρ_2 . In particular, second-order finite-volume methods employ piecewise linear reconstructions:

$$\tilde{\mathbf{q}}(x, y; t) = \sum_{j,k} \mathbb{I}_{j,k}(t) \left[\bar{\mathbf{q}}_{j,k}(t) + (\mathbf{q}_x(t))_{j,k}(x - x_{j,k}(t)) + (\mathbf{q}_y(t))_{j,k}(y - y_{j,k}(t)) \right], \quad (2.1)$$

where $\mathbb{I}_{j,k}(t)$ is the characteristic function of cell $I_{j,k}(t)$, $\mathbf{q} := (\rho_1, \rho_2)^\top$, and $(\mathbf{q}_x(t))_{j,k}$ and $(\mathbf{q}_y(t))_{j,k}$ are appropriately defined slopes, which should be computed in a non-oscillatory manner using a nonlinear limiter (see Section 2.4 for a particular example). The corresponding one-sided point values at the cell interfaces are then obtained using the interpolant (2.1) as follows:

$$\begin{aligned} \mathbf{q}_{j+\frac{1}{2},k}^-(t) &:= \lim_{\substack{\mathbf{z} \rightarrow \mathbf{z}_{j+\frac{1}{2},k}(t) \\ \mathbf{z} \in I_{j,k}(t)}} \tilde{\mathbf{q}}(\mathbf{z}), & \mathbf{q}_{j+\frac{1}{2},k}^+(t) &:= \lim_{\substack{\mathbf{z} \rightarrow \mathbf{z}_{j+\frac{1}{2},k}(t) \\ \mathbf{z} \in I_{j+1,k}(t)}} \tilde{\mathbf{q}}(\mathbf{z}), \\ \mathbf{q}_{j,k+\frac{1}{2}}^-(t) &:= \lim_{\substack{\mathbf{z} \rightarrow \mathbf{z}_{j,k+\frac{1}{2}}(t) \\ \mathbf{z} \in I_{j,k}(t)}} \tilde{\mathbf{q}}(\mathbf{z}), & \mathbf{q}_{j,k+\frac{1}{2}}^+(t) &:= \lim_{\substack{\mathbf{z} \rightarrow \mathbf{z}_{j,k+\frac{1}{2}}(t) \\ \mathbf{z} \in I_{j,k+1}(t)}} \tilde{\mathbf{q}}(\mathbf{z}). \end{aligned} \quad (2.2)$$

2.2. Parabolic–elliptic case

We begin with a numerically more challenging parabolic–elliptic ($\tau = 0$) case.

2.2.1. Presentation of the general algorithm

When $\tau = 0$, the system (1.1) is composed of a time-independent elliptic equation, defining the concentration of the chemoattractant, and of a set of evolution equations for the species densities. To start our computation, we assume to be given a set of initial cell averages $\{(\overline{\rho_1})_{j,k}(0), (\overline{\rho_2})_{j,k}(0)\}$ of the densities, associated with the initial mesh defined by the nodes $\{z_{j \pm \frac{1}{2}, k \pm \frac{1}{2}}(0)\}$. For all $t \geq 0$, we propose the following discretization of (1.1).

1. Evolution step:

- Compute the cell-averaged concentrations $\overline{c}_{j,k}$ at the current time level t using efficient and robust relaxation iterations to numerically solve the third (time-independent) equation in the system (1.1). Besides the iteration scheme, this step also involves appropriate definitions of the directional derivatives of c along face normals as well as of reconstructed values of c on the faces.
- Compute new values of the cell-averaged densities $(\overline{\rho_1})_{j,k}, (\overline{\rho_2})_{j,k}$ at the next time level $t + \Delta t$ using a conservative upwind finite volume explicit approximation. This requires the definition of the upwind fluxes, including directional derivatives of the densities along face normals, as well as of the reconstruction strategy to evaluate the one-sided values of the densities (and their derivatives) at face midpoints. If a multi-stage scheme (such as a Runge–Kutta one) is used to integrate the resulting ODE, the evolution step is repeated to obtain intermediate values of the concentration.

- Move the mesh according to an appropriate error sensor depending on the smoothness of the densities. This requires the definition of the error sensor as well as of a strategy to define and control nodal displacements, and of a projection of cell averages of densities and concentration from one mesh to the other.

The following sections are devoted to the description of the main bricks of the method. In particular, Section 2.2.2 discusses the iterative procedure to solve the time-independent elliptic equation for the concentration, while the basic finite-volume semi-discrete upwind method is discussed in Section 2.2.3. Complements to complete the previous two aspects are discussed in Section 2.4 devoted to the evaluation of midpoint values and directional derivatives on cell faces. The only missing element is the AMM strategy, which is thoroughly explained in Section 2.5.

In order to simplify the notation, in the following paragraphs we will suppress the dependence on time of the considered quantities whenever possible.

2.2.2. Relaxation iterations for the concentration c

For a given distribution in space of ρ_1 and ρ_2 , we compute c by discretizing the third equation in (1.1) and then solving the obtained nonlinear algebraic system using Jacobi iterations. To this end, we first integrate the third equation in (1.1) over cell $I_{j,k}$ to obtain its finite-volume approximation:

$$\zeta \overline{c}_{j,k} - \nu \Delta_{j,k} c = \gamma_1 (\overline{\rho_1})_{j,k} + \gamma_2 (\overline{\rho_2})_{j,k}, \quad (2.3)$$

where

$$\Delta_{j,k} c := \frac{\ell_{j+\frac{1}{2},k} D_n c_{j+\frac{1}{2},k} - \ell_{j-\frac{1}{2},k} D_n c_{j-\frac{1}{2},k} + \ell_{j,k+\frac{1}{2}} D_n c_{j,k+\frac{1}{2}} - \ell_{j,k-\frac{1}{2}} D_n c_{j,k-\frac{1}{2}}}{|I_{j,k}|} \quad (2.4)$$

is the numerical approximation of Δc , and D_n is an approximation of the directional derivative along the normal direction. Details on the evaluation of D_n will be given in Section 2.4. The discretization (2.3), (2.4) can be rewritten as

$$\zeta \overline{c}_{j,k} - \sum_{-1 \leq m, n \leq 1} \alpha_{j+m, k+n} \overline{c}_{j+m, k+n} = \gamma_1 (\overline{\rho_1})_{j,k} + \gamma_2 (\overline{\rho_2})_{j,k}, \quad (2.5)$$

so that the Jacobi iterations applied to (2.5) read as

$$\overline{c}_{j,k}^* = \frac{\sum_{-1 \leq m, n \leq 1, |m|+|n|>0} \alpha_{j+m, k+n} \overline{c}_{j+m, k+n}^{(i)} + \gamma_1 (\overline{\rho_1})_{j,k} + \gamma_2 (\overline{\rho_2})_{j,k}}{\zeta - \alpha_{0,0}}. \quad (2.6)$$

In practice, each iteration step (2.6) is followed by the relaxation one:

$$\overline{c}_{j+m, k+n}^{(i+1)} = (1 - \theta) \overline{c}_{j+m, k+n}^{(i)} + \nu \overline{c}_{j,k}^*, \quad (2.7)$$

where θ is the relaxation parameter chosen as follows. If $\overline{c}_{j,k}^* \geq 0$ for all j, k , we take $\theta = \frac{2}{3}$; otherwise, we select θ such that $\overline{c}_{j+m, k+n}^{(i+1)} > \frac{1}{2} \overline{c}_{j+m, k+n}^{(i)}$.

We run the iterations (2.6), (2.7) until $\|\overline{c}^{(i+1)} - \overline{c}^{(i)}\|$ becomes smaller than a prescribed tolerance, while keeping $i < i_{\max}$, where $i_{\max}$ is an a-priori prescribed upper bound on the number of iterations.

