Journal of Computational Physics 466 (2022) 111421

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

Provable convergence of blow-up time of numerical approximations for a class of convection-diffusion equations ☆

Hui Guo [a], Xueting Liang [a], Yang Yang [b,*]

[a] College of Science, China University of Petroleum, Qingdao 266580, China
[b] Department of Mathematical Sciences, Michigan Technological University, Houghton, MI 49931, United States of America

ARTICLE INFO

Article history:
Received 6 February 2022
Received in revised form 31 May 2022
Accepted 23 June 2022
Available online 30 June 2022

Keywords:
Blow-up time
Blow-up solutions
Convection-diffusion equations
Positivity-preserving
L^1-stability

ABSTRACT

In this paper, we investigate the numerical algorithms to capture the blow-up time for a class of convection-diffusion equations with blow-up solutions. The numerical methods for such equations may not be straightforward to construct due to the lack of stability. Moreover, the blow-up time is more difficult to capture since we cannot distinguish whether the blow-up is physical or is due to the instability of the numerical methods. In this paper, we consider a class of convection-diffusion equations with positive blow-up solutions and the blow-up is due to the formation of δ-singularities. We use the positivity-preserving technique to enforce the L^1-stability and the L^2-norm of the numerical approximations to detect the blow-up phenomenon. We propose two ways to define the numerical blow-up time and prove their convergence to the exact one. As an application, we extend this method to calculate when the shock appears for scalar hyperbolic conservation laws. Three model problems will be discussed and tested to confirm the convergence numerically. Finally, the method can also be used to test whether an equation has a blow-up solution in finite time.

1. Introduction

For certain convection-diffusion partial differential equations the exact solutions may become unbounded during time evolution. The time T at which infinity appears in the exact solution is called the blow-up time. Blow-up theory is one of the most important contents in the study of partial differential equations and it has many applications in astrophysics, chemistry and environmental engineering, etc. It mainly describes the problems of heat accumulation and material advection. There are two basic mechanisms to trigger the blow-up for convection-diffusion equations: the superlinear growth of the source and the strong advection of the flow. One of the most interesting and challenging topics in the blow-up theory is to capture the blow-up time.

The semi-linear heat equation is one of the most significant equations that yields blow-up solutions. In 1963, Kaplan [30] first proposed some sufficient conditions for the blow-up phenomenon. In fact, if the source is superlinear, then the exact solution will grow up to infinity during time evolution. To capture the blow-up time, most theoretical works use the method of upper and lower solutions to find an interval containing the exact blow-up time, e.g. [30,17,18]. However,

☆ The first and second authors were supported by the Fundamental Research Funds for the Central Universities 20CX05011A and the Major Scientific and Technological Projects of CNPC under Grant ZD2019-183-008. The third author was supported by NSF grant DMS-1818467.

* Corresponding author.
E-mail addresses: sdugh@163.com (H. Guo), liangxt97@163.com (X. Liang), yyang7@mtu.edu (Y. Yang).

https://doi.org/10.1016/j.jcp.2022.111421
0021-9991/© 2022 Elsevier Inc. All rights reserved.

it is not straightforward to calculate the exact blow-up time directly. As an alternative, numerical methods can be used to approximate the exact blow-up time. There are many works discussing the numerical blow-up time in the literature. In 1975, Nakagawa [36] considered a finite difference (FD) discretization of the semi-linear heat equation. The basic idea is to define a sequence of numerical blow-up time and prove the convergence to the exact one during mesh refinement. This seems to be the first algorithm that can compute the blow-up time numerically. Later, the idea was generalized in [1,4,9–11,22]. Moreover, finite element methods [24,38,37] were also proposed, yet no theoretical support was available.

Besides the above, some convection-diffusion equations can also yield blow-up solutions, such as the pressureless Euler equation and the chemotaxis model, etc. These models are used to describe some transport phenomenon, and the density functions are always nonnegative. Different from the semilinear heat equations that the blow-up is due to the source, such models yield δ-singularities during finite time due to the strong advection.

The system of pressureless Euler equations reads

$$\begin{aligned} &\rho_t + \nabla\cdot(\rho u) = 0, \\ &(\rho u)_t + \nabla\cdot(\rho u \otimes u) = 0, \end{aligned} \tag{1.1}$$

where ρ is the density function and u is the velocity. System (1.1) is a weakly hyperbolic system and both eigenvalues of the Jacobian are u. The model can be viewed as a simplification of the corresponding usual system of Euler equations when the effects of pressure are neglected. The system may also arise modeling the collision of sticky particles, and if the particles are stuck together, then δ-functions appear in the density. During the last three decades, there were many theoretical contributions to the system, e.g. [5,16,26,47]. Moreover, numerical methods were also studied by several authors [6,7,21,49]. In [7], the authors added an artificial viscosity and built a diffusive scheme. In [6], Berthon et al. investigated a relaxation scheme for the pressureless gasses system. Gosse and James [21] analyzed the upwind schemes and the Lax-Friedrichs schemes. In [49], the high-order discontinuous Galerkin (DG) methods with the bound-preserving technique [50] were proposed to obtain physically relevant numerical approximations.

Another commonly used model that yields blow-up solutions in finite time is the following Keller-Segel (KS) chemotaxis model [34]

$$\begin{aligned} &u_t - div(\nabla u - \chi u \nabla v) = 0, \\ &v_t - \Delta v = u - v. \end{aligned} \tag{1.2}$$

This model represents the evolution of a cell density $u(\cdot,t)$ in the presence of a chemical substrate $v(\cdot,t)$. The boundary condition is set to be homogeneous Neumann boundary condition

$$\nabla u \cdot \mathbf{n} = \nabla v \cdot \mathbf{n} = 0, \tag{1.3}$$

where $\mathbf{n}$ is the outer normal of the boundary $\partial\Omega$. The theory and mathematical model of chemotaxis can be traced back to the pioneering work of Patlak [39] in the 1950s and Keller and Segel [31–33] in the 1970s. Chemotaxis is the highly nonlinear terminology which indicates movements by cells in reaction to a chemical substance, where cells approach chemically favorable environments and avoid unpleasant ones [34]. One of the most significant properties of chemotaxis behavior is the ability to display cell aggregation [3]. This phenomenon has shown to result in finite time blow-up under certain initial conditions for problems in two and three space dimensions [15,19,25,35], while it has proved that no blow-up occurs for 1D problems [35]. Thus, capturing the blow-up behavior of the numerical solution is an interesting and challenging task since we have to test whether a spike in the numerical approximation is a δ-function or not. Some existing numerical methods for solving chemotaxis equations provide high-resolution and positivity-preserving schemes, including finite volume methods [2,15], finite element methods [40,43], flux-corrected finite element methods [44], and DG methods [12,13,34,23,14].

There are not too many works discussing the blow-up time of the chemotaxis equations. Budd [8] used a high-order moving mesh method to obtain a careful resolution of the collapse behavior. The blow-up time T was estimated to high precision by computing until $u(0,t^*) \approx 10^{21}$ and approximating T by t^*, which means that the blow-up occurs when the numerical solution is large enough. Li [34] numerically demonstrated how to find the approximate blow-up time by using the L^2-norm of the numerical solutions. However, the above works do not have theoretical evidence to support the results.

In this paper, we consider general convection-diffusion equations with blow-up solutions. We assume the exact solution to be positive and the L^q $(q > 1)$ norm of the exact solution becomes unbounded during finite time. To capture the exact blow-up time is not an easy task since we have no idea about the blow-up rate, blow-up set, etc, as this information may not be available. The only information we can use is the error estimates, and this estimate fails to work if the time is close to the blow-up time. Therefore, the numerical approximations are not reliable when the blow-up is about to appear. In this paper, we will extend upon [34] and define a special numerical blow-up time for general convection-diffusion equations with blow-up solutions, and theoretically prove its convergence to the exact blow-up time during mesh refinement under some reasonable assumptions. To construct each numerical blow-up time in the sequence, we first apply the positivity-preserving technique to the proposed scheme to guarantee the L^1 stability of the numerical approximations. Then we compute the numerical approximations on the meshes with different refinement levels $n = 1, 2, 3$, etc. Subsequently, we trace the L^q $(q > 1)$ norms of the numerical solutions with respect to time. If the terminal time t is small, the exact solution is smooth, then the L^q-norms of the numerical approximations should be almost identical thanks to the error estimates. On the other hand, if t is large, the exact solution may contain large gradients. Then a sufficiently refined mesh is necessary to resolve the problem. Therefore, the L^q-norms of the numerical approximations under two different resolutions may deviate at some time $t^\star$. Then the numerical blow-up time for the numerical approximation on the coarse mesh is expected to be $t^\star$. Unfortunately, it is impossible to calculate $t^\star$ exactly since it is not straightforward to predict the difference of the L^q-norms of the numerical approximations. Therefore, we define the numerical blow-up time, denoted as η_n, as the smallest time that the difference between the two L^q-norms is equal to some threshold c_n. It is easy to see that the smaller the c_n, the smaller the η_n. If c_n is too small, say c_n is less than the initial error between the numerical and exact solutions, then $\eta_n = 0$, and this is definitely not the numerical blow-up time we expected. On the other hand, if c_n is too large, η_n may be larger than T. Numerical experiments also verify the facts given above. Therefore, we have to choose the threshold c_n carefully. In this paper, we will propose two ways to choose c_n. The basic idea is to consider the series $\sum_{n=1}^{\infty} c_n$. We will show that if the series $\sum_{n=1}^{\infty} c_n$ is convergent, i.e. the threshold is not too large, then $\liminf\limits_{n\to\infty} \eta_n \le T$. To demonstrate the other direction of the inequality, we choose the decay rate of c_n to be strictly less than the desired accuracy of the numerical scheme. Hence the numerical blow-up time cannot be the one when the solution is still smooth, i.e. the error estimate is still valid. Therefore, it is possible to use p-series $\sum_{n=1}^{\infty} \frac{1}{n^p}$ and geometric series $\sum_{n=1}^{\infty} \frac{1}{m^n}$ with suitable p and m for $\sum_{n=1}^{\infty} c_n$. Thanks to the selection of c_n, we have the convergence of a subsequence of the numerical blow-up time. Unfortunately, it is not straightforward to prove the convergence of the whole sequence, since we have no information about the blow-up rate, and we will discuss this in the future. Finally, the proposed blow-up time requires several levels of mesh refinement, and the computational cost is large if we do not apply suitable adaptive method, such as the moving mesh method given in [8]. However, the main target of this paper is to test the convergence of the proposed numerical blow-up time, not the good strategies for h-adaptivity. Therefore, we only use uniform mesh in all the numerical experiments. With suitable mesh adaptation, we can only increase the level of mesh refinement near the blow-up sets, leading to significantly reduced computational cost. We do not expect any difficulties in extending the algorithm to the numerical methods with mesh adaptations.

The above algorithm can be used to predict when the δ-function appears. If the exact solution may develop a discontinuity, such as nonlinear hyperbolic equations, the derivative of the numerical approximation can be considered as a δ-function. As an extension, we discuss the following Burgers' equation

$$u_t + (\frac{u^2}{2})_x = 0, \tag{1.4}$$

and use the algorithm proposed above to compute when the shock appears. Different from the procedure discussed above, the derivative of the numerical approximations may not be positive, and the positivity-preserving technique does not work. To obtain bounded derivatives, instead of pursuing the L^1 stability, we develop the total variation boundedness (TVB) of the numerical approximations. Thanks to the TVB technique [45,46], we can extend the proposed algorithm to compute when the shock appears by calculating the blow-up time of the derivative of the numerical solutions.

The organization of this paper is as follows. In Section 2, we demonstrate how to define the numerical blow-up time and its convergence to the exact one for general convection-diffusion equations with blow-up solutions. Some reasonable assumptions on the exact solutions and numerical approximations will be made. In Section 3, we discuss the pressureless Euler equations and the chemotaxis model in details and present how the assumptions made in Section 2 are satisfied. For problems with discontinuous solutions, we extend the algorithm to calculate when the shock appears in Sections 4. In Section 5, numerical experiments are given to demonstrate the effectiveness of the methods. Some concluding remarks are given in Section 6.

