



---

*Research article*

## High-order finite difference approximation of the Keller-Segel model with additional self- and cross-diffusion terms and a logistic source

Panpan Xu<sup>1</sup>, Yongbin Ge<sup>1,\*</sup> and Lin Zhang<sup>2</sup>

<sup>1</sup> Institute of Applied Mathematics and Mechanics, Ningxia University, Yinchuan, 750021, China

<sup>2</sup> School of Mathematics and Information Science, Guangzhou University, Guangzhou, 510006, China

\* **Correspondence:** Email: [gybnxu@yeah.net](mailto:gybnxu@yeah.net).

**Abstract:** In this paper, we consider the Keller-Segel chemotaxis model with self- and cross-diffusion terms and a logistic source. This system consists of a fully nonlinear reaction-diffusion equation with additional cross-diffusion. We establish some high-order finite difference schemes for solving one- and two-dimensional problems. The truncation error remainder correction method and fourth-order Padé compact schemes are employed to approximate the spatial and temporal derivatives, respectively. It is shown that the numerical schemes yield second-order accuracy in time and fourth-order accuracy in space. Some numerical experiments are demonstrated to verify the accuracy and reliability of the proposed schemes. Furthermore, the blow-up phenomenon and bacterial pattern formation are numerically simulated.

**Keywords:** Keller-Segel chemotaxis model; logistic source; self-diffusion terms; cross-diffusion terms; finite difference method; high-order accuracy

---

### 1. Introduction

In this paper, we focus on the numerical solution of the following Keller-Segel (K-S) chemotaxis model including self- and cross-diffusion terms and a logistic source, i.e.,

$$\begin{cases} \partial_t u = \nabla \cdot (D(u)\nabla u) - \nabla \cdot (\chi(v)u\nabla v) + f(u), & (\mathbf{x}, t) \in \Omega_t, \\ \partial_t v = p_1 \Delta v + p_2 \Delta u + \alpha u - \beta v, & (\mathbf{x}, t) \in \Omega_t, \end{cases} \quad (1.1)$$

with the initial conditions

$$u(\mathbf{x}, 0) = u_0(\mathbf{x}), \quad v(\mathbf{x}, 0) = v_0(\mathbf{x}), \quad \mathbf{x} \in \Omega, \quad (1.2)$$

and Neumann boundary conditions

$$\nabla u \cdot \mathbf{n} = 0, \quad \nabla v \cdot \mathbf{n} = 0, \quad \mathbf{x} \in \partial\Omega, \quad t \in (0, T], \quad (1.3)$$

or Dirichlet boundary conditions

$$u(\mathbf{x}, t) = \varphi(\mathbf{x}, t), \quad v(\mathbf{x}, t) = \psi(\mathbf{x}, t), \quad \mathbf{x} \in \partial\Omega, \quad t \in (0, T], \quad (1.4)$$

where  $\Omega_t = \Omega \times (0, T]$  is a given bounded rectangular domain,  $\Omega \in \mathbb{R}^d$  ( $d = 1, 2$ ),  $T > 0$ ,  $\partial\Omega$  is the boundary of  $\Omega$ ,  $\mathbf{n}$  is the unit normal vector of  $\partial\Omega$ ,  $\mathbf{x}$  represents the space variable and  $t$  stands for time variable.  $u = u(\mathbf{x}, t)$  represents the cell (or organism) density and  $v = v(\mathbf{x}, t)$  denotes the concentration of the chemical signal. The function where  $D(u)$  describes the random diffusion rate of the cells,  $D(u) = \lambda + \theta u$ ,  $\lambda, \theta \geq 0$ .  $\chi(v)$  represents the chemotactic sensitivity function, in this paper,  $\chi(v)$  is treated as a positive constant.  $u\nabla v$  represents the chemotactic flux of cells. The function  $f(u)$  represents the logistic source that describes cell growth and death. We take  $f(u) = \delta u(1 - u)$ .  $p_1, p_2, \alpha$  and  $\beta$  are positive constants, they are expressed as the diffusivity, the degradation rate and the formation rate of the chemoattractant, respectively.  $\alpha u - \beta v$  is a kinetic function that describes the production and degradation of a chemical signal.  $\varphi(\mathbf{x}, t)$  and  $\psi(\mathbf{x}, t)$  are given smooth functions. In [17, 25, 28], the authors have proved, for arbitrarily small cross diffusion, the global time existence and uniqueness of weak solutions for the K-S chemotaxis model (1.1) through the use of an entropy function such as for  $t \rightarrow \infty$ .

One of the key characteristics of living organisms is that when they are stimulated by external environmental factors [1], they make directional movements towards or away from the stimulus. Chemotaxis refers to the directional movement of a cell or microorganism in response to a chemical stimulus along a concentration gradient of the stimulus. Chemotaxis plays an important role in bacterial aggregation, self-organization, and some physiological and pathological processes, such as, bacterial aggregation patterns, slime mold formation, tumor invasion, wound healing, embryonic development and angiogenesis [2]. The important properties of Eq (1.1) are the self-aggregation phenomenon and spatial pattern formation. The mathematical model and theory of chemotaxis can be traced back to the pioneering work of Patlak in the 1950s [3], as well as Keller and Segel in the 1970s [4–6]. For an in-depth understanding of the background of chemotaxis models, we can consult Refs. [7–12] and the references therein. It is well known that the solution of Eq (1.1) has great variation. It can develop sharp structures and exhibit finite time blow-up patterns within small regions of the domain [13–16]; thus, the numerical solution of the K-S system is challenging. How to avoid the blow-up problem of cells has been researched by many researchers. For example, four methods were mentioned in [17]: modify the chemotactic sensitivity, modify the cell diffusion, add a cross-diffusion term in the equation for the chemical signal or consider nonvanishing growth-death models. In the present review, we carefully discuss the K-S chemotaxis system with self- and cross-diffusion terms and a logistic source, as well as its related issues.

The model is a cross-coupled system of time-diffusion-chemotaxis partial differential equations with a complex structure and strong non-linearity, which makes it difficult to obtain an analytical solution. Numerical calculations and simulations play a key role in understanding these cellular dynamics mentioned above. Although a large body of research work in the literature already exists for these models, only a few numerical methods have been proposed; therefore, it is necessary to study effective numerical methods for the modeling and analysis of chemotactic systems. Below we review

the previous references for solving the K-S chemotaxis system: Akhmouch and Amine [1] discussed a new linearized implicit finite volume method for solving K-S chemotaxis growth models, developed a linearized semi-exponentially fitted scheme and verified it to be first-order convergent. Saito [15] established a conservative upwind finite element scheme for the simplified K-S model. The method is the first-order convergent. Sulman and Nguyen [18] used an adaptive moving mesh finite element method to study the K-S chemotaxis model, and numerically simulated the finite time blow-up and pattern formation characteristics of the model solution in the boundary and central regions and verified the second-order accuracy of numerical scheme. Qiu et al. [19] used the Galerkin method to solve the classical K-S chemotaxis model and developed a fully explicit scheme that has the third-order accuracy in time. Borsche et al. [20] used a first-order upwind scheme to solve the one-dimensional (1D) K-S chemotaxis model on general network topologies. Shen and Xu [21] proposed first-order and second-order schemes based on the gradient flow structure to solve parabolic-elliptic equations, and they developed a first-order scheme by using convex splitting to solve parabolic-parabolic equations. Braun et al. [22] proposed a first-order accuracy in time and second-order accuracy in space scheme for the doubly-parabolic systems at the inner nodes of the network connecting the two-dimensional (2D) chambers with the 1D channels. Xiao et al. [23] proposed a new semi-implicit scheme discretization of the characteristic derivative surface for the chemotaxis model using the finite element method with second-order accuracy. Zhang et al. [24] proposed a five-step fully implicit compact difference scheme for solving the K-S chemotaxis model. The scheme has fourth-order accuracy in both spatial and temporal directions. Hassan and Harfash [25] established a full finite element approximation scheme for the K-S model with self- and cross-diffusion terms and a logistic source, and numerically simulated the explosion phenomenon of high concentration solution of bacteria aggregation. Most of the above numerical methods produce results with low precision, which often leads to large computational errors. If we want to obtain high-resolution computed results to meet the needs of practical problems, we need to compute on very fine grid. In addition, high-order discretization usually needs non-compact stencils, which increases the band-width of the coefficient matrix of the algebraic equations. Both grid refinement and increasing the width of the matrix generate a great number of arithmetic calculations. From this viewpoint, neither the low-order method with fine meshes nor the high-order method with non-compact meshes appear to be computationally cost-effective. This is why high-order compact (HOC) finite difference methods has become important. On the basis of this viewpoint, in this paper, we aim to use the finite difference method to establish HOC difference schemes for Eq (1.1). The truncation error remainder correction method is used to deal with the temporal derivative and the fourth-order Padé compact difference schemes are used to treat the spatial derivatives. It is shown that they yield temporally second-order and spatially fourth-order accuracy.

The rest of this article is designed as follows: Section 2 establishes the difference scheme for 1D K-S chemotaxis models; Section 3 proposes the difference scheme for 2D K-S chemotaxis models. Section 4 verifies the accuracy and reliability of the presented schemes, and it shows the simulation of the blow-up phenomenon and bacterial pattern formation. Section 5 is the conclusion of the whole work.

**2. High-order finite difference scheme for 1D K-S model**

First, we consider the 1D model of Eq (1.1) as follows:

$$u_t = D(u)_x u_x + D(u)u_{xx} - \chi u_x v_x - \chi u v_{xx} + f(u), \quad (x, t) \in \Omega_t, \tag{2.1}$$

$$v_t = p_1 v_{xx} + p_2 u_{xx} + \alpha u - \beta v, \quad (x, t) \in \Omega_t, \tag{2.2}$$

with the initial conditions

$$u(x, 0) = u_0(x), \quad v(x, 0) = v_0(x), \quad x \in \Omega, \tag{2.3}$$

and Neumann boundary conditions

$$u_x(l, t) = 0, \quad v_x(l, t) = 0, \quad l \in \partial\Omega, \quad t > 0, \tag{2.4}$$

or Dirichlet boundary conditions

$$u(l, t) = \varphi(l, t), \quad v(l, t) = \psi(l, t), \quad l \in \partial\Omega, \quad t > 0. \tag{2.5}$$

Consider the domain  $[a, b] \times (0, T]$ , in which  $[a, b]$  is divided into  $M$  uniform intervals,  $a = x_0 < x_1 < x_2 < \dots < x_M = b$ , and  $(0, T]$  is divided into  $N$  uniform intervals. The spatial step size and temporal step size are  $h = \frac{b-a}{M}$  and  $\tau = \frac{T}{N}$ , respectively.

Denote the mesh points as  $(x_i, t_n)$ ,  $x_i = a + ih, t_n = n\tau, i = 0, 1, \dots, M, n = 0, 1, \dots, N$ , and denote  $\Omega_\tau = \{t_n | 0 \leq n \leq N\}$ ,  $\Omega = \{x_i | 0 \leq i \leq M\}$ ,  $\Omega_t = \Omega \times \Omega_\tau$ . Then, define grid function  $u = \{u_i^n | 0 \leq i \leq M, 0 \leq n \leq N\}$ ,  $v = \{v_i^n | 0 \leq i \leq M, 0 \leq n \leq N\}$  on  $\Omega_t$ . For convenience, we denote  $u_i^n = u(x_i, t_n), v_i^n = v(x_i, t_n), D_i^n = D(u_i^n), f_i^n = f(u_i^n)$ .