2.2.3. Semi-discrete upwind scheme for the evolution of ρ_1 and ρ_2

In this section, we develop a second-order semi-discrete upwind scheme for the first two equations in (1.1), which we first rewrite in the following flux form:

$$\begin{cases} (\rho_1)_t + \nabla \cdot F_1(\rho_1, c) = 0, & F_1(\rho_1, c) = \chi_1 \rho_1 \nabla c - v_1 \nabla \rho_1, \\ (\rho_2)_t + \nabla \cdot F_2(\rho_2, c) = 0, & F_2(\rho_2, c) = \chi_2 \rho_2 \nabla c - v_2 \nabla \rho_2. \end{cases} \quad (2.8)$$

We then approximate the analytical fluxes F_i , $i = 1, 2$ at the cell interfaces $(x_{j+\frac{1}{2}}, y_k)$ and $(x_j, y_{k+\frac{1}{2}})$ by

$$\begin{aligned} F_i|_{(x_{j+\frac{1}{2}}, y_k)} &\approx (\mathcal{F}_i)_{j+\frac{1}{2}, k} = \chi_i (\rho_i)_{j+\frac{1}{2}, k} D_{\mathbf{n}} c_{j+\frac{1}{2}, k} - v_i D_{\mathbf{n}} (\rho_i)_{j+\frac{1}{2}, k}, \\ F_i|_{(x_j, y_{k+\frac{1}{2}})} &\approx (\mathcal{F}_i)_{j, k+\frac{1}{2}} = \chi_i (\rho_i)_{j, k+\frac{1}{2}} D_{\mathbf{n}} c_{j, k+\frac{1}{2}} - v_i D_{\mathbf{n}} (\rho_i)_{j, k+\frac{1}{2}}, \end{aligned} \quad i = 1, 2. \quad (2.9)$$

As before, $D_{\mathbf{n}}$ denotes the discrete directional derivatives in the corresponding normal directions, which will be explicitly defined in Section 2.5. The values at the midpoint of cell interfaces $(\rho_i)_{j+\frac{1}{2}, k}$ and $(\rho_i)_{j, k+\frac{1}{2}}$ are defined in an upwind manner as follows:

$$(\rho_i)_{j+\frac{1}{2}, k} = \begin{cases} (\rho_i)_{j+\frac{1}{2}, k}^- & \text{if } D_{\mathbf{n}} c_{j+\frac{1}{2}, k} \geq 0, \\ (\rho_i)_{j+\frac{1}{2}, k}^+ & \text{if } D_{\mathbf{n}} c_{j+\frac{1}{2}, k} < 0, \end{cases} \quad (\rho_i)_{j, k+\frac{1}{2}} = \begin{cases} (\rho_i)_{j, k+\frac{1}{2}}^- & D_{\mathbf{n}} c_{j, k+\frac{1}{2}} \geq 0, \\ (\rho_i)_{j, k+\frac{1}{2}}^+ & D_{\mathbf{n}} c_{j, k+\frac{1}{2}} < 0, \end{cases} \quad i = 1, 2,$$

with $(\rho_i)_{j+\frac{1}{2}, k}^\pm$ and $(\rho_i)_{j, k+\frac{1}{2}}^\pm$, $i = 1, 2$, being the corresponding one-sided point values reconstructed using the interpolant (2.1) and formula (2.2).

Finally, integrating (2.8) over $I_{j,k}$ and using the numerical fluxes (2.9), we arrive at the following semi-discrete upwind scheme:

$$\frac{d}{dt} (\bar{\rho}_i)_{j,k} = - \frac{\ell_{j+\frac{1}{2}, k} (\mathcal{F}_i)_{j+\frac{1}{2}, k} - \ell_{j-\frac{1}{2}, k} (\mathcal{F}_i)_{j-\frac{1}{2}, k} + \ell_{j, k+\frac{1}{2}} (\mathcal{F}_i)_{j, k+\frac{1}{2}} - \ell_{j, k-\frac{1}{2}} (\mathcal{F}_i)_{j, k-\frac{1}{2}}}{|I_{j,k}|}. \quad (2.10)$$

This is a system of time-dependent ODEs, which should be numerically solved using a sufficiently accurate and stable ODE solver. In the numerical results reported in Section 3, we have used the three-stage third-order strong-stability preserving Runge–Kutta (SSP-RK) method (see, e.g., [15,16]) implemented together with the “draining” time-step strategy to ensure the positivity of ρ_1 and ρ_2 (this strategy was originally proposed in [17] in the context of the Saint-Venant system of shallow water equations, applied to the chemotaxis systems in [10], and extended to structured quadrilateral meshes in [18]).

2.3. Parabolic–parabolic case

We now consider the parabolic–parabolic ($\tau = 1$) case, in which ρ_1 and ρ_2 are still evolved in time using the semi-discrete upwind scheme presented in Section 2.2.3. However, the chemoattractant concentration c now satisfies the time-dependent parabolic equation, which is discretized in space as follows:

$$\frac{d}{dt} \bar{c}_{j,k} = \sum_{-1 \leq m, n \leq 1} \alpha_{j+m, k+n} \bar{c}_{j+m, k+n} + \gamma_1 (\bar{\rho}_1)_{j,k} + \gamma_2 (\bar{\rho}_2)_{j,k} - \zeta \bar{c}_{j,k}, \quad (2.11)$$

where the coefficients $\alpha_{j+m, k+n}$ of the discrete Laplacian are the same as in formula (2.5).

In this case, one needs to numerically solve the system of ODEs (2.10), (2.11) for both ρ_1 , ρ_2 and c using an accurate and stable ODE solver. As outlined above, in our numerical experiments we have used the three-stage third-order SSP-RK method (notice that the initial conditions should now be given for all three variables). One can show that the positivity of c can be easily guaranteed by choosing a sufficiently small time step.

2.4. Positivity preserving reconstruction and normal derivatives

We now describe how the piecewise linear reconstruction (2.1) as well as required directional derivatives, $D_{\mathbf{n}} \mathbf{q}_{j+\frac{1}{2}, k}$ and $D_{\mathbf{n}} \mathbf{q}_{j, k+\frac{1}{2}}$, as well as $D_{\mathbf{n}} c_{j+\frac{1}{2}, k}$ and $D_{\mathbf{n}} c_{j, k+\frac{1}{2}}$, can be obtained.

We note that both the global polynomial in (2.1) and directional derivatives are computed in the componentwise manner and therefore, for the sake of simplicity, we restrict our consideration to the ρ_1 -component only.

We start with the reconstruction procedure and recall that the slopes $((\rho_1)_x)_{j,k}$ and $((\rho_1)_y)_{j,k}$ in (2.1) are to be computed using a nonlinear limiter needed to prevent oscillations and appearance of unphysical negative reconstructed point values. To this end, we build four linear interpolations: $L_{j,k}^{+,+}(x, y)$, $L_{j,k}^{+,-}(x, y)$, $L_{j,k}^{-,+}(x, y)$ and $L_{j,k}^{-,-}(x, y)$ outlined in Fig. 2.1. Each one of these linear interpolations is obtained by passing a plane through the point $(\mathbf{z}_{j,k}, (\bar{\rho}_1)_{j,k})$ and the corresponding points in the two neighboring cells. For example, $L_{j,k}^{+,+}(x, y)$ is obtained using the following three points: $(\mathbf{z}_{j,k}, (\bar{\rho}_1)_{j,k})$,

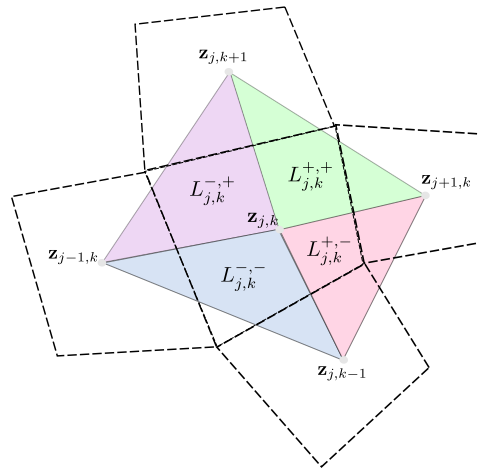


Fig. 2.1. Four linear interpolants over cell $I_{j,k}$.