2. Blow-up time

In this section, we proceed to construct the sequence of numerical blow-up time and prove its convergence to the exact one T. For simplicity, we consider DG methods in this section. Actually the idea also works for finite volume and FD methods. We consider the problem in one space dimension, and denote Ω to be the computational domain with $\Omega_h = \{K\}$ being a uniform partition of Ω. The extension to problems in two space dimensions are similar. Suppose u is the exact solution which may yield δ-functions during time evolution and $u_h^n = u_h(x, t^n)$ is the piecewise polynomial approximation at time level $t = t^n$. Moreover, we denote C to be a generic positive constant that can take different values at different occurrences, but it must be independent of the mesh size h. First of all, we would like to make the following assumptions of the exact solution u and its numerical approximation u_h.

1. The exact solution u is non-negative and $\displaystyle\frac{d}{dt}\int_\Omega u(x,t)\ dx = 0$.
2. The exact solution u is smooth and bounded for $t < T$ and

$$\lim_{t\to T^-} \|u\|_{L^q} = \infty,$$

where $\|\cdot\|_{L^q}$ is the standard L^q-norm on the whole computational domain ($1 < q \le \infty$). For problems with δ-functions, we choose $q = 2$.

3. The numerical approximation $u_h \geq 0$ and $\int_\Omega u_h^n \, dx = \int_\Omega u_h^0 \, dx$, for all $n \geq 0$.
4. For any $\epsilon > 0$, there exists $H^\epsilon > 0$ such that for any $h < H^\epsilon$ and $t^n < T - \epsilon$, we have $\|u - u_h\|_{L^q} \leq Ch^{k+1}$ for some $k \geq 0$, where C does not depend on h but may depend on ϵ, H^ϵ, the regularity of the exact solution and time t. In general, for finite element methods, we choose k to be the polynomial degree used in the finite element space.

Remark 2.1. The Hypothesis 1 is satisfied by general convection-diffusion equations with homogeneous Neumann boundary conditions. Actually, for nonhomogeneous Neumann boundary conditions, we do not need the total mass to be a constant but bounded within the target time period. The Hypothesis 2 gives the particular blow-up phenomenon we are interested in and defines the exact blow-up time T. Hypothesis 3 requires the positivity-preserving technique and it yields the L^1 stability of the scheme. Actually, we can use other stability to substitute the L^1 stability. Finally, the Hypothesis 4 is the standard error estimate for general nonlinear convection-diffusion equations with smooth exact solutions. The four Hypotheses are the basic assumptions we need to study the blow-up phenomenon.

The above assumption is for fully discretized schemes. To obtain the numerical approximations between t^n and t^{n+1}, we may consider linear function interpolation and define $u_h(x,t) = \theta u_h^n + (1-\theta)u_h^{n+1}$ for $t^n \leq t \leq t^{n+1}$ with $\theta = (t^{n+1} - t)/(t^{n+1} - t^n)$. Then it is easy to check that $u_h(x,t)$ defined above satisfies the last two hypotheses for some k.

Next we proceed to define the numerical blow-up time. Notice that it is impossible to obtain the error estimates between the numerical and exact solutions under the L^q-norm near $t = T$ due to Hypothesis 2. Therefore, we need to construct a sequence of numerical blow-up time that converges to the exact one during mesh refinement. To construct each numerical blow-up time, we would like to use the information of two numerical approximations under different resolutions. We compute the L^q-norms of the numerical solutions with respect to time. Due to Hypothesis 4, the L^q-norms should be almost identical if t is small while we anticipate a significant difference if t is large. Therefore, we say the numerical blow-up for the approximation under the coarse mesh occurs if the difference between the L^q-norms of the two simulations is large enough. For simplicity, we consider uniform meshes, and the whole algorithm in one space dimension is given below.

1. Divide the computational domain into N_0 cells.
2. Compute the L^q-norm of the numerical approximation u_h with respect to time t, denoted as $S(N_0, t)$. In general, we take $q = 2$.
3. Suppose we have computed $S(N, t)$ then equally split each cell into two subcells, repeat the simulation on the new mesh and calculate $S(2N, t)$.
4. Let $N = N_0 2^n$ for some positive integer n, where n is the level of mesh refinement. Use $S(2N, t)$ as reference and calculate the numerical blow-up time η_n as

$$\eta_n = \inf\{t \geq t_\star : S(2N, t) \geq S(N, t) + D(n)\}, \tag{2.1}$$

   where $t_\star$ is some known lower bound of the blow-up time and $D(n) > 0$ is the threshold in defining the numerical blow-up time. If we do not have any information of the lower bound of the blow-up time, it is possible to choose $t_\star = 0$.
5. Repeat steps 3 and 4, until η_n does not change much during mesh refinement.

Remark 2.2. In step 4, $S(2N,t) \approx S(N,t)$ if t is small due to the error estimates, while we anticipate $S(2N,t) >> S(N,t)$ if t is close to the exact blow-up time T. To get (2.1), we can select the first time level, denoted as t^{m+1}, such that the inequality in (2.1) is satisfied. Then we anticipate $\eta_n \in [t^m, t^{m+1}]$. We can either compute the $S(N,t)$ and $S(2N,t)$ for all $t \in [t^m, t^{m+1}]$ and select the t such that the equal sign in (2.1) has achieved or simply take $\eta_n = t^{m+1}$. The former way is not difficult to compute due to the linear interpolation we used in the time interval and $S(N,t)$ and $S(2N,t)$ are both quadratic functions in t. In practice, we can simply choose the latter way, as the error made in computing η_n is less than Δt, the time step size, and this error does not affect the convergence as $\Delta t \to 0$ during mesh refinement.

Remark 2.3. In (2.1), we cannot prove the existence of η_n for all n. For simplicity of presentation, we assume $\eta_n = \infty$ if it does not exist, i.e. $S(2N,t) < S(N,t) + D(n)$ for all $t > 0$.

Before we prove the convergence of the numerical blow-up time η_n, we would like to demonstrate the following lemma.

Lemma 2.1. *Suppose the computational domain Ω is divided into N uniform cells and the Hypothesis 3 is satisfied, then there exists $C > 0$ such that $S(N,t) = \|u_h\|_{L^q} \leq Ch^{\frac{1}{q}-1}$, where $h = \frac{1}{N}$ and $1 \leq q \leq \infty$.*

Proof. By the norm equivalence in finite dimensional spaces, we have for $1 < q < \infty$

$$\|u_h\|_{L^q}^q = \sum_{K\in\Omega_h} \|u_h\|_{L^q(K)}^q \le C \sum_{K\in\Omega_h} h^{1-q}\|u_h\|_{L^1(K)}^q \le C \left(\sum_{K\in\Omega_h} h^{\frac{1}{q}-1}\|u_h\|_{L^1(K)}\right)^q,$$

then

$$\|u_h\|_{L^q} \le C \sum_{K\in\Omega_h} h^{\frac{1}{q}-1}\|u_h\|_{L^1(K)} = Ch^{\frac{1}{q}-1}\|u_h\|_{L^1}.$$

By Hypothesis 3, $\|u_h\|_{L^1} = \int_\Omega u_h \, dx$ is a constant that does not depend on t or h. The proof for $q=\infty$ is similar, so we omit it. □

For problems in d space dimensions, we can obtain similar results following the same lines. Therefore, we only state the lemma below without proof.

Lemma 2.2. *Suppose the computational domain is divided into N^d uniform hyperrectangular cells and the Hypothesis 3 is satisfied, then there exists $C>0$ such that $S(N,t) = \|u_h\|_{L^q} \le Ch^{\frac{d}{q}-d}$, where $h=\frac{1}{N}$.*

Remark 2.4. The above two lemmas also work for FD methods. Actually, let $\{x_j\}_{j=1}^N$ be the uniform distributed grid points with mesh size h and $\{u_j\}_{j=1}^N$ be the numerical approximation at the grid points. We denote $I_j = [x_j - \frac{h}{2}, x_j + \frac{h}{2}]$ to be the cells and it is easy to check $\cup I_j = \Omega$. Finally, we define $u_h(x) = u_j$ if $x \in I_j$ and $\|u_h\|_{L^q}^q = h\sum_{j=1}^N u_j^q$. Then we can follow the proof in Lemma 2.1 to obtain the conclusion for FD methods. For the rest of this section, we can also use these definitions to extend the results to FD methods.

To prove the convergence of the numerical blow-up time, we need two lemmas.

Lemma 2.3. *Suppose the hypotheses 1-4 are satisfied, and*

$$\sum_{n=1}^{\infty} D(n) < \infty, \tag{2.2}$$

then there exists a subsequence $\{\eta_{n_k}\} \subseteq \{\eta_n\}$ such that $\eta_n \le T$.

Proof. If false, then there exists $M>0$ such that $t_\ell > T$ for all $\ell \ge M$. By (2.1), we have

$$S(N_0 2^{\ell+1}, t) < S(N_0 2^\ell, t) + D(\ell), \quad \forall\, t_\star \le t \le T,\ \ell \ge M.$$

We sum up over ℓ to obtain

$$\sum_{\ell=M}^n S(N_0 2^{\ell+1}, t) < \sum_{\ell=M}^n S(N_0 2^\ell, t) + \sum_{\ell=M}^n D(\ell),$$

which further implies

$$S(N_0 2^n, t) < S(N_0 2^M, t) + \sum_{\ell=M}^{\infty} D(\ell), \quad \forall\, n > M.$$

By Lemma 2.1 with $h = \frac{1}{N_0 2^M}$, we have $S(N_0 2^M, t) \le C$. Using the fact that $\sum_{n=1}^{\infty} D(n) < \infty$, we can obtain $S(N_0 2^n, t) \le C$ for all $n > M$ and $t_\star \le t \le T$. We will show that this contradicts Hypothesis 2 and finish the proof. Actually, if Hypothesis 2 is still valid, then there exists $T_0 \ge t_\star$ such that $\|u(t)\|_{L^q} > 2C$, for all $t > T_0$. Given $t > T_0$, we take $n > M$ to be sufficiently large, with h being sufficiently small, such that $\|u - u_h\|_{L^q} \le C$ by Hypothesis 4. Then by triangle inequality, we have

$$2C < \|u(t)\|_{L^q} \le \|u_h\|_{L^q} + \|u - u_h\|_{L^q} = S(N_0 2^n, t) + \|u - u_h\|_{L^q} \le 2C,$$

which is a contradiction. Therefore, Hypothesis 2 is no longer satisfied.

By Hypothesis 4 and triangle inequality, we have $\|u(t)\|_{L^q} \le C$, for all $t_\star \le t < T$, which contradicts Hypothesis 2. □

Lemma 2.3 has a straightforward corollary.

Corollary 2.1. *Suppose the hypotheses 1-4 are satisfied, and the numerical blow-up time is defined as* (2.1) *with* $\sum_{n=1}^{\infty} D(n) < \infty$, *then*

$$\liminf_{n\to\infty} \eta_n \le T.$$

Now we proceed to discuss the other direction of the above inequality and the result is given below.

Lemma 2.4. *Suppose the hypotheses 1-4 are satisfied, and the numerical blow-up time is defined as* (2.1) *with*

$$D(n) \ge C 2^{-\tilde{k}n} \tag{2.3}$$

for some $\tilde{k} < k+1$ *if* n *is large, then*

$$\liminf_{n\to\infty} \eta_n \ge T.$$

Proof. If false, there exists $\epsilon > 0$ such that

$$\liminf_{n\to\infty} \eta_n = T - 2\epsilon.$$

Without loss of generality, we assume

$$\lim_{n\to\infty} \eta_n = T - 2\epsilon.$$

Then we have the following two facts:

1. There exists $M^\epsilon > 0$ such that for any $n > M^\epsilon$, we have $\eta_n < T - \epsilon$;
2. By hypotheses 2 and 4, there exists H^ϵ such that for any $h < H^\epsilon$ and $t < T - \epsilon$, we have $\|u - u_h\|_{L^q} \le Ch^{k+1}$.

Take n to be sufficiently large such that $n > M^\epsilon$ and $h = \frac{1}{N} = \frac{1}{N_0 2^n} < H^\epsilon$, we have

$$|S(2N,t) - S(N,t)| \le Ch^{k+1} = \frac{C}{2^{n(k+1)}}, \quad \forall\, t_\star \le t < T - \epsilon.$$

Here we can take $t \ge \eta_\star$ because $\eta_n \ge \eta_\star$ and hence $\eta_\star \le T - 2\epsilon < T - \epsilon$. By (2.1), we have

$$S(2N, \eta_n) - S(N, \eta_n) = D(n). \tag{2.4}$$

Therefore,

$$\frac{C}{2^{\tilde{k}n}} \le D(n) = S(2N,\eta_n) - S(N,\eta_n) \le |S(2N,\eta_n) - S(N,\eta_n)| \le \frac{C}{2^{n(k+1)}}, \tag{2.5}$$

which is a contradiction if n is large. □

Remark 2.5. We can use the idea given in Remark 2.2 to get the identity (2.4). In practice, it is not necessary to compute η_n exactly, and we can simply use the time level as the numerical blow-up time, as the error is within Δt, the time step size, and such an error does not affect the convergence as $\Delta t \to 0$.

