A SECOND-ORDER BOUND-PRESERVING EXPONENTIAL SCHEME FOR DEGENERATE PARABOLIC EQUATIONS *

CHANG CHEN† AND CHI-WANG SHU‡

Abstract. This paper proposes a second-order numerical method for solving nonlinear parabolic equations with degenerate mobility. The intrinsic degenerate mobility in the equation yields a globally bounded solution. A pivotal feature of our methodology is an appropriate reformulation of the equation into an equivalent form. After applying a discontinuous Galerkin spatial discretization method, we derive a fully nonlinear ordinary differential equation (ODE) with a splitting structure. By introducing a linear term into the ODE, an exponential temporal discretization method, which involves only linear solvers, is proposed based on integrating factors and strong stability preserving (SSP) Runge-Kutta methods. Our approach is proven to exhibit second-order accuracy, ensures bound preservation and mass conservation, and demonstrates a favorable CFL condition $\tau \sim h$, where τ and h are the temporal and spatial mesh sizes respectively. Comprehensive numerical tests validate the second-order accuracy and bound-preserving behaviors of our method.

Key words. bound preservation, discontinuous Galerkin method, exponential scheme, SSP Runge-Kutta, degenerate parabolic equations

MSC codes. 65M12, 65M60, 35K65

1. Introduction. In this paper, we focus on designing numerical schemes for solving a class of degenerate parabolic equations of the form

$$\begin{cases} \rho_t = \nabla \cdot \left(f(\rho) \nabla \left(H'(\rho) + V(\boldsymbol{x}) + W * \rho \right) \right), \quad \boldsymbol{x} \in \Omega, \quad t > 0, \\ \rho(\boldsymbol{x}, 0) = \rho_0(\boldsymbol{x}), \quad \boldsymbol{x} \in \Omega. \end{cases} \tag{1.1}$$

Here $\Omega \subset \mathbb{R}^d$ with $d = 1$ or 2, and $\rho(\boldsymbol{x}, t) : \Omega \times [0, +\infty) \to \mathbb{R}$ is the unknown particle density depending on time and space; $f(\rho)$ is a nonlinear degenerate mobility function that may cause solutions to exhibit boundedness or non-smoothness; the term $H'(\rho)$, the derivative of a sufficiently smooth function $H(\rho)$, can be seen as an internal energy term. The function $V(\boldsymbol{x})$ is an external potential and the term $W * \rho$ is a non-local interaction term where W is an interaction kernel. In this work, we assume that $H(\rho)$ is convex and $W(\boldsymbol{x}) = W(-\boldsymbol{x})$. The equation (1.1) can be written as a gradient flow

$$\rho_t = \nabla \cdot \left(f(\rho) \nabla \left(\frac{\delta E(\rho)}{\delta \rho} \right) \right)$$

with respect to the free energy functional

$$E(\rho) = \int_\Omega H(\rho) + V\rho + \frac{1}{2}(W * \rho)\rho \, d\boldsymbol{x}.$$

The equation (1.1) has been utilized extensively across a variety of fields such as gas interactions [12], granular material behavior [4], cell migration, and chemotaxis phenomena in biology [9], as well as population swarming [7]. Numerous renowned equations can be considered as specific instances of equation (1.1). For example, when

* **Funding:** The second author was supported by NSF grant DMS-2309249.

†Yau Mathematical Sciences Center, Tsinghua University, Beijing, 100084, China. (cc20@mails.tsinghua.edu.cn).

‡Division of Applied Mathematics, Brown University, Providence, RI 02912, USA. (chi-wang_shu@brown.edu).

$f(\rho) = \rho$, the equation is reduced to the Wasserstein flow of the energy functional E. To be more specific, it encompasses well-known equations such as the porous medium equation [13] and the aggregation–diffusion equations [3].

The presence of the degenerate mobility $f(\rho)$ introduces a complex layer of intricacy to the equation. Unlike its constant case, this type of mobility exerts a substantial influence over the global bound and continuity of solutions (see Section 4). Consider the mobility function $f(\rho) = \rho(1-\rho)$. It remains positive within the interval (0,1) and becomes degenerate at the endpoints 0 and 1. This characteristic ensures that the solution to equation (1.1) with this specific mobility always stays confined within the range $[0, 1]$ as long as the initial $\rho_0 \in [0, 1]$ (such as [19]). For our subsequent analysis, we make the assumption, without loss of generality, that the exact solution ρ of equation (1.1) adheres to the condition that

$$\rho(\boldsymbol{x}, t) \in [0, 1] \text{ for all } (\boldsymbol{x}, t) \in \Omega \times [0, +\infty) \tag{1.2}$$

and $f(\rho) \geq 0$ for all $\rho \in [0, 1]$. The restriction regarding the nonnegativity of mobility $f(\rho)$ is necessary for the well-posedness of model (1.1). Given this framework, bound preservation of numerical schemes becomes important when attempting to resolve (1.1), as the model become ill-posed if values fall outside of the prescribed bounds.

There are extensive bound-preserving and high-order spatial numerical methods for equation (1.1), such as the finite-difference method [28], finite-volume method [3, 5], discontinuous Galerkin method [25, 34] and references therein. For temporal discretization of (1.1), the implicit methods are commonly used to maintain stability and preserve physical properties [3, 14, 29]. However, linear implicit schemes (such as classical implicit Runge-Kutta methods and linear multistep methods) are usually limited to first-order in time to unconditionally maintain boundness [33] and necessitate the solution of a large non-linear algebraic system at each time-step. Explicit temporal discretizations are computationally efficient at each iteration, but a parabolic CFL condition $\tau \sim h^2$, where τ and h are temporal and spatial mesh sizes respectively, is normally required for explicit schemes [34]. To strike a balance between efficiency and stability, implicit-explicit (IMEX) methods are frequently utilized when the equation exhibits a splitting structure. The work [10] rewrote the model (1.1) as a splitting form

$$\rho_t = \underbrace{\Delta\Phi(\rho)}_{\mathcal{C}(\rho)} + \underbrace{\nabla \cdot \left(f(\rho)\nabla\left(V(\boldsymbol{x}) + W * \rho\right)\right)}_{\mathcal{D}(\rho)} = \mathcal{C}(\rho) + \mathcal{D}(\rho) \tag{1.3}$$

with $\Phi(\rho) = \int_0^\rho f(s)H''(s)\, ds$. Then, they dealt implicitly with $\mathcal{C}(\rho)$ and explicitly with $\mathcal{D}(\rho)$ following the idea as in [8]. However, nonlinear systems were still required to be solved using this splitting, and the preservation of positivity was only proven for the first-order temporal scheme. This issue of limited accuracy is pervasive in the development of bound-preserving schemes using classical IMEX-RK schemes, owing to the infeasibility of implicit SSP-RK schemes with an order higher than one while having an infinite SSP coefficient [21]. Recently, the IMEX-RK incorporating multi-derivatives [16, 20, 24] has offered the potential to obtain high-order bound-preserving schemes. However, the assumption of high-order derivatives is not easily applicable to the equation (1.1). Furthermore, a variety of novel techniques have been developed for bound preservation, including Lagrange multiplier approaches [15, 36] and the energy variational approach [18]. These approaches primarily confront challenges in solving fully nonlinear implicit schemes or conducting rigorous error analysis for general models.