$(\mathbf{z}_{j+1,k}, (\overline{\rho_1})_{j+1,k})$ and $(\mathbf{z}_{j,k+1}, (\overline{\rho_1})_{j,k+1})$. Notice that since $\mathbf{z}_{j,k}$ is the geometric center of cell $I_{j,k}$, the obtained linear interpolants are conservative in $I_{j,k}$, that is,

$$\frac{1}{|I_{j,k}|} \iint_{I_{j,k}} L_{j,k}^{\pm,\pm}(x, y) dx dy = (\overline{\rho_1})_{j,k}.$$

The corresponding numerical derivatives $((\rho_1)_x)_{j,k}$ and $((\rho_1)_y)_{j,k}$ are then computed by using a nonlinear limiter

$$\begin{aligned} ((\rho_1)_x)_{j,k} &= \frac{\sigma_{j,k}}{4} \left[(L_{j,k}^{+,+})_x + (L_{j,k}^{-,+})_x + (L_{j,k}^{-,-})_x + (L_{j,k}^{+,-})_x \right], \\ ((\rho_1)_y)_{j,k} &= \frac{\sigma_{j,k}}{4} \left[(L_{j,k}^{+,+})_y + (L_{j,k}^{-,+})_y + (L_{j,k}^{-,-})_y + (L_{j,k}^{+,-})_y \right], \end{aligned}$$

where $\sigma_{j,k}$ is a positivity enforcing parameter given by

$$\sigma_{j,k} = \min \left\{ 1, \left| \frac{(\overline{\rho_1})_{j,k}}{m_{j,k} - (\overline{\rho_1})_{j,k}} \right| \right\},$$

where

$$m_{j,k} = \min \left\{ (\widehat{\rho_1})_{j,k}(\mathbf{z}_{j+\frac{1}{2},k+\frac{1}{2}}), (\widehat{\rho_1})_{j,k}(\mathbf{z}_{j-\frac{1}{2},k+\frac{1}{2}}), (\widehat{\rho_1})_{j,k}(\mathbf{z}_{j+\frac{1}{2},k-\frac{1}{2}}), (\widehat{\rho_1})_{j,k}(\mathbf{z}_{j-\frac{1}{2},k-\frac{1}{2}}) \right\}$$

and $(\widehat{\rho_1})_{j,k} = (L_{j,k}^{+,+} + L_{j,k}^{-,+} + L_{j,k}^{-,-} + L_{j,k}^{+,-})/4$.

The directional derivatives of ρ_1 are defined as

$$D_{\mathbf{n}}(\rho_1)_{j+\frac{1}{2},k} = \nabla(\rho_1)(x_{j+\frac{1}{2},k}, y_{j+\frac{1}{2},k}) \cdot \mathbf{n}_{j+\frac{1}{2},k} \quad \text{and} \quad D_{\mathbf{n}}(\rho_1)_{j,k+\frac{1}{2}} = \nabla(\rho_1)(x_{j,k+\frac{1}{2}}, y_{j,k+\frac{1}{2}}) \cdot \mathbf{n}_{j,k+\frac{1}{2}},$$

and they can be estimated by

$$D_{\mathbf{n}}(\rho_1)_{j+\frac{1}{2},k} = \frac{(\rho_1)_{j+1,k}^W - (\rho_1)_{j,k}^E}{|\mathbf{z}_{j+1,k}^W - \mathbf{z}_{j,k}^E|} \quad \text{and} \quad D_{\mathbf{n}}(\rho_1)_{j,k+\frac{1}{2}} = \frac{(\rho_1)_{j,k+1}^S - (\rho_1)_{j,k}^N}{|\mathbf{z}_{j,k+1}^S - \mathbf{z}_{j,k}^N|}, \quad (2.12)$$

respectively. In (2.12), $\mathbf{z}_{j,k}^E$, for example, is the intersection between the orthogonal bisector of the cell interface between $I_{j,k}$ and $I_{j+1,k}$ and one of the line segments: either $\overline{\mathbf{z}_{j,k}\mathbf{z}_{j,k+1}}$ or $\overline{\mathbf{z}_{j,k}\mathbf{z}_{j,k-1}}$ (see Fig. 2.2), and

$$(\rho_1)_{j,k}^E = \widetilde{\rho_1}(\mathbf{z}_{j,k}^E),$$

where $\widetilde{\rho_1}(x, y)$ is a piecewise linear reconstruction of ρ_1 given by (2.1). All of the other values $(\mathbf{z}_{j+1,k}^W, \mathbf{z}_{j,k}^N, \mathbf{z}_{j,k+1}^S, (\rho_1)_{j,k}^W, (\rho_1)_{j,k}^N$ and $(\rho_1)_{j,k+1}^S)$ in (2.12) are computed in a similar manner.

Finally, we describe how the normal derivatives of c , needed in the discrete Laplacian (2.4), are evaluated. In fact, we use the same formula as (2.12), namely,

$$D_{\mathbf{n}}c_{j+\frac{1}{2},k} = \frac{c_{j+1,k}^W - c_{j,k}^E}{|\mathbf{z}_{j+1,k}^W - \mathbf{z}_{j,k}^E|} \quad \text{and} \quad D_{\mathbf{n}}c_{j,k+\frac{1}{2}} = \frac{c_{j,k+1}^S - c_{j,k}^N}{|\mathbf{z}_{j,k+1}^S - \mathbf{z}_{j,k}^N|}, \quad (2.13)$$

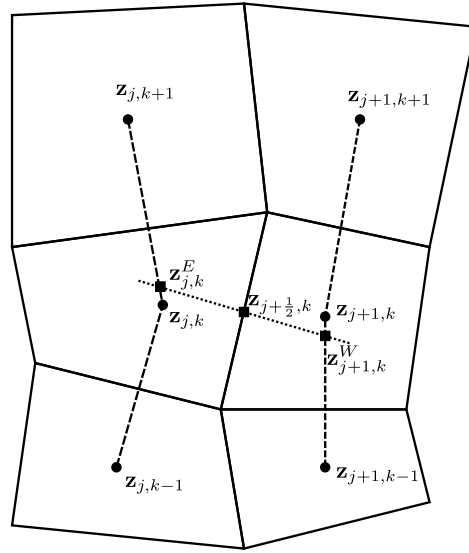


Fig. 2.2. Set-up for numerical computations of normal derivatives.

but the point values $c_{j,k}^E$, $c_{j,k}^W$, $c_{j,k}^N$ and $c_{j,k+1}^S$ are now computed in a different, simpler way. For instance, in order to compute $c_{j,k}^E$, we distinguish between the following two cases:

$$c_{j,k}^E = \begin{cases} \frac{(z_{j,k+1} - z_{j,k}^E) \bar{c}_{j,k} + (z_{j,k}^E - z_{j,k}) \bar{c}_{j,k+1}}{|z_{j,k+1} - z_{j,k}|}, & \text{if } z_{j,k}^E \in \overline{z_{j,k} z_{j,k+1}} \text{ (as in Fig. 2.2),} \\ \frac{(z_{j,k-1} - z_{j,k}^E) \bar{c}_{j,k} + (z_{j,k}^E - z_{j,k}) \bar{c}_{j,k-1}}{|z_{j,k-1} - z_{j,k}|}, & \text{if } z_{j,k}^E \in \overline{z_{j,k} z_{j,k-1}}. \end{cases}$$

All of the other values ($c_{j,k}^W$, $c_{j,k}^N$ and $c_{j,k+1}^S$) in (2.13) are computed in a similar manner.

2.5. Adaptive moving mesh (AMM) procedure

In this section, we briefly describe the AMM procedure used to adapt the structured finite-volume mesh so that the size of the cells are automatically getting substantially smaller near the spiky structures developed by the solution of the system (1.1). For a complete description, we refer the reader to [13].

We introduce the uniform rectangular logical mesh

$$\left\{ (\xi_{j+\frac{1}{2}}, \eta_{k+\frac{1}{2}}) \mid \xi_{j+\frac{1}{2}} = j\Delta\xi, \eta_{k+\frac{1}{2}} = k\Delta\eta \right\}, \quad j = 0, \dots, N, \quad k = 0, \dots, M,$$

where $\Delta\xi = 1/N$ and $\Delta\eta = 1/M$ are the spatial scales in the ξ - and η -directions, respectively. Let us denote the one-to-one coordinate transformation from the logical domain to the computational one by

$$(x, y) = (x(\xi, \eta), y(\xi, \eta)), \quad (\xi, \eta) \in [0, 1] \times [0, 1],$$

so that $x_{j+\frac{1}{2}, k+\frac{1}{2}} = x(\xi_{j+\frac{1}{2}}, \eta_{k+\frac{1}{2}})$ and $y_{j+\frac{1}{2}, k+\frac{1}{2}} = y(\xi_{j+\frac{1}{2}}, \eta_{k+\frac{1}{2}})$. We assume that $x(0, \eta) = x_{\min}$ and $x(1, \eta) = x_{\max}$ for all η as well as $y(\xi, 0) = y_{\min}$ and $y(\xi, 1) = y_{\max}$ for all ξ .