Remark 2.6. Based on the proof given above, we have to choose $\tilde{k}$ to be as small as possible to observe contradiction for smaller n. Based on Corollary 2.1, we need to choose $\tilde{k} > 0$. Therefore, in practice, we need to choose $\tilde{k}$ to be a small positive constant. Numerical experiments in Section 5 also support such expectation.

Now, we can demonstrate the main theorem.

Theorem 2.1. *Suppose the conditions in Lemmas 2.3 and 2.4 are satisfied, then*

$$\liminf_{n\to\infty} \eta_n = T.$$

Proof. The conclusion follows from Corollary 2.1 and Lemma 2.4. □

Following the proof given above with some minor changes, we can obtain the theorem for a more general case.

Theorem 2.2. *Suppose the exact solution satisfies $\|u(\cdot,t)\| \le C$ for all $t \ge 0$ under some norm $\|\cdot\|$ and Hypothesis 2. The numerical approximation u_h satisfies $\|u_h(\cdot,t)\| \le C$ for all $t \ge 0$ and Hypothesis 4. Moreover, assume the numerical blow-up time is defined as (2.1), with $D(n)$ satisfying (2.2) and (2.3), then*

$$\liminf_{n\to\infty} \eta_n = T.$$

Remark 2.7. Actually, Theorem 2.1 is a special case of Theorem 2.2 with $\|\cdot\| = \|\cdot\|_{L^1}$. Theorem 2.1 will be used for the blow-up of the primitive functions while Theorem 2.2 will be used for the blow-up of the derivative of the primitive functions.

Now, we demonstrate two ways to define the numerical blow-up time

1. The first approach is given as

$$\eta_n = \inf\{t \ge t_\star : S(2N,t) \ge S(N,t) + \frac{C}{N_0 m^n}\}, \tag{2.6}$$

with $1 < m < 2^{k+1}$. In most of the numerical experiments we take $k = 1$. Therefore, $1 < m < 4$ would be the suitable choices.

2. Another approach is to define the blow-up time as

$$\eta_n = \inf\{t \ge t_\star : S(2N,t) \ge S(N,t) + \frac{C}{N_0 n^m}\}, \tag{2.7}$$

with $m > 1$. Clearly, $D(n) = \frac{C}{N_0 n^m}$ satisfies (2.2). Moreover, notice the fact that $n^m << m^n$ if n is large, we can show that the conditions given in Theorem 2.1 are all satisfied.

3. Numerical discretizations

In this section, we construct the numerical methods for the pressureless Euler equations and the chemotaxis model to be used in this paper.

3.1. The discontinuous Galerkin methods for pressureless Euler equations

In this subsection, we apply DG methods to pressureless Euler equations subject to periodic boundary conditions in one space dimension. First we rewrite (1.1) as

$$\begin{aligned} &\mathbf{w}_t + \mathbf{f}(\mathbf{w})_x = 0, \quad t > 0,\ x \in \Omega, \\ &\mathbf{w} = \begin{pmatrix} \rho \\ m \end{pmatrix}, \qquad \mathbf{f}(\mathbf{w}) = \begin{pmatrix} m \\ \rho u^2 \end{pmatrix}, \end{aligned} \tag{3.1}$$

where $m = \rho u$, ρ is the density and u is the velocity. Let τ be the time step size, N_x be the number of cells, and $\{I_j\}$, $j = 1, ..., N_x$ be a uniform partition of Ω with h being the mesh size and $I_j = [x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}}]$. The finite element space is defined as

$$\mathbf{V}_h = \{\mathbf{v} : \text{each of its components } v_i|_{I_j} \in P^k(I_j), 1 \le j \le N_x\},$$

where $P^k(I_j)$ denotes the space of polynomials in I_j of degree at most k. The DG scheme for (3.1) is to find $\mathbf{w}_h \in \mathbf{V}_h$ such that for any $\mathbf{v}_h \in \mathbf{V}_h$

$$((\mathbf{w}_h)_t, \mathbf{v}_h)_j = (\mathbf{f}(\mathbf{w}_h), (\mathbf{v}_h)_x)_j + \hat{\mathbf{f}}_{j-\frac{1}{2}} \mathbf{v}_h^+|_{j-\frac{1}{2}} - \hat{\mathbf{f}}_{j+\frac{1}{2}} \mathbf{v}_h^-|_{j+\frac{1}{2}}, \tag{3.2}$$

where $(\mathbf{w}, \mathbf{v})_j = \int_{I_j} \mathbf{w} \cdot \mathbf{v} dx$, and $\mathbf{v}_h^- = \mathbf{v}_h(x_{j+\frac{1}{2}}^-)$ denotes the left limit of the vector $\mathbf{v}_h$ at $x_{j+\frac{1}{2}}$. Likewise for $\mathbf{v}_h^+$. Moreover,

$$\hat{\mathbf{f}}_{j+\frac{1}{2}} = \hat{\mathbf{f}}(\mathbf{w}_h(x_{j+\frac{1}{2}}^-), \mathbf{w}_h(x_{j+\frac{1}{2}}^+))$$

is the numerical flux. If we take $\mathbf{v}_h = 1$ in (3.2) and Euler forward time discretization, we have the equation satisfied by the numerical cell averages

$$\overline{\mathbf{w}}_j^{n+1} = \overline{\mathbf{w}}_j^n + \frac{\tau}{h}(\widehat{\mathbf{f}}_{j-\frac{1}{2}} - \widehat{\mathbf{f}}_{j+\frac{1}{2}}), \tag{3.3}$$

where $\overline{\mathbf{w}}_j^n$ is the numerical cell average is cell I_j at time level n. Moreover, we also denote $\mathbf{w}_j^n$ to be the numerical approximation in cell I_j at time level n. Physically, the density is positive and the velocity satisfies the maximum principle. In [49], the admissible set was defined to be

$$G = \left\{ \mathbf{w} = \begin{pmatrix} \rho \\ m \end{pmatrix} : \rho > 0,\ a\rho \le m \le b\rho \right\},$$

where

$$a = \min u_0(x) \quad \text{and} \quad b = \max u_0(x), \tag{3.4}$$

with u_0 being the initial velocity. Let α_i, $i = 0, \cdots, M$ be the Legendre Gauss-Lobatto quadrature weights for the interval $[-\frac{1}{2}, \frac{1}{2}]$ such that $\sum_{i=0}^{M} \alpha_i = 1$, with $2M - 3 \ge k$, and denote the corresponding Gauss-Lobatto points in cell I_j as x_i^j, then the Gauss-Lobatto quadrature yields

$$\overline{\mathbf{w}}_j^n = \sum_{i=0}^{M} \alpha_i \mathbf{w}_j^n(x_i^j).$$

The following theorem was demonstrated in [49].

Theorem 3.1. *Suppose $\mathbf{w}_h^n(x_i^j) \in G$ in (3.3), then with the Godunov numerical fluxes and under the CFL condition*

$$\lambda = \frac{\tau}{h} < \frac{1}{2\max(|\, a\,|, |\, b\,|)},$$

where a and b are defined in (3.4), we have $\overline{\mathbf{w}}_j^{n+1} \in G$, for all $j = 1, \cdots, N_x$.

Thanks to the above theorem, we have $\overline{\mathbf{w}}_j^{n+1} \in G$. However, the numerical approximation $\mathbf{w}_j^{n+1}$ may not be in G. Therefore, a bound-preserving limiter is necessary to find an updated one $\widetilde{\mathbf{w}}_j \in G$. For simplicity, we drop the superscript $n+1$ in the following procedure.

1. Set up a small number $\varepsilon = 10^{-13}$.
2. Let ρ_j be the numerical approximation of the density in cell I_j. If its cell average $\overline{\rho}_j > \varepsilon$, then proceed to the following steps. Otherwise, take $\widetilde{\mathbf{w}}_j = \overline{\mathbf{w}}_j$ as the numerical solution and skip the following steps.
3. Modify the density: Compute $m_j = \min_i \rho_j(x_i^j)$, then we take

   $$\tilde{\rho}_j = \overline{\rho}_j + \theta(\rho_j - \overline{\rho}_j)$$

   with

   $$\theta = \frac{\overline{\rho}_j - \min(\varepsilon, \overline{\rho}_j)}{\overline{\rho}_j - m_j}$$

   as the new numerical density, also denoted as ρ_j.
4. Modify the velocity: Define $\mathbf{q}_i^j = \mathbf{w}_j(x_i^j)$ and

   $$G^{\varepsilon} = \left\{ \mathbf{w} = \begin{pmatrix} \rho \\ m \end{pmatrix} : \rho \ge \varepsilon, a - \varepsilon \le \frac{m}{\rho} \le b + \varepsilon \right\}.$$

   If $\mathbf{q}_i^j \in G^{\varepsilon}$, then take $\theta_i^j = 1$. Otherwise, take

   $$\theta_i^j = \frac{\|\, \overline{\mathbf{w}}_j - \mathbf{s}_i^j \,\|}{\|\, \overline{\mathbf{w}}_j - \mathbf{q}_i^j \,\|},$$

   where $\|\cdot\|$ is the Euclidean norm, and $\mathbf{s}_i^j$ is the intersection of the line

   $$\mathbf{s}(t) = (1-t)\overline{\mathbf{w}}_j + t\mathbf{q}_i^j, \qquad 0 \le t \le 1,$$

   and

   $$\partial G^{\varepsilon} = \left\{ \mathbf{w} = \begin{pmatrix} \rho \\ m \end{pmatrix} : \rho \ge \varepsilon,\ \frac{m}{\rho} = a - \varepsilon \text{ or } b + \varepsilon \right\}.$$

   Define $\theta_j = \min_{i=0,\ldots,M} \theta_i^j$, and use $\widetilde{\mathbf{w}}_j = \overline{\mathbf{w}}_j + \theta_j(\mathbf{w}_j - \overline{\mathbf{w}}_j)$, as the updated approximation.

Remark 3.1. With the bound-preserving technique we can obtain positive solutions, hence Hypothesis 3 is satisfied. Hypothesis 4 was also verified by numerical experiments in [49].