The difference operator is defined as follows:

$$\delta_t^+ u_i^n = \frac{u_i^{n+1} - u_i^n}{\tau}, \quad \delta_x u_i^n = \frac{u_{i+1}^n - u_{i-1}^n}{2h}, \quad \delta_x^2 u_i^n = \frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{h^2}.$$

We consider the values of Eqs (2.1) and (2.2) at the  $n$ -th level of time. By the Taylor expansion for the first-order derivatives in time, it becomes

$$(u_t)_i^n = \delta_t^+ u_i^n - \frac{\tau}{2} (u_{tt})_i^n + O(\tau^2), \tag{2.6}$$

$$(v_t)_i^n = \delta_t^+ v_i^n - \frac{\tau}{2} (v_{tt})_i^n + O(\tau^2). \tag{2.7}$$

To construct second-order accuracy scheme in the temporal direction,  $u_{tt}$  and  $v_{tt}$  in Eqs (2.6) and (2.7) need to be processed. Therefore, we take the derivatives of both sides of Eqs (2.1) and (2.2) with respect to  $t$ , and get

$$u_{tt} = D_{xt}u_x + D_x u_{xt} + D_t u_{xx} + D u_{xxt} - \chi u_{xt} v_x - \chi u_x v_{xt} - \chi u_t v_{xx} - \chi u v_{xxt} + f_t, \tag{2.8}$$

$$v_{tt} = p_1 v_{xxt} + p_2 u_{xxt} + \alpha u_t - \beta v_t. \tag{2.9}$$

To discretize Eqs (2.8) and (2.9) at point  $(x_i, t_n)$ , we use the central difference quotient for the spatial derivatives and forward difference quotient for the temporal derivatives,

$$(u_{tt})_i^n = \delta_t^+ \delta_x D_i^n \delta_x u_i^n + \delta_x D_i^n \delta_t^+ \delta_x u_i^n + \delta_t^+ D_i^n \delta_x^2 u_i^n + D_i^n \delta_t^+ \delta_x^2 u_i^n - \chi \delta_t^+ \delta_x u_i^n \delta_x v_i^n$$

$$-\chi\delta_x u_i^n \delta_t^+ \delta_x v_i^n - \chi\delta_t^+ u_i^n \delta_x^2 v_i^n - \chi u_i^n \delta_t^+ \delta_x^2 v_i^n + \delta_t^+ f_i^n + O(\tau + h^2), \quad (2.10)$$

$$(v_u)_i^n = p_1 \delta_t^+ \delta_x^2 v_i^n + p_2 \delta_t^+ \delta_x^2 u_i^n + \alpha \delta_t^+ u_i^n - \beta \delta_t^+ v_i^n + O(\tau + h^2), \quad (2.11)$$

and substituting Eqs (2.10) and (2.11) into Eqs (2.6) and (2.7), respectively, then discretizing Eqs (2.1) and (2.2) at point  $(x_i, t_n)$ , substituting Eqs (2.6) and (2.7) into Eqs (2.1) and (2.2), we obtain

$$\begin{aligned} & \delta_t^+ u_i^n - \frac{\tau}{2} (\delta_t^+ \delta_x D_i^n \delta_x u_i^n + \delta_x D_i^n \delta_t^+ \delta_x u_i^n + \delta_t^+ D_i^n \delta_x^2 u_i^n + D_i^n \delta_t^+ \delta_x^2 u_i^n - \chi \delta_t^+ \delta_x u_i^n \delta_x v_i^n \\ & - \chi \delta_x u_i^n \delta_t^+ \delta_x v_i^n - \chi \delta_t^+ u_i^n \delta_x^2 v_i^n - \chi u_i^n \delta_t^+ \delta_x^2 v_i^n + \delta_t^+ f_i^n = (D_x)_i^n (u_x)_i^n + D_i^n (u_{xx})_i^n \\ & - \chi (u_x)_i^n (v_x)_i^n - \chi u_i^n (v_{xx})_i^n + f_i^n + O(\tau^2 + \tau h^2), \end{aligned} \quad (2.12)$$

$$\begin{aligned} & \delta_t^+ v_i^n - \frac{\tau}{2} (p_1 \delta_t^+ \delta_x^2 v_i^n + p_2 \delta_t^+ \delta_x^2 u_i^n + \alpha \delta_t^+ u_i^n - \beta \delta_t^+ v_i^n) = p_1 (v_{xx})_i^n + p_2 (u_{xx})_i^n + \alpha u_i^n \\ & - \beta v_i^n + O(\tau^2 + \tau h^2). \end{aligned} \quad (2.13)$$

Substituting the definitions of  $\delta_t^+$ ,  $\delta_x$  and  $\delta_x^2$  into Eqs (2.12) and (2.13), and omitting the truncation error, we get

$$\begin{aligned} & \left( \frac{1}{8h^2} (D_{i+1}^n - D_{i-1}^n) - \frac{1}{2h^2} D_i^n - \frac{\chi}{8h^2} (v_{i+1}^n - v_{i-1}^n) \right) u_{i-1}^{n+1} + \left( \frac{1}{\tau} + \frac{1}{h^2} D_i^n + \frac{\chi}{2h^2} (v_{i+1}^n \right. \\ & \left. - 2v_i^n + v_{i-1}^n) \right) u_i^{n+1} + \left( -\frac{1}{8h^2} (D_{i+1}^n - D_{i-1}^n) - \frac{1}{2h^2} D_i^n + \frac{\chi}{8h^2} (v_{i+1}^n - v_{i-1}^n) \right) u_{i+1}^{n+1} = \\ & \left( -\frac{1}{8h^2} (D_{i+1}^{n+1} - 2D_{i+1}^n - D_{i-1}^{n+1} + 2D_{i-1}^n) + \frac{1}{2h^2} (D_i^{n+1} - 2D_i^n) + \frac{\chi}{8h^2} (v_{i+1}^{n+1} - 2v_{i+1}^n \right. \\ & \left. - v_{i-1}^{n+1} + 2v_{i-1}^n) \right) u_{i-1}^{n+1} + \left( \frac{1}{\tau} - \frac{1}{h^2} (D_i^{n+1} - 2D_i^n) - \frac{\chi}{2h^2} (v_{i+1}^{n+1} - 2v_{i+1}^n - 2v_i^{n+1} + 4v_i^n \right. \\ & \left. + v_{i-1}^{n+1} - 2v_{i-1}^n) - \chi (v_{xx})_i^n \right) u_i^{n+1} + \left( \frac{1}{8h^2} (D_{i+1}^{n+1} - 2D_{i+1}^n - D_{i-1}^{n+1} + 2D_{i-1}^n) + \frac{1}{2h^2} (D_i^{n+1} \right. \\ & \left. - 2D_i^n) - \frac{\chi}{8h^2} (v_{i+1}^{n+1} - 2v_{i+1}^n - v_{i-1}^{n+1} + 2v_{i-1}^n) \right) u_{i+1}^{n+1} + \frac{1}{2} (f_i^{n+1} + f_i^n) + (D_x)_i^n (u_x)_i^n \\ & + D_i^n (u_{xx})_i^n - \chi (u_x)_i^n (v_x)_i^n, \end{aligned} \quad (2.14)$$

$$\begin{aligned} & - \frac{p_1}{2h^2} v_{i+1}^{n+1} + \left( \frac{1}{\tau} + \frac{p_1}{h^2} + \frac{\beta}{2} \right) v_i^{n+1} - \frac{p_1}{2h^2} v_{i-1}^{n+1} = - \frac{p_1}{2h^2} v_{i+1}^n + \left( \frac{1}{\tau} + \frac{p_1}{h^2} - \frac{\beta}{2} \right) v_i^n - \\ & \frac{p_1}{2h^2} v_{i-1}^n + \left( \frac{\alpha}{2} - \frac{p_2}{h^2} \right) u_i^{n+1} + \left( \frac{\alpha}{2} + \frac{p_2}{h^2} \right) u_i^n + \frac{p_1}{2h^2} (u_{i+1}^{n+1} + u_{i-1}^{n+1} - u_{i+1}^n - u_{i-1}^n) \\ & + p_1 (v_{xx})_i^n + p_2 (u_{xx})_i^n. \end{aligned} \quad (2.15)$$

For the first- and second-order derivatives in Eqs (2.14) and (2.15), we use the following fourth-order compact Padé schemes [26],

$$\frac{1}{6} (\phi_x)_{i+1}^n + \frac{2}{3} (\phi_x)_i^n + \frac{1}{6} (\phi_x)_{i-1}^n = \frac{\phi_{i+1}^n - \phi_{i-1}^n}{2h} + O(h^4), \quad (2.16)$$

$$\frac{1}{12} (\phi_{xx})_{i+1}^n + \frac{5}{6} (\phi_{xx})_i^n + \frac{1}{12} (\phi_{xx})_{i-1}^n = \frac{\phi_{i+1}^n - 2\phi_i^n + \phi_{i-1}^n}{h^2} + O(h^4), \quad (2.17)$$

where  $\phi \in \{D, u, v\}$ , for  $i = 1, 2, \dots, M-1$ .

Because the values of  $(\phi_x)_0^n$ ,  $(\phi_x)_M^n$ ,  $(\phi_{xx})_0^n$  and  $(\phi_{xx})_M^n$  in Eqs (2.16) and (2.17) are unknown, and the values of  $u_0^{n+1}$ ,  $u_M^{n+1}$ ,  $v_0^{n+1}$  and  $v_M^{n+1}$  in Eqs (2.14) and (2.15) are also unknown, they need to be calculated. We discuss this on a case-by-case basis based on boundary conditions as follows.

For the Dirichlet boundary conditions, the values of  $u_0^{n+1}$ ,  $u_M^{n+1}$ ,  $v_0^{n+1}$  and  $v_M^{n+1}$  are computed by Eq (2.5). And the values of  $(\phi_x)_0^n$ ,  $(\phi_x)_M^n$ ,  $(\phi_{xx})_0^n$  and  $(\phi_{xx})_M^n$  are calculated by the consistent fourth-order boundary schemes [27],

$$(\phi_x)_0^n + \frac{14}{15}(\phi_x)_1^n = \frac{1}{h} \left( -\frac{184}{75}\phi_0^n + \frac{703}{180}\phi_1^n - \frac{89}{30}\phi_2^n + \frac{67}{30}\phi_3^n - \frac{77}{90}\phi_4^n + \frac{41}{300}\phi_5^n \right) + O(h^4), \quad (2.18)$$

$$(\phi_x)_M^n - \frac{14}{15}(\phi_x)_{M-1}^n = \frac{1}{h} \left( \frac{52}{25}\phi_M^n - \frac{1067}{180}\phi_{M-1}^n + \frac{67}{10}\phi_{M-2}^n - \frac{41}{10}\phi_{M-3}^n + \frac{133}{90}\phi_{M-4}^n - \frac{69}{300}\phi_{M-5}^n \right) + O(h^4), \quad (2.19)$$

$$(\phi_{xx})_0^n + \frac{51}{52}(\phi_{xx})_1^n = \frac{1}{h^2} \left( \frac{12293}{2340}\phi_0^n - \frac{18903}{1040}\phi_1^n + \frac{2891}{104}\phi_2^n - \frac{23941}{936}\phi_3^n + \frac{387}{26}\phi_4^n - \frac{5063}{1040}\phi_5^n + \frac{494}{720}\phi_6^n \right) + O(h^4), \quad (2.20)$$

$$(\phi_{xx})_M^n - \frac{51}{52}(\phi_{xx})_{M-1}^n = \frac{1}{h^2} \left( \frac{17599}{4680}\phi_M^n - \frac{17237}{1040}\phi_{M-1}^n + \frac{795}{26}\phi_{M-2}^n - \frac{28735}{936}\phi_{M-3}^n + \frac{1871}{104}\phi_{M-4}^n - \frac{6117}{1040}\phi_{M-5}^n + \frac{596}{720}\phi_{M-6}^n \right) + O(h^4). \quad (2.21)$$

For the Neumann boundary conditions, we use Eq (2.4) directly to calculate the values of  $(\phi_x)_0^n$  and  $(\phi_x)_M^n$ . The values of  $(\phi_{xx})_0^n$  and  $(\phi_{xx})_M^n$  are computed by Eqs (2.20) and (2.21). The values of  $u_0^{n+1}$ ,  $u_M^{n+1}$ ,  $v_0^{n+1}$  and  $v_M^{n+1}$  are computed by following fourth-order approximation constructed in [24].

$$\eta_0^{n+1} = \frac{1}{25}(48\eta_1^{n+1} - 36\eta_2^{n+1} + 16\eta_3^{n+1} - 3\eta_4^{n+1}) + O(h^4), \quad (2.22)$$

$$\eta_M^{n+1} = \frac{1}{25}(48\eta_{M-1}^{n+1} - 36\eta_{M-2}^{n+1} + 16\eta_{M-3}^{n+1} - 3\eta_{M-4}^{n+1}) + O(h^4), \quad (2.23)$$

where  $\eta \in \{u, v\}$ .

**Remark 1:** Eqs (2.14) and (2.15) are finite difference schemes after discretization of Eqs (2.1) and (2.2). The schemes involve only three grid points, and the local truncation errors are  $O(\tau^2 + \tau h^2 + h^4)$ . So the schemes yield the second-order accuracy in time and fourth-order accuracy in space.

The algorithm for solving Eqs (2.1)–(2.4) is described as follows:

---

**Algorithm 1.** The 1D algorithm for solving Eqs (2.1)–(2.4).

---

**Step 1:** Let  $n = 0$ , the initial time layers  $u_i^0, v_i^0$  are computed by Eq (2.3);

**Step 2:**  $(\phi_x)_i^n$  and  $(\phi_{xx})_i^n$  are computed by Eqs (2.16) and (2.17).  $(\phi_x)_0^n$  and  $(\phi_x)_M^n$  are given by Neumann boundary conditions Eq (2.4), the values of  $(\phi_{xx})_0^n$  and  $(\phi_{xx})_M^n$  are computed by Eqs (2.20) and (2.21), where  $\phi \in \{D, u, v\}$ ,  $i = 1, 2, \dots, M - 1$ ;

**Step 3:** The values of  $\eta_0^{n+1}$  and  $\eta_M^{n+1}$  are calculated by Eqs (2.22) and (2.23), where  $\eta \in \{u, v\}$ ;

**Step 4:** Given initial guess values, i.e.,  $u_i^{n+1(0)} \leftarrow u_i^{n(0)}$  and  $v_i^{n+1(0)} \leftarrow v_i^{n(0)}$ , and the iteration tolerance is  $Tol = 10^{-12}$ .

do  $k = 1, 2, \dots$  ( $k$  represents the number of iterations).