Mesh redistribution. Using a variational approach (see, e.g., [12]), one can obtain the following system of moving mesh PDEs (MMPDEs):

$$(\omega \mathbf{z}_\xi)_\xi + (\omega \mathbf{z}_\eta)_\eta = \mathbf{0}, \quad (2.14)$$

where ω is the monitor function, which is supposed to be dependent on a differential operator applied, for example, to one of the components of the computed solution $\mathbf{q}$. In Examples 1–4 considered in Section 3, we have taken the following ρ_2 -based monitor function:

$$\omega(\rho_2) = 1 + \delta\varphi(|D\rho_2|), \quad (2.15)$$

where $D\rho_2 = ((\rho_2)_\xi, (\rho_2)_\eta)$. When the MMPDEs (2.14) is discretized, we compute the corresponding numerical derivatives at (ξ_j, η_k) by using the second-order centered differences:

$$(D\rho_2)_{j,k} = \left(\frac{(\overline{\rho_2})_{j+1,k} - (\overline{\rho_2})_{j-1,k}}{2\Delta\xi}, \frac{(\overline{\rho_2})_{j,k+1} - (\overline{\rho_2})_{j,k-1}}{2\Delta\eta} \right).$$

The function φ in (2.15) is a smoothing filter employed to prevent the appearance of sharp gradients in the functions $\xi = \xi(x, y)$ and $\eta = \eta(x, y)$; for details, see [13]. Finally, α in (2.15) is an intensity parameter needed to control the mesh concentration. In our numerical experiments, we have chosen α to be

$$\delta = \left(\frac{1 - \beta}{\beta|\Omega|} \iint_{\Omega} \varphi(|D\rho_2|) \, dx \, dy \right)^{-1},$$

where $\beta \in (0, 1)$ is the prescribed fraction of mesh points to be concentrated at the “rough” areas of the solution and $|\Omega|$ is the total area of the computational domain. In Examples 1–4 considered in Section 3, we have used $\beta = 0.2$.

Remark 2.1. We note that the ρ_2 -based monitor function ω in (2.15) is a reasonable choice as ρ_2 expects to develop a spiky structure faster than ρ_1 . However, there are situations (like in Example 5 considered in Section 3, when the solution develops a multi-spiky structure with several spikes of substantially different magnitude), in which the use of ρ_2 -based monitor function may lead to an excessive concentration of the mesh cells near the highest spike. In order to prevent this undesirable scenario, one may replace (2.15) with

$$\omega(\rho_2) = 1 + \delta \varphi(|D(\ln(1 + \rho_2))|),$$

whose use would help to redistribute the mesh cells more uniformly so that they concentrate in the vicinities of all of the spikes. Also, in the multi-spiky Example 5, we have used a larger value of $\beta = 0.6$ in order to allocate a larger portion of the mesh cells in the spiky parts of the computed solution.

Equipped with the monitor function ω , we discretize the MMPDEs (2.14) using the centered difference approximation, which results in a linear algebraic system for the mesh points locations. The obtained system is numerically solved using the Jacobi iterations combined with the mesh relaxation procedure to avoid rapid change of mesh. Also, we limit the mesh movement at each iteration (see [13] for details) to ensure that

$$\min_{j,k} \left\{ \min_{i \in \{-1, 0, 1\}} \left(x_{j+\frac{3}{2}, k+\frac{1}{2}+i} - x_{j+\frac{1}{2}, k+\frac{1}{2}}, x_{j+\frac{1}{2}, k+\frac{1}{2}} - x_{j-\frac{1}{2}, k+\frac{1}{2}+i} \right) \right\} \geq \kappa \frac{x_{\max} - x_{\min}}{N}$$

and

$$\min_{j,k} \left\{ \min_{i \in \{-1, 0, 1\}} \left(y_{j+\frac{1}{2}, k+\frac{3}{2}+i} - y_{j+\frac{1}{2}, k+\frac{1}{2}}, y_{j+\frac{1}{2}, k+\frac{1}{2}} - y_{j+\frac{1}{2}, k-\frac{1}{2}+i} \right) \right\} \geq \kappa \frac{y_{\max} - y_{\min}}{M},$$

where κ is taken to be 0.1 in all of the numerical examples reported in Section 3.

Conservative solution projection. After obtaining the new mesh, we follow the approach in [14] and project the solution from the cells $I_{j,k}^{\text{old}}$, whose vertices are $\mathbf{z}_{j\pm\frac{1}{2}, k\pm\frac{1}{2}}^{\text{old}}$, to the new cells $I_{j,k}^{\text{new}}$ with the vertices $\mathbf{z}_{j\pm\frac{1}{2}, k\pm\frac{1}{2}}^{\text{new}}$.

Let $\overline{\mathbf{q}}_{j,k}^{\text{old}}$ and $\overline{\mathbf{q}}_{j,k}^{\text{new}}$ be the cell averages over the cells $I_{j,k}^{\text{old}}$ and $I_{j,k}^{\text{new}}$, respectively. We use the conservative solution projection step from [14] given by

$$|I_{j,k}^{\text{new}}| \overline{\mathbf{q}}_{j,k}^{\text{new}} = |I_{j,k}^{\text{old}}| \overline{\mathbf{q}}_{j,k}^{\text{old}} + \mu_{j+\frac{1}{2}, k} \mathbf{q}_{j+\frac{1}{2}, k}^{\text{old}} - \mu_{j-\frac{1}{2}, k} \mathbf{q}_{j-\frac{1}{2}, k}^{\text{old}} + \mu_{j, k+\frac{1}{2}} \mathbf{q}_{j, k+\frac{1}{2}}^{\text{old}} - \mu_{j, k-\frac{1}{2}} \mathbf{q}_{j, k-\frac{1}{2}}^{\text{old}},$$

where

$$\begin{aligned} \mu_{j+\frac{1}{2}, k} &= \frac{1}{2} \left[(x_{j+\frac{1}{2}, k-\frac{1}{2}}^{\text{new}} - x_{j+\frac{1}{2}, k+\frac{1}{2}}^{\text{old}})(y_{j+\frac{1}{2}, k+\frac{1}{2}}^{\text{new}} - y_{j+\frac{1}{2}, k-\frac{1}{2}}^{\text{old}}) \right. \\ &\quad \left. - (x_{j+\frac{1}{2}, k+\frac{1}{2}}^{\text{new}} - x_{j+\frac{1}{2}, k-\frac{1}{2}}^{\text{old}})(y_{j+\frac{1}{2}, k-\frac{1}{2}}^{\text{new}} - y_{j+\frac{1}{2}, k+\frac{1}{2}}^{\text{old}}) \right], \\ \mu_{j, k+\frac{1}{2}} &= \frac{1}{2} \left[(x_{j+\frac{1}{2}, k+\frac{1}{2}}^{\text{new}} - x_{j-\frac{1}{2}, k+\frac{1}{2}}^{\text{old}})(y_{j-\frac{1}{2}, k+\frac{1}{2}}^{\text{new}} - y_{j+\frac{1}{2}, k+\frac{1}{2}}^{\text{old}}) \right. \\ &\quad \left. - (x_{j-\frac{1}{2}, k+\frac{1}{2}}^{\text{new}} - x_{j+\frac{1}{2}, k+\frac{1}{2}}^{\text{old}})(y_{j+\frac{1}{2}, k+\frac{1}{2}}^{\text{new}} - y_{j-\frac{1}{2}, k+\frac{1}{2}}^{\text{old}}) \right], \\ \mathbf{q}_{j+\frac{1}{2}, k}^{\text{old}} &= \begin{cases} \mathbf{q}_{j+\frac{1}{2}, k}^+ & \text{if } \mu_{j+\frac{1}{2}, k} > 0, \\ \mathbf{q}_{j+\frac{1}{2}, k}^- & \text{if } \mu_{j+\frac{1}{2}, k} < 0, \end{cases} \quad \mathbf{q}_{j, k+\frac{1}{2}}^{\text{old}} = \begin{cases} \mathbf{q}_{j, k+\frac{1}{2}}^+ & \text{if } \mu_{j, k+\frac{1}{2}} > 0, \\ \mathbf{q}_{j, k+\frac{1}{2}}^- & \text{if } \mu_{j, k+\frac{1}{2}} < 0, \end{cases} \end{aligned}$$

and $\mathbf{q}_{j+\frac{1}{2}, k}^{\pm}$ and $\mathbf{q}_{j, k+\frac{1}{2}}^{\pm}$ are the point values reconstructed over the grid $I_{j,k}^{\text{old}}$ using the piecewise linear reconstruction (2.1) and formula (2.2).