The above work is for Euler forward time discretization. For high-order ones, we consider the strong stability preserving (SSP) time discretizations [20,41,42] for the ODE system $u_t = Lu$. The second order SSP Runge-Kutta method is given as [20]

$$\begin{aligned} u^{(1)} &= u^n + \tau L(u^n), \\ u^{n+1} &= \frac{1}{2}u^n + \frac{1}{2}\left(u^{(1)} + \tau L(u^{(1)})\right). \end{aligned} \tag{3.5}$$

The third order SSP Runge-Kutta method is given as [42]

$$\begin{aligned} u^{(1)} &= u^n + \tau L(u^n), \\ u^{(2)} &= \frac{3}{4}u^n + \frac{1}{4}\left(u^{(1)} + \tau L(u^{(1)})\right), \\ u^{n+1} &= \frac{1}{3}u^n + \frac{2}{3}\left(u^{(2)} + \tau L(u^{(2)})\right). \end{aligned} \tag{3.6}$$

3.2. The FD scheme for chemotaxis model

According to Appendix B, the chemotaxis model in two space dimensions (1.2) under the polar coordinates can be rewritten as follows

$$\begin{aligned} \tilde{u}_t &= \Delta\tilde{u} - \left(\frac{\tilde{u}}{r}\right)_r - \chi\left(\tilde{u}\frac{r\tilde{v}_r - \tilde{v}}{r^2}\right)_r, \\ \tilde{v}_t &= \Delta\tilde{v} - \left(\frac{\tilde{u}}{r}\right)_r + \tilde{u} - \tilde{v}, \end{aligned} \tag{3.7}$$

on the domain $\Omega = [0, 1]$, where $\tilde{u} = ru$, $\tilde{v} = rv$, $\Delta\tilde{u} = \tilde{u}_{rr}$ and $\Delta\tilde{v} = \tilde{v}_{rr}$. Homogeneous Neumann boundary condition is considered at $r = 0$. Based on the above notations, the L^1-norm of u is defined as

$$\|u\|_{L^1} = \iint u\, dxdy = \int 2\pi ur\, dr = \int 2\pi\tilde{u}\, dr.$$

Therefore, we use

$$\sum_{j=1}^{N}\tilde{u}_j 2\pi h$$

to compute the L^1-norm of the numerical approximation, where $\tilde{u}_j$ is the numerical solution at grid point r_j, $j = 1, \cdots, N$ and h is the mesh size. Similarly, the L^2-norm can be defined as

$$\sqrt{\sum_{j=1}^{N}\tilde{u}_j^2 2\pi\frac{h}{r}}$$

We follow the second-order integrating factor Runge-Kutta [27] method and the scheme can be written as

$$\begin{aligned} \tilde{u}^{(1)} &= e^{-A\tau}\left[\tilde{u}^n - \tau\left(\frac{\widehat{\tilde{u}^n}}{r}\right)_r - \tau\chi\left(\tilde{u}^n\frac{\widehat{r\tilde{v}_r^n - \tilde{v}^n}}{r^2}\right)_r\right], \\ \tilde{v}^{(1)} &= e^{-A\tau}\left[\tilde{v}^n - \tau\left(\frac{\widehat{\tilde{v}^n}}{r}\right)_r + \tau\tilde{u}^n - \tau\tilde{v}^n\right]; \end{aligned} \tag{3.8a}$$

$$\begin{aligned} \tilde{u}^{n+1} &= \frac{1}{2}e^{-A\tau}\tilde{u}^n + \frac{1}{2}\left[\tilde{u}^{(1)} - \tau\left(\frac{\widehat{\tilde{u}^{(1)}}}{r}\right)_r - \tau\chi\left(\tilde{u}^{(1)}\frac{\widehat{r\tilde{v}_r^{(1)} - \tilde{v}^{(1)}}}{r^2}\right)_r\right], \\ \tilde{v}^{n+1} &= \frac{1}{2}e^{-A\tau}\tilde{v}^n + \frac{1}{2}\left[\tilde{v}^{(1)} - \tau\left(\frac{\widehat{\tilde{v}^{(1)}}}{r}\right)_r + \tau\tilde{u}^{(1)} - \tau\tilde{v}^{(1)}\right], \end{aligned} \tag{3.8b}$$

where $\tilde{u}$ and $\tilde{v}$ are vectors containing the numerical approximations at the grid points, A is the matrix of the Δ operator, all the derivatives with respect to r are regarded as some FD spatial discretizations, $\widehat{*}$ is the numerical flux. More details will be given in Appendix B. To obtain positive approximations of $\tilde{u}$ and $\tilde{v}$, we employ the positivity-preserving limiter proposed in [45,46], detailed implementation will be given in Appendix A, to get positive quantities within the square brackets. Moreover, the exponential term keeps the positivity of the numerical approximations. Actually, in [29] the authors proved that the exponential term preserves the maximum-principle. The results can easily be extended to keep the positivity of the numerical approximations, since the exponential term maps the vector e to e itself, where e is the vector whose components are all 1.

4. Blow-up of the derivatives

It is well-known that nonlinear hyperbolic equations may develop shocks in finite time though the initial condition is smooth. When the shock appears, the derivative of the discontinuity of the exact solution can be considered as a δ-function, and this is called the blow-up of the derivative. In this section, we extend the idea proposed in Section 2 to capture the blow-up time of the derivative, i.e. the time when the shock appears. We consider the following scalar hyperbolic conservation law in one space dimension subject to periodic boundary conditions

$$u_t + f(u)_x = 0, \quad u(x,0) = u_0(x), \tag{4.1}$$

where $u_0(x)$ is a function with a bounded variation. The main difficulty we are facing is how to find a suitable norm as demonstrated in Theorem 2.2 for the derivatives of the numerical approximations, since they may not be positive and the integrals may not be a constant during time evolution. Notice that the total variation of a differentiable function is the L^1-norm of the derivative. As an alternative, we use the total variation to control the primitive function, and apply the total variation bounded (TVB) FD schemes [46] for (4.1).

The definition of total variation of a real-valued function $p(x)$ over $[a,b]$ is

$$Var(p) = \sup_{N_x} \sup_{a=x_0<x_1<\ldots<x_{N_x}=b} \sum_{j=0}^{N_x-1} |p(x_{j+1}) - p(x_j)|, \tag{4.2}$$

which equals $\int_a^b |p'(x)|dx$ when the function is differentiable. Therefore, the total variation can be viewed as the L^1-norm of the derivative. If the initial condition has a bounded total variation, then the total variation of the exact solution is decreasing during time evolution, i.e.

$$Var(u(\cdot,t_2)) \le Var(u(\cdot,t_1)) \text{ for any } t_2 > t_1, \tag{4.3}$$

which further yields

$$\|u_x(\cdot,t)\|_{L^1} \le \|u_x(\cdot,0)\|_{L^1}, \ \forall\, t > 0.$$

To construct the FD approximations of (4.1), we define x_j, $j = 1, \cdots, N_x$ as the uniformly distributed grid points of the computational domain Ω with mesh size h. We use u_j^n for the numerical approximation of the exact solution $u(x_j, t^n)$, the total variation of the numerical solution is measured by

$$TV(u^n) = \sum_j |u_{j+1}^n - u_j^n|. \tag{4.4}$$

We construct the approximation of u_x, the derivative of the exact solution, at $x = x_{j+\frac{1}{2}} = x_j + \frac{h}{2}$ as

$$(u_x)_{j+\frac{1}{2}}^n \approx \frac{u_{j+1}^n - u_j^n}{h} \tag{4.5}$$

and define

$$\|u_x^n\|_{L^1} = \sum_{j=1}^{N_x-1} \left| \frac{u_{j+1}^n - u_j^n}{h} \right| \cdot h = TV(u^n), \quad \|u_x^n\|_{L^2}^2 = \sum_{j=1}^{N_x-1} \left| \frac{u_{j+1}^n - u_j^n}{h} \right|^2 \cdot h.$$

In this paper, we would like to construct u to be TVB, namely

$$TV(u^{n+1}) \le TV(u^n) + M\tau \quad or \quad TV(u^{n+1}) \le (1 + M\tau) TV(u^n), \tag{4.6}$$

for some constant M. The TVB flux limiter is one of the most effective technique for high-order FD methods [45,46] to obtain the total variation stability. In this paper, we will follow the TVB flux limiter proposed in [46] to obtain the boundedness of the total variation of the numerical approximations during time evolution, which further implies

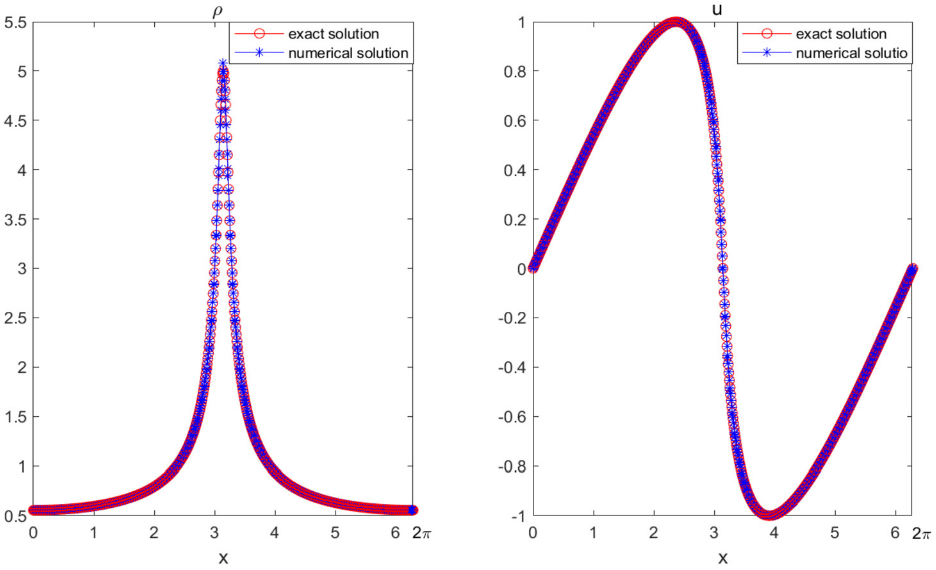

Fig. 1. The numerical approximations of density and velocity at $t = 0.8$ with $\mathcal{P}^1$ polynomials and $N_x = 160$.

$$\|u_x(\cdot,t)\|_{L^1} \le C, \ \forall\, t > 0.$$

Notice that (4.5) is only second-order accurate. Therefore, we can choose $k < 1$ in Hypothesis 4 in Section 2.

Based on the above analysis, the conditions required in Theorem 2.2 are satisfied. We denote $S(N,t) = \|u_x(\cdot,t)\|_{L^2}$ and the numerical blow-up time η_n is also defined as (2.6) or (2.7).

5. Numerical experiments

In this section, we use numerical experiments to demonstrate the convergence of the numerical blow-up time. The theoretical analysis can neither guarantee the convergence of the full sequence of the numerical blow-up time nor provide the convergence rate of the numerical blow-up time. Therefor, in all the numerical experiments, we check (1) whether the numerical blow-up time is getting closer to the exact one and (2) whether the difference between two adjacent numerical blow-up time is getting smaller.

5.1. Pressureless Euler equations

Example 5.1. We solve (3.1) with the following initial data

$$\rho_0(x) = 1, \qquad u_0(x) = sin(x), \tag{5.1}$$

on the domain $\Omega = [0, 2\pi]$. Clearly, the exact solution is

$$u(x,t) = u_0(x_0), \quad \rho(x,t) = \frac{\rho_0(x_0)}{1 + u_0'(x_0)t},$$

where x_0 is given implicitly by

$$x_0 + tu_0(x_0) = x,$$

and at $t = 1$, a δ-function appears in the density at $x = \pi$.

We use the third order SSP RK method with $\tau = 0.05h$ and test the example by using P^k polynomials with $k = 1$. We divide Ω into N uniform cells. Tables 1 and 2 show the L^2-norm of the error between the numerical and exact solutions at $t = 0.8$. We can achieve second order accuracy. Fig. 1 shows the numerical approximations of the density and velocity. We observe a large gradient in the velocity and a potential blow-up in the density at $x = \pi$.

To calculate the numerical blow-up time, we compute the L^2-norm of the numerical approximations of the density with respect to time. We take $N_0 = 10$ as the initial resolution and gradually refine the mesh and take $N = N_0 2^n$ as the number of grid points with n being the level of mesh refinement. Fig. 2 shows the time evolution of the L^2-norm of the numerical approximations of the density with different values of n ($n = 1, \cdots, 9$), denoted as "sn". Here a base-10 log scale is used for the vertical axis. We observe significant differences among the curves when t is close to 1.

Table 1
L^1-norm and L^2-norm of the error at $t = 0.8$ between the numerical velocity and the exact velocity for Example 5.1 with initial condition (5.1).

N	L^1-norm	order	L^2-norm	order
$10 \cdot 2^2$	1.01E-01	-	1.32E-01	-
$10 \cdot 2^3$	3.59E-02	1.49	6.21E-02	1.12
$10 \cdot 2^4$	1.19E-02	1.60	1.99E-02	1.08
$10 \cdot 2^5$	3.20E-03	1.89	5.56E-03	1.64
$10 \cdot 2^6$	7.98E-04	2.00	1.45E-03	1.84
$10 \cdot 2^7$	1.97E-04	2.02	3.66E-04	1.97
$10 \cdot 2^8$	4.94E-05	2.00	9.56E-05	1.98

Table 2
L^1-norm and L^2-norm of the error at $t = 0.8$ between the numerical density and the exact density for Example 5.1 with initial condition (5.1).