(1) Let  $D(u_i^{n+1(k)}) \leftarrow D(u_i^{n+1(k-1)})$ ,  $f(u_i^{n+1(k)}) \leftarrow f(u_i^{n+1(k-1)})$ ,  $i = 1, 2, \dots, M - 1$ ;

(2)  $u_i^{n+1(k)}$  are computed by Eq (2.14),  $v_i^{n+1(k)}$  are computed by incorporating  $u_i^{n+1(k)}$  into Eq (2.15). Coupled iteratively solved  $u_i^{n+1}$  and  $v_i^{n+1}$  by Eqs (2.14) and (2.15).

When  $\|u_i^{n+1(k)} - u_i^{n+1(k-1)}\| < Tol$  and  $\|v_i^{n+1(k)} - v_i^{n+1(k-1)}\| < Tol$ , iteration stops;

**Step 5:** Let  $n \leftarrow n + 1$ , repeat **Steps 2–4**, until the termination time.

---

**Remark 2:** The above algorithm is applicable for the 1D K-S chemotaxis model given by Eqs (2.1) and (2.2) with Neumann boundary conditions. In the numerical example, we also use the Dirichlet boundary conditions. It needs to move to Step 2, where the values of  $(\phi_x)_0^n$ ,  $(\phi_x)_M^n$ ,  $(\phi_{xx})_0^n$  and  $(\phi_{xx})_M^n$  are computed by using Eqs (2.18)–(2.21); then, in Step 3, the values of  $\eta_0^{n+1}$  and  $\eta_M^{n+1}$  are directly computed by using Eq (2.5).

### 3. High-order finite difference scheme for 2D K-S model

Then, we consider the 2D K-S model of Eq (1.1) as follows:

$$u_t = D(u)_x u_x + D(u)u_{xx} + D(u)y u_y + D(u)u_{yy} - \chi u_x v_x - \chi u v_{xx} - \chi u_y v_y - \chi u v_{yy} + f(u), \quad (x, y, t) \in \Omega_t, \quad (3.1)$$

$$v_t = p_1 v_{xx} + p_1 v_{yy} + p_2 u_{xx} + p_2 u_{yy} + \alpha u - \beta v, \quad (x, y, t) \in \Omega_t, \quad (3.2)$$

with the initial conditions

$$u(x, y, 0) = u_0(x, y), \quad v(x, y, 0) = v_0(x, y), \quad (x, y) \in \Omega, \quad (3.3)$$

and Neumann boundary conditions

$$\nabla u \cdot \mathbf{n} = 0, \quad \nabla v \cdot \mathbf{n} = 0, \quad (x, y) \in \partial\Omega, \quad t > 0, \quad (3.4)$$

or Dirichlet boundary conditions

$$u(x, y, t) = \varphi(x, y, t), \quad v(x, y, t) = \psi(x, y, t), \quad (x, y) \in \partial\Omega, \quad t > 0, \quad (3.5)$$

where  $D(u)$  and  $f(u)$  are similar to the 1D case.

Consider the domain  $[a, b]^2 \times (0, T]$ , in which  $[a, b]$  is divided into  $M$  uniform intervals,  $a = x_0 < x_1 < x_2 < \dots < x_M = b$ ,  $a = y_0 < y_1 < y_2 < \dots < y_M = b$ , and  $(0, T]$  is divided into  $N$  uniform intervals. The spatial step size and time step size are  $h = \frac{b-a}{M}$  and  $\tau = \frac{T}{N}$ , respectively.

Denote the mesh points as  $(x_i, y_j, t_n)$ ,  $x_i = a + ih, y_j = a + jh, t_n = n\tau, i = 0, 1, \dots, M, j = 0, 1, \dots, M, n = 0, 1, \dots, N$ , and  $\Omega_\tau = \{t_n | 0 \leq n \leq N\}$ ,  $\Omega = \{(x_i, y_j) | 0 \leq i \leq M, 0 \leq j \leq M\}$ ,  $\Omega_t = \Omega \times \Omega_\tau$ . Then, define the grid functions  $u = \{u_{i,j}^n | 0 \leq i \leq M, 0 \leq j \leq M, 0 \leq n \leq N\}$ ,  $v = \{v_{i,j}^n | 0 \leq i \leq M, 0 \leq j \leq M, 0 \leq n \leq N\}$  on  $\Omega_t$ . For convenience, we denote  $u_{i,j}^n = u(x_i, y_j, t_n)$ ,  $v_{i,j}^n = v(x_i, y_j, t_n)$ ,  $D_{i,j}^n = D(u_{i,j}^n)$  and  $f_{i,j}^n = f(u_{i,j}^n)$ .

The difference operator is defined as follows:

$$\delta_t^+ u_{i,j}^n = \frac{u_{i,j}^{n+1} - u_{i,j}^n}{\tau}, \quad \delta_x u_{i,j}^n = \frac{u_{i+1,j}^n - u_{i-1,j}^n}{2h}, \quad \delta_x^2 u_{i,j}^n = \frac{u_{i+1,j}^n - 2u_{i,j}^n + u_{i-1,j}^n}{h^2}.$$

Similarly,  $\delta_y u_{i,j}^n$  and  $\delta_y^2 u_{i,j}^n$  can be defined.

We consider the value of Eq (3.1) at the  $n$ -th level of time, use the Taylor expansion for the first-order derivative in time, we get

$$(u_t)_{i,j}^n = \delta_t^+ u_{i,j}^n - \frac{\tau}{2}(u_{tt})_{i,j}^n + O(\tau^2). \quad (3.6)$$

To gain second-order accuracy in the temporal dimension,  $u_{tt}$  in Eq (3.6) needs to be processed. We take the derivative of both sides of Eq (3.1) with respect to  $t$ , and then we have

$$\begin{aligned} u_{tt} = & D_{xt}u_x + D_x u_{xt} + D_t u_{xx} + D u_{xxt} + D_y u_y + D_y u_{yt} + D_t u_{yy} + D u_{yyt} - \chi u_{xt} v_x \\ & - \chi u_x v_{xt} - \chi u_t v_{xx} - \chi u v_{xxt} - \chi u_{yt} v_y - \chi u_y v_{yt} - \chi u_t v_{yy} - \chi u v_{yyt} + f_t. \end{aligned} \quad (3.7)$$

To discretize Eq (3.7) at point  $(x_i, y_j, t_n)$ , we use the forward difference quotient approximation for the time derivatives and central difference quotient approximation for the spatial derivatives, respectively. Then substituting it into Eq (3.6), we have

$$\begin{aligned} (u_t)_{i,j}^n = & \delta_t^+ u_{i,j}^n - \frac{\tau}{2} [\delta_t^+ \delta_x D_{i,j}^n \delta_x u_{i,j}^n + \delta_x D_{i,j}^n \delta_t^+ \delta_x u_{i,j}^n + \delta_t^+ D_{i,j}^n \delta_x^2 u_{i,j}^n + D_{i,j}^n \delta_t^+ \delta_x^2 u_{i,j}^n \\ & + \delta_t^+ \delta_y D_{i,j}^n \delta_y u_{i,j}^n + \delta_y D_{i,j}^n \delta_t^+ \delta_y u_{i,j}^n + \delta_t^+ D_{i,j}^n \delta_y^2 u_{i,j}^n + D_{i,j}^n \delta_t^+ \delta_y^2 u_{i,j}^n - \chi \delta_t^+ \delta_x u_{i,j}^n \delta_x v_{i,j}^n \\ & - \chi \delta_x u_{i,j}^n \delta_t^+ \delta_x v_{i,j}^n - \chi \delta_t^+ u_{i,j}^n \delta_x^2 v_{i,j}^n - \chi u_{i,j}^n \delta_t^+ \delta_x^2 v_{i,j}^n - \chi \delta_t^+ \delta_y u_{i,j}^n \delta_y v_{i,j}^n - \chi \delta_y u_{i,j}^n \delta_t^+ \delta_y v_{i,j}^n \\ & - \chi \delta_t^+ u_{i,j}^n \delta_y^2 v_{i,j}^n - \chi u_{i,j}^n \delta_t^+ \delta_y^2 v_{i,j}^n + \delta_t^+ f_{i,j}^n] + O(\tau^2 + \tau h^2), \end{aligned} \quad (3.8)$$

considering Eq (3.1) at point  $(x_i, y_j, t_n)$  and combining with Eq (3.8), we obtain

$$\begin{aligned} & \delta_t^+ u_{i,j}^n - \frac{\tau}{2} [\delta_t^+ \delta_x D_{i,j}^n \delta_x u_{i,j}^n + \delta_x D_{i,j}^n \delta_t^+ \delta_x u_{i,j}^n + \delta_t^+ D_{i,j}^n \delta_x^2 u_{i,j}^n + D_{i,j}^n \delta_t^+ \delta_x^2 u_{i,j}^n + \delta_t^+ \delta_y D_{i,j}^n \\ & \delta_y u_{i,j}^n + \delta_y D_{i,j}^n \delta_t^+ \delta_y u_{i,j}^n + \delta_t^+ D_{i,j}^n \delta_y^2 u_{i,j}^n + D_{i,j}^n \delta_t^+ \delta_y^2 u_{i,j}^n - \chi \delta_t^+ \delta_x u_{i,j}^n \delta_x v_{i,j}^n - \chi \delta_x u_{i,j}^n \delta_t^+ \\ & \delta_x v_{i,j}^n - \chi \delta_t^+ u_{i,j}^n \delta_x^2 v_{i,j}^n - \chi u_{i,j}^n \delta_t^+ \delta_x^2 v_{i,j}^n - \chi \delta_t^+ \delta_y u_{i,j}^n \delta_y v_{i,j}^n - \chi \delta_y u_{i,j}^n \delta_t^+ \delta_y v_{i,j}^n - \chi \delta_t^+ u_{i,j}^n \delta_y^2 \\ & v_{i,j}^n - \chi u_{i,j}^n \delta_t^+ \delta_y^2 v_{i,j}^n + \delta_t^+ f_{i,j}^n] = (D_x)_{i,j}^n (u_x)_{i,j}^n + D_{i,j}^n (u_{xx})_{i,j}^n + (D_y)_{i,j}^n (u_y)_{i,j}^n + D_{i,j}^n (u_{yy})_{i,j}^n \\ & - \chi (u_x)_{i,j}^n (v_x)_{i,j}^n - \chi u_{i,j}^n (v_{xx})_{i,j}^n - \chi (u_y)_{i,j}^n (v_y)_{i,j}^n - \chi u_{i,j}^n (v_{yy})_{i,j}^n + f_{i,j}^n + O(\tau^2 + \tau h^2). \end{aligned} \quad (3.9)$$

Similarly, using the same approach for Eq (3.2), we obtain

$$\delta_t^+ v_{i,j}^n - \frac{\tau}{2} (p_1 \delta_t^+ \delta_x^2 v_{i,j}^n + p_1 \delta_t^+ \delta_y^2 v_{i,j}^n + p_2 \delta_t^+ \delta_x^2 u_{i,j}^n + p_2 \delta_t^+ \delta_y^2 u_{i,j}^n + \alpha \delta_t^+ u_{i,j}^n - \beta \delta_t^+ v_{i,j}^n)$$



$$= p_1(v_{xx})_{i,j}^n + p_1(v_{yy})_{i,j}^n + p_2(u_{xx})_{i,j}^n + p_2(u_{yy})_{i,j}^n + \alpha u_{i,j}^n - \beta v_{i,j}^n + O(\tau^2 + \tau h^2); \quad (3.10)$$

substituting the definitions of  $\delta_t^+$ ,  $\delta_x$ ,  $\delta_y$ ,  $\delta_x^2$  and  $\delta_y^2$  into Eqs (3.9) and (3.10), and omitting the truncation error, we have