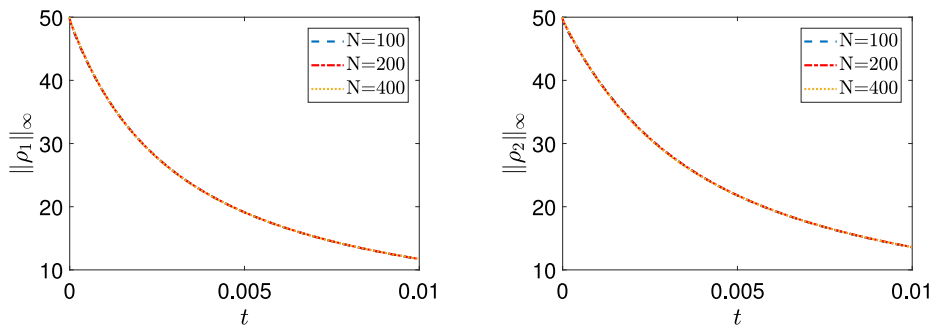


Fig. 3.1. Example 1 ($\chi_2 = 2$): The maximums of ρ_1 (left) and ρ_2 (right) as functions of time.

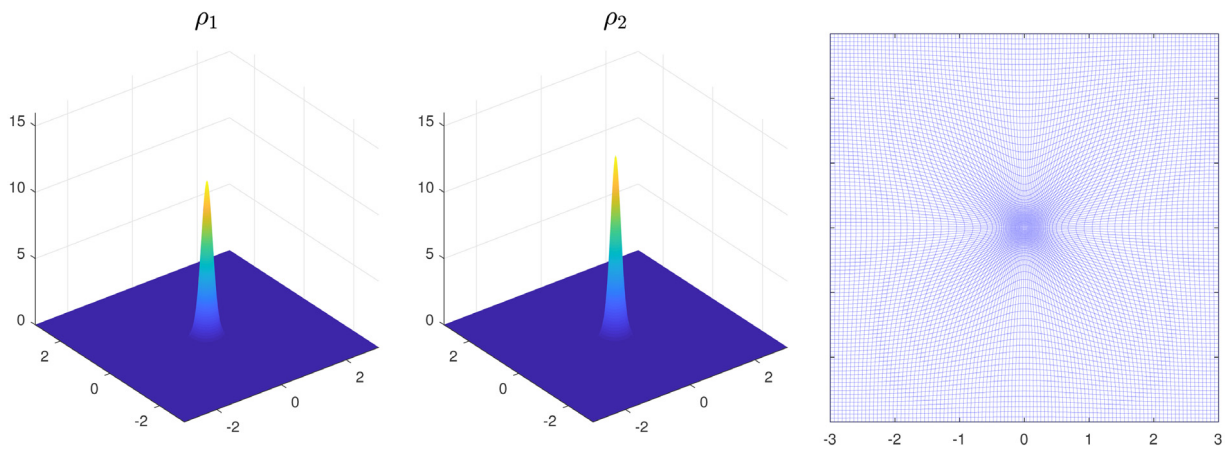


Fig. 3.2. Example 1 ($\chi_2 = 2$): $\rho_1(x, y, 0.01)$ (left) and $\rho_2(x, y, 0.01)$ (middle) and the mesh distribution at time $T = 0.01$ (right), obtained using $N = 100$.

3. Numerical examples

In this section, we test the designed AMM scheme on a number of numerical examples. In all of them, we consider the two-species chemotaxis system (1.1) with $\gamma_1 = \gamma_2 = \zeta = \nu_1 = \nu_2 = 1$ and numerically solved it on the domain $\Omega = [-3, 3] \times [-3, 3]$ subject to the homogeneous Neumann boundary conditions.

3.1. Parabolic–elliptic case

Here, we consider the system (1.1) with $\tau = 0$, $\nu = \chi_1 = 1$ and following initial data:

$$\rho_1(x, y, 0) = \rho_2(x, y, 0) = 50e^{-100(x^2+y^2)}.$$

The computational domain is initially split into $N \times N$ uniform cells and the numerical solution is computed on several different grids.

Example 1—Smooth solutions

The purpose of the first example is to test the accuracy of the designed AMM method. To this end, we consider two different values of χ_2 , for which the solution remains smooth.

We first choose $\chi_2 = 2$ and compute the numerical solutions until the final time $T = 0.01$ using $N = 100, 200$ and 400 . In Fig. 3.1, we show that the maximum values of both ρ_1 and ρ_2 decrease in time. As one can see, these curves are almost indistinguishable for different values of N . In Fig. 3.2, we plot the profiles of ρ_1 and ρ_2 along with the final time mesh distribution for $N = 100$.

We then choose a larger $\chi_2 = 10$ and use four different grids with $N = 50, 100, 200$ and 400 to compute the numerical solutions until the final time $T = 0.04$. In Fig. 3.3, where the maximum values of ρ_1 and ρ_2 are plotted as functions of time, one can see the effect of a larger chemotactic sensitivity parameter: ρ_2 does not decay in the same monotone manner as in the case of a smaller $\chi_2 = 2$. The final time numerical solution (ρ_1 and ρ_2) computed using $N = 100$, as well as the corresponding mesh distribution are presented in Fig. 3.4. It should be observed that in the case of a larger $\chi_2 = 10$, the

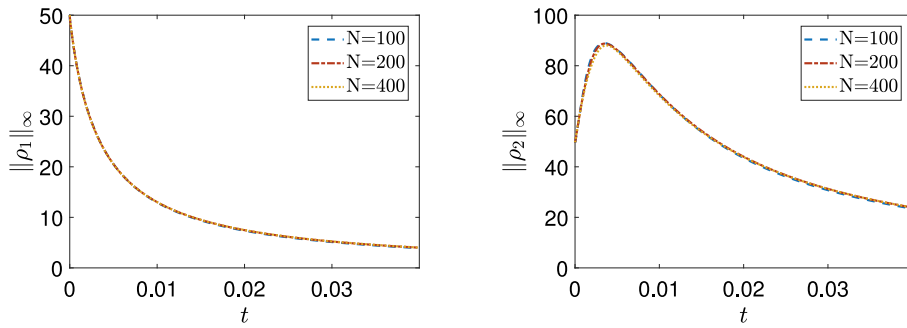


Fig. 3.3. Example 1 ($\chi_2 = 10$): The maximums of ρ_1 (left) and ρ_2 (right) as functions of time.

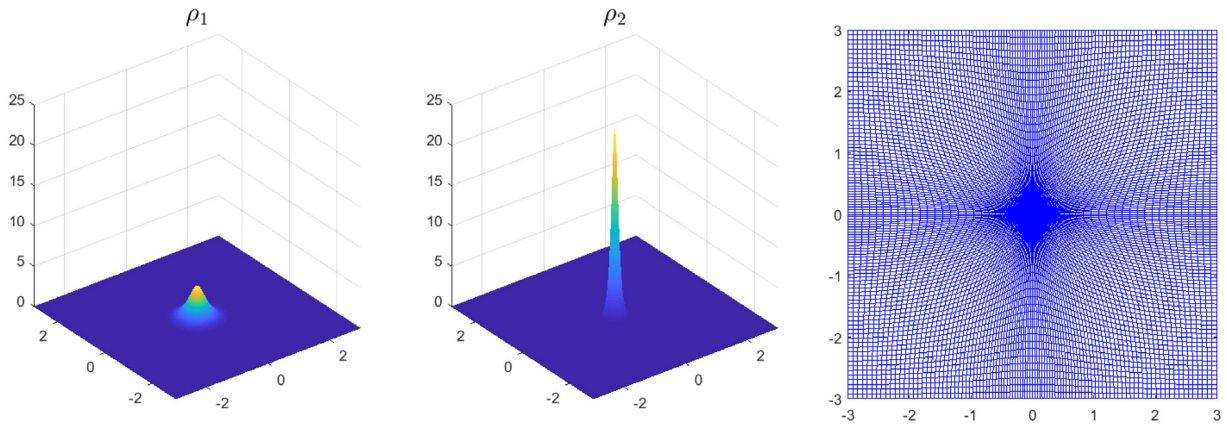


Fig. 3.4. Example 1 ($\chi_2 = 10$): $\rho_1(x, y, 0.04)$ (left) and $\rho_2(x, y, 0.04)$ (middle) and the mesh distribution at time $T = 0.04$ (right), obtained using $N = 100$.

Table 3.1

Example 1 ($\chi_2 = 10$): The L^1 - and L^∞ -convergence rates for ρ_1 and ρ_2 .

N	$r_2^N(\rho_1)$	$r_\infty^N(\rho_1)$	$r_2^N(\rho_2)$	$r_\infty^N(\rho_2)$
50	1.62	0.62	-0.06	0.10
100	1.47	1.08	1.03	0.62
200	1.83	1.01	2.16	1.06

difference in the magnitude between ρ_1 and ρ_2 is quite large (though they both are smooth and bounded) and the mesh is more concentrated near the origin.