N	L^1-norm	order	L^2-norm	order
$10 \cdot 2^2$	1.01E-01	-	2.08E-01	-
$10 \cdot 2^3$	1.00E-02	2.11	1.32E-02	2.19
$10 \cdot 2^4$	2.48E-03	2.02	3.32E-03	1.99
$10 \cdot 2^5$	6.22E-04	1.99	8.78E-04	1.92
$10 \cdot 2^6$	1.57E-04	1.99	2.29E-04	1.94
$10 \cdot 2^7$	3.94E-05	1.99	5.85E-05	1.97
$10 \cdot 2^8$	9.88E-06	2.00	1.48E-05	1.98

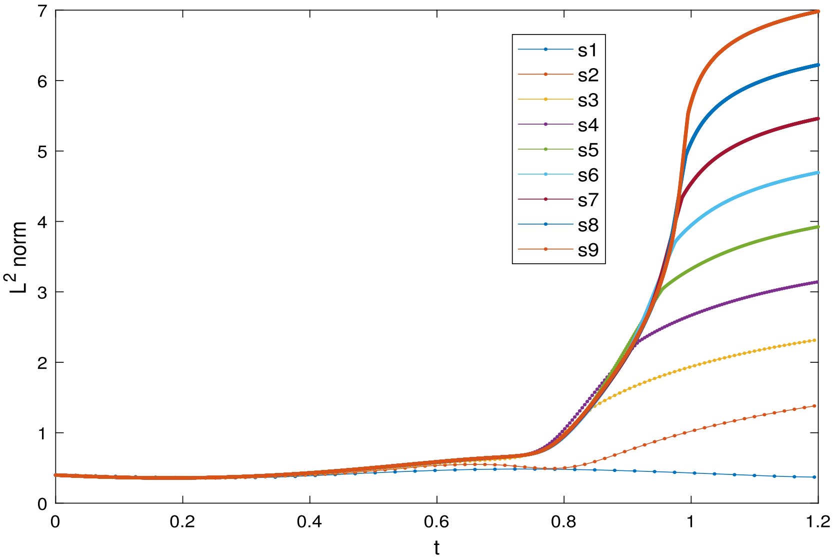

Fig. 2. Log-scaled L^2-norm of the numerical approximations for Example 5.1 under different resolutions. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

We compute the numerical blow-up time η_n by (2.6) with $\tau = 0.05h$ and different choices of m, and the results are given in Table 3. We have the following observations: (1) For fixed m, the larger the C, the better the approximation, since the numerical blow-up time is closer to 1; (2) For fixed C, the smaller the m, the better the convergence, and this is the same as we expected in Remark 2.6; (3) If $m < 4$, we can observe convergence of the numerical blow-up time. If $m = 5$, the numerical blow-up time does not converge to 1. This result agrees with Lemma 2.4 since we chose $k = 1$ and $m = 2^{\tilde{k}} = 2^{k+1} = 4$ is the threshold. Moreover, since $m = 5$ yields $\tilde{k} = \ln_2 m > 2$ which is greater than the designed accuracy of the numerical approximation. Therefore, the decay rate of $D(n)$ is faster than that of the error during mesh refinements. Hence, we expect $\eta_n \to 0$ as $n \to \infty$, which can be observed when $C = 100$. We anticipate a more refined mesh is necessary to observe such convergence for larger values of C.

Next, we calculate the numerical blow-up time η_n by (2.7) and the results are given in Table 4, similar observations can be obtained.

Table 3
Numerical blow-up time by (2.6) with different values of m and C and under different mesh sizes.

m	N	10	20	40	80	160	320	640	1280	2560
1.1	C=100	-	-	0.833	0.848	0.891	0.929	0.954	0.971	0.9815
	C=1000	-	-	-	0.911	0.915	0.935	0.955	0.971	0.9817
	C=10000	-	-	-	-	0.954	0.956	0.962	0.973	0.9820
	C=100000	-	-	-	-	-	-	0.979	0.980	0.9840
1.5	C=100	-	-	0.817	0.833	0.888	0.927	0.954	0.971	0.9815
	C=1000	-	-	-	0.888	0.903	0.931	0.954	0.971	0.9815
	C=10000	-	-	-	-	0.939	0.943	0.957	0.971	0.9817
	C=100000	-	-	-	-	-	-	0.967	0.974	0.9820
2	C=100	-	-	0.817	0.825	0.884	0.927	0.954	0.971	0.9815
	C=1000	-	-	0.958	0.872	0.895	0.929	0.954	0.971	0.9815
	C=10000	-	-	-	-	0.923	0.935	0.954	0.971	0.9815
	C=100000	-	-	-	-	-	0.954	0.959	0.971	0.9817
3	C=100	-	-	0.801	0.817	0.884	0.927	0.954	0.971	0.9815
	C=1000	-	-	-	0.856	0.888	0.927	0.954	0.971	0.9815
	C=10000	-	-	-	-	0.907	0.929	0.954	0.971	0.9815
	C=100000	-	-	-	-	-	0.939	0.955	0.971	0.9815
5	C=100	-	-	0.770	0.801	0.884	0.522	0.365	0.229	0.1198
	C=1000	-	-	-	0.833	0.884	0.927	0.761	0.748	0.5044
	C=10000	-	-	-	0.888	0.891	0.927	0.807	0.782	0.7682
	C=100000	-	-	-	-	0.915	0.929	0.954	0.841	0.8183

Table 4
Numerical blow-up time by (2.7)with different values of m and C and under different mesh sizes.

m	N	10	20	40	80	160	320	640	1280	2560
1.1	C=100	-	-	0.833	0.840	0.888	0.927	0.954	0.971	0.9815
	C=1000	-	-	-	0.888	0.903	0.931	0.954	0.971	0.9815
	C=10000	-	-	-	-	0.939	0.943	0.957	0.971	0.9817
	C=100000	-	-	-	-	-	0.970	0.970	0.975	0.9822
1.5	C=100	-	-	0.833	0.833	0.884	0.927	0.954	0.971	0.9815
	C=1000	-	-	-	0.880	0.899	0.929	0.954	0.971	0.9815
	C=10000	-	-	-	-	0.931	0.939	0.956	0.971	0.9817
	C=100000	-	-	-	-	-	0.962	0.964	0.973	0.9820
2	C=100	-	-	0.833	0.825	0.884	0.927	0.954	0.971	0.9815
	C=1000	-	-	-	0.872	0.891	0.929	0.954	0.971	0.9815
	C=10000	-	-	-	-	0.923	0.935	0.955	0.971	0.9815
	C=100000	-	-	-	-	-	0.954	0.960	0.972	0.9817
3	C=100	-	-	0.833	0.834	0.884	0.927	0.954	0.971	0.9815
	C=1000	-	-	-	0.856	0.888	0.927	0.954	0.971	0.9815
	C=10000	-	-	-	-	0.907	0.929	0.954	0.971	0.9815
	C=100000	-	-	-	-	-	0.941	0.956	0.971	0.9815

5.2. 1D KS chemotaxis model

Example 5.2. We consider the following KS chemotaxis model in one space dimension on the computational domain $\Omega = [-0.4, 0.4]$

$$\begin{aligned} &u_t - u_{xx} - (uv_x)_x = 0, \\ &v_t - v_{xx} = u - v, \end{aligned} \tag{5.2}$$

subject to periodic boundary conditions. The initial conditions are given as

$$\begin{cases} u_0 = 840^{-84x^2}, \\ v_0 = 420e^{-42x^2}. \end{cases} \tag{5.3}$$

The initial data are the one dimensional version of the initial data given in [14], where a blow-up is anticipated. We use the second-order centered difference for the diffusion term and the second-order FD methods with positivity-preserving flux limiter [45,46] for the convection term. The time integration was also given as the second-order integrating factor Runge-Kutta method discussed in Section 3.2. The numerical approximations of u at different terminal time with $N_x = 160$ grid points are given in Fig. 3. From the figure, we observe a δ-like structure at $x = 0$. However, it has been proved that the blow-up will never happen for chemotaxis model in one space dimension. Therefore, this structure cannot be a δ-function. To test this, we compute the L^2-norm of the numerical approximations on different grids and the results are given in Fig. 4, where a based-10 log scale is applied for the vertical axis and the L^2 norm of the numerical approximation under the n-th

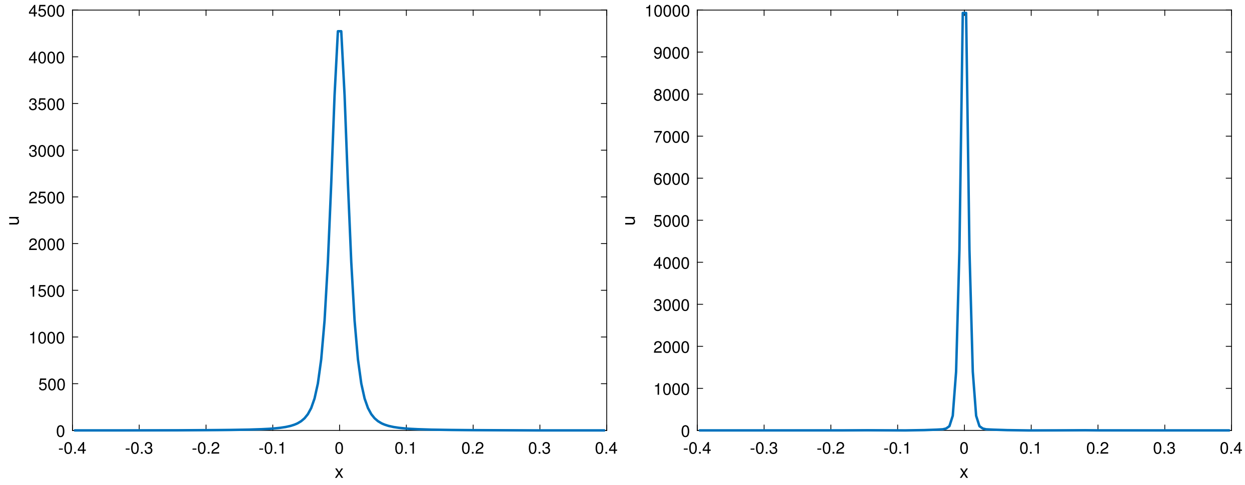

Fig. 3. Example 5.2: numerical approximations of u at $t = 5 * 10^{-5}$ (left) and $t = 1 * 10^{-4}$ (right) with positivity-preserving limiter for FD scheme and $N = 160$.

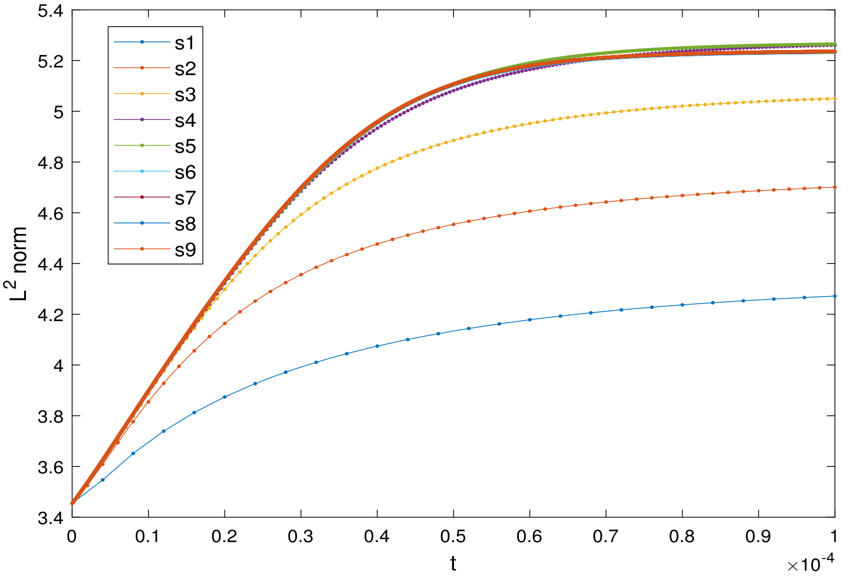

Fig. 4. Log-scaled L^2-norm of the numerical approximations for Example 5.2 under different resolutions.

level mesh refinement is denoted as "sn". We observe that the curves are basically the same if $n > 4$ $(N > 80)$. Hence this is not a blow-up solution.

5.3. 2D KS chemotaxis models

We use the second order FD method given in section 3.2 to solve the 2D KS chemotaxis equations (3.7) on the computational domain Ω in polar coordinates. For simplicity, we always assume the solution keeps the same along the angular coordinate. Hence Ω is a one dimensional domain containing the radial coordinate only.