An alternative prevalent way to explore high-order bound-preserving schemes is by employing the exponential integrator (see [22] for a review). To the best of our knowledge, no previous work has studied an exponential bound-preserving scheme for (1.1), but numerous research studies have explored similar models from the insights of exponential integrators. In contrast to (1.3), we reformulate (1.1) with a distinct splitting structure

$$\rho_t = \underbrace{\nabla \cdot (F(\rho)\nabla\rho)}_{\mathcal{L}(\rho)\rho} + \underbrace{\nabla \cdot (f(\rho)\nabla (V(\boldsymbol{x}) + W * \rho))}_{\mathcal{N}(\rho)} = \mathcal{L}(\rho)\rho + \mathcal{N}(\rho), \tag{1.4}$$

where $F(\rho) = f(\rho)H^{''}(\rho) \geq 0$. By introducing a term $L\rho$ as an approximation of $\mathcal{L}(\rho)\rho$, the equation (1.4) can be further written as

$$\left(e^{-\mathcal{T}(t)}\rho\right)_t = e^{-\mathcal{T}(t)}\left((\mathcal{L}(\rho) - L)\rho + \mathcal{N}(\rho)\right). \tag{1.5}$$

where $\mathcal{T}(t) = \int_0^t L(s)\, ds$ is the integrating factor associated with L. The work [25] took $L = \mu I$ with a constant $\mu > 0$, which makes its computation similar to that of explicit methods. However, their scheme required the parabolic CFL condition $\tau \sim h^2$ analogous to explicit methods when attempting to apply it to our equation (1.4). In cases where $\mathcal{L}(\rho)$ is independent of ρ, several exponential linear schemes employing $L = \mathcal{L}$ have been effectively utilized for (1.4) to generate bounded numerical solutions [17, 26, 27]. In these works, the operator $\mathcal{L}$ possesses a specific special structure, resulting in a symmetry of either Toeplitz matrices or circulant matrices. Consequently, FFT-based algorithms are adequate to achieve enhanced efficiency. Moreover, the very recent study [11] also implemented a fixed constant approximation to $\mathcal{L}$ that depends only on the spatial vector $\boldsymbol{x}$, although the issue of bound-preservation has not been addressed in their context. For $\mathcal{L}(\rho)$ depending on ρ, the work [23] introduced a second-order exponential scheme that incorporates $L = \mathcal{L}(\rho)$. This scheme needed three to four evaluations of the fully nonlinear equation $\rho_t = L(\rho)\rho$ per iteration. When applied to our model (1.4), the computational burden generally resembles that of implicit methods, despite their provision of detailed techniques for distinct stiff kinetic equations.

Consequently, it remains a significant challenge to compute a bound-preserving solution to (1.1) while achieving both high-order accuracy and efficiency. In this paper, we adopt the splitting (1.4) and present a second-order scheme using an exponential integrator associated with $L = \mathcal{L}(\rho^*)$, where ρ^* is explicitly given. At each iteration, our scheme only requires three times of computation for the linear equation $\rho_t = L(\rho^*)\rho$, which can be efficiently evaluated by numerous existing algorithms (see the review [30]). Importantly, we prove that our method is mass-conservative, bound-preserving and benefits from a favorable CFL condition $\tau \sim h$.

2. Spatial discretization. In this section, we present a spatial discretization method for parabolic equations (1.1) using a discontinuous Galerkin (DG) approach. We only consider the one-dimensional case ($d = 1$) as an demonstration and the two-dimensional case ($d = 2$) can be derived in the same way (see [34, Section 3]). In our following statement, we denote $x = \boldsymbol{x}$ as a spatial variable in one-dimensional space.

Let $I_i = (x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}})$ and $I = \cup_{i=1}^N I_i$ be a partition of the domain Ω. For simplicity, we consider uniform meshes $h = x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}$, but this assumption is not essential. The discontinuous piecewise polynomial space is defined as

$$V_h = \{v_h : v_h|_{I_i} \in P^k(I_i), i = 1, 2 \cdots, N\},$$

where $P^k(I_i)$ is the space of k-th order polynomial. We use the notation v_h^+ and v_h^- as the right and left limit of $v_h \in V_h$ respectively. For a function $s = s(\rho)$ or $s = s(\rho, x)$, we denote by $s_h = s(\rho_h)$ or $s_h = s(\rho_h, x)$ respectively. Furthermore, the notation s_h^+ stands for $s(\rho_h^+)$ or $s(\rho_h^+, x)$, and s_{h^-} stands for $s(\rho_h^-)$ or $s(\rho_h^-, x)$. For the purpose of this paper, we only consider the second-order DG scheme ($k = 1$) and choose

$$\phi_i^{(1)} = \frac{x_{i+\frac{1}{2}} - x}{h}, \quad \phi_i^{(2)} = \frac{x - x_{i-\frac{1}{2}}}{h}, \tag{2.1}$$

as the basis for $P^1(I_i)$. Note that $x_{i\pm\frac{1}{2}}$ are the $k+1$ Gauss–Lobatto quadrature points on I_i when $k = 1$. The Gauss–Lobatto quadrature on $x_{i\pm\frac{1}{2}}$ can be defined by: for $\eta, \zeta \in V_h$

$$\widetilde{\int}_{I_i} \eta\zeta \, dx = \frac{h}{2}\left((\eta\zeta)^+_{i-\frac{1}{2}} + (\eta\zeta)^-_{i+\frac{1}{2}}\right)$$

and

$$\widetilde{\int}_{I_i} \eta\partial_x\zeta \, dx = \frac{h}{2}\left((\eta\partial_x(\mathcal{I}\zeta))^+_{i-\frac{1}{2}} + (\eta\partial_x(\mathcal{I}\zeta))^-_{i+\frac{1}{2}}\right),$$

where the operator $\mathcal{I}$ returns the first-order polynomial interpolating at $x_{i\pm\frac{1}{2}}$. As a convention, $\widetilde{\int}_\Omega$ stands for $\sum_i \widetilde{\int}_{I_i}$.

To define the DG method, we first introduce auxiliary variables to split the original problem (1.4) into the following system of first-order equations:

$$\begin{aligned}
\rho_t &= \partial_x\left(F(\rho)\eta\right) + \partial_x\left(f(\rho)u\right),\\
\eta &= \partial_x\rho,\\
u &= \partial_x\xi,\\
\xi &= V(x) + W * \rho.
\end{aligned}$$

A periodic or compactly supported boundary condition is considered in this work, but our approach can be extended to zero-flux (inflow) and Dirichlet (outflow) boundary conditions, and more complex boundary conditions. Then our DG approximation can be described as: Find $\rho_h, \eta_h, u_h, \xi_h \in V_h$ such that for any $\phi_h, \psi_h, \varphi_h \in V_h$,

(2.2)

$$\begin{aligned}
\widetilde{\int}_{I_i} (\rho_h)_t\phi_h \, dx &= -\widetilde{\int}_{I_i} (F_h\eta_h)\partial_x\phi_h \, dx + (\widehat{F_h\eta_h})_{i+\frac{1}{2}}(\phi_h)^-_{i+\frac{1}{2}} - (\widehat{F_h\eta_h})_{i-\frac{1}{2}}(\phi_h)^+_{i-\frac{1}{2}}\\
&\quad - \widetilde{\int}_{I_i} (f_hu_h)\partial_x\phi_h \, dx + (\widehat{f_hu_h})_{i+\frac{1}{2}}(\phi_h)^-_{i+\frac{1}{2}} - (\widehat{f_hu_h})_{i-\frac{1}{2}}(\phi_h)^+_{i-\frac{1}{2}}\\
\widetilde{\int}_{I_i} \eta_h\psi_h \, dx &= -\widetilde{\int}_{I_i} \rho_h\partial_x\psi_h \, dx + (\widehat{\rho_h})_{i+\frac{1}{2}}(\psi_h)^-_{i+\frac{1}{2}} - (\widehat{\rho_h})_{i-\frac{1}{2}}(\psi_h)^+_{i-\frac{1}{2}}\\
\widetilde{\int}_{I_i} u_h\varphi_h \, dx &= -\widetilde{\int}_{I_i} \xi_h\partial_x\varphi_h \, dx + (\widehat{\xi_h})_{i+\frac{1}{2}}(\varphi_h)^-_{i+\frac{1}{2}} - (\widehat{\xi_h})_{i-\frac{1}{2}}(\varphi_h)^+_{i-\frac{1}{2}}\\
(\xi_h)_i &= V(x_i) + (W * \rho_h)(x_i)
\end{aligned}$$

When W is smooth, the convolution can be approximated by

$$(W * \rho_h)(x_i) \approx \widetilde{\int}_\Omega W(x_i - y)\rho_h(y) \, dy.$$

The numerical fluxes are chosen in the following way

$$\widehat{F_h\eta_h} = (F_h\eta_h)^+, \widehat{\rho_h} = \rho_h^- \quad \text{or} \quad \widehat{F_h\eta_h} = (F_h\eta_h)^-, \widehat{\rho_h} = \rho_h^+, \tag{2.3a}$$

$$\widehat{f_hu_h} = \frac{1}{2}\left((f_hu_h)^+ + (f_hu_h)^- + \alpha(g^+ - g^-)\right), \quad \alpha = \max\{|u_h|^+, |u_h|^-\}, \tag{2.3b}$$

$$\widehat{\xi_h} = \frac{1}{2}\left(\xi_h^+ + \xi_h^-\right), \tag{2.3c}$$

where g is chosen to satisfy $\mathrm{sign}[g_h] = \mathrm{sign}[f_h]$ or $\mathrm{sign}[g_h] = 0$, and $\alpha g \pm fu \in [0, \alpha]$ when all $\rho \in [0, 1]$. For the case $f = \rho(1-\rho)$, we can take $g = \rho$. Using the fact $\alpha \geq |u|$, it is straightforward to confirm that $\alpha g \pm fu \in [0, \alpha]$ for $\rho \in [0, 1]$. Here α is a key parameter associated with bound preservation, as discussed in Lemma 3.1.