We also use this example (with $\chi_2 = 10$) to study the convergence of the proposed AMM method and compute the experimental rates of convergence by

$$r_2^N(\rho_i) := \log_2 \left[\frac{E_2^N(\rho_i)}{E_2^{2N}(\rho_i)} \right] \quad \text{and} \quad r_\infty^N(\rho_i) := \log_2 \left[\frac{E_\infty^N(\rho_i)}{E_\infty^{2N}(\rho_i)} \right], \quad i = 1, 2,$$

where the L^2 - and L^∞ -errors are defined by

$$E_2^N(\rho_i) := \|\rho_i^N - \rho_i^{800}\|_2 = \sum_{j=1}^N \sum_{k=1}^N \left((\overline{\rho_i^N})_{j,k} - \varrho_i^{800}(x_{j,k}^N, y_{j,k}^N) \right)^2 |I_{j,k}^N|,$$

$$E_\infty^N(\rho_i) := \|\rho_i^N - \rho_i^{800}\|_\infty = \max_{1 \leq j,k \leq N} |(\overline{\rho_i^N})_{j,k} - \varrho_i^{800}(x_{j,k}^N, y_{j,k}^N)|.$$

Here, ϱ_i^{800} is a linear interpolant for the reference solution ρ_i^{800} computed using 800×800 cells, $(\overline{\rho_i^N})_{j,k}$ is the cell average of ρ_i over cell $I_{j,k}^N$ in the $N \times N$ mesh, and $(x_{j,k}^N, y_{j,k}^N)$ is the geometric center of cell $I_{j,k}^N$.

In Table 3.1, the experimental convergence rates are shown. As one can see, while the L^2 -errors exhibit the second-order decay, the L^∞ -errors decay much slower. We note that these results are satisfactory as the mesh is nonuniform and its size, especially near the origin, does not depend linearly on $1/N$.

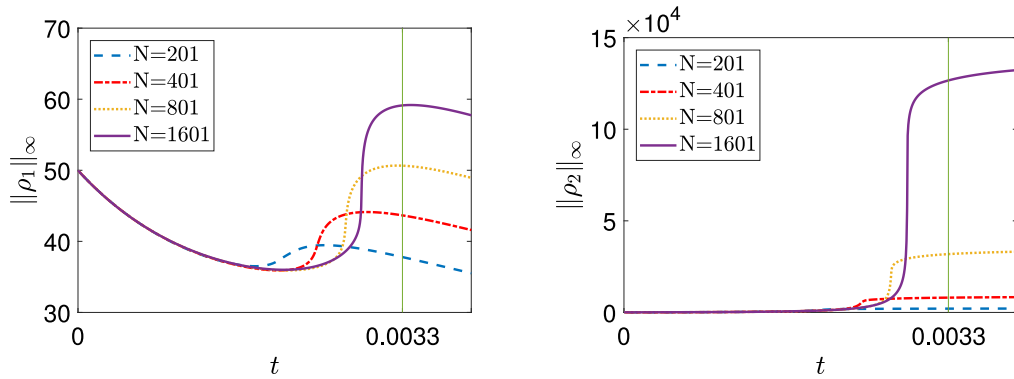


Fig. 3.5. Example 2: The maximums of ρ_1 (left) and ρ_2 (right) computed using the fourth-order hybrid finite-volume–finite-difference scheme from [10].

Example 2—Blowing up solution

In this example, we take even larger $\chi_2 = 20$. In this case, the solution behavior will be drastically different from what has been observed in Example 1. In particular, according to [3,5] both ρ_1 and ρ_2 will blow up in finite time and only simultaneous blowup is possible. Moreover, as it was first demonstrated in [9] and then confirmed in [10], ρ_1 and ρ_2 are expected to blow up in a different manner: while ρ_2 is supposed to develop a δ -type singularity, ρ_1 would grow up algebraically. Thus, capturing this blowup behavior is a challenging task. To cite an example, we refer the reader to [10], where the solution of the studied initial–boundary value problem was computed by both second- and a fourth-order hybrid finite-volume–finite-difference schemes on fixed uniform grids. The results obtained there indicated that even though both the second- and fourth-order schemes asymptotically behaved in a similar way in the blowup regime, the magnitudes of ρ_1 , which grows algebraically, were substantially larger in the fourth-order computations. On the other hand, the results (maximum values of ρ_1 and ρ_2 as functions of time) obtained by the aforementioned fourth-order scheme and presented in Fig. 3.5 show that the ρ_1 -component of the blowing up solution is still not well captured even when a very fine mesh is used. Moreover, as one can see in this figure, there are some discrepancy among the solutions on different grids before the approximated blowup time, which may imply that the actual blowup time is earlier than $t = 3.3 \times 10^{-3}$ predicted based on the experiments reported in [10]. It is therefore advantageous to use AMM techniques to numerically detect the blowup in a more convincing way.

In this work, we compute the solution until the final time $t = 0.004$ using the designed AMM method with $N = 101, 201, 401$ and 801 . As one can see in Fig. 3.6, our AMM scheme achieves much higher resolution compared with the fourth-order scheme from [10]. In addition, much less discrepancy between the solutions obtained on different grids is observed when the solution is still smooth, which suggests that the AMM method is capable of providing a better estimate of the blowup time. This is illustrated in Fig. 3.7, where we plot the ratios $\|\rho_2^{2N+1}\|_\infty / \|\rho_2^{N+1}\|_\infty$ as functions of time for both the fourth-order hybrid finite-volume–finite-difference scheme from [10] and the proposed AMM scheme.

The final (post blowup) time numerical solution (ρ_1 and ρ_2) computed using $N = 101$ are presented in Fig. 3.8 (left and middle). One can clearly see both qualitative and quantitative differences between ρ_1 and ρ_2 . A high resolution is achieved thanks to the AMM concentration ρ in the small neighborhood of the origin; see Fig. 3.8 (right).

3.2. Parabolic–parabolic case

We now consider the system (1.1) with $\tau = 1$, $\nu = 10$, $\chi_1 = 5$ and $\chi_2 = 60$. In all of the computations below, we use an initially uniform mesh with $N = 101$.

Example 3—Blowing up solution: ρ_2 blows up faster than ρ_1

In this example, taken from [9, Example 8], we choose the following initial conditions:

$$\rho_1(x, y, 0) = \rho_2(x, y, 0) = 500e^{-100(x^2+y^2)}, \quad c(x, y, 0) = 1.$$

For this setting, numerical results reported in [9] suggest that ρ_2 blows up faster than ρ_1 . In Fig. 3.9, we plot both densities at the final time $T = 0.001$ and the corresponding mesh distribution. Compared with the results obtained in [9] on a 400×400 uniform grid, the proposed AMM upwind scheme provides a higher resolution of the computed solution. Indeed, the maximum values of ρ_1 and ρ_2 are now $3.5758 \cdot 10^3$ and $2.8726 \cdot 10^5$, respectively (compare to $2.8660 \cdot 10^3$ and $1.8027 \cdot 10^5$).

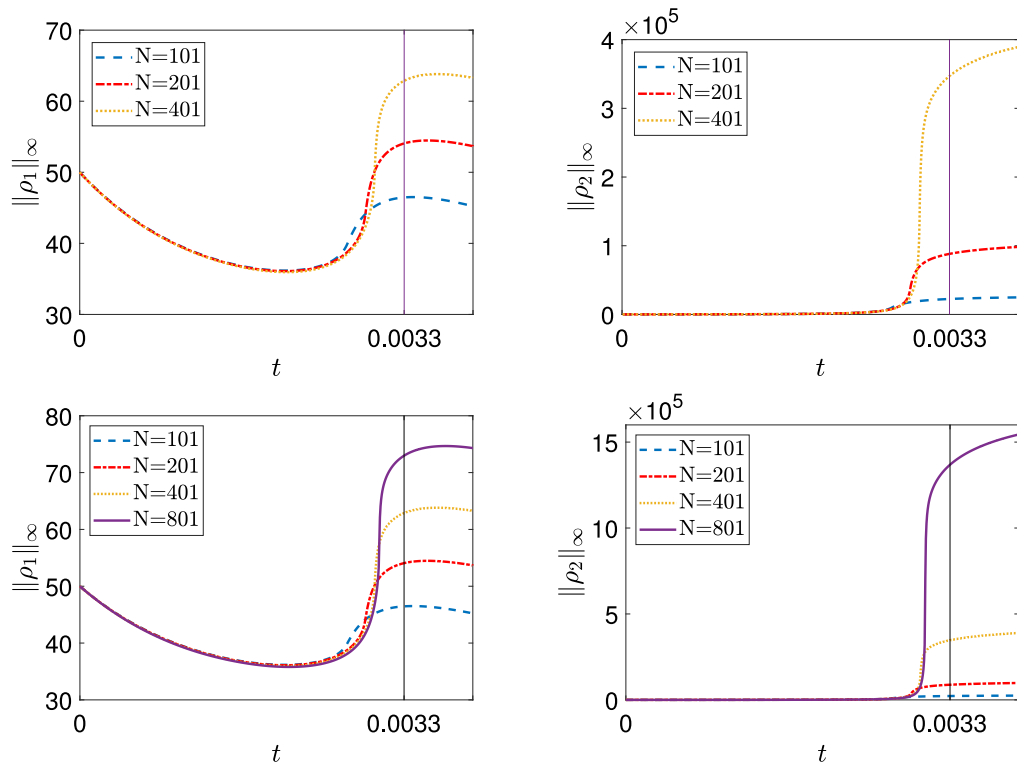


Fig. 3.6. Example 2: The maximums of ρ_1 (left) and ρ_2 (right) computed using the AMM scheme.