$$\begin{aligned} & \left[ -\frac{1}{8h^2}(D_{i+1,j}^n - D_{i-1,j}^n + 4D_{i,j}^n) + \frac{\chi}{8h^2}(v_{i+1,j}^n - v_{i-1,j}^n) \right] u_{i+1,j}^{n+1} + \left[ \frac{1}{8h^2}(D_{i+1,j}^n - D_{i-1,j}^n - 4D_{i,j}^n) \right. \\ & \left. - \frac{\chi}{8h^2}(v_{i+1,j}^n - v_{i-1,j}^n) \right] u_{i-1,j}^{n+1} + \left[ \frac{1}{\tau} + \frac{2}{h^2}D_{i,j}^n + \frac{\chi}{2h^2}(v_{i+1,j}^n + v_{i-1,j}^n - 4v_{i,j}^n + v_{i,j+1}^n + v_{i,j-1}^n) \right] \\ & u_{i,j}^{n+1} + \left[ -\frac{1}{8h^2}(D_{i,j+1}^n - D_{i,j-1}^n + 4D_{i,j}^n) + \frac{\chi}{8h^2}(v_{i,j+1}^n - v_{i,j-1}^n) \right] u_{i,j+1}^{n+1} + \left[ \frac{1}{8h^2}(D_{i,j+1}^n - D_{i,j-1}^n \right. \\ & \left. - 4D_{i,j}^n) - \frac{\chi}{8h^2}(v_{i,j+1}^n - v_{i,j-1}^n) \right] u_{i,j-1}^{n+1} = \left[ \frac{1}{8h^2}(D_{i+1,j}^{n+1} - 2D_{i+1,j}^n - D_{i-1,j}^{n+1} + 2D_{i-1,j}^n) + \frac{1}{2h^2} \right. \\ & (D_{i,j}^{n+1} - 2D_{i,j}^n) - \frac{\chi}{8h^2}(v_{i+1,j}^{n+1} - 2v_{i+1,j}^n - v_{i-1,j}^{n+1} + 2v_{i-1,j}^n) \left. \right] u_{i+1,j}^n + \left[ -\frac{1}{8h^2}(D_{i+1,j}^{n+1} - 2D_{i+1,j}^n - \right. \\ & \left. D_{i-1,j}^{n+1} + 2D_{i-1,j}^n) + \frac{1}{2h^2}(D_{i,j}^{n+1} - 2D_{i,j}^n) + \frac{\chi}{8h^2}(v_{i+1,j}^{n+1} - 2v_{i+1,j}^n - v_{i-1,j}^{n+1} + 2v_{i-1,j}^n) \right] u_{i-1,j}^n + \end{aligned} \quad (3.11)$$

$$\begin{aligned} & \left[ \frac{1}{\tau} - \frac{2}{h^2}(D_{i,j}^{n+1} - 2D_{i,j}^n) - \frac{\chi}{2h^2}(v_{i+1,j}^{n+1} - 2v_{i+1,j}^n - 4v_{i,j}^{n+1} + 8v_{i,j}^n + v_{i-1,j}^{n+1} - 2v_{i-1,j}^n + v_{i,j+1}^{n+1} \right. \\ & \left. - 2v_{i,j+1}^n + v_{i,j-1}^{n+1} - 2v_{i,j-1}^n) - \chi(v_{xx})_{i,j}^n - \chi(v_{yy})_{i,j}^n \right] u_{i,j}^n + \left[ \frac{1}{8h^2}(D_{i,j+1}^{n+1} - 2D_{i,j+1}^n - D_{i,j-1}^{n+1} \right. \\ & \left. + 2D_{i,j-1}^n) + \frac{1}{2h^2}(D_{i,j}^{n+1} - 2D_{i,j}^n) - \frac{\chi}{8h^2}(v_{i,j+1}^{n+1} - 2v_{i,j+1}^n - v_{i,j-1}^{n+1} + 2v_{i,j-1}^n) \right] u_{i,j+1}^n + \left[ -\frac{1}{8h^2} \right. \\ & (D_{i,j+1}^{n+1} - 2D_{i,j+1}^n - D_{i,j-1}^{n+1} + 2D_{i,j-1}^n) + \frac{1}{2h^2}(D_{i,j}^{n+1} - 2D_{i,j}^n) + \frac{\chi}{8h^2}(v_{i,j+1}^{n+1} - 2v_{i,j+1}^n - v_{i,j-1}^{n+1} \\ & \left. + 2v_{i,j-1}^n) \right] u_{i,j-1}^n + \frac{1}{2} [f_{i,j}^{n+1} + f_{i,j}^n] + (D_x)_{i,j}^n (u_x)_{i,j}^n + D_{i,j}^n (u_{xx})_{i,j}^n + (D_y)_{i,j}^n (u_y)_{i,j}^n + D_{i,j}^n (u_{yy})_{i,j}^n \\ & - \chi(u_x)_{i,j}^n (v_x)_{i,j}^n - \chi(u_y)_{i,j}^n (v_y)_{i,j}^n, \\ & - \frac{p_1}{2h^2} v_{i+1,j}^{n+1} - \frac{p_1}{2h^2} v_{i-1,j}^{n+1} + \left( \frac{1}{\tau} + 2\frac{p_1}{h^2} + \frac{\beta}{2} \right) v_{i,j}^{n+1} - \frac{p_1}{2h^2} v_{i,j+1}^{n+1} - \frac{p_1}{2h^2} v_{i,j-1}^{n+1} = -\frac{p_1}{2h^2} v_{i+1,j}^n - \\ & \frac{p_1}{2h^2} v_{i-1,j}^n + \left( \frac{1}{\tau} + 2\frac{p_1}{h^2} - \frac{\beta}{2} \right) v_{i,j}^n - \frac{p_1}{2h^2} v_{i,j+1}^n - \frac{p_1}{2h^2} v_{i,j-1}^n + \frac{p_2}{2h^2} u_{i+1,j}^{n+1} + \frac{p_2}{2h^2} u_{i-1,j}^{n+1} + \left( -2\frac{p_2}{h^2} \right. \\ & \left. + \frac{\alpha}{2} \right) u_{i,j}^{n+1} + \frac{p_2}{2h^2} u_{i,j+1}^{n+1} + \frac{p_2}{2h^2} u_{i,j-1}^{n+1} - \frac{p_2}{2h^2} u_{i+1,j}^n - \frac{p_2}{2h^2} u_{i-1,j}^n + \left( 2\frac{p_2}{h^2} + \frac{\alpha}{2} \right) u_{i,j}^n - \frac{p_2}{2h^2} u_{i,j+1}^n \\ & - \frac{p_2}{2h^2} u_{i,j-1}^n + p_1(v_{xx})_{i,j}^n + p_1(v_{yy})_{i,j}^n + p_2(u_{xx})_{i,j}^n + p_2(u_{yy})_{i,j}^n. \end{aligned} \quad (3.12)$$

For the first- and second-order derivatives in Eqs (3.11) and (3.12), we use the following fourth-order Padé schemes [26],

$$\frac{1}{6}(\phi_x)_{i+1,j}^n + \frac{2}{3}(\phi_x)_{i,j}^n + \frac{1}{6}(\phi_x)_{i-1,j}^n = \frac{\phi_{i+1,j}^n - \phi_{i-1,j}^n}{2h_x} + O(h^4), \quad (3.13)$$

$$\frac{1}{6}(\phi_y)_{i,j+1}^n + \frac{2}{3}(\phi_y)_{i,j}^n + \frac{1}{6}(\phi_y)_{i,j-1}^n = \frac{\phi_{i,j+1}^n - \phi_{i,j-1}^n}{2h} + O(h^4), \quad (3.14)$$

$$\frac{1}{12}(\phi_{xx})_{i+1,j}^n + \frac{5}{6}(\phi_{xx})_{i,j}^n + \frac{1}{12}(\phi_{xx})_{i-1,j}^n = \frac{\phi_{i+1,j}^n - 2\phi_{i,j}^n + \phi_{i-1,j}^n}{h^2} + O(h^4), \quad (3.15)$$

$$\frac{1}{12}(\phi_{yy})_{i,j+1}^n + \frac{5}{6}(\phi_{yy})_{i,j}^n + \frac{1}{12}(\phi_{yy})_{i,j-1}^n = \frac{\phi_{i,j+1}^n - 2\phi_{i,j}^n + \phi_{i,j-1}^n}{h^2} + O(h^4), \quad (3.16)$$

where  $\phi \in \{D, u, v\}$ , and  $i, j = 1, 2, \dots, M-1$ .

The values of  $(\phi_x)_{0,j}^n, (\phi_x)_{M,j}^n, (\phi_y)_{i,0}^n, (\phi_y)_{i,M}^n, (\phi_{xx})_{0,j}^n, (\phi_{xx})_{M,j}^n, (\phi_{yy})_{i,0}^n$  and  $(\phi_{yy})_{i,M}^n$  are unknown. In addition, the values of  $u_{0,j}^{n+1}, u_{M,j}^{n+1}, u_{i,0}^{n+1}, u_{i,M}^{n+1}, v_{0,j}^{n+1}, v_{M,j}^{n+1}, v_{i,0}^{n+1}$  and  $v_{i,M}^{n+1}$  are also unknown and they need to be calculated. Same with the 1D method, they are discussed according to the boundary conditions.

For the Dirichlet boundary conditions, the values of  $u_{0,j}^{n+1}, u_{M,j}^{n+1}, u_{i,0}^{n+1}, u_{i,M}^{n+1}, v_{0,j}^{n+1}, v_{M,j}^{n+1}, v_{i,0}^{n+1}$  and  $v_{i,M}^{n+1}$  are computed by using Eq (3.5), and the values of  $(\phi_x)_{0,j}^n, (\phi_x)_{M,j}^n, (\phi_y)_{i,0}^n, (\phi_y)_{i,M}^n, (\phi_{xx})_{0,j}^n, (\phi_{xx})_{M,j}^n, (\phi_{yy})_{i,0}^n$  and  $(\phi_{yy})_{i,M}^n$  are computed with the consistent fourth-order boundary schemes [27].

$$(\phi_x)_{0,j}^n + \frac{14}{15}(\phi_x)_{1,j}^n = \frac{1}{h} \left( -\frac{184}{75}\phi_{0,j}^n + \frac{703}{180}\phi_{1,j}^n - \frac{89}{30}\phi_{2,j}^n + \frac{67}{30}\phi_{3,j}^n - \frac{77}{90}\phi_{4,j}^n + \frac{41}{300}\phi_{5,j}^n \right) + O(h^4), \quad (3.17)$$

$$(\phi_x)_{M,j}^n - \frac{14}{15}(\phi_x)_{M-1,j}^n = \frac{1}{h} \left( \frac{52}{25}\phi_{M,j}^n - \frac{1067}{180}\phi_{M-1,j}^n + \frac{67}{10}\phi_{M-2,j}^n - \frac{41}{10}\phi_{M-3,j}^n + \frac{133}{90}\phi_{M-4,j}^n - \frac{69}{300}\phi_{M-5,j}^n \right) + O(h^4), \quad (3.18)$$

$$(\phi_y)_{i,0}^n + \frac{14}{15}(\phi_y)_{i,1}^n = \frac{1}{h} \left( -\frac{184}{75}\phi_{i,0}^n + \frac{703}{180}\phi_{i,1}^n - \frac{89}{30}\phi_{i,2}^n + \frac{67}{30}\phi_{i,3}^n - \frac{77}{90}\phi_{i,4}^n + \frac{41}{300}\phi_{i,5}^n \right) + O(h^4), \quad (3.19)$$

$$(\phi_y)_{i,M}^n - \frac{14}{15}(\phi_y)_{i,M-1}^n = \frac{1}{h} \left( \frac{52}{25}\phi_{i,M}^n - \frac{1067}{180}\phi_{i,M-1}^n + \frac{67}{10}\phi_{i,M-2}^n - \frac{41}{10}\phi_{i,M-3}^n + \frac{133}{90}\phi_{i,M-4}^n - \frac{69}{300}\phi_{i,M-5}^n \right) + O(h^4), \quad (3.20)$$

$$(\phi_{xx})_{0,j}^n + \frac{51}{52}(\phi_{xx})_{1,j}^n = \frac{1}{h^2} \left( \frac{12293}{2340}\phi_{0,j}^n - \frac{18903}{1040}\phi_{1,j}^n + \frac{2891}{104}\phi_{2,j}^n - \frac{23941}{936}\phi_{3,j}^n + \frac{387}{26}\phi_{4,j}^n - \frac{5063}{1040}\phi_{5,j}^n + \frac{494}{720}\phi_{6,j}^n \right) + O(h^4), \quad (3.21)$$

$$(\phi_{xx})_{M,j}^n - \frac{51}{52}(\phi_{xx})_{M-1,j}^n = \frac{1}{h^2} \left( \frac{17599}{4680}\phi_{M,j}^n - \frac{17237}{1040}\phi_{M-1,j}^n + \frac{795}{26}\phi_{M-2,j}^n - \frac{28735}{936}\phi_{M-3,j}^n + \frac{1871}{104}\phi_{M-4,j}^n - \frac{6117}{1040}\phi_{M-5,j}^n + \frac{596}{720}\phi_{M-6,j}^n \right) + O(h^4), \quad (3.22)$$

$$(\phi_{yy})_{i,0}^n + \frac{51}{52}(\phi_{yy})_{i,1}^n = \frac{1}{h^2} \left( \frac{12293}{2340}\phi_{i,0}^n - \frac{18903}{1040}\phi_{i,1}^n + \frac{2891}{104}\phi_{i,2}^n - \frac{23941}{936}\phi_{i,3}^n + \frac{387}{26}\phi_{i,4}^n - \frac{5063}{1040}\phi_{i,5}^n + \frac{494}{720}\phi_{i,6}^n \right) + O(h^4), \quad (3.23)$$

$$(\phi_{yy})_{i,M}^n - \frac{51}{52}(\phi_{yy})_{i,M-1}^n = \frac{1}{h^2} \left( \frac{17599}{4680}\phi_{i,M}^n - \frac{17237}{1040}\phi_{i,M-1}^n + \frac{795}{26}\phi_{i,M-2}^n - \frac{28735}{936}\phi_{i,M-3}^n + \frac{1871}{104}\phi_{i,M-4}^n - \frac{6117}{1040}\phi_{i,M-5}^n + \frac{596}{720}\phi_{i,M-6}^n \right) + O(h^4), \quad (3.24)$$

in which  $i, j = 0, 1, 2, \dots, M$ .