Remark 2.1. If only positivity preservation is required, g in (2.3) can be relaxed to satisfy $\alpha g \pm fu \geq 0$ as suggested by [34].

Remark 2.2. In this work, we employ the Lax-Friedrichs (LF)-type flux due to its simplicity, widespread use, and ease of implementation. Moreover, bound-preserving flux limiters are well-studied for LF-type fluxes [37], which is helpful for our following analysis. However, our results can also be obtained using alternative fluxes, such as the Harten–Lax–van Leer (HLL)-type flux [35].

Subsequently, we reformulate (2.2) into a vector ODE corresponding to the values on all Gauss-Lobatto nodes. Let

$$\rho_h(x,t) = \rho_i^{(1)}(t)\phi_i^{(1)}(x) + \rho_i^{(2)}(t)\phi_i^{(2)}(x), \quad x \in I_i,\ t \geq 0, \tag{2.4}$$

where $\phi_i^{(1)}$, $\phi_i^{(2)}$ are given in (2.1). Define $\boldsymbol{\rho}_h(t) = (\rho_1^{(1)}(t), \rho_1^{(2)}(t), \ldots, \rho_N^{(1)}(t), \rho_N^{(2)}(t))^\top$ as the vector representing the values of ρ_h on all Gauss-Lobatto nodes. By employing the basis functions $\phi_i^{(1)}$ and $\phi_i^{(2)}$ in the DG discretization (2.2), we can derive an ODE of the following form:

$$(\boldsymbol{\rho}_h)_t = \mathcal{L}_h(\boldsymbol{\rho}_h)\boldsymbol{\rho}_h + \mathcal{N}_h(\boldsymbol{\rho}_h), \tag{2.5}$$

where $\mathcal{L}_h(\boldsymbol{\rho}_h) \in \mathbb{R}^{2N\times 2N}$ is a matrix linked to the discretization of $\partial_x F(\rho)\partial_x$, and $\mathcal{N}_h(\boldsymbol{\rho}) \in \mathbb{R}^{2N}$ is a vector associated with the discretization of $\partial_x(f(\rho)u)$. Employing the flux (2.3a) along with periodic boundary conditions, two non-symmetric operator $\mathcal{L}_h(\boldsymbol{\rho})$ are obtained with off-diagonal elements involving difference $F(\rho_i^{(1)}) - F(\rho_i^{(2)})$ or $F(\rho_i^{(2)}) - F(\rho_i^{(1)})$, which can be negative. In this work, we average the two non-symmetric operators to obtain a symmetric one

(2.6)

$$\mathcal{L}_h(\boldsymbol{\rho}_h) = \frac{1}{h^2}\begin{pmatrix}
d_1^1 & 0 & F_1^2 & & & & & & F_N^2 & a_1 \\
0 & d_1^2 & a_2 & F_2^1 & & & & & & F_1^1 \\
F_1^2 & a_2 & d_2^1 & 0 & F_2^2 & & & & & \\
 & F_2^1 & 0 & d_2^2 & a_3 & F_3^1 & & & & \\
 & & \ddots & \ddots & \ddots & \ddots & \ddots & & & \\
 & & & F_{i-1}^2 & a_i & d_i^1 & 0 & F_i^2 & & \\
 & & & & F_i^1 & 0 & d_i^2 & a_{i+1} & F_{i+1}^1 & \\
 & & & & & \ddots & \ddots & \ddots & \ddots & \\
F_N^2 & & & & & & F_{N-1}^2 & a_{N-1} & d_N^1 & 0 \\
a_1 & F_1^1 & & & & & & F_N^1 & 0 & d_N^2
\end{pmatrix}_{2N\times 2N}$$

where $F_i^\ell = F(\rho_i^{(\ell)})$ and

$$d_i^1 = -(F_i^2 + F_i^1 + 2F_{i-1}^2), \quad d_i^2 = -(2F_{i+1}^1 + F_i^2 + F_i^1), \quad a_i = F_i^1 + F_{i-1}^2.$$

Consequently, using the fact $F = fH'' \geq 0$, one obtains that $\mathcal{L}_h(\boldsymbol{\rho}_h) = (\ell_{ij})_{2N\times 2N}$ is symmetric and has the following properties:

- Zero row sums: $\mathcal{L}_h(\boldsymbol{\rho}_h)\mathbf{1} = 0$ with $\mathbf{1} = (1, 1, \cdots, 1)^\top \in \mathbb{R}^{2N}$.
- Pattern of signs: $\ell_{ij} \leq 0$ if $i = j$, and $\ell_{ij} \geq 0$ if $i \neq j$.

Matrices exhibiting both of these characteristics are known as graph Laplacians [6], which lead to an ODE that guarantees both mass conservation and bound preservation. For a vector $\boldsymbol{\rho}$, we denote $\boldsymbol{\rho} \in [0, 1]$ or $0 \leq \boldsymbol{\rho} \leq 1$ to indicate that every element of $\boldsymbol{\rho}$ falls within the range of 0 to 1. Based on this notation, we present the following classical conclusion:

LEMMA 2.3 ([6, Proposition 1.1]). *Let the matrix $\mathcal{L}_h(\boldsymbol{\rho}_h)$ be defined as* (2.6) *and $\boldsymbol{\rho}(t)$ be a solution to the ODE*

$$\boldsymbol{\rho}_t = \mathcal{L}_h(\boldsymbol{\rho}_h)\boldsymbol{\rho}, \quad \boldsymbol{\rho}(0) = \boldsymbol{\rho}_0 \tag{2.7}$$

with $\boldsymbol{\rho}_0 \in [0, 1]$. Then for all $t \geq 0$, $\boldsymbol{\rho}(t) \in [0, 1]$ and $\mathbf{1}^\top\boldsymbol{\rho}(t) = \mathbf{1}^\top\boldsymbol{\rho}_0$.

Alternatively, we can express the solution of the ODE (2.7) using an exponential matrix as

$$\boldsymbol{\rho}(t) = e^{t\mathcal{L}_h(\boldsymbol{\rho}_h)}\boldsymbol{\rho}_0.$$

Therefore, Lemma 2.3 demonstrates that the exponential matrix $e^{t\mathcal{L}_h(\boldsymbol{\rho}_h)}$ ensures both mass conservation and bound preservation.