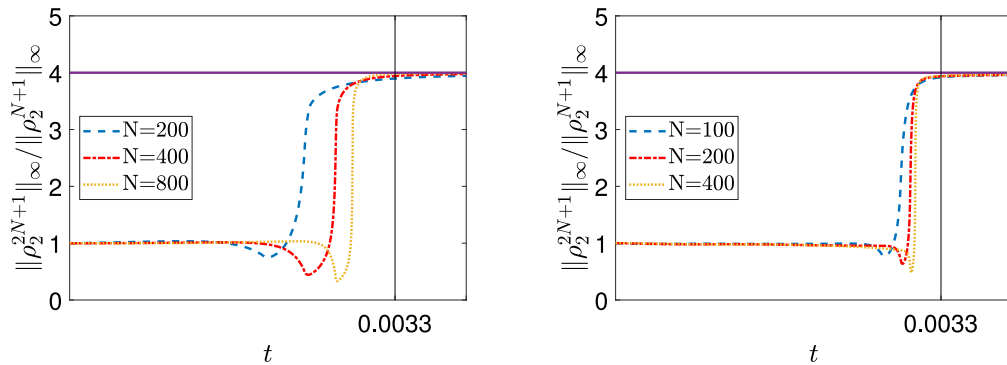


Fig. 3.7. Example 2: $\|\rho_2^{2N+1}\|_\infty / \|\rho_2^{N+1}\|_\infty$ as functions of time for different values of N , computed using the fourth-order hybrid finite-volume–finite-difference scheme from [10] (left) and the proposed AMM scheme (right).

Example 4—Blowing up solution: large initial data

In this example, taken from [9, Example 9], we choose a larger initial cell densities:

$$\rho_1(x, y, 0) = \rho_2(x, y, 0) = 5000e^{-100(x^2+y^2)}, \quad c(x, y, 0) = 1.$$

In this case, both ρ_1 and ρ_2 blow up much earlier than in Example 3; see Fig. 3.10, where both densities and the corresponding mesh distribution are plotted at the final time $T = 0.0002$. Once again, the proposed AMM upwind scheme produced more accurate results than those computed in [9] on a 400×400 uniform grid.

Example 5—Multi-spiky structures

In the last example, we take noisy initial data,

$$\rho_1(x, y, 0) \equiv \rho_2(x, y, 0) = 10(1 + \psi), \quad c(x, y, 0) \equiv 1,$$

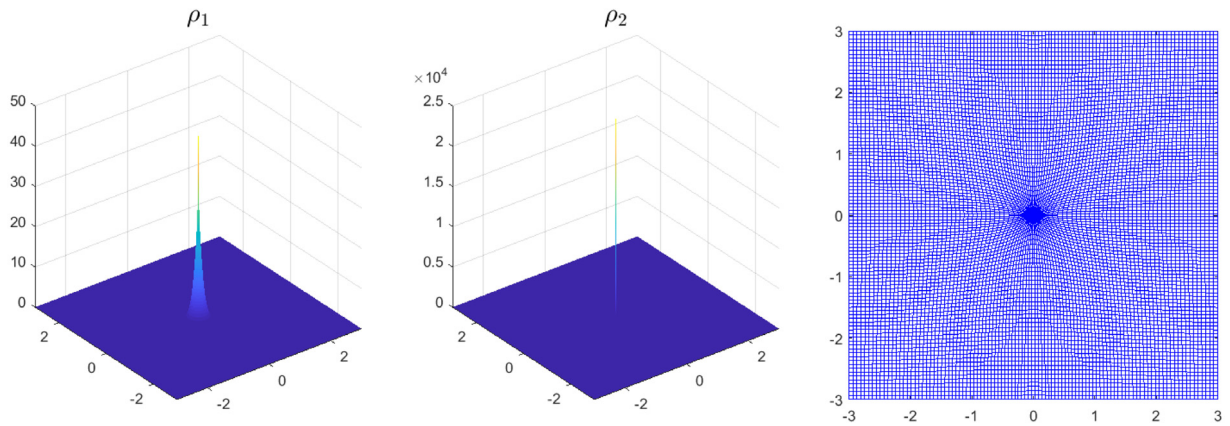


Fig. 3.8. Example 2: $\rho_1(x, y, 0.004)$ (left) and $\rho_2(x, y, 0.004)$ (middle) and the mesh distribution at time $T = 0.004$ (right), obtained using the AMM scheme with $N = 101$.

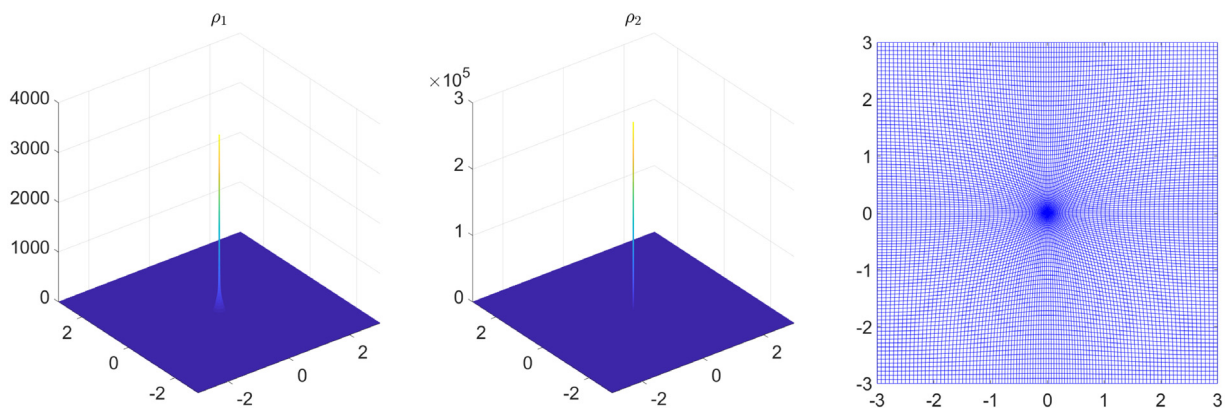


Fig. 3.9. Example 3: $\rho_1(x, y, 0.001)$ (left) and $\rho_2(x, y, 0.001)$ (middle) and the mesh distribution at time $T = 0.001$ (right), obtained using the AMM scheme with $N = 101$.