For the Neumann boundary conditions, we can directly use Eq (3.4) to evaluate  $(\phi_x)_{0,j}^n, (\phi_x)_{M,j}^n, (\phi_y)_{i,0}^n$  and  $(\phi_y)_{i,M}^n$ , and use Eqs (3.21) and (3.24) to compute  $(\phi_{xx})_{0,j}^n, (\phi_{xx})_{M,j}^n, (\phi_{yy})_{i,0}^n$  and  $(\phi_{yy})_{i,M}^n$ . The values of  $u_{0,j}^{n+1}, u_{M,j}^{n+1}, u_{i,0}^{n+1}, u_{i,M}^{n+1}, v_{0,j}^{n+1}, v_{M,j}^{n+1}, v_{i,0}^{n+1}$  and  $v_{i,M}^{n+1}$  are evaluated by the following fourth-order schemes constructed in [24].

$$\eta_{0,j}^{n+1} = \frac{1}{25} \left( 48\eta_{1,j}^{n+1} - 36\eta_{2,j}^{n+1} + 16\eta_{3,j}^{n+1} - 3\eta_{4,j}^{n+1} \right) + O(h^4), \quad (3.25)$$

$$\eta_{M,j}^{n+1} = \frac{1}{25} (48\eta_{M-1,j}^{n+1} - 36\eta_{M-2,j}^{n+1} + 16\eta_{M-3,j}^{n+1} - 3\eta_{M-4,j}^{n+1}) + O(h^4), \quad (3.26)$$

$$\eta_{i,0}^{n+1} = \frac{1}{25} (48\eta_{i,1}^{n+1} - 36\eta_{i,2}^{n+1} + 16\eta_{i,3}^{n+1} - 3\eta_{i,4}^{n+1}) + O(h^4), \quad (3.27)$$

$$\eta_{i,M}^{n+1} = \frac{1}{25} (48\eta_{i,M-1}^{n+1} - 36\eta_{i,M-2}^{n+1} + 16\eta_{i,M-3}^{n+1} - 3\eta_{i,M-4}^{n+1}) + O(h^4), \quad (3.28)$$

in which  $\eta \in \{u, v\}$ ,  $i, j = 0, 1, 2, \dots, M$ .

**Remark 3:** Equations (3.11) and (3.12) are finite difference schemes after discretization of Eqs (3.1) and (3.2). It is easy to find that the schemes are only involved in five grid points, and that the local truncation errors of Eqs (3.11) and (3.12) are  $O(\tau^2 + \tau h^2 + h^4)$ , i.e., the scheme has second-order accuracy in time direction and fourth-order accuracy in space direction.

The algorithm for solving Eqs (3.1)–(3.4) is described in Algorithm 2.

---

**Algorithm 2** The 2D algorithm for solving Eqs (3.1)–(3.4).

---

**Step 1:** Let  $n = 0$ , the initial time layers  $u_{i,j}^0$  and  $v_{i,j}^0$  are calculated by Eq (3.3);

**Step 2:**  $(\phi_x)_{i,j}^n$  and  $(\phi_{xx})_{i,j}^n$  are computed by incorporating Eqs (3.13)–(3.16).

The values of  $(\phi_x)_{0,j}^n$ ,  $(\phi_x)_{M,j}^n$ ,  $(\phi_y)_{i,0}^n$  and  $(\phi_y)_{i,M}^n$  are given by Neumann boundary conditions Eq (3.4), the values of  $(\phi_{xx})_{0,j}^n$ ,  $(\phi_{xx})_{M,j}^n$ ,  $(\phi_{yy})_{i,0}^n$  and  $(\phi_{yy})_{i,M}^n$  are computed by Eqs (3.21)–(3.24), where  $\phi \in \{D, u, v\}$ ,  $i, j = 1, 2, \dots, M - 1$ ;

**Step 3:** The values of  $\eta_{0,j}^{n+1}$ ,  $\eta_{M,j}^{n+1}$ ,  $\eta_{i,0}^{n+1}$  and  $\eta_{i,M}^{n+1}$  are computed by Eqs (3.25) and (3.28), where  $\eta \in \{u, v\}$ ,  $i, j = 0, 1, \dots, M$ ;

**Step 4:** Given initial guess values, i.e.  $u_{i,j}^{n+1(0)} \leftarrow u_{i,j}^{n(0)}$  and  $v_{i,j}^{n+1(0)} \leftarrow v_{i,j}^{n(0)}$ , and the iteration tolerance is  $Tol = 10^{-11}$ .

do  $k = 1, 2, \dots$  ( $k$  represents the number of iterations).

(1) Let  $D(u_{i,j}^{n+1(k)}) \leftarrow D(u_{i,j}^{n+1(k-1)})$ ,  $f(u_{i,j}^{n+1(k)}) \leftarrow f(u_{i,j}^{n+1(k-1)})$ ,  $i, j = 1, 2, \dots, M - 1$ ;

(2)  $u_{i,j}^{n+1(k)}$  are computed by Eq (3.11),  $v_{i,j}^{n+1(k)}$  are computed by incorporating  $u_{i,j}^{n+1(k)}$  into Eq (3.12). Coupled iteratively solved  $u_{i,j}^{n+1}$  and  $v_{i,j}^{n+1}$  by Eqs (3.11) and (3.12).

When  $\|u_{i,j}^{n+1(k)} - u_{i,j}^{n+1(k-1)}\| < Tol$  and  $\|v_{i,j}^{n+1(k)} - v_{i,j}^{n+1(k-1)}\| < Tol$ , iteration stops;

**Step 5:** Let  $n \leftarrow n + 1$ , repeat **Steps 2–4**, until the termination time.

---

**Remark 4:** The above algorithm is described for the 2D K-S chemotaxis model (3.1) and (3.2) with Neumann boundary conditions. In the numerical example, we are also going to use Dirichlet boundary conditions. It needs to compute the values of  $(\phi_x)_{0,j}^n$ ,  $(\phi_x)_{M,j}^n$ ,  $(\phi_y)_{i,0}^n$  and  $(\phi_y)_{i,M}^n$  with Eqs (3.17)–(3.24) in the Step 2, and the values of  $\eta_{0,j}^{n+1}$ ,  $\eta_{M,j}^{n+1}$ ,  $\eta_{i,0}^{n+1}$  and  $\eta_{i,M}^{n+1}$  are calculated by Eq (3.5) in the Step 3.

#### 4. Numerical experiments

In this section, we employ several numerical examples to verify the performances of the proposed method for solving the K-S chemotaxis system give by Eqs (1.1)–(1.4). We verify the accuracy of the numerical solution of the K-S chemotaxis model with Dirichlet boundary conditions in 1D and 2D, and we compare it with the results in existing literature. We also study the numerical solution of the K-S chemotaxis model with Neumann boundary conditions in 2D.

To measure accuracy, we simply modify the problem Eq (1.1) by adding the source terms  $s(\mathbf{x}, t)$  and  $g(\mathbf{x}, t)$  and transform it into the following equations:

$$\begin{cases} u_t = \nabla \cdot (D(u)\nabla u) - \nabla \cdot (\chi u \nabla v) + \delta u(1 - u) + s(\mathbf{x}, t), & (\mathbf{x}, t) \in \Omega_t. \\ v_t = p_1 \Delta v + p_2 \Delta u + \alpha u - \beta v + g(\mathbf{x}, t). & (\mathbf{x}, t) \in \Omega_t. \end{cases} \quad (4.1)$$

The computations are programmed in Fortran 77 language with double precision arithmetic and are implemented on a personal computer equipped with Intel Core i7-7500U CPU @ 2.70 GHz 2.90 GHz, 4.00 GB RAM Windows workstation. The convergence rate of errors are defined as follows:

$$Rate = \frac{\log(\|e_{h_1}\|_p / \|e_{h_2}\|_p)}{\log(h_1/h_2)}$$

where,  $p$  could be  $\infty$  or 2.  $\|e_h^u\|_2 = \sqrt{h \sum_{i=0}^M [U(x_i, t_N) - u(x_i, t_N)]^2}$ ,  $\|e_h^v\|_2 = \sqrt{h \sum_{i=0}^M [V(x_i, t_N) - v(x_i, t_N)]^2}$ ,  
 $\|e_h^u\|_\infty = \max_{0 \leq i \leq M} |U(x_i, t_N) - u(x_i, t_N)|$  and  $\|e_h^v\|_\infty = \max_{0 \leq i \leq M} |V(x_i, t_N) - v(x_i, t_N)|$  for 1D.  
 $\|e_h^u\|_2 = \sqrt{h^2 \sum_{i,j=0}^M [U(x_i, y_j, t_N) - u(x_i, y_j, t_N)]^2}$ ,  $\|e_h^v\|_2 = \sqrt{h^2 \sum_{i,j=0}^M [V(x_i, y_j, t_N) - v(x_i, y_j, t_N)]^2}$ ,  
 $\|e_h^u\|_\infty = \max_{0 \leq i,j \leq M} |U(x_i, y_j, t_N) - u(x_i, y_j, t_N)|$  and  $\|e_h^v\|_\infty = \max_{0 \leq i,j \leq M} |V(x_i, y_j, t_N) - v(x_i, y_j, t_N)|$  for 2D.  
 $u(x_i, t_N)$ ,  $v(x_i, t_N)$ ,  $u(x_i, y_j, t_N)$  and  $v(x_i, y_j, t_N)$  represent the numerical solutions at points  $(x_i, t_N)$  and  $(x_i, y_j, t_N)$ ,  $U(x_i, t_N)$ ,  $V(x_i, t_N)$ ,  $U(x_i, y_j, t_N)$  and  $V(x_i, y_j, t_N)$  represent the exact solutions at points  $(x_i, t_N)$  and  $(x_i, y_j, t_N)$ .

#### 4.1. Problem 1 [25]

First, we consider the 1D K-S chemotaxis model as follows:

$$\begin{cases} u_t = D(u)_x u_x + D(u) u_{xx} - \chi u_x v_x - \chi u v_{xx} + \delta u(1 - u) + s(x, t), \\ v_t = v_{xx} + u_{xx} + u - v + g(x, t), \end{cases}$$

where

$$D(u) = 1 + u, \quad g(x, t) = -2e^{-t+x} - 2(1 + 2x^2)e^{-t+x^2},$$

and

$$s(x, t) = -3e^{-t+x} - e^{-2t+2x} + 2\chi(1 + x + 2x^2)e^{-2t+x+x^2},$$

with the initial conditions

$$u(x, 0) = e^x, \quad v(x, 0) = e^{x^2},$$

and boundary conditions

$$u(0, t) = e^{-t}, \quad u(1, t) = e^{-t+1}, \quad v(0, t) = e^{-t}, \quad v(1, t) = e^{-t+1}.$$

The exact solutions are  $u(x, t) = e^{-t+x}$  and  $v(x, t) = e^{-t+x^2}$ .