3. Temporal discretization. In this section, we present first- and second-order bound-preserving temporal discretizations for the ODE (2.5). Our schemes are designed by incorporating an auxiliary linear term independent of ρ_h and then employing an exponential integrator. For simplicity, the time domain is discretized using equispaced points with time-stepping $\tau > 0$, and we define the nth point given by $t_n = n\tau$ $(n = 0, 1, 2, \cdots)$. Considering the ODE (2.5) on the interval $[t_n, t_{n+1}]$, we reformulate it as

$$(\boldsymbol{\rho}_h)_t = L_p(t)\boldsymbol{\rho}_h + \mathcal{N}_h(\boldsymbol{\rho}_h) + (\mathcal{L}_h(\boldsymbol{\rho}_h) - L_p(t))\boldsymbol{\rho}_h, \quad (x, t) \in \Omega \times [t_n, t_{n+1}], \tag{3.1}$$

where $L_p(t)$ is a pth-order approximation of $\mathcal{L}_h(\boldsymbol{\rho}_h(t))$ satisfying

$$\mathcal{L}_h(\boldsymbol{\rho}_h(t_n + s)) = L_p(t_n + s) + O(s^p), \qquad s \in [0, \tau]. \tag{3.2}$$

Define the integrating factor

$$\mathcal{T}(t) = \int_0^t L_p(s)\, ds. \tag{3.3}$$

Then, the equation (3.1) can be written as

$$\left(e^{-\mathcal{T}(t)}\boldsymbol{\rho}_h(t)\right)_t = e^{-\mathcal{T}(t)}\left(\mathcal{N}_h(\boldsymbol{\rho}_h(t)) + (\mathcal{L}_h(\boldsymbol{\rho}_h(t)) - L_p(t))\boldsymbol{\rho}_h(t)\right) \tag{3.4}$$

Define $\boldsymbol{w}(t) = e^{-\mathcal{T}(t)}\boldsymbol{\rho}_h(t)$. We get

$$(\boldsymbol{w}(t))_t = e^{-\mathcal{T}(t)}\left(\mathcal{N}_h(\boldsymbol{\rho}_h(t)) + (\mathcal{L}_h(\boldsymbol{\rho}_h(t)) - L_p(t))\boldsymbol{\rho}_h(t)\right) = H(\boldsymbol{w}, t). \tag{3.5}$$

Next, we introduce first- and second-order bound-preserving schemes for (3.5) using SSP-RK methods [21]. It is noteworthy that our proposed schemes are simplified to the exponential SSP-RK methods [26] when $\mathcal{L}_h(\boldsymbol{\rho}_h)$ is a constant matrix independent of $\boldsymbol{\rho}_h$.

3.1. First-order scheme. Consider the approximation $L_0(t) = \mathcal{L}_h(\boldsymbol{\rho}_h(t_n))$. Then the integrating factor (3.3) becomes $\mathcal{T}(t) = t\mathcal{L}_h(\boldsymbol{\rho}_h(t_n))$. By applying the first-order forward Euler scheme to (3.5), we derive the scheme

$$\boldsymbol{\rho}_h^{n+1} = e^{\tau \mathcal{L}_h(\boldsymbol{\rho}_h^n)} \left(\boldsymbol{\rho}_h^n + \tau \mathcal{N}_h(\boldsymbol{\rho}_h^n)\right), \tag{3.6}$$

which is equivalent to

$$\boldsymbol{\rho}_h^{n+1,1} = \boldsymbol{\rho}_h^n + \tau \mathcal{N}_h(\boldsymbol{\rho}_h^n) \tag{3.7a}$$

$$\boldsymbol{\rho}_h^{n+1} = e^{\tau \mathcal{L}_h(\boldsymbol{\rho}_h^n)} \boldsymbol{\rho}_h^{n+1,1}. \tag{3.7b}$$

Let $\boldsymbol{\rho}_h(t)$ be an exact solution of (2.5). Substituting $\boldsymbol{\rho}_h^n = \boldsymbol{\rho}_h(t_n)$ into (3.6), one has

$$\begin{aligned}
& e^{\tau \mathcal{L}_h(\boldsymbol{\rho}_h(t_n))} \left(\boldsymbol{\rho}_h(t_n) + \tau \mathcal{N}_h(\boldsymbol{\rho}_h(t_n))\right) \\
= & \left(I + \tau \mathcal{L}_h(\boldsymbol{\rho}_h(t_n)) + O(\tau^2)\right) \left(\boldsymbol{\rho}_h(t_n) + \tau \mathcal{N}_h(\boldsymbol{\rho}_h(t_n))\right) \\
= & \left(\boldsymbol{\rho}_h(t_n) + \tau \left(\mathcal{L}_h(\boldsymbol{\rho}_h(t_n))\boldsymbol{\rho}_h(t_n) + \mathcal{N}_h(\boldsymbol{\rho}_h(t_n))\right)\right) + O(\tau^2) \\
= & \left(\boldsymbol{\rho}_h(t_n) + \tau \left.\frac{d\boldsymbol{\rho}_h(t)}{dt}\right|_{t=t_n}\right) + O(\tau^2).
\end{aligned}$$

Using the Taylor's expansion, the local truncation error of the scheme (3.6) is

$$\boldsymbol{\rho}_h(t_{n+1}) - e^{\tau \mathcal{L}_h(\boldsymbol{\rho}_h(t_n))} \left(\boldsymbol{\rho}_h(t_n) + \tau \mathcal{N}_h(\boldsymbol{\rho}_h(t_n))\right) = O(\tau^2).$$

Hence. the scheme (3.6) is a first-order temporal discretization for (2.5).

Alternatively, we can express the scheme (3.7) in a weak form: Find $\rho_h^{n+1,1}, \rho_h^{n+1} \in V_h$ such that for all $\phi_h, \psi_h \in V_h$,

$$\widetilde{\int}_{I_i} \frac{\rho_h^{n+1,1} - \rho_h^n}{\tau} \phi_h \, dx = -\widetilde{\int}_{I_i} (f_h^n u_h^n) \partial_x \phi_h \, dx + (\widehat{f_h^n u_h^n})_{i+\frac{1}{2}} (\phi_h)^-_{i+\frac{1}{2}} - (\widehat{f_h^n u_h^n})_{i-\frac{1}{2}} (\phi_h)^+_{i-\frac{1}{2}}, \tag{3.8a}$$

$$\widetilde{\int}_{I_i} u_h^n \varphi_h \, dx = -\widetilde{\int}_{I_i} \xi_h^n \partial_x \varphi_h \, dx + (\widehat{\xi_h^n})_{i+\frac{1}{2}} (\varphi_h)^-_{i+\frac{1}{2}} - (\widehat{\xi_h^n})_{i-\frac{1}{2}} (\varphi_h)^+_{i-\frac{1}{2}}, \tag{3.8b}$$

$$(\xi_h^n)_i = V(x_i) + (W * \rho_h^n)(x_i), \tag{3.8c}$$

and $\rho_h^{n+1} = (\tilde{\rho}_h^L(\tau) + \tilde{\rho}_h^R(\tau))/2$ with $\tilde{\rho}_h : [0, \tau] \to V_h$ solving the system

$$\widetilde{\int}_{I_i} (\tilde{\rho}_h)_t \phi_h \, dx = -\widetilde{\int}_{I_i} (F_h^n \eta_h) \, \partial_x \phi \, dx + (\widehat{F_h^n \eta_h})_{i+\frac{1}{2}} \phi^-_{i+\frac{1}{2}} - (\widehat{F_h^n \eta_h})_{i-\frac{1}{2}} \phi^+_{i-\frac{1}{2}}, \tag{3.9a}$$

$$\widetilde{\int}_{I_i} \eta_h \psi_h \, dx = -\widetilde{\int}_{I_i} \tilde{\rho}_h \partial_x \psi_h \, dx + (\widehat{\tilde{\rho}_h})_{i+\frac{1}{2}} (\psi_h)^-_{i+\frac{1}{2}} - (\widehat{\tilde{\rho}_h})_{i-\frac{1}{2}} (\psi_h)^+_{i-\frac{1}{2}}, \tag{3.9b}$$

$$\tilde{\rho}_h(0) = \rho_h^{n+1,1}. \tag{3.9c}$$

and $\tilde{\rho}_h^L$, $\tilde{\rho}_h^R$ correspond to the solutions using two different fluxes according to (2.3a).