Example 5.3. We choose $\Omega = [\pi, 3\pi]$, $\chi = 1$ and consider the following initial conditions

$$\begin{cases} u_0 = \sin^2 r \\ v_0 = \cos r + 2. \end{cases} \tag{5.4}$$

We compute the numerical approximations of u at $t = 0.1$ and calculate the error between the numerical and reference solutions (obtained by the finest resolution) under L^1 and L^2 norms, and the results are given in Table 5. Clearly, we can observe optimal order of accuracy.

Table 5
L^1-norm and L^2-norm of the error at $t = 0.1$ for Example 5.3.

N	L^1-norm	order	L^2-norm	order
2^4	4.35E-02	-	2.26E-02	-
2^5	1.18E-02	1.88	5.93E-03	1.93
2^6	2.38E-03	2.31	1.14E-03	2.38
2^7	4.55E-04	2.38	2.10E-04	2.44
2^8	9.90E-05	2.20	4.47E-05	2.23

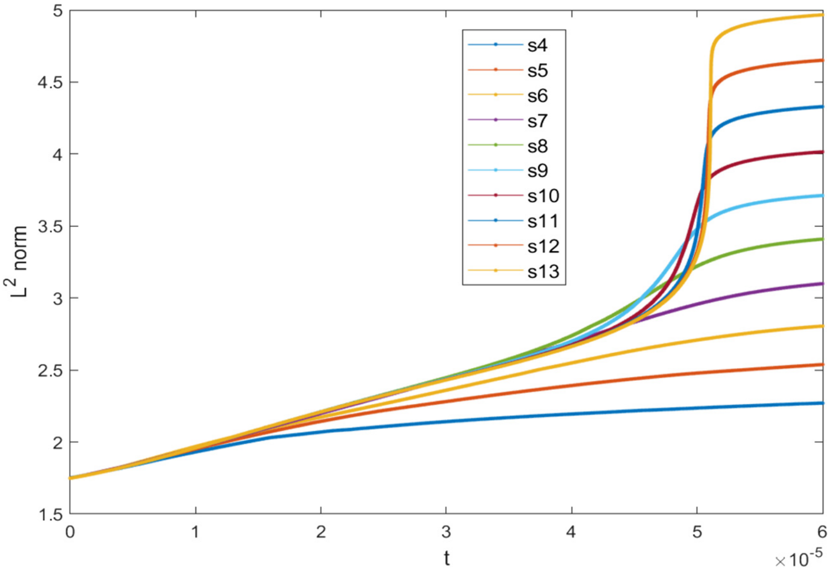

Fig. 5. L^2-norm of the numerical approximation for Example 5.4 under different resolutions.

Next, we test the convergence of the numerical blow-up time proposed in (2.6) and (2.7).

Example 5.4. We take the domain $\Omega = [0, 1]$, $\chi = 8$ and consider the following initial conditions

$$\begin{cases} u_0 = 1000e^{-500r^2}, \\ v_0 = 10e^{-500r^2}. \end{cases} \tag{5.5}$$

We compute the L^2-norm of the numerical approximations with respect to time. The initial mesh contains $N_0 = 10$ grid points and double the total grid points during mesh refinement with n being the refinement level. Fig. 5 shows the time evolution of the L^2-norm of the numerical solutions with $N = N_0 \cdot 2^n$ ($n = 4$ to 13) grid points, where a base-10 log scale is used for the vertical axis and the L^2-norm of the numerical approximation under the n-th level of mesh refinement is denoted as “sn”. We can see that the curves are basically identical if $t < 5 \times 10^{-5}$, and this agrees with the results presented in [8], where the true blow-up time is anticipated to be 5.115×10^{-5}. We compute the numerical blow-up time η_n by (2.6) with $\tau = 10^{-6}h$ and $m = 2, 3$. The results are given in Tables 6-7. We can observe that the numerical blow-up time converges to the exact one during mesh refinement. Moreover, we compute the numerical blow-up time η_n by (2.7) with $m = 2, 3$ and similar results can also be obtained from Tables 8-9. Moreover, we conclude that the choice of the constant C does not have significant effects on the convergence.

5.4. Burgers' equations

In this section, we test the convergence of the numerical blow-up time of the derivative.

Example 5.5. We solve (4.1) with initial condition $u_0(x) = \sin(x)$. The exact solution is given implicitly as $u(x, t) = sin(x - ut)$. The initial total variation is $Var(u_0) = 4$ and the shock appears at $t = 1$.

Table 6
The convergence of blow-up time by (2.6) with $m = 2$ and different constant C.

	n	160	320	640	1280	2560
Blow-up time	C=100	4.913E-06	5.300E-06	1.465E-05	1.042E-05	1.064E-05
	C=1000	-	1.501E-05	1.683E-05	1.453E-05	4.556E-05
	C=10000	-	-	2.710E-05	3.587E-05	4.564E-05
	C=100000	-	-	-	4.346E-05	4.638E-05
	C=1000000	-	-	-	-	4.883E-05
	N	5120	10240	20480	40960	81920
Blow-up time	C=100	4.917E-05	5.042E-05	5.089E-05	5.108E-05	5.115E-05
	C=1000	4.917E-05	5.042E-05	5.089E-05	5.108E-05	5.115E-05
	C=10000	4.917E-05	5.042E-05	5.089E-05	5.108E-05	5.115E-05
	C=100000	4.921E-05	5.042E-05	5.089E-05	5.108E-05	5.115E-05
	C=1000000	4.948E-05	5.044E-05	5.089E-05	5.108E-05	5.115E-05

Table 7
The convergence of blow-up time by (2.6) with $m = 3$ and different constant C.

	N	160	320	640	1280	2560
Blow-up time	C=100	2.125E-06	4.356E-06	1.263E-06	3.500E-07	1.031E-07
	C=1000	-	6.256E-06	1.475E-05	1.034E-05	9.609E-06
	C=10000	-	-	1.743E-05	1.407E-05	4.556E-05
	C=100000	-	-	2.939E-05	3.526E-05	4.561E-05
	C=1000000	-	-	-	4.304E-05	4.604E-05
	N	5120	10240	20480	40960	81920
Blow-up time	C=100	4.917E-05	5.042E-05	5.089E-05	5.108E-05	5.115E-05
	C=1000	4.917E-05	5.042E-05	5.089E-05	5.108E-05	5.115E-05
	C=10000	4.917E-05	5.042E-05	5.089E-05	5.108E-05	5.115E-05
	C=100000	4.917E-05	5.042E-05	5.089E-05	5.108E-05	5.115E-05
	C=1000000	4.918E-05	5.042E-05	5.089E-05	5.108E-05	5.115E-05

Table 8
The convergence of blow-up time by (2.7) with $m = 2$ and different constant C.

	N	160	320	640	1280	2560
Blow-up time	C=100	4.425E-06	5.300E-06	1.473E-05	1.089E-05	4.556E-05
	C=1000	-	1.501E-05	1.737E-05	1.724E-05	4.558E-05
	C=10000	-	-	2.914E-05	3.804E-05	4.577E-05
	C=100000	-	-	-	4.543E-05	4.714E-05
	C=1000000	-	-	-	-	-
	N	5120	10240	20480	40960	81920
Blow-up time	C=100	4.917E-05	5.042E-05	5.089E-05	5.108E-05	5.115E-05
	C=1000	4.917E-05	5.042E-05	5.089E-05	5.108E-05	5.115E-05
	C=10000	4.918E-05	5.042E-05	5.089E-05	5.108E-05	5.115E-05
	C=100000	4.931E-05	5.043E-05	5.089E-05	5.108E-05	5.115E-05
	C=1000000	5.004E-05	5.053E-05	5.091E-05	5.108E-05	5.115E-05

Table 9
The convergence of blow-up time by (2.7) with $m = 3$ and different constant C.

	N	160	320	640	1280	2560
Blow-up time	C=100	2.125E-06	4.431E-06	1.438E-06	9.942E-06	9.040E-06
	C=1000	9.575E-06	6.750E-06	1.510E-05	1.154E-05	4.556E-05
	C=10000	-	-	1.942E-05	2.151E-05	4.559E-05
	C=100000	-	-	3.521E-05	3.960E-05	4.585E-05
	C=1000000	-	-	-	-	4.751E-05
	N	5120	10240	20480	40960	81920
Blow-up time	C=100	4.917E-05	5.042E-05	5.089E-05	5.108E-05	5.115E-05
	C=1000	4.917E-05	5.042E-05	5.089E-05	5.108E-05	5.115E-05
	C=10000	4.917E-05	5.042E-05	5.089E-05	5.108E-05	5.115E-05
	C=100000	4.919E-05	5.042E-05	5.089E-05	5.108E-05	5.115E-05
	C=1000000	4.934E-05	5.044E-05	5.089E-05	5.108E-05	5.115E-05

Table 10
L^1, L^2 norms of the error with TVB flux limiters.

N	L^1-norm	order	L^2-norm	order
2^5	8.85E-03	-	6.42E-03	-
2^6	1.16E-03	2.933	8.94E-04	2.844
2^7	1.47E-04	2.978	1.15E-04	2.962
2^8	1.84E-05	2.996	1.44E-05	2.991
2^9	2.31E-06	2.998	1.81E-06	2.998
2^{10}	2.88E-07	3.000	2.26E-07	2.999

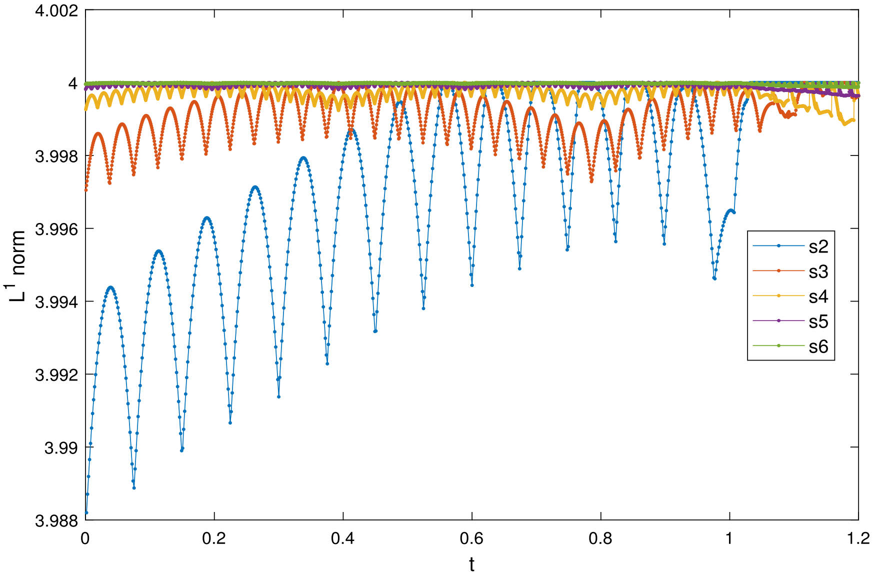

Fig. 6. L^1-norm of the numerical approximation for Example 5.5 under different resolutions.

We apply the ENO3 [42] method with the TVB flux limiter [45,46]. The time integration is given as the SSP RK3 method (3.6) with $\tau = 0.01h$. We compute the error between the numerical and exact solutions at $t = 0.5$. The results are given in Table 10. From the table, we can observe optimal convergence rates.

Next, we compute the blow-up time of the derivatives. Fig. 6 shows the L^1-norm of the numerical approximations of u_x on different grids. Only the results based on $n = 2, 3, 4, 5, 6$ are plotted, since the curves for larger values of n are too close to 4. Moreover, we also plot the L^2-norm of the numerical approximations of u_x in Fig. 7.

We choose (2.6) with $m = 2$, 3, 4 and 5 to compute the blow-up time, and the results are given in Table 11. Since (4.5) is only second-order accurate, we can only observe convergence from the results for $m = 2$, 3. Though the convergence for $m = 3$ is quite slow. The results for $m = 5$ may not converge, since the numerical blow-up time is decreasing, especially if C is small. $m = 4$ is the threshold, and it is hard to determine whether the numerical blow time is convergent or divergent.

Moreover, we also use (2.7) with $m = 2$ and 3 to compute the numerical blow-up time and the results are given in Table 12. We can clearly observe that η_n converges to 1, the exact blow-up time, during mesh refinement.

6. Concluding remarks

In this work, we considered the general convection-diffusion equations with blow-up solutions. We proposed two methods to find the numerical blow-up time by using the L^2-norm of the solutions. The convergence of the numerical blow-up time can be obtained under some reasonable assumptions. We extended this method to calculate when the shock appears for nonlinear hyperbolic equations thanks to the TVB flux limiters. Numerical experiments verified the theoretical analysis. Finally, the proposed method can also be used to test whether the equation has blow-up solutions or not.