**Table 1.**  $\|e_h^u\|_2$ ,  $\|e_h^u\|_\infty$  and convergence rates of  $u$  for Problem 1.

$N$	Ref. [25]				Present				
	$\ e_h^u\ _2$	Rate	$\ e_h^u\ _\infty$	Rate	$\ e_h^u\ _2$	Rate	$\ e_h^u\ _\infty$	Rate	CPU
10	6.714E-05		2.595E-04		3.958E-07		5.520E-07		0.124
20	1.138E-05	2.56	6.563E-05	1.98	6.816E-08	2.53	9.349E-08	2.56	0.843
25	6.444E-06	2.55	4.200E-05	2.00	3.028E-08	3.64	4.129E-08	3.66	1.609
40	1.901E-06	2.60	1.592E-05	2.06	4.987E-09	3.84	6.772E-09	3.85	6.123
50	9.797E-07	2.97	9.239E-06	2.44	2.073E-09	3.93	2.812E-09	3.94	11.481
100	1.647E-07	2.58	2.221E-06	2.07	1.323E-10	3.97	1.771E-10	3.99	93.259

**Table 2.**  $\|e_h^v\|_2$ ,  $\|e_h^v\|_\infty$  and convergence rates of  $v$  for Problem 1.

$N$	Ref. [25]				Present				
	$\ e_h^v\ _2$	Rate	$\ e_h^v\ _\infty$	Rate	$\ e_h^v\ _2$	Rate	$\ e_h^v\ _\infty$	Rate	CPU
10	5.169E-05		2.028E-03		4.500E-06		8.622E-06		0.124
20	8.713E-05	2.57	5.102E-04	1.99	5.622E-07	3.00	7.943E-07	3.44	0.843
25	4.940E-06	2.54	3.273E-04	1.99	2.531E-07	3.57	3.547E-07	3.61	1.609
40	1.505E-06	2.51	1.283E-04	1.99	4.216E-08	3.81	5.869E-08	3.83	6.123
50	8.607E-07	2.50	8.241E-05	1.98	1.755E-08	3.92	2.442E-08	3.93	11.481
100	1.509E-07	2.52	2.064E-05	2.00	1.115E-10	3.98	1.549E-09	3.99	93.259

To verify the accuracy of the 1D scheme, i.e., Eqs (2.14) and (2.15), we chose an example with an exact solution. Table 1 and Table 2 show the  $\|e_h\|_2$  and  $\|e_h\|_\infty$  values of numerical solutions  $u$  and  $v$  using the scheme proposed in this paper and comparisons with the results in [25] with different values of  $N$  in 1D, with  $\tau = 0.1h^2$ ,  $Tol = 10^{-12}$  for Problem 1. We have used the same problem parameters as [25], i.e.,  $T = 1$ ,  $\delta = \chi = 1$ ,  $\Omega = [0, 1]$  for comparison. From data in Tables 1 and 2, we can see that the convergence order of the difference scheme in this paper reaches the fourth-order in space, which is higher than the method in [25]. To satisfy the stability conditions of the scheme in this paper, we have used  $\tau=0.1h^2$  and in [25], it was  $\tau = 10^{-6}$ . It means that we used a larger time step size than that used in [25]. The advantage of our method is that we calculate fewer layers of time to reach the same time moment. Therefore, the present compact method can accurately predict the solution of the 1D K-S model.

#### 4.2. Problem 2 [25]

Then, consider the following 2D K-S chemotaxis model:

$$\begin{cases} u_t = D(u)_x u_x + D(u)u_{xx} + D(u)_y u_y + D(u)u_{yy} - \chi u_x v_x \\ \quad - \chi u v_{xx} - \chi u_y v_y - \chi u v_{yy} + \delta u(1 - u) + s(x, y, t), \\ v_t = v_{xx} + v_{yy} + u_{xx} + u_{yy} + u - v + g(x, y, t), \end{cases}$$

where

$$D(u) = 1 + u, \quad g(x, y, t) = -3e^{-t+x+y} - 4(1 + x^2 + y^2)e^{-t+x^2+y^2},$$

and

$$s(x, y, t) = -4e^{-t+x+y} - 3e^{(-2t+2x^2+2y^2)} + 2\chi(2 + x + y + 2x^2 + 2y^2)e^{-t+x+y}e^{-t+x^2+y^2},$$

with the initial conditions

$$u(x, y, 0) = e^{x+y}, \quad v(x, y, 0) = e^{x^2+y^2},$$

and boundary conditions

$$\begin{aligned} u(0, y, t) &= e^{-t+y}, & u(1, y, t) &= e^{-t+1+y}, & u(x, 0, t) &= e^{-t+x}, & u(x, 1, t) &= e^{-t+x+1}, \\ v(0, y, t) &= e^{-t+y^2}, & v(1, y, t) &= e^{-t+1+y^2}, & v(x, 0, t) &= e^{-t+x^2}, & v(x, 1, t) &= e^{-t+x^2+1}. \end{aligned}$$

The exact solutions are  $u(x, y, t) = e^{-t+x+y}$  and  $v(x, y, t) = e^{-t+x^2+y^2}$ .

**Table 3.**  $\|e_h^u\|_2$ ,  $\|e_h^u\|_\infty$  and convergence rates of  $u$  for Problem 2.

$N$	Ref. [25]				Present				
	$\ e_h^u\ _2$	Rate	$\ e_h^u\ _\infty$	Rate	$\ e_h^u\ _2$	Rate	$\ e_h^u\ _\infty$	Rate	CPU
20	3.231E-06		1.115E-04		1.019E-07		1.864E-07		34.616
25	1.470E-06	3.53	6.639E-05	2.32	4.521E-08	3.64	8.199E-08	3.68	86.075
40	2.337E-07	3.91	2.287E-05	2.27	7.443E-09	3.84	1.338E-08	3.86	593.237
50	1.82E-07	1.12	1.66E-05	1.43	3.094E-09	3.93	5.552E-09	3.94	1458.047

**Table 4.**  $\|e_h^v\|_2$ ,  $\|e_h^v\|_\infty$  and convergence rates of  $v$  for Problem 2.

$N$	Ref. [25]				Present				
	$\ e_h^v\ _2$	Rate	$\ e_h^v\ _\infty$	Rate	$\ e_h^v\ _2$	Rate	$\ e_h^v\ _\infty$	Rate	CPU
20	2.734E-05		9.178E-04		7.356E-07		1.431E-06		34.616
25	1.385E-05	3.04	5.922E-04	1.96	3.329E-07	3.55	6.385E-07	3.62	86.075
40	3.397E-06	2.99	2.377E-04	1.94	5.577E-08	3.80	1.061E-08	3.82	593.237
50	1.64E-06	3.26	1.48E-04	2.12	2.325E-08	3.92	4.412E-08	3.93	1458.047

Similarly, to test the accuracy of the scheme for the 2D cases, we also chose an example with an exact solution in [25]. Tables 3 and 4 list the  $\|e_h\|_2$ ,  $\|e_h\|_\infty$  and convergence orders given by the scheme of [25] and the presented scheme. We set  $Tol = 10^{-11}$ , and the time step  $\tau = 0.1h^2$ . We fix the values of the problem parameters same as that in [25]:  $T = 1$ ,  $\delta = \chi = 1$ . From Tables 3 and 4, we can clearly see that the proposed scheme achieves approximately fourth-order convergence in space. By comparing the results in Tables 3 and 4, we find the computational results in this paper are more accurate than those in [25], both in the case of  $\|e_h\|_2$  and in the case of  $\|e_h\|_\infty$ . In addition, we take  $\tau = 0.1h^2$  to satisfy the stability conditions of the scheme in this paper, while  $\tau = 10^{-7}$  was used in [25]. Namely, we use a larger time step size than that in the [25], but the numerical results we calculated at the same  $N$  are better than those in [25].

#### 4.3. Problem 3 [25]

Next, we consider the following 2D K-S chemotaxis model with a logistic source.

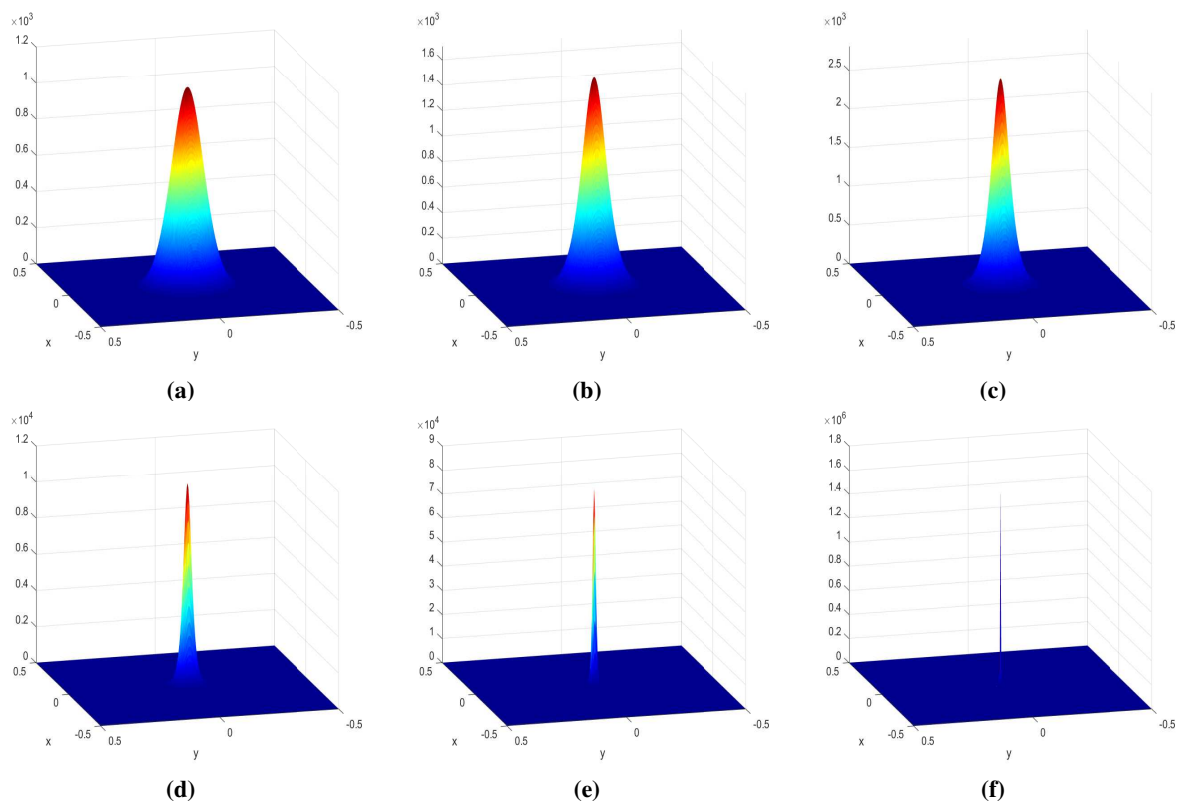
$$\begin{cases} u_t = u_{xx} + u_{yy} - u_x v_x - u v_{xx} - u_y v_y - u v_{yy} + \delta u(1 - u), \\ v_t = v_{xx} + v_{yy} + u - v. \end{cases}$$

The initial values of this problem are

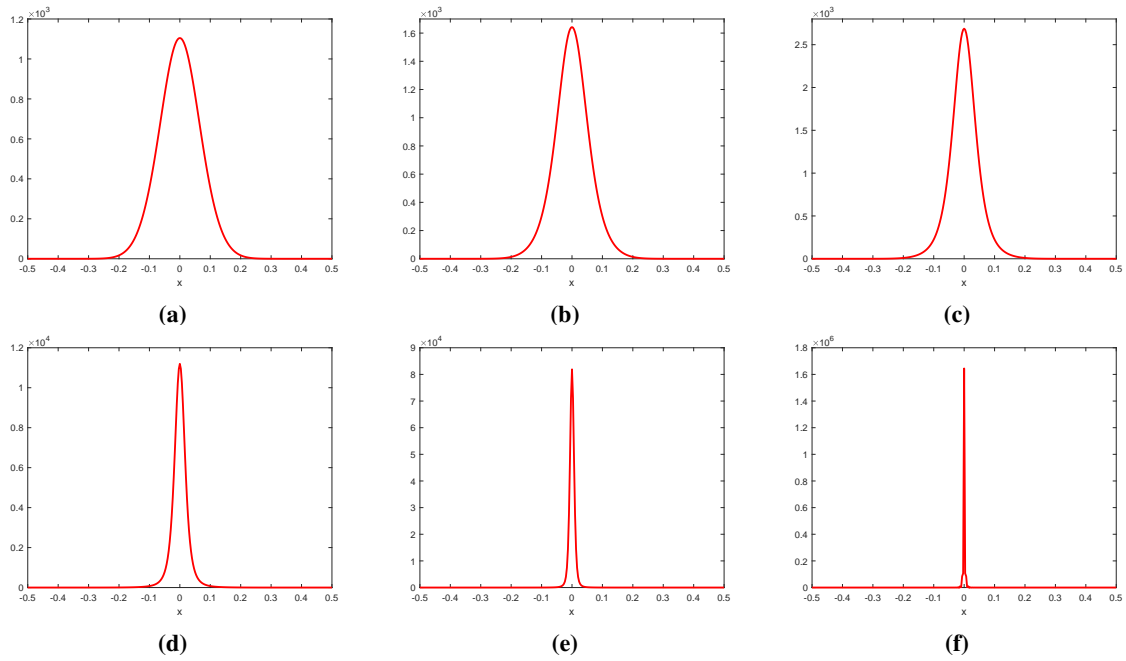
$$u(x, y, 0) = 1000e^{-100(x^2+y^2)}, \quad v(x, y, 0) = 500e^{-50(x^2+y^2)},$$

with homogeneous Neumann boundary conditions.