LEMMA 3.1. *Given ρ_h^n, the $\rho_h^{n+1,1}$ solved in (3.8) satisfies*

- *Mass conservation: $\widetilde{\int}_\Omega \rho_h^{n+1,1} \, dx = \widetilde{\int}_\Omega \rho_h^n \, dx$;*

- *Bound-preservation for the cell average: Assuming that ρ_h^n is within the range $[0,1]$ at the Gauss-Lobatto quadrature, i.e., $\boldsymbol{\rho}_h^n \in [0,1]$, the cell average $(\bar{\rho}_h)_i^{n+1,1} = \frac{1}{h}\widetilde{\int}_{I_i} \rho_h^{n+1,1}\, dx \in [0,1]$ if*

$$\alpha\lambda g \le \rho \text{ and } \alpha\lambda(1-g) \le 1-\rho, \tag{3.10}$$

where α, g are defined in the numerical flux (2.3) and $\lambda = \tau/h$. Specifically, when $g = \rho$, it reduces to the CFL condition

$$\tau \le \frac{h}{\max\limits_{i=1,\cdots,N} \alpha^n_{i\pm\frac{1}{2}}}. \tag{3.11}$$

Proof. Let $\phi_h \equiv 1$ in (3.8a), we obtain that

$$\widetilde{\int}_{I_i} \rho_h^{n+1,1}\, dx = \widetilde{\int}_{I_i} \rho_h^n\, dx + \tau\left((\widehat{f_h^n u_h^n})_{i+\frac{1}{2}} - (\widehat{f_h^n u_h^n})_{i-\frac{1}{2}}\right), \quad i = 1, 2, \cdots, N. \tag{3.12}$$

Summing i from 1 to N, one get

$$\widetilde{\int}_{\Omega} \rho_h^{n+1,1}\, dx = \widetilde{\int}_{\Omega} \rho_h^n\, dx + \tau\left((\widehat{f_h^n u_h^n})_{N+\frac{1}{2}} - (\widehat{f_h^n u_h^n})_{-\frac{1}{2}}\right).$$

The numerical flux at the right boundary is equal to the numerical flux at the left boundary, as required by the periodic boundary condition and the definition of the numerical flux. Therefore, we can conclude that $\widetilde{\int}_I \rho_h^{n+1,1}\, dx = \widetilde{\int}_I \rho_h^n\, dx$.

Multiplying both side of (3.12) with $1/h$, it becomes

$$(\bar{\rho}_h)_i^{n+1,1} = (\bar{\rho}_h)_i^n + \lambda\left((\widehat{f_h^n u_h^n})_{i+\frac{1}{2}} - (\widehat{f_h^n u_h^n})_{i-\frac{1}{2}}\right), \quad i = 1, 2, \cdots, N.$$

Note that the Gauss–Lobatto quadrature is exact for evaluating the cell average $(\bar{\rho}_h)_i^n$ since $(\rho_h)_i^n$ is a first-order polynomial. For simplicity, we omit the indices n and h for the notation ρ_h^n in the subsequent proof. Then

$$\begin{aligned}
(\bar{\rho})_i^{n+1,1} =& \frac{1}{2}\left(\rho^-_{i+\frac{1}{2}} + \rho^+_{i-\frac{1}{2}}\right) + \frac{\lambda}{2}\left((fu)^+_{i+\frac{1}{2}} + (fu)^-_{i+\frac{1}{2}} + \alpha_{i+\frac{1}{2}}(g^+_{i+\frac{1}{2}} - g^-_{i+\frac{1}{2}})\right) \\
& - \frac{\lambda}{2}\left((fu)^+_{i-\frac{1}{2}} + (fu)^-_{i-\frac{1}{2}} + \alpha_{i-\frac{1}{2}}(g^+_{i-\frac{1}{2}} - g^-_{i-\frac{1}{2}})\right) \\
=& \frac{1}{2}\Big[\rho^-_{i+\frac{1}{2}} + \lambda\left((fu)^+_{i+\frac{1}{2}} + (fu)^-_{i+\frac{1}{2}} + \alpha_{i+\frac{1}{2}}(g^+_{i+\frac{1}{2}} - g^-_{i+\frac{1}{2}})\right) \\
& -\lambda\left((fu)^-_{i+\frac{1}{2}} + (fu)^+_{i-\frac{1}{2}} + \alpha_{i+\frac{1}{2}}(g^-_{i+\frac{1}{2}} - g^+_{i-\frac{1}{2}})\right)\Big] + \\
& \frac{1}{2}\Big[\rho^+_{i-\frac{1}{2}} + \lambda\left((fu)^-_{i+\frac{1}{2}} + (fu)^+_{i-\frac{1}{2}} + \alpha_{i+\frac{1}{2}}(g^-_{i+\frac{1}{2}} - g^+_{i-\frac{1}{2}})\right) \\
& -\lambda\left((fu)^+_{i-\frac{1}{2}} + (fu)^-_{i-\frac{1}{2}} + \alpha_{i-\frac{1}{2}}(g^+_{i-\frac{1}{2}} - g^-_{i-\frac{1}{2}})\right)\Big] \\
=& \frac{1}{2}\Big[\lambda\alpha_{i+\frac{1}{2}}K^+(\rho^+_{i+\frac{1}{2}}, \alpha_{i+\frac{1}{2}}) + \lambda\alpha_{i+\frac{1}{2}}K^-(\rho^+_{i-\frac{1}{2}}, \alpha_{i+\frac{1}{2}}) \\
& + \left(\rho^-_{i+\frac{1}{2}} - 2\lambda\alpha_{i+\frac{1}{2}}g^-_{i+\frac{1}{2}}\right)\Big] \\
& \frac{1}{2}\Big[\lambda\alpha_{i-\frac{1}{2}}K^-(\rho^-_{i-\frac{1}{2}}, \alpha_{i-\frac{1}{2}}) + \lambda\alpha_{i+\frac{1}{2}}K^+(\rho^-_{i+\frac{1}{2}}, \alpha_{i+\frac{1}{2}}) \\
& + \left(\rho^+_{i-\frac{1}{2}} - \lambda(\alpha_{i+\frac{1}{2}} + \alpha_{i-\frac{1}{2}})g^+_{i-\frac{1}{2}}\right)\Big],
\end{aligned}$$

where $K^{\pm} = g \pm fu/\alpha$. Note that $K^{\pm} \in [0,1]$ by the choice of g. To make $(\bar{\rho}_h)_i^{n+1,1} \in [0,1]$, it is sufficient to ensure $\rho - \lambda\alpha g \in [0, 1-\lambda\alpha]$, which reduces to the restriction (3.10). ∎

Remark 3.2. In the definition of ξ_h, it is continuous and (3.8b) gives $u_h = \partial_x \mathcal{I}(\xi_h)$ on I_i after integration by parts. Hence the CFL condition (3.11) becomes

$$\tau \leq \frac{h}{\max_i \|\partial_x \mathcal{I}(\xi_h)\|_{L^\infty(I_i)}} \sim h. \tag{3.13}$$

which is a favorable CFL condition for parabolic equations. Particularly, when $W = 0$ and $V = x$, this corresponds to the usual CFL condition $\tau \leq h$ for hyperbolic conservation laws.

Remark 3.3. For efficient implementation, we can approximate the exponential matrix in (3.7b) as

$$\boldsymbol{\rho}_h^{n+1} = \left(I - \tau \mathcal{L}_h(\boldsymbol{\rho}_h^n)\right)^{-1} \boldsymbol{\rho}_h^{n+1,1}, \tag{3.14}$$

which is also a first-order scheme [8]. The bound preservation of this approximation still holds using a sufficient condition [31]: a matrix $M = (m_{ij})$ is inverse positive, meaning that every entry of M^{-1} is non-negative, if $m_{ij} \leq 0$ for $i \neq j$, $m_{ii} > 0$, and M is strictly diagonally dominant ($m_{ii} > \sum_{i \neq j} |m_{ij}|$). The first-order approximation $M = I - \tau \mathcal{L}_h(\boldsymbol{\rho}_h^n)$ unconditionally satisfies this condition since the matrix $\mathcal{L}_h(\boldsymbol{\rho}_h^n)$ is a graph Laplacian, as previously noted. This approximation (3.14) will be employed in our subsequent investigation of the second-order scheme.