In the future, we will discuss how to improve the convergence rate of the numerical blow-up time as the current algorithm may result in slow convergence. One idea was to use adaptive methods and refine the meshes locally near the singularities.

CRediT authorship contribution statement

Hui Guo: Funding acquisition, Investigation, Writing – review & editing. **Xueting Liang:** Software, Writing – original draft. **Yang Yang:** Conceptualization, Funding acquisition, Investigation, Methodology.

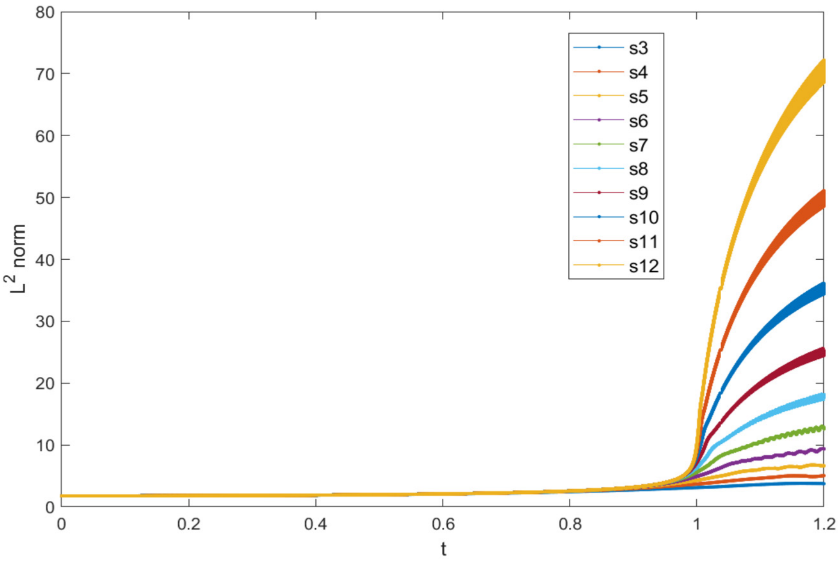

Fig. 7. L^2-norm of the numerical approximation for Example 5.5 under different resolutions.

Table 11
The blow-up time by (2.6) with different values of m.

m	N	40	80	160	320	640	1280	2560	5120	10240	20480
2	C=1	-	0.683	0.753	0.808	0.851	0.884	0.910	0.921	0.945	0.9570
	C=10	-	0.910	0.899	0.908	0.922	0.936	0.949	0.960	0.969	0.9755
	C=100	-	-	1.063	0.999	0.979	0.974	0.975	0.979	0.982	0.9858
	C=300	-	-	-	1.119	1.009	0.992	0.986	0.986	0.987	0.9894
	C=400	-	-	-	1.160	1.019	0.997	0.989	0.988	0.989	0.9903
3	C=1	-	0.592	0.651	0.701	0.742	0.775	0.801	0.823	0.842	0.8577
	C=10	-	0.830	0.826	0.841	0.861	0.880	0.897	0.910	0.921	0.9302
	C=100	-	1.126	0.967	0.933	0.929	0.934	0.942	0.950	0.958	0.9626
	C=500	-	-	1.179	0.999	0.968	0.961	0.962	0.966	0.970	0.9747
	C=1000	-	-	-	1.034	0.989	0.972	0.969	0.972	0.975	0.9784
4	C=1	-	0.514	0.547	0.572	0.587	0.596	0.601	0.603	0.605	0.6052
	C=100	-	1.012	0.912	0.890	0.889	0.896	0.904	0.910	0.915	0.9173
	C=1000	-	-	1.170	0.979	0.950	0.944	0.946	0.950	0.954	0.9567
	C=10000	-	-	-	-	1.011	0.981	0.973	0.972	0.973	0.9753
	C=20000	-	-	-	-	1.041	0.992	0.980	0.977	0.977	0.9790
5	C=10	-	0.732	0.719	0.722	0.722	0.713	0.697	0.675	0.648	0.6162
	C=100	-	0.961	0.875	0.853	0.852	0.854	0.857	0.850	0.842	0.8320
	C=1000	-	-	1.021	0.943	0.922	0.919	0.920	0.922	0.921	0.9191
	C=10000	-	-	-	1.052	0.979	0.960	0.955	0.955	0.957	0.9575

Table 12
The numerical blow-up time by (2.7) with different values of m.

m	N	40	80	160	320	640	1280	2560	5120	10240	20480
2	C=0.1	-	0.330	0.492	0.633	0.740	0.817	0.875	0.911	0.938	0.9572
	C=1	-	0.683	0.745	0.808	0.860	0.899	0.929	0.950	0.965	0.9756
	C=10	-	0.910	0.891	0.908	0.928	0.946	0.961	0.972	0.980	0.9858
	C=200	-	-	1.250	1.035	1.005	0.995	0.992	0.993	0.994	0.9950
	C=300	-	-	-	1.120	1.018	1.002	0.997	0.995	0.995	0.9961
	C=400	-	-	-	1.160	1.029	1.008	1.000	0.997	0.997	0.9970
3	C=1	-	0.606	0.651	0.720	0.786	0.841	0.884	0.916	0.940	0.9572
	C=10	-	0.840	0.826	0.851	0.883	0.912	0.935	0.953	0.966	0.9756
	C=100	-	1.199	0.967	0.943	0.945	0.955	0.965	0.974	0.981	0.9858
	C=500	-	-	1.179	1.010	0.985	0.980	0.982	0.985	0.988	0.9910
	C=1000	-	-	-	1.048	1.005	0.992	0.989	0.989	0.991	0.9930

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A. The positivity-preserving limiter

In this section, we introduce the general parametrized flux limiter proposed in [46,48]. The limiter can preserve the global maximum principle while maintaining the high order accuracy of the underlying scheme. In this paper, we will introduce the one-dimensional case, and assume the numerical consider the following form

$$\frac{d}{dt}u_j + \frac{1}{h}(\widehat{H}_{j+\frac{1}{2}} - \widehat{H}_{j-\frac{1}{2}}) = 0, \tag{A.1}$$

where u_j is the numerical approximation at the grid point x_j, and $\widehat{H}$ is the high-order numerical flux. With the third order SSP-RK method (3.5), the updated scheme can be written as

$$u_j^{n+1} = u_j^n - \lambda(\widehat{H}^{rk}_{j+\frac{1}{2}} - \widehat{H}^{rk}_{j-\frac{1}{2}}), \tag{A.2}$$

where $\lambda = \tau/h$, and

$$\widehat{H}^{rk}_{j+\frac{1}{2}} = \frac{1}{6}\widehat{H}^{n}_{j+\frac{1}{2}} + \frac{2}{3}\widehat{H}^{(2)}_{j+\frac{1}{2}} + \frac{1}{6}\widehat{H}^{(1)}_{j+\frac{1}{2}}, \tag{A.3}$$

with $\widehat{H}^*$ being the numerical flux obtained by u_h^* at each RK stage for $* = n, (1), (2)$, respectively.

The general parametrized flux-limiting procedure proposed in [48] is to modify the flux $\widetilde{H}^{rk}_{j+\frac{1}{2}}$:

$$\widetilde{H}^{rk}_{j+\frac{1}{2}} = \theta_{j+\frac{1}{2}}(\widehat{H}^{rk}_{j+\frac{1}{2}} - \widehat{h}_{j+\frac{1}{2}}) + \widehat{h}_{j+\frac{1}{2}}, \tag{A.4}$$

where $\widehat{h}_{j+\frac{1}{2}}$ is a low order monotone flux with which the scheme is positivity-preserving, i.e.

$$0 \le u_j^n - \lambda(\widehat{h}_{j+\frac{1}{2}} - \widehat{h}_{j-\frac{1}{2}}).$$

The limiting parameter $\theta_{j+\frac{1}{2}}$ is a number between 0 and 1, and it is defined to ensure

$$0 \le u_j^n - \lambda(\widetilde{H}^{rk}_{j+\frac{1}{2}} - \widetilde{H}^{rk}_{j-\frac{1}{2}}). \tag{A.5}$$

Detailed steps are given in [48] to find the limiting parameters through the inequality (A.5). The basic idea is to find a pair $(\Lambda_{-\frac{1}{2},I_j}, \Lambda_{+\frac{1}{2},I_j})$ such that any pair $(\theta_{j-\frac{1}{2}}, \theta_{j+\frac{1}{2}}) \in [0, \Lambda_{-\frac{1}{2},I_j}] \times [0, \Lambda_{+\frac{1}{2},I_j}]$ satisfies (A.5). Then the following inequality can be obtained:

$$\lambda\theta_{j-\frac{1}{2}}(\widetilde{H}^{rk}_{j-\frac{1}{2}} - \widehat{h}_{j-\frac{1}{2}}) - \lambda\theta_{j+\frac{1}{2}}(\widetilde{H}^{rk}_{j+\frac{1}{2}} - \widehat{h}_{j+\frac{1}{2}}) - \Gamma_j^m \ge 0, \tag{A.6}$$

where we have

$$\Gamma_j^m := 0 - [u_j^n - \lambda(\widehat{h}_{j+\frac{1}{2}} - \widehat{h}_{j-\frac{1}{2}})] \le 0. \tag{A.7}$$

Let $F_{j\pm\frac{1}{2}} := \widehat{H}^{rk}_{j\pm\frac{1}{2}} - \widehat{h}_{j\pm\frac{1}{2}}$, next we get the parameter $\theta_{j+\frac{1}{2}}$ as follows:

(a) if $F_{j-\frac{1}{2}} \ge 0$, $F_{j+\frac{1}{2}} \le 0$,

$$(\Lambda^m_{-\frac{1}{2},I_j}, \Lambda^m_{+\frac{1}{2},I_j}) = (1, 1);$$

(b) if $F_{j-\frac{1}{2}} \ge 0$, $F_{j+\frac{1}{2}} > 0$,

$$(\Lambda^m_{-\frac{1}{2},I_j}, \Lambda^m_{+\frac{1}{2},I_j}) = (1, \min(1, \frac{\Gamma_j^m}{-\lambda F_{j+\frac{1}{2}}}));$$

(c) if $F_{j-\frac{1}{2}} < 0$, $F_{j+\frac{1}{2}} \le 0$,

$$(\Lambda^m_{-\frac{1}{2},I_j}, \Lambda^m_{+\frac{1}{2},I_j}) = (\min(1, \frac{\Gamma_j^m}{\lambda F_{j-\frac{1}{2}}}), 1);$$

(d) if $F_{j-\frac{1}{2}} < 0$, $F_{j+\frac{1}{2}} > 0$,

$$(\Lambda^m_{-\frac{1}{2},I_j}, \Lambda^m_{+\frac{1}{2},I_j}) = (\frac{\Lambda^m_{-\frac{1}{2},I_j}}{\lambda F_{j-\frac{1}{2}} - \lambda F_{j+\frac{1}{2}}}, \frac{\Lambda^m_{-\frac{1}{2},I_j}}{\lambda F_{j-\frac{1}{2}} - \lambda F_{j+\frac{1}{2}}}).$$

The local parameter $\theta_{j+\frac{1}{2}}$ is determined to be

$$\theta_{j+\frac{1}{2}} = \min(\Lambda^m_{+\frac{1}{2},I_j}, \Lambda^m_{-\frac{1}{2},I_{j+1}}). \tag{A.8}$$

Therefore, the modified positivity-preserving numerical flux will be

$$\widetilde{H}^{rk}_{j+\frac{1}{2}} = \theta_{j+\frac{1}{2}}(\widehat{H}^{rk}_{j+\frac{1}{2}} - \widehat{h}_{j+\frac{1}{2}}) + \widehat{h}_{j+\frac{1}{2}}. \tag{A.9}$$

Appendix B. The chemotaxis model

We use polar coordinates to rewrite (1.2). For simplicity, we assume the solution is a constant along the angular coordinate, i.e. $u_\theta = 0$, then it is easy to check that $u_x = u_r \cos\theta$ and $u_y = u_r \sin\theta$. Therefore, we have

$$\begin{aligned} ru_t &= (ru_r)_r - \chi(ruv_r)_r, \\ rv_t &= (rv_r)_r + ru - rv, \end{aligned} \tag{B.1}$$

on the domain $r \in [0, 1]$. Let $\tilde{u} = ru$ and $\tilde{v} = rv$, we can further get

$$\begin{aligned} \tilde{u}_t &= \Delta\tilde{u} - \left(\frac{\tilde{u}}{r}\right)_r - \chi\left(\tilde{u}\frac{r\tilde{v}_r - \tilde{v}}{r^2}\right)_r, \\ \tilde{v}_t &= \Delta\tilde{v} - \left(\frac{\tilde{v}}{r}\right)_r + \tilde{u} - \tilde{v}, \end{aligned} \tag{B.2}$$

where $\Delta\tilde{u} = \tilde{u}_{rr}$ and $\Delta\tilde{v} = \tilde{v}_{rr}$.