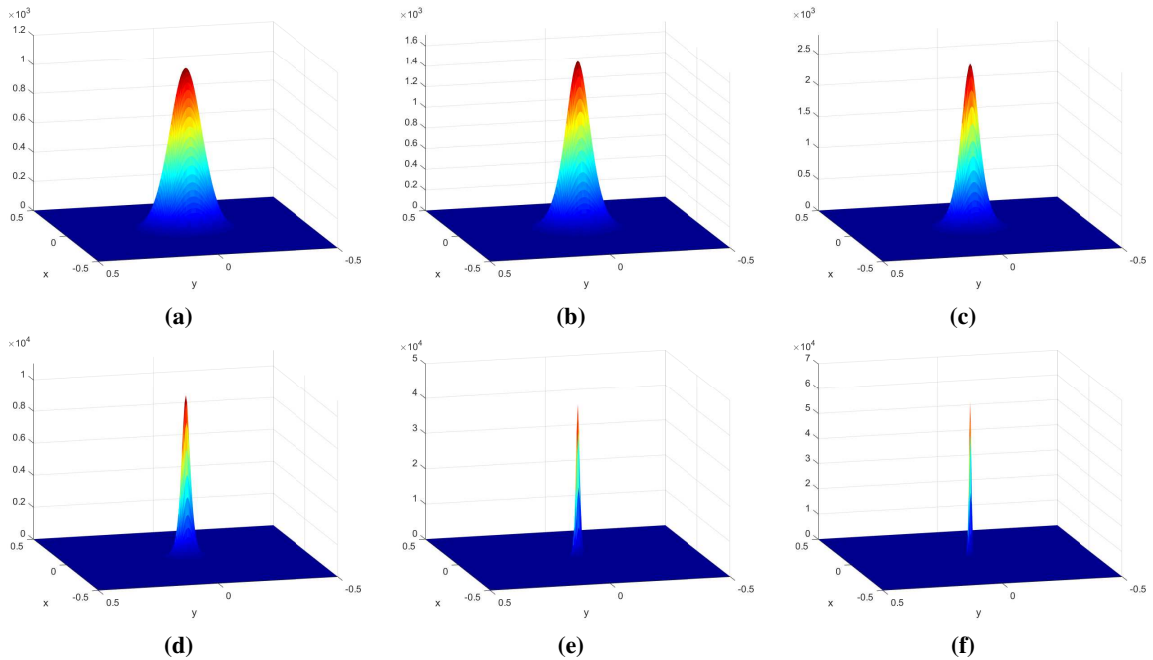
To further verify the effectiveness of the proposed finite difference method for 2D K-S chemotaxis models, we used  $(x, y) \in [-1/2, 1/2]$ ,  $\tau = 10^{-7}$ ,  $Tol = 10^{-12}$  for Problem 3. For this problem, the exact solution is not available. Reference [14] proved in theory that the solutions would blow-up in a finite time, but the blow-up solution is difficult to capture. The computed results were obtained by the presented difference schemes and compared with those in [25]. The approximated solutions of the cell density  $u$  at  $N = 300$ ,  $T = 10^{-6}$ ,  $5 \times 10^{-6}$ ,  $10^{-5}$ ,  $2.5 \times 10^{-5}$ ,  $5 \times 10^{-5}$  and  $8 \times 10^{-5}$  are plotted in Figure 1. The corresponding slices along  $y = 0$  of approximated cell density  $u$  are plotted in Figure 2. We find that as time increases the phenomenon of the peak becomes more and more obvious. We observe that the blow-up time is the same as the conjecture in [25].



**Figure 1.** The numerical solution of the cell density  $u_{i,j}^n$  for  $N = 300$  at (a)  $T = 10^{-6}$ , (b)  $T = 5 \times 10^{-6}$ , (c)  $T = 10^{-5}$ , (d)  $T = 2.5 \times 10^{-5}$ , (e)  $T = 5 \times 10^{-5}$ , (f)  $T = 8 \times 10^{-5}$ .



**Figure 2.** The 1D slice along  $y = 0$  of the numerical solution of the cell density  $u_i^n$  for  $N = 300$  at (a)  $T = 10^{-6}$ , (b)  $T = 5 \times 10^{-6}$ , (c)  $T = 10^{-5}$ , (d)  $T = 2.5 \times 10^{-5}$ , (e)  $T = 5 \times 10^{-5}$ , (f)  $T = 8 \times 10^{-5}$



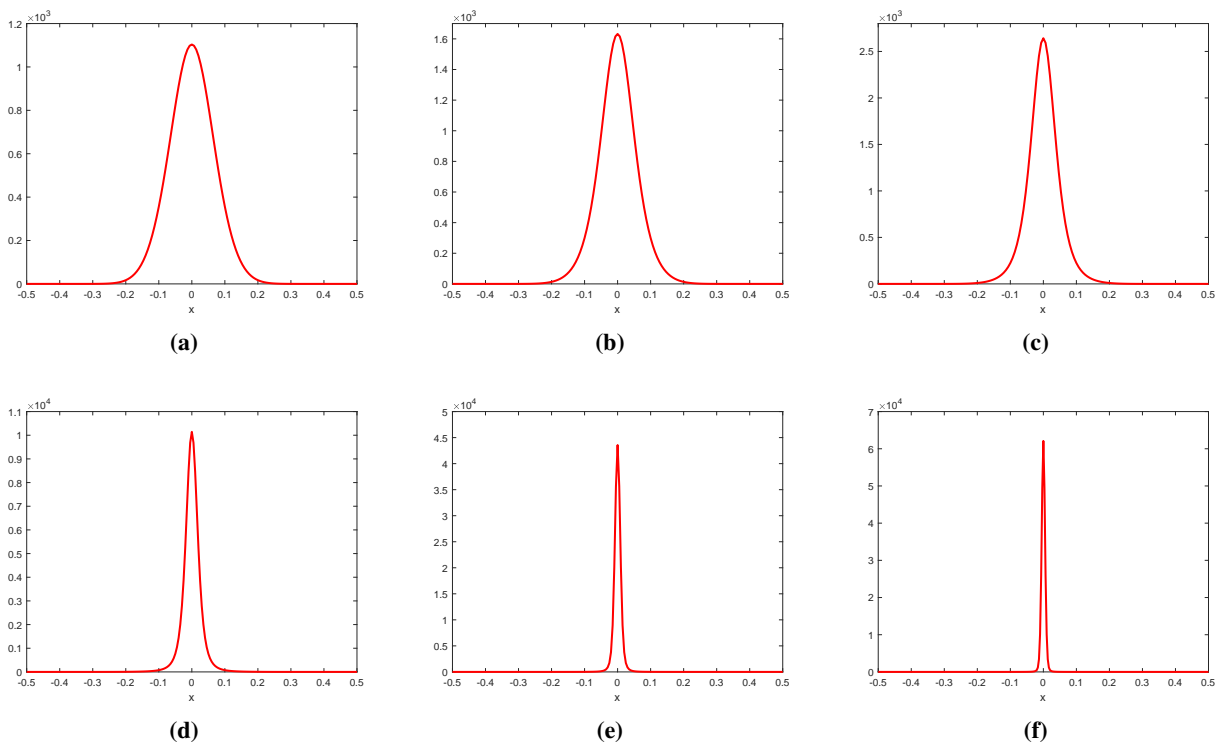
**Figure 3.** The numerical solution of the cell density  $u_{i,j}^n$  at (a)  $N = 150, T = 10^{-6}$ , (b)  $N = 150, T = 5 \times 10^{-6}$ , (c)  $N = 150, T = 10^{-5}$ , (d)  $N = 200, T = 2.5 \times 10^{-5}$ , (e)  $N = 200, T = 5 \times 10^{-5}$ , (f)  $N = 300, T = 8 \times 10^{-5}$



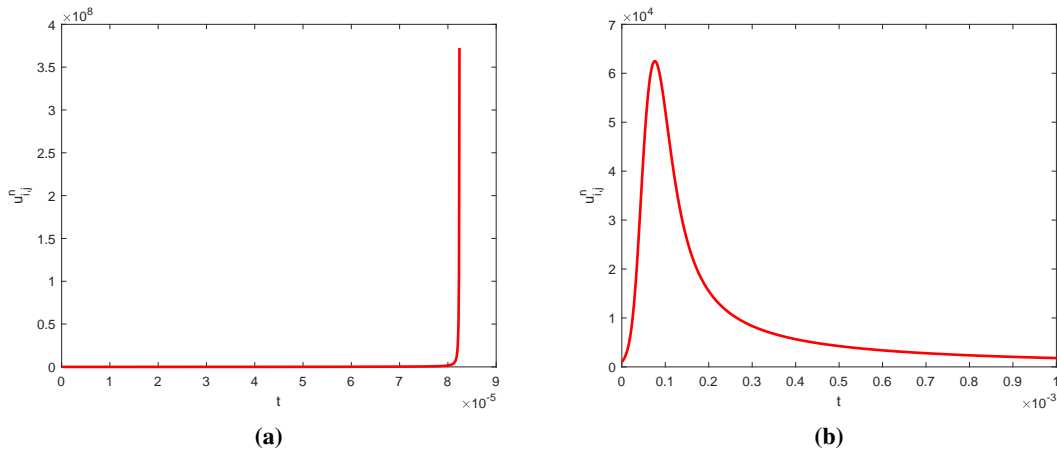
The approximated solution of the cell density  $u$  at  $N = 150, T = 10^{-6}$ ,  $N = 150, T = 5 \times 10^{-6}$ ,  $N = 150, T = 10^{-5}$ ,  $N = 200, T = 2.5 \times 10^{-5}$ ,  $N = 200, T = 5 \times 10^{-5}$  and  $N = 300, T = 8 \times 10^{-5}$  are plotted in Figure 3. The corresponding slices along  $y = 0$  of the approximation of the cell density  $u$  are plotted Figure 4.

Through the experiment, we find that a negative cell density  $u$  may occur for different values of  $N$  at the same time, which is unreasonable. In order to preserve the positivity of the cell density  $u$ , for time smaller than  $T = 5 \times 10^{-5}$ , we applied  $N = 150$ , and for time values larger than  $T = 5 \times 10^{-5}$ , we applied  $N = 300$ . It has a peak in the center of the domain  $\Omega$ , where the blow-up of the cell density  $u$  occurs in finite time. As the number of mesh points increases, the change in the accuracy of the solution becomes not significant [25].

By comparison of  $\delta = 0$  and  $\delta = 1$ , we can clearly see from Figures 2d–f and Figures 4d–f that when  $\delta = 1$ , the value of the cell density  $u$  decreases significantly. For further comparison, variation of the maximum value of approximated cell density  $u$  with time  $t$  when  $\delta = 0$  and  $\delta = 1$  is plotted in Figure 5. We note that in Figure 5a, when  $\delta = 0$  the value of approximated cell density  $u$  at time  $8.24e - 05$  is  $3.72e + 08$ , the value at this moment is very large and the gradient around it varies considerably. However, in Figure 5b, the value of approximated cell density  $u$  is  $6.15e + 04$  compared to that with  $\delta = 0$ , which is significantly smaller. It is verified that the logical source term can avoid the blow-up, that is, the logical source term is important for the model. This point is consistent with the fourth-order method for the cells to avoid the blow-up problems mentioned in [17].



**Figure 4.** The 1D slice along  $y = 0$  of the numerical solution of cell density  $u_i^n$  at (a)  $N = 150, T = 10^{-6}$ , (b)  $N = 150, T = 5 \times 10^{-6}$ , (c)  $N = 150, T = 10^{-5}$ , (d)  $N = 200, T = 2.5 \times 10^{-5}$ , (e)  $N = 200, T = 5 \times 10^{-5}$ , (f)  $N = 300, T = 8 \times 10^{-5}$ .



**Figure 5.** Variation of the maximum value of the numerical solution of the cell density  $u_{i,j}^n$  with time  $t$ , (a)  $\delta = 0$ , (b)  $\delta = 1$ .