As stated in Lemma 3.1, the Euler forward stage (3.7a) only maintains bound-preservation on the cell average, not across all Gauss-Lobatto quadrature points. Following the methodology developed by [37], we can apply a bound-preserving limiter to enforce the boundedness of nodal values on all Gauss–Lobatto quadrature points without violating the mass conservation and accuracy. Precisely, let

$$\rho_h^{n+1,2}(x_{i\pm\frac{1}{2}}^{\mp}) = (\bar{\rho}_h)_i^{n+1,1} + \theta_i \left(\rho_h^{n+1,1}(x_{i\pm\frac{1}{2}}^{\mp}) - (\bar{\rho}_h)_i^{n+1,1}\right),$$

with

$$\begin{aligned} \theta_i &= \min\left\{ \frac{(\bar{\rho}_h)_i^{n+1,1}}{(\bar{\rho}_h)_i^{n+1,1} - m_i}, \frac{1 - (\bar{\rho}_h)_i^{n+1,1}}{M_i - (\bar{\rho}_h)_i^{n+1,1}}, 1 \right\}, \\ m_i &= \min \rho_h^{n+1,1}(x_{i\pm\frac{1}{2}}^{\mp}),\ M_i = \max \rho_h^{n+1,1}(x_{i\pm\frac{1}{2}}^{\mp}). \end{aligned}$$

Then we get $\rho_h^{n+1,2}(x_{i\pm\frac{1}{2}}^{\mp}) \in [0,1]$ and $(\bar{\rho}_h)_i^{n+1,2} = (\bar{\rho}_h)_i^{n+1,1}$ (see [37]). Furthermore, the interpolation polynomial of $\{\rho_h^{n+1,2}(x_{i\pm\frac{1}{2}}^{\mp})\}$ on I_i satisfies mnloodsxwe

$$\left|\rho_h^{n+1,2}(x) - \rho_h^{n+1,1}(x)\right| \leq C_k \max_{x \in \{x_{i\pm\frac{1}{2}}^{\mp}\}} \left|\rho(x, t_{n+1}) - \rho_h^{n+1,1}(x)\right|,$$

where $\rho(x, t_{n+1})$ is the exact solution at time t_{n+1} and C_k is a constant depending only on the polynomial degree k. Consequently, we can update

$$\boldsymbol{\rho}_h^{n+1} = e^{\tau \mathcal{L}(\boldsymbol{\rho}_h^n)} \boldsymbol{\rho}_h^{n+1,2}.$$

and conclude that $\boldsymbol{\rho}_h^{n+1} \in [0,1]$ by Lemma 2.3. Denote the bound limiter as $\mathcal{P}$. The first-order bound-preserving scheme can be summarized as

$$\boldsymbol{\rho}_h^{n+1} = e^{\tau \mathcal{L}(\boldsymbol{\rho}_h^n)} \mathcal{P}\left(\boldsymbol{\rho}_h^n + \tau \mathcal{N}_h(\boldsymbol{\rho}_h^n)\right). \tag{3.15}$$

Using the approximation (3.14), a more efficient first-order scheme is given by

$$\boldsymbol{\rho}_h^{n+1} = \left(I - \tau \mathcal{L}(\boldsymbol{\rho}_h^n)\right)^{-1} \mathcal{P}\left(\boldsymbol{\rho}_h^n + \tau \mathcal{N}_h(\boldsymbol{\rho}_h^n)\right). \tag{3.16}$$

THEOREM 3.4. *The updates* (3.15) *and* (3.16) *are bound preserving and mass conservative, provided the time step restriction specified in* (3.13) *is satisfied.*

3.2. Second-order scheme. Based on the first-order scheme (3.15) or (3.16), we can construct a first-order approximation for $\mathcal{L}_h(\boldsymbol{\rho}_h)$. Specially, let

$$\boldsymbol{\rho}_h^{n+1,1} = \left(I - \tau \mathcal{L}_h(\boldsymbol{\rho}_h^n)\right)^{-1} \left(\boldsymbol{\rho}_h^n + \tau \mathcal{N}_h(\boldsymbol{\rho}_h^n)\right) \tag{3.17}$$

and introduce the linear interpolation

$$L_1(t) = \frac{t - t_n}{\tau} \mathcal{L}_h(\boldsymbol{\rho}_h^{n+1,1}) + \left(1 - \frac{t - t_n}{\tau}\right) \mathcal{L}_h(\boldsymbol{\rho}_h^n), \tag{3.18}$$

which provides a first-order approximation for $\mathcal{L}_h(\boldsymbol{\rho}_h)$ on the interval $[t_n, t_{n+1}]$. Consequently, the integrating factor (3.3) becomes

$$\mathcal{T}(t) = \frac{(t - t_n)^2}{2\tau} \mathcal{L}_h(\boldsymbol{\rho}_h^{n+1,1}) - \frac{\tau}{2} \left(1 - \frac{t - t_n}{\tau}\right)^2 \mathcal{L}_h(\boldsymbol{\rho}_h^n). \tag{3.19}$$

By substituting (3.18) and (3.19) into (3.4) and applying the second-order SSP Runge-Kutta scheme [21, equation (4.1)], we derive the following scheme:

$$\boldsymbol{\rho}_h^{n+1,2} = e^{\tau \tilde{\mathcal{L}}_h^{n+1}} \left(\boldsymbol{\rho}_h^n + \tau \mathcal{N}_h(\boldsymbol{\rho}_h^n)\right), , \tag{3.20a}$$

$$\boldsymbol{\rho}_h^{n+1} = \frac{1}{2} e^{\tau \tilde{\mathcal{L}}_h^{n+1}} \boldsymbol{\rho}_h^n + \frac{1}{2} \left(\boldsymbol{\rho}_h^{n+1,2} + \tau \mathcal{N}_h(\boldsymbol{\rho}_h^{n+1,2})\right), \tag{3.20b}$$

where

$$\tilde{\mathcal{L}}_h^{n+1} = \frac{1}{2}(\mathcal{L}_h(\boldsymbol{\rho}_h^{n+1,1}) + \mathcal{L}_h(\boldsymbol{\rho}_h^n)),$$

and a high-order term $\frac{1}{2}\tau(\mathcal{L}_h(\boldsymbol{\rho}_h^{n+1,2}) - \mathcal{L}_h(\boldsymbol{\rho}_h^{n+1,1}))\boldsymbol{\rho}_h^n$ is dropped in (3.20b).

Remark 3.5. In update (3.17) and (3.20), the stiff term $\mathcal{L}_h(\boldsymbol{\rho}_h)\boldsymbol{\rho}_h$ does not appear in the explicit Euler steps, even though it exists in the ODE (3.5). Hence, the stability of the explicit Euler step only depends on the nonlinear term $\mathcal{N}_h$.

Let $\boldsymbol{\rho}_h(t)$ be an exact solution for (2.5), we next show that (3.20) is second-order in time. Denote $\mathcal{G}(\boldsymbol{\rho}_h) = \mathcal{L}_h(\boldsymbol{\rho}_h)\boldsymbol{\rho}_h + \mathcal{N}_h(\boldsymbol{\rho}_h)$ and $\boldsymbol{\rho} = \boldsymbol{\rho}_h(t_n)$ for simplicity. The Taylor's expansion gives that

(3.21)

$$\begin{aligned}
\boldsymbol{\rho}_h(t_{n+1}) &= \boldsymbol{\rho} + \tau \mathcal{G}(\boldsymbol{\rho}) + \frac{\tau^2}{2} \mathcal{G}_t(\boldsymbol{\rho}) + O(\tau^3) \\
&= \boldsymbol{\rho} + \tau \mathcal{G}(\boldsymbol{\rho}) + \frac{\tau^2}{2} \left(\left(\mathcal{L}_h'(\boldsymbol{\rho})\mathcal{G}(\boldsymbol{\rho})\right)\boldsymbol{\rho} + \mathcal{L}_h(\boldsymbol{\rho})\mathcal{G}(\boldsymbol{\rho})\right) + \tau^2 \mathcal{N}_h'(\boldsymbol{\rho})\mathcal{G}(\boldsymbol{\rho}) + O(\tau^3)
\end{aligned}$$

Next, we substitute the exact solution at t_n into (3.17) and (3.20) to calculate the local truncation error. We start by obtaining $\boldsymbol{\rho}^{(1)}$ by substituting $\boldsymbol{\rho}_h^n = \boldsymbol{\rho}$ into (3.17), yielding

$$\boldsymbol{\rho}^{(1)} = (I - \tau\mathcal{L}(\boldsymbol{\rho}))^{-1}(\boldsymbol{\rho} + \tau\mathcal{N}(\boldsymbol{\rho})) = \boldsymbol{\rho} + \tau\mathcal{G}(\boldsymbol{\rho}) + O(\tau^2),$$
$$\mathcal{L}_h(\boldsymbol{\rho}^{(1)}) = \mathcal{L}_h(\boldsymbol{\rho}) + \tau\mathcal{L}_h'(\boldsymbol{\rho})\mathcal{G}(\boldsymbol{\rho}) + O(\tau^2).$$