Next, we show that the numerical approximations are positive provided Δt is small enough. Due to the SSP structure of the proposed schemes (3.8), we only need to discuss (3.8a). We consider the following first order time discretization

$$\tilde{u}^{n+1} = e^{-A\tau}\left[\tilde{u}^n - \tau\left(\widehat{\frac{\tilde{u}^n}{r}}\right)_r - \tau\chi\left(\widehat{\tilde{u}^n\frac{r\tilde{v}^n_r - \tilde{v}^n}{r^2}}\right)_r\right], \tag{B.3a}$$

$$\tilde{v}^{n+1} = e^{-A\tau}\left[\tilde{v}^n - \tau\left(\widehat{\frac{\tilde{v}^n}{r}}\right)_r + \tau\tilde{u}^n - \tau\tilde{v}^n\right], \tag{B.3b}$$

with $\widehat{*}$ being the numerical flux with flux limiters discussed in Appendix A. To perform the spatial discretization, we denote r_j, $j = 1, \cdots, N$, to be the uniformly distributed grid points with mesh size h and $r_1 = \frac{h}{2}$, $r_N = 1 - \frac{h}{2}$. The spatial discretization of $\widehat{*}_r$ at r_j is given as $(\widehat{*}_{j+\frac{1}{2}} - \widehat{*}_{j-\frac{1}{2}})/h$, where $\widehat{*}_{j-\frac{1}{2}}$ is the numerical flux at $r_j - \frac{h}{2}$. In this paper, the high-order numerical flux is obtained by the third-order WENO FD method with flux splitting [28], and the low-order monotone flux is given as the upwind flux with $\tilde{v}^n$ given as the average of the values at the two adjacent grid points and $\tilde{v}_r$ given by the centered difference. To enforce the homogeneous Neumann boundary condition at $r = 0$, we take $\widehat{*}_{\frac{1}{2}} = \widehat{*}_{\frac{3}{2}}$. For simplicity, we use u for $\tilde{u}$ and v for $\tilde{v}$ in this section, and let $f(u) = \frac{u}{r}$, $g(u) = u\frac{rv_r - v}{r^2}$ and $h(v) = \frac{v}{r}$. Next, we use u_j as the numerical solution at the jth grid point, likewise for v_j. Then we have the following lemma, whose proof follows from direct computation, hence we skip it.

Lemma B.1. *Suppose $u^n \geq 0$ and $v^n \geq 0$, then $u^{n+1} \geq 0$ under the CFL condition*

$$1 - \lambda a - 2\lambda\chi b \geq 0, \tag{B.4}$$

where $a = \max|f'(u)|$, $b = \max|g'(u)|$. In addition, we have $v^{n+1} \geq 0$ under the CFL condition

$$1 - \tau - \lambda c \geq 0, \tag{B.5}$$

where $c = \max|h'(v)|$.

References

[1] L.M. Abia, J.C. López-Marcos, J. Martínez, The Euler method in the numerical integration of reaction-diffusion problems with blow-up, Appl. Numer. Math. 38 (2001) 287–313.

[2] B. Andreianov, M. Bendahmane, M. Saad, Finite volume methods for degenerate chemotaxis model, J. Comput. Appl. Math. 235 (14) (2011) 4015–4031.

[3] G. Arumugam, J. Tyagi, Keller-Segel chemotaxis models: a review, Acta Appl. Math. 171 (1) (2021) 1–82.

[4] K.A. Adou, K.A. Touré, A. Coulibaly, On the numerical quenching time at blow-up, Adv. Math. Sci. 8 (2019) 71–85.

[5] F. Berthelin, Existence and weak stability for a pressureless model with unilateral Cconstraint, Math. Models Methods Appl. Sci. 12 (2) (2008) 249–272.

[6] C. Berthon, M. Breu, M.O. Titeux, A relaxation scheme for the approximation of the pressureless Euler equations, Numer. Methods Partial Differ. Equ. 22 (2) (2010) 484–505.

[7] L. Boudin, J. Mathiaud, A numerical scheme for the one-dimensional pressureless gases system, Numer. Methods Partial Differ. Equ. 28 (6) (2012) 1729–1746.

[8] C.J. Budd, R. Carretero-González, R.D. Russell, Precise computations of chemotactic collapse using moving mesh methods, J. Comput. Phys. 202 (2) (2005) 463–487.

[9] C.H. Cho, On the convergence of numerical blow-up time for a second order nonlinear ordinary differential equation, Appl. Math. Lett. 24 (1) (2011) 49–54.

[10] C.H. Cho, S. Hamada, H. Okamoto, On the finite difference approximation for a parabolic blow-up problem, Jpn. J. Ind. Appl. Math. 24 (2) (2007) 131–160.

[11] C.H. Cho, Y.J. Lu, On the numerical solutions for a parabolic system with blow-up, AIMS Math. 6 (2021) 11749–11777.

[12] Y. Epshteyn, Discontinuous Galerkin methods for the chemotaxis and haptotaxis models, J. Comput. Appl. Math. 224 (1) (2009) 168–181.

[13] Y. Epshteyn, A. Izmirlioglu, Fully discrete analysis of a discontinuous finite element method for the Keller-Segel chemotaxis model, J. Sci. Comput. 40 (1–3) (2009) 211–256.

[14] Y. Epshteyn, A. Kurganov, New interior penalty discontinuous Galerkin methods for the Keller-Segel Chemotaxis model, SIAM J. Numer. Anal. 47 (2008) 368–408.

[15] F. Filbet, A finite volume scheme for the Patlak-Keller-Segel chemotaxis model, Numer. Math. 104 (4) (2006) 457–488.

[16] B. François, J. François, Duality solutions for pressureless gases, monotone scalar conservation laws, and uniqueness, Commun. Partial Differ. Equ. 24 (11–12) (1999) 2173–2189.

[17] H. Fujita, On the blowing up of solutions to the Cauchy problem for $u_t = \triangle u + u^{1+\alpha}$, J. Fac. Sci. Univ. Tokyo Sect. 13 (1966) 109–124.

[18] H. Fujita, On some nonexistence and nonuniqueness theorems for nonlinear parabolic equations, Proc. Symp. Pure Math. 18 (1970) 105–113.

[19] H. Gajewski, K. Zacharias, Global behaviour of a reaction-diffusion system modelling chemotaxis, Math. Nachr. 195 (1) (1998) 77–114.

[20] S. Gottlieb, C.W. Shu, E. Tadmor, Strong stability-preserving high-order time discretization methods, SIAM Rev. 43 (1) (2001) 89–112.

[21] L. Gosse, F. James, Numerical approximations of one-dimensional linear conservation equations with discontinuous coefficients, Math. Comput. 59 (231) (2000) 987–1015.

[22] P. Groisman, Totally discrete explicit and semi-implicit Euler methods for a Blow-up problem in several space dimensions, Computing 76 (3/4) (2006) 325–352.

[23] L. Guo, X. Li, Y. Yang, Energy dissipative local discontinuous Galerkin methods for Keller-Segel chemotaxis model, J. Sci. Comput. 78 (2019) 1387–1404.

[24] L. Guo, Y. Yang, Positivity-preserving high-order local discontinuous Galerkin method for parabolic equations with blow-up solutions, J. Comput. Phys. 289 (2015) 181–195.

[25] M.A. Herrero, J.L. Juan Velázquez, Singularity patterns in a chemotaxis mode, Math. Ann. 306 (1) (1996) 583–623.

[26] F. Huang, Z. Wang, Well posedness for pressureless flow, Commun. Math. Phys. 222 (1) (2001) 117–146.

[27] L. Isherwood, Z.J. Grant, S. Gottlieb, Strong stability preserving integrating factor Runge-Kutta methods, SIAM J. Numer. Anal. 56 (2018) 3276–3307.

[28] G. Jiang, C.-W. Shu, Efficient implementation of weighted ENO schemes, J. Comput. Phys. 126 (1996) 202–228.

[29] L. Ju, X. Li, Z. Qiao, J. Yang, Maximum bound principle preserving integrating factor Runge-Kutta methods for semilinear parabolic equations, J. Comput. Phys. 439 (2021) 110450.

[30] S. Kaplan, On the growth of solutions of quasilinear parabolic equations, Commun. Pure Appl. Math. 16 (1963) 305–330.

[31] E.F. Keller, L.A. Segel, Initiation of some mold aggregation viewed as an instability, J. Theor. Biol. 26 (1970) 399–415.

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

[33] E.F. Keller, L.A. Segel, Traveling bands of chemotactic bacteria: a theoretical analysis, J. Theor. Biol. 30 (1971) 377–380.

[34] X. Li, C.W. Shu, Y. Yang, Local discontinuous Galerkin method for the Keller-Segel chemotaxis model, J. Sci. Comput. 73 (2–3) (2017) 943–967.

[35] T. Nagai, Blow-up of radially symmetric solutions to a chemotaxis system, Adv. Math. Sci. Appl. 5 (2) (1995).

[36] T. Nakagawa, Blowing up of a finite difference solution to $u_t = u_{xx} + u^2$, Appl. Math. Optim. 2 (4) (1975) 337–350.

[37] T. Nakagawa, T. Ushijima, Numerical analysis of the semi-linear equation of blow-up type, Publ. Math. Inform. Rennes S5 (1976) 1–24.

[38] Nakanishi, N. Saito, Finite element method for radially symmetric solution of a multidimensional semilinear heat equation, Jpn. J. Ind. Appl. Math. 37 (2020) 165–191.

[39] C.S. Patlak, Random walk with persistence and external bias, Bull. Math. Biol. 15 (1953) 311–338.

[40] N. Saito, Conservative upwind finite-element method for a simplified Keller-Segel system modelling chemotaxis, IMA J. Numer. Anal. 27 (2) (2007) 332–365.

[41] C.W. Shu, Total-variation-diminishing time discretizations, SIAM J. Sci. Stat. Comput. 9 (1998) 1073–1084.

[42] C.W. Shu, S. Osher, Efficient implementation of essentially non-oscillatory shock-capturing schemes, J. Comput. Phys. 77 (1988) 439–471.

[43] R. Strehl, A. Sokolov, D. Kuzmin, et al., A positivity-preserving finite element method for chemotaxis problems in 3D, J. Comput. Appl. Math. 239 (2013) 290–303.

[44] R. Strehl, A. Sokolov, D. Kuzmin, et al., A flux-corrected finite element method for chemotaxis problems, Comput. Methods Appl. Math. 10 (2) (2021) 219–232.

[45] P.K. Sweby, High resolution schemes using flux limiters for hyperbolic conservation laws, SIAM J. Numer. Anal. 21 (5) (1984) 995–1011.

[46] S. Wang, Z. Xu, Total variation bounded flux limiters for high order finite difference schemes solving one-dimensional scalar conservation laws, Math. Comput. 88 (316) (2018) 691–716.

[47] E. Weinan, Y.G. Rykov, Y.G. Sinai, Generalized variational principles, global weak solutions and behavior with random initial data for systems of conservation laws arising in adhesion particle dynamics, Commun. Math. Phys. 177 (1996) 349–380.

[48] Z. Xu, Parametrized maximum principle preserving flux limiters for high order schemes solving hyperbolic conservation laws: one-dimensional scalar problem, Math. Comput. 83 (289) (2014) 2213–2238.

[49] Y. Yang, D. Wei, C.W. Shu, Discontinuous Galerkin method for Krauses consensus models and pressureless Euler equations, J. Comput. Phys. 252 (2013) 109–127.

[50] X. Zhang, C.W. Shu, On positivity preserving high order discontinuous Galerkin schemes for compressible Euler equations on rectangular meshes, J. Comput. Phys. 229 (2010) 8918–8934.