4.4. Problem 4 [1]

Finally, we consider the following 2D chemotaxis growth model for bacterial pattern formation [31] with homogeneous Neumann boundary conditions,

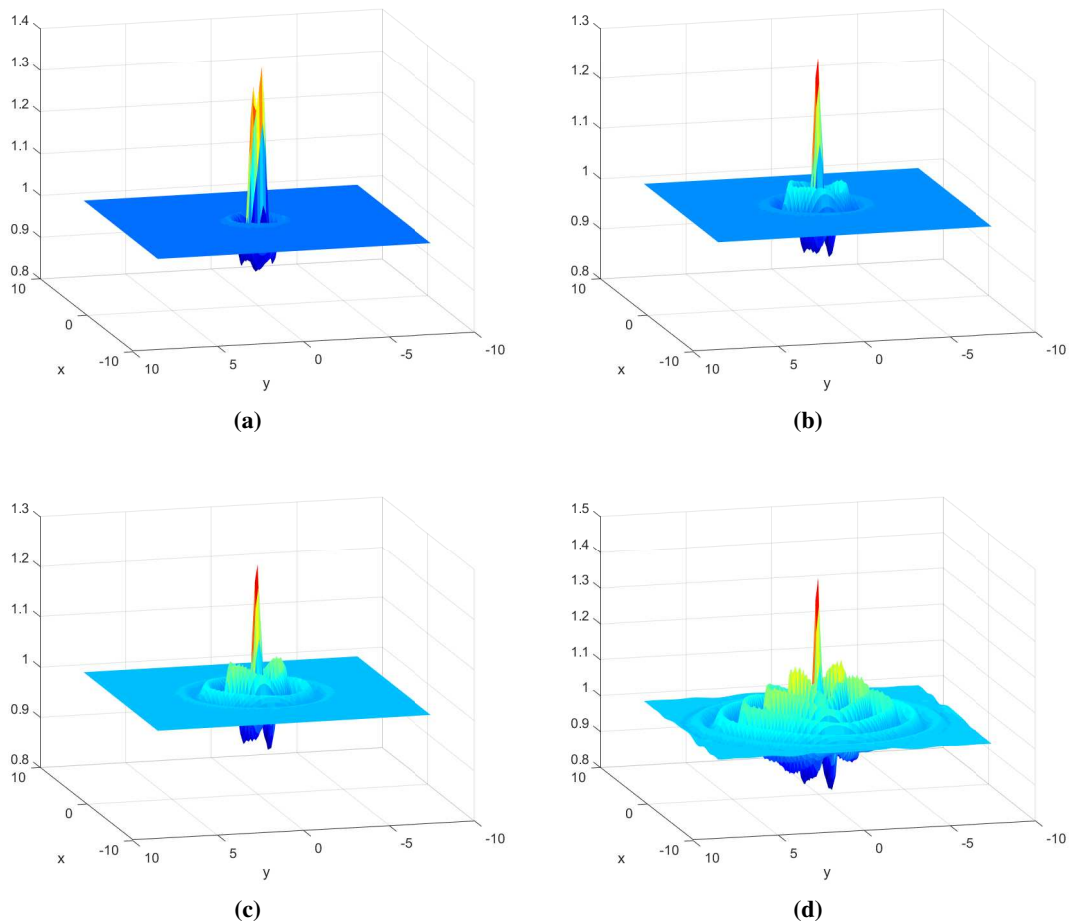
$$\begin{cases} u_t = Du_{xx} + Du_{yy} - \chi u_x v_x - \chi uv_{xx} - \chi u_y v_y - \chi uv_{yy} + 2u(1 - u), \\ v_t = v_{xx} + v_{yy} + u - \beta v. \end{cases}$$

with the initial values

$$u(x, y, 0) = \begin{cases} 1 + rand, & \text{if } \|x\|_2 < 0.7, \\ 1, & \text{otherwise,} \end{cases}$$

and  $v(x, y, 0) = 1/32$ .

The term  $2u(1 - u)$  is a logistic source representing the growth rate of  $u$ . For this model, the computational domain is the square  $\Omega = (-8, 8)^2$ , since the model is quite sensitive to the choice of the parameters, we applied the values of the parameters to be the same as in [1]:  $D = 0.0625$ ,  $\chi = 6$ ,  $\beta = 16$ . To understand the evolution of the solution, we take  $T = 1, 5, 10, 30$ , where  $rand$  is a random perturbation which is a constant between 0 and 1 on each cell of the mesh and  $\|x\|_2$  denotes the Euclidean distance to the center node  $(0, 0, 0)$  of the domain  $\Omega$ . The time step  $\tau = 10^{-5}$  was chosen. Figure 6 describes approximate solution of  $u$  for  $N = 100$  at different times. As we notice from Figure 6, the initial concentration of the cell  $u$  slowly expands from the center to the boundaries of the domain  $\Omega$  with a periodic form of continuous rings. The behavior of our calculated solution is consistent with that simulated in [1].



**Figure 6.** The numerical solution of cell density  $u_{i,j}^n$  for  $N = 100$  at (a)  $T = 1$ , (b)  $T = 5$ , (c)  $T = 10$ , (d)  $T = 30$ .

## 5. Conclusions

Based on the fourth-order Padé compact difference schemes for the spatial derivatives and the truncation error remainder correction method for the temporal derivative, we have proposed a highly accurate numerical scheme based on the finite difference method to solve the K-S chemotaxis system with self- and cross-diffusion terms and a logistic source in 1D and 2D. The local truncation errors of the schemes are  $O(\tau^2 + \tau h^2 + h^4)$ , which means that the schemes yield temporally second-order and spatially fourth-order accuracy. The accuracy and effectiveness of the 1D and 2D schemes have been verified by numerical examples. Moreover, we simulated the blow-up phenomenon and bacterial pattern formation of K-S chemotaxis model (1.1) by using the presented numerical method.

We note that positivity preserving nature of numerical methods for K-S chemotaxis models is a very important aspect, which was not taken into account in this paper. We will consider it in our ongoing work. Meanwhile, the three-dimensional (3D) K-S chemotaxis model has been numerically studied by some researchers [29, 30]. Zhao et al. [29] investigated the improved Petrov-Galerkin finite element

method defined on general static surfaces. A semi-implicit second-order scheme based on gradient and Laplace recoveries has been used to obtain a completely discrete form. Huang et al. [30] used the operator-splitting finite-element method combined with the flux-corrected transport algorithm to solve the 3D chemotaxis model. The method, which is used to solve several 1D sub-problems can keep the numerical solution non-negative. The scheme has second-order accuracy in the spatial direction. We are trying to extend the present method to the 3D K-S chemotaxis model. This work is in progress and the research results will be reported in the near future.

### Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

### Acknowledgments

This work was partially supported by the National Natural Science Foundation of China (12161067, 11961054), Natural Science Foundation of Ningxia (2022AAC02023), National Youth Top-notch Talent Support Program of Ningxia, and Graduate Innovation Program of Ningxia University (CXXM2022-12).

### Conflict of interest

The authors declare that there is no conflict of interest.

### References

1. M. Akhmouch, M. B. Amine, A time semi-exponentially fitted scheme for chemotaxis-growth models, *Calcolo*, **54** (2017), 609–641. <https://doi.org/10.1007/s10092-016-0201-4>
2. J. T. Bonner, M. E. Hoffman, Evidence for a substance responsible for the spacing pattern of aggregation and fruiting in the cellular slime molds, *J. Embryol. Exp. Morph*, **11** (1963), 571–589. <https://doi.org/10.1242/dev.11.3.571>
3. C. S. Patlak, Random walk with persistence and external bias, *Bull. Math. Biophys*, **15** (1953), 311–338. <https://doi.org/10.1007/BF02476407>
4. E. F. Keller, L. A. Segel, Initiation of slime mold aggregation viewed as an instability, *J. Theor. Biol*, **26** (1970), 399–415. [https://doi.org/10.1016/0022-5193\(70\)90092-5](https://doi.org/10.1016/0022-5193(70)90092-5)
5. E. F. Keller, L. A. Segel, Traveling bands of chemotactic bacteria: A Theoretical Analysis, *J. Theor. Biol*, **30** (1971), 235–248. [https://doi.org/10.1016/0022-5193\(71\)90051-8](https://doi.org/10.1016/0022-5193(71)90051-8)
6. E. F. Keller, L. A. Segel, Model for chemotaxis, *J. Theor. Biol*, **30** (1971), 225–234. [https://doi.org/10.1016/0022-5193\(71\)90050-6](https://doi.org/10.1016/0022-5193(71)90050-6)
7. J. Adler, Chemotaxis in bacteria, *Annu. Rev. Biochem*, **44** (1975), 341–356. <https://doi.org/10.1146/annurev.bi.44.070175.002013>
8. J. T. Bonner, *The cellular slime molds*, Princeton: Princeton University Press, 1959. <https://doi.org/10.1515/9781400876884>

9. E. O. Budrene, H. C. Berg, Complex patterns formed by motile cells of *Escherichia coli*, *Nature*, **349** (1991), 630–633. <https://doi.org/10.1038/349630a0>
10. S. Childress, J. K. Percus, Nonlinear aspects of chemotaxis, *Math. Biosci*, **56** (1981), 217–237. [https://doi.org/10.1016/0025-5564\(81\)90055-9](https://doi.org/10.1016/0025-5564(81)90055-9)
11. M. H. Cohen, A. Robertson, Wave propagation in the early stages of aggregation of cellular slime molds, *J. Theor. Biol*, **31** (1971), 101–118. [https://doi.org/10.1016/0022-5193\(71\)90124-X](https://doi.org/10.1016/0022-5193(71)90124-X)
12. B. Perthame, *Transport equations in biology*, Berlin: Springer Science & Business Media, 2007. <https://doi.org/10.1007/978-3-7643-7842-4>
13. M. A. Herrero, E. Medina, J. J. L. Velázquez, Finite-time aggregation into a single point in a reaction-diffusion system, *Nonlinearity*, **10** (1977), 1739–1754. <https://doi.org/10.1088/0951-7715/10/6/016>
14. M. A. Herrero, J. J. L. Velázquez, A blow-up mechanism for a chemotaxis model, *Ann. Scuola. Norm-Sci*, **24** (1997), 633–683.
15. N. Saito, Conservative upwind finite-element method for a simplified Keller-Segel system modelling chemotaxis, *IMA J. Numer. Anal*, **27** (2007), 332–365. <https://doi.org/10.1093/imanum/drl018>
16. A. Chertock, A. Kurganov, A second-order positivity preserving central-upwind scheme for chemotaxis and haptotaxis models, *Numer. Math*, **111** (2008), 457–488. <https://doi.org/10.1007/s00211-008-0188-0>
17. J. A. Carrillo, S. Hittmeir, A. Jüngel, Cross diffusion and nonlinear diffusion preventing blow up in the Keller-Segel model, *Math. Models Methods Appl. Sci*, **22** (2011), 1–35. <https://doi.org/10.1142/S0218202512500418>
18. M. Sulman, T. Nguyen, A Positivity preserving moving mesh finite element method for the Keller-Segel Chemotaxis Model, *J. Sci. Comput*, **80** (2019), 1–18. <https://doi.org/10.1007/s10915-019-00951-0>
19. C. Qiu, Q. Liu, J. Yan, Third order positivity-preserving direct discontinuous Galerkin method with interface correction for chemotaxis Keller-Segel equations, *J. Comput. Phys*, **433** (2021), 110–191. <https://doi.org/10.1016/j.jcp.2021.110191>
20. S. Borsche, S. Göttlich, A. Klar, P. Schillen, The scalar Keller-Segel model on networks, *Math Mod Meth Appls*, **24** (2014), 221–247. <https://doi.org/10.1142/S0218202513400071>
21. J. Shen, J. Xu, Unconditionally bound preserving and energy dissipative schemes for a class of Keller-Segel equations, *SIAM J. Numer. Anal*, **58** (2020), 1674–1695. <https://doi.org/10.1137/19M1246705>
22. E. C. Braun, G. Bretti, R. Natalini, Mass-preserving approximation of a chemotaxis multi-domain transmission model for microfluidic chips, *Mathematics*, **9** (2021), 688. <https://doi.org/10.3390/math9060688>
23. X. Xiao, X. Feng, Y. He, Numerical simulations for the chemotaxis models on surfaces via a novel characteristic finite element method, *Comput. Math. Appl*, **78** (2019), 20–34. <https://doi.org/10.1016/j.camwa.2019.02.004>

24. L. Zhang, Y. Ge, Z. Wang, Positivity-preserving high-order compact difference method for the Keller-Segel chemotaxis model, *Math. Biosci. Eng.*, **19** (2020), 6764–6794. <https://doi.org/10.3934/mbe.2022319>
25. S. M. Hassan, A. J. Harfash, Finite element approximation of a Keller-Segel model with additional self- and cross-diffusion terms and a logistic source, *Commun. Nonlinear Sci. Numer. Simul.*, **104** (2022), 106063. <https://doi.org/10.1016/j.cnsns.2021.106063>
26. S. K. Lele, Compact finite difference schemes with spectral-like resolution, *J. Comput. Phys.*, **103** (1992), 16–42. [https://doi.org/10.1016/0021-9991\(92\)90324-R](https://doi.org/10.1016/0021-9991(92)90324-R)
27. T. Wang, T. Liu, A consistent fourth-order compact scheme for solving convection-diffusion equation, *Math. Numer. Sin.*, **38** (2016), 391–403. <https://doi.org/10.12286/jssx.2016.4.391>
28. S. Hittmeir, A. Jüngel, Cross-diffusion preventing blow up in the two-dimensional Keller-Segel model, *SIAM J. Math. Anal.*, **43** (2011), 997–1022. <https://doi.org/10.1137/100813191>
29. S. Zhao, X. Xiao, J. Zhao, X. Feng, A Petrov-Galerkin finite element method for simulating chemotaxis models on stationary surfaces, *Comput. Math. Appl.*, **79** (2020), 3189–3205. <https://doi.org/10.1016/j.camwa.2020.01.019>
30. X. Huang, X. Xiao, J. Zhao, X. Feng, An efficient operator-splitting FEM-FCT algorithm for 3D chemotaxis models, *Comput. Math. Appl.*, **36** (2020), 1393–1404. <https://doi.org/10.1007/s00366-019-00771-8>
31. M. Aida, T. Tsujikawa, M. Efendiev, A. Yagi, M. Mimura, Lower Estimate of the Attractor Dimension for a Chemotaxis Growth System, *J. Lond. Math. Soc.*, **74** (2006), 453–474. <https://doi.org/10.1112/S0024610706023015>



AIMS Press

©2023 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>)