Using this, we substitute $\boldsymbol{\rho}_h^n = \boldsymbol{\rho}$ and $\boldsymbol{\rho}_h^{n+1,1} = \boldsymbol{\rho}^{(1)}$ into (3.20), and denote the results from (3.20a) and (3.20b) as $\boldsymbol{\rho}^{(2)}$ and $\boldsymbol{\rho}^{(3)}$, respectively. Defining $\tilde{\mathcal{L}} = \frac{1}{2}(\mathcal{L}_h(\boldsymbol{\rho}^{(1)}) + \mathcal{L}_h(\boldsymbol{\rho}))$, we obtain

$$\begin{aligned}\boldsymbol{\rho}^{(2)} &= \boldsymbol{\rho} + \tau\left(\tilde{\mathcal{L}}\boldsymbol{\rho} + \mathcal{N}_h(\boldsymbol{\rho})\right) + \frac{\tau^2}{2}\tilde{\mathcal{L}}\left(\tilde{\mathcal{L}}\boldsymbol{\rho} + 2\mathcal{N}_h(\boldsymbol{\rho})\right) + O(\tau^3)\\ &= \boldsymbol{\rho} + \tau\mathcal{G}(\boldsymbol{\rho}) + \frac{\tau^2}{2}\mathcal{L}_h(\boldsymbol{\rho})\left(\mathcal{G}(\boldsymbol{\rho}) + \mathcal{N}_h(\boldsymbol{\rho})\right) + \frac{\tau^2}{2}\left(\mathcal{L}_h'(\boldsymbol{\rho})\mathcal{G}(\boldsymbol{\rho})\right)\boldsymbol{\rho} + O(\tau^3)\end{aligned}$$

Similarly, we calculate $\mathcal{N}_h(\boldsymbol{\rho}^{(2)}) = \mathcal{N}_h(\boldsymbol{\rho}) + \tau\mathcal{N}_h'(\boldsymbol{\rho})\mathcal{G}(\boldsymbol{\rho}) + O(\tau^2)$, and for $\boldsymbol{\rho}^{(3)}$, we have

$$\begin{aligned}\boldsymbol{\rho}^{(3)} &= \frac{1}{2}\left(I + \tau\tilde{\mathcal{L}} + \frac{\tau^2}{2}\tilde{\mathcal{L}}^2\right)\boldsymbol{\rho} + \frac{1}{2}\left(\boldsymbol{\rho}^{(2)} + \tau\mathcal{N}_h(\boldsymbol{\rho}^{(2)})\right) + O(\tau^3)\\ &= \boldsymbol{\rho} + \tau\mathcal{G}(\boldsymbol{\rho}) + \frac{\tau^2}{2}\left(\left(\mathcal{L}_h'(\boldsymbol{\rho})\mathcal{G}(\boldsymbol{\rho})\right)\boldsymbol{\rho} + \mathcal{L}_h(\boldsymbol{\rho})\mathcal{G}(\boldsymbol{\rho})\right) + \frac{\tau^2}{2}\mathcal{N}_h'(\boldsymbol{\rho})\mathcal{G}(\boldsymbol{\rho}) + O(\tau^3).\end{aligned}$$

Finally, combining with (3.21), we arrive at

$$\boldsymbol{\rho}_h(t_{n+1}) - \boldsymbol{\rho}^{(3)} = O(\tau^3),$$

which confirms that the scheme (3.20) is a second-order temporal discretization for ODE (2.5). When the bound-preserving limiter is applied immediately after each Euler forward stage, we obtain a second-order bound-preserving scheme

$$\boldsymbol{\rho}_h^{n+1,1} = (I - \tau\mathcal{L}_h(\boldsymbol{\rho}_h^n))^{-1}\mathcal{P}\left(\boldsymbol{\rho}_h^n + \tau\mathcal{N}_h(\boldsymbol{\rho}_h^n)\right), \tag{3.22a}$$

$$\boldsymbol{\rho}_h^{n+1,2} = e^{\frac{\tau}{2}\left(\mathcal{L}_h(\boldsymbol{\rho}_h^n) + \mathcal{L}_h(\boldsymbol{\rho}_h^{n+1,1})\right)}\mathcal{P}\left(\boldsymbol{\rho}_h^n + \tau\mathcal{N}_h(\boldsymbol{\rho}_h^n)\right), \tag{3.22b}$$

$$\boldsymbol{\rho}_h^{n+1} = \frac{1}{2}e^{\frac{\tau}{2}\left(\mathcal{L}_h(\boldsymbol{\rho}_h^n) + \mathcal{L}_h(\boldsymbol{\rho}_h^{n+1,1})\right)}\boldsymbol{\rho}_h^n + \frac{1}{2}\mathcal{P}\left(\boldsymbol{\rho}_h^{n+1,2} + \tau\mathcal{N}_h(\boldsymbol{\rho}_h^{n+1,2})\right). \tag{3.22c}$$

Similar to the first-order scheme (3.15), we have the following theorem for (3.22).

THEOREM 3.6. *The time discretization* (3.22) *of the semi-discrete scheme* (2.5) *is bound preserving and mass conservative as long as the time step restriction* (3.13) *is satisfied.*

Proof. The mass conservation trivially holds. As shown in the derivation of the first-order scheme, given any $\boldsymbol{\rho}_h \in [0,1]$, the step $\mathcal{P}(\boldsymbol{\rho}_h + \tau\mathcal{N}_h(\boldsymbol{\rho}_h))$ remains within the bounds $[0,1]$ when the time step restriction (3.13) is satisfied. Then the desired result is obtained by the unconditional bound-preserving properties of operators $(I - \tau\mathcal{L}_h(\boldsymbol{\rho}_h))^{-1}$ and $e^{-\tau\mathcal{L}_h(\boldsymbol{\rho}_h)}$. ∎

Remark 3.7. For practical implementation, one can use the second-order approximation

$$e^{\tau\mathcal{L}} \approx \left(I - \tau\mathcal{L} + \frac{\tau^2}{2}\mathcal{L}^2\right)^{-1} \tag{3.23}$$

in (3.22) to optimize computational efficiency. However, the guarantee of bound-preservation is compromised due to the inclusion of $\mathcal{L}^2$. A pragmatic approach would be to initially utilize the approximation from (3.23). If a value surpasses the expected density bounds, the respective step should be discarded. Subsequently, one should revert to computing using the exponential matrix.

4. Numerical results. In this section, we examine the performance and accuracy of our proposed numerical schemes (3.22) for computing several examples on domain $\Omega = [-q, q]^d$ with $q > 0$ and $d = 1, 2$. In practical, the technique mentioned in Remark 3.7 is applied to (3.22). The error is measured in the discrete norms

$$
\begin{aligned}
err_{L^1} &= \widetilde{\int_\Omega} |\rho_h(\boldsymbol{x}, t) - \rho(\boldsymbol{x}, t)| \, d\boldsymbol{x} = \left(\frac{h}{2}\right)^d \|\boldsymbol{\rho}_h - \boldsymbol{\rho}\|_{\ell^1}, \\
err_{L^2} &= \sqrt{\widetilde{\int_\Omega} |\rho_h(\boldsymbol{x}, t) - \rho(\boldsymbol{x}, t)|^2 \, d\boldsymbol{x}} = \left(\frac{h}{2}\right)^{\frac{d}{2}} \|\boldsymbol{\rho}_h - \boldsymbol{\rho}\|_{\ell^2}, \\
err_{L^\infty} &= \|\boldsymbol{\rho}_h - \boldsymbol{\rho}\|_\infty.
\end{aligned} \tag{4.1}
$$

Here ρ_h is the numerical solution obtained by the scheme (3.22), and ρ is the exact solution for (1.1) or a reference solution computed by our approach in a finer mesh. Vectors $\boldsymbol{\rho}_h$ and $\boldsymbol{\rho}$ are the values of ρ_h and ρ on all Gauss–Lobatto quadrature points, respectively. We chose $g = \rho$ in the flux (2.3).