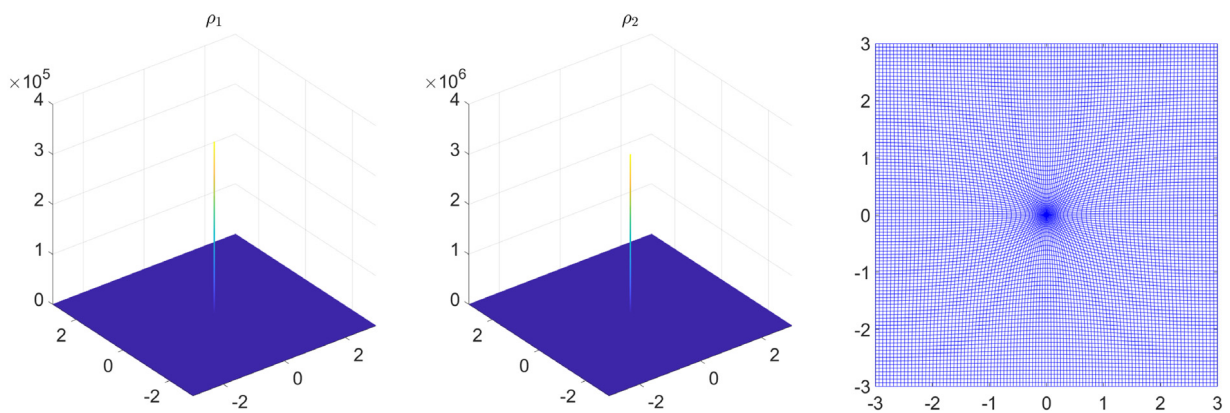


Fig. 3.10. Example 4: $\rho_1(x, y, 0.0002)$ (left) and $\rho_2(x, y, 0.0002)$ (middle) and the mesh distribution at time $T = 0.0002$ (right), obtained using the AMM scheme with $N = 101$.

where ψ is a random variable uniformly distributed on $[0, 1]$. The solution, reported in Fig. 3.11, develops complicated multi-spiky structures, which, as one can see, are captured by the our AMM upwind scheme with a high resolution.

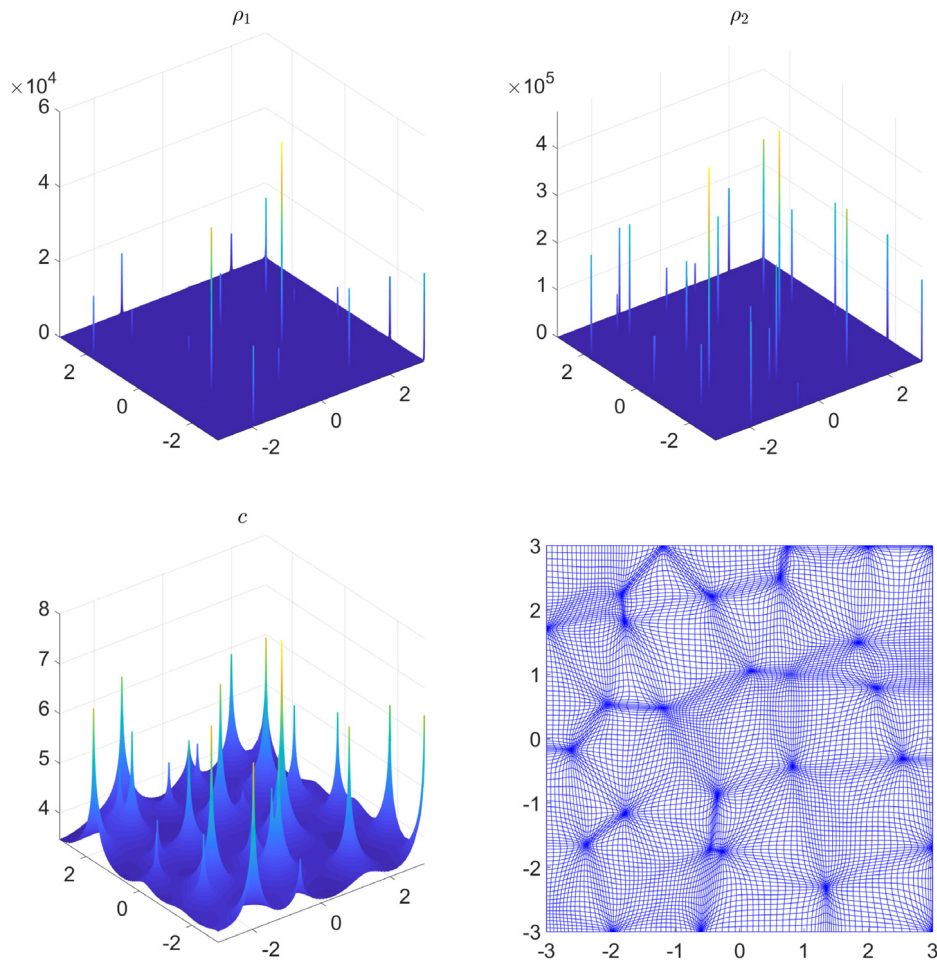


Fig. 3.11. Example 5: $\rho_1(x, y, 0.1)$ (top left), $\rho_2(x, y, 0.1)$ (top right), $c(x, y, 0.1)$ (bottom left) and the mesh distribution at time $t = 0.1$ (bottom right), obtained using the AMM scheme with $N = 101$.

Acknowledgments

The work of A. Chertock was supported in part by NSF (National Science Foundation) grants DMS-1521051 and DMS-1818684. The work of A. Kurganov was supported in part by NSFC grant 11771201 and NSF grant DMS-1521009.

References

- [1] G. Wolansky, Multi-components chemotactic system in the absence of conflicts, *European J. Appl. Math.* 13 (2002) 641–661.
- [2] C. Conca, E. Espejo, K. Vilches, Remarks on the blowup and global existence for a two species chemotactic Keller-Segel system in $\mathbb{R}^2$, *European J. Appl. Math.* 22 (6) (2011) 553–580.
- [3] E.E. Espejo, A. Stevens, T. Suzuki, Simultaneous blowup and mass separation during collapse in an interacting system of chemotactic species, *Differential Integral Equations* 25 (3–4) (2012) 251–288.
- [4] E.E. Espejo, A. Stevens, J.J.L. Velázquez, A note on non-simultaneous blow-up for a drift-diffusion model, *Differential Integral Equations* 23 (5–6) (2010) 451–462.
- [5] E.E. Espejo, K. Vilches, C. Conca, Sharp condition for blow-up and global existence in a two species chemotactic Keller-Segel system in $\mathbb{R}^2$, *European J. Appl. Math.* 24 (2013) 297–313.
- [6] E.E. Espejo Arenas, A. Stevens, J.J.L. Velázquez, Simultaneous finite time blow-up in a two-species model for chemotaxis, *Analysis (Munich)* 29 (3) (2009) 317–338.
- [7] A. Fasano, A. Mancini, M. Primicerio, Equilibrium of two populations subject to chemotaxis, *Math. Models Methods Appl. Sci.* 14 (2004) 503–533.
- [8] M. Kurokiba, T. Ogawa, Finite time blow-up of the solution for a nonlinear parabolic equation of drift-diffusion type, *Differential Integral Equations* 4 (2003) 427–452.
- [9] A. Kurganov, M. Lukáčová-Medviďová, Numerical study of two-species chemotaxis models, *Discrete Contin. Dyn. Syst. Ser. B* 19 (2014) 131–152.
- [10] A. Chertock, Y. Epshteyn, H. Hu, A. Kurganov, High-order positivity-preserving hybrid finite-volume-finite-difference methods for chemotaxis systems, *Adv. Comput. Math.* 44 (2018) 327–350.
- [11] L. Arpaia, M. Ricchiuto, r -adaptation for shallow water flows: conservation, well balancedness, efficiency, *Comput. Fluids* 160 (2018) 175–203.

- [12] W. Huang, R.D. Russell, Adaptive Moving Mesh Methods, in: *Applied Mathematical Sciences*, vol. 174, Springer, New York, 2011, p. xviii+432.
- [13] A. Kurganov, Z. Qu, O. Rozanova, T. Wu, Adaptive moving mesh central-upwind schemes for hyperbolic system of PDEs. Applications to compressible Euler equations and granular hydrodynamics, Preprint available at <http://129.81.170.14/~kurganov/Kurganov-Qu-Rozanova-Wu.pdf>.
- [14] H. Tang, T. Tang, Adaptive mesh methods for one-and two-dimensional hyperbolic conservation laws, *SIAM J. Numer. Anal.* 41 (2) (2003) 487–515.
- [15] S. Gottlieb, C.-W. Shu, E. Tadmor, Strong stability-preserving high-order time discretization methods, *SIAM Rev.* 43 (2001) 89–112.
- [16] S. Gottlieb, D. Ketcheson, C.-W. Shu, Strong Stability Preserving Runge-Kutta and Multistep Time Discretizations, World Scientific Publishing Co. Pte. Ltd., Hackensack, NJ, 2011, p. xii+176.
- [17] A. Bollermann, S. Noelle, M. Lukáčová-Medviďová, Finite volume evolution Galerkin methods for the shallow water equations with dry beds, *Commun. Comput. Phys.* 10 (2) (2011) 371–404.
- [18] A. Kurganov, Z. Qu, T. Wu, Well-balanced positivity preserving adaptive moving mesh central-upwind schemes for the Saint-Venant systems of shallow water equations, in preparation.