While (3.13) provides a theoretical time step restriction, we observed that our methods remain robust and maintain the boundedness even with larger timesteps. To enhance efficiency, we implement an adaptive timestep strategy, commonly used to allow for larger timesteps. Specifically, given $\boldsymbol{\rho}_h^n$, we start with a relaxed timestep $\tau_n = h$. At each explicit Euler step, we verify whether all new cell averages remain within $[0, 1]$. If this condition is satisfied, we proceed with the computation. If not, we revert to initial state $\boldsymbol{\rho}_h^n$ and restart the computation with the timestep τ_n halved. This adaptive timestep strategy is highly efficient, and Lemma 3.1 ensures that only a finite number of reductions are necessary to ensure that cell averages of explicit Euler step stay within $[0, 1]$. We apply this strategy in all subsequent numerical experiments, unless otherwise indicated.

4.1. Computing matrix exponential. As our scheme only requires the evaluation of $e^{-\tau \mathcal{L}_h} \boldsymbol{\rho}$, we employ the method proposed in [1] to efficiently compute the action of the matrix exponential. Specifically, for a matrix $A \in \mathbb{R}^{K \times K}$ and a vector $b \in \mathbb{R}^K$, we compute $e^A b$ by expressing it as

$$
e^A b = \left(e^{s^{-1}A}\right)^s b = \underbrace{e^{s^{-1}A} e^{s^{-1}A} \cdots e^{s^{-1}A}}_{s \text{ times}} b,
$$

where $s \geq 1$ is an integer chosen such that $e^{s^{-1}A}$ can be well approximated by the truncated Taylor series

$$
r_m(s^{-1}A) = \sum_{i=0}^{m} \frac{(s^{-1}A)^i}{i!}.
$$

The algorithm can then be understood as the recurrence relation

$$
b_{j+1} = r_m\left(s^{-1}A\right) b_j, \quad j = 0, 1, \ldots, s-1, \quad b_0 = b.
$$

The computational cost of evaluating $e^A b$ using this recurrence is $s \times m$ matrix-vector products of the form Ab. Given a specific tolerance, [1, Code Fragment 3.1] provides

a method to determine appropriate values for s and m such that the error is below the desired threshold. The upper bound on $s \times m$ is proportional to $\|A^p\|_1^{1/p}$ for some integer $p \geq 1$.

In our scheme, we have $A = -\tau\mathcal{L}_h \in \mathbb{R}^{2N\times 2N}$ and $b = \boldsymbol{\rho}_h \in \mathbb{R}^{2N}$ with N to be the number of spatial partition. According to (2.6), the structure of $\mathcal{L}_h$ suggests that the ℓ_1 norm of $\mathcal{L}_h$ is dependent on h, specially $\|(\mathcal{L}_h)^p\|_1^{1/p} = O(h^{-2})$. Provided a CFL condition $\tau \sim h$ is applied, the number of matrix-vector products in our case is proportional to

$$\|A^p\|_1^{1/p} = \tau\|\mathcal{L}_h^p\|_1^{1/p} = O(h^{-1}) = O(N).$$

Since $\mathcal{L}_h$ is a sparse matrix with $O(N)$ non-zero elements, the computational cost of each matrix-vector product is $O(N)$. As a result, the total computational cost of computing the action of the matrix exponential is $O(N^2)$. If the matrix exponential $e^{-\tau\mathcal{L}_h}$ is given explicitly, the cost of multiplying $e^{-\tau\mathcal{L}_h}$ by a vector $\boldsymbol{\rho}$ is also $O(N^2)$, as $e^{-\tau\mathcal{L}_h}$ is generally a full matrix, even if $\mathcal{L}_h$ is sparse. This implies that the theoretical complexity of the algorithm for computing the matrix exponential action is acceptable within the context of our model. In our numerical experiments, we find that the algorithm consistently demonstrates efficiency with $s \times m \ll N$. Combined with the technique in Remark 3.7, the number of computations involving the exponential matrix can be significantly reduced.

4.2. Accuracy test. We first examine the accuracy of (3.22) on an initial value problem with a source term S:

(4.2)
$$\begin{cases} \rho_t = \nabla\cdot\big(\rho(1-\rho)\nabla(\rho + \sin(\mathbf{1}^\top\boldsymbol{x}) + W * \rho)\big) + S(\boldsymbol{x},t), \quad \boldsymbol{x}\in[-\pi,\pi]^d, t>0, \\ \rho(\boldsymbol{x},0) = \dfrac{1}{4}\left(\sin(\mathbf{1}^\top\boldsymbol{x})+1\right), \end{cases}$$

where $\mathbf{1} = (1,\cdots,1)^\top \in \mathbb{R}^d$ and $W(\boldsymbol{x}) = \cos(\mathbf{1}^\top\boldsymbol{x})/(2\pi)$. Here periodic boundary conditions are applied and the source term S is used to ensure that the exact solution is

$$\rho(\boldsymbol{x},t) = \frac{1}{4}\left(\sin(\mathbf{1}^\top\boldsymbol{x}+t)+1\right). \tag{4.3}$$

In this test, $F = f = \rho(1-\rho)$, $H' = \rho$ and $V = \sin(\mathbf{1}^\top\boldsymbol{x})$. We calculate the error at $T = 1$ with the fixed timestep $\tau = \frac{1}{8\pi}h$. The adaptive timestep strategy is not employed here because our theoretical results regarding the timestep do not incorporate the additional source term S in model (4.2), which means we cannot ensure that the adaptive timestep strategy will stop within a finite number of reductions. Table 1 and Table 2 show the corresponding second-order convergence in one and two dimensions.

TABLE 1
Accuracy test in one dimension for computing a solution to the equation (4.2). *The error is calculated at* $T = 1$.

N	L^1 error	Order	L^2 error	Order	L^∞ error	Order
20	2.074e-02		9.966e-03		8.508e-03	
40	5.032e-03	2.043	2.442e-03	2.029	2.639e-03	1.689
80	1.232e-03	2.030	6.006e-04	2.024	6.788e-04	1.959
160	3.029e-04	2.024	1.463e-04	2.038	1.752e-04	1.954

TABLE 2
Accuracy test in two dimensions for computing a solution to the equation (4.2)*. The error is calculated at* $T = 1$*.*

N	L^1 error	Order	L^2 error	Order	L^∞ error	Order
10×10	9.177e-01		1.884e-01		1.051e-01	
20×20	2.182e-01	2.073	4.636e-02	2.023	3.046e-02	1.787
40×40	5.095e-02	2.098	1.038e-02	2.159	6.019e-03	2.339

4.3. Saturation experiment. To demonstrate the bound-preserving property of our numerical scheme, we consider the saturation experiment given by

$$\begin{cases} \rho_t = \nabla \cdot \left(\rho(1-\rho) \nabla \left(D \ln(\rho) + \frac{C}{2} |\boldsymbol{x}|^2 \right) \right), \\ \rho(\boldsymbol{x}, 0) = \rho_0. \end{cases} \tag{4.4}$$

Here D and C are positive numbers. In this case, $f = \rho(1-\rho)$, $H = D\left(\rho \ln(\rho) - \rho\right)$, $F = fH'' = D(1-\rho)$, $V(x) = \frac{C}{2}|\boldsymbol{x}|^2$ and $W(\boldsymbol{x}) = 0$. The exact solution of (4.4) is bounded on $[0, 1]$ and the steady state depends on the initial mass $m = \|\rho_0\|_{L^1}$. There exists a threshold $m_c = (\frac{2\pi D}{C})^{d/2}$ such that the steady state can be written as

$$\rho_\infty(\boldsymbol{x}) = \begin{cases} A \exp\left(-\frac{C}{2D} |\boldsymbol{x}|^2 \right), & m \le m_c \\ B \exp\left(-\frac{C}{2D} \max\left\{ |\boldsymbol{x}|^2 - \ell^2, 0 \right\} \right), & m > m_c \end{cases} \tag{4.5}$$

where A, B are positive constants such that $\|\rho_\infty\|_{L^1} = \|\rho_0\|_{L^1} = m$, and ℓ can be determined from the initial datum (see reference [2]). Numerically, we solve the equation (4.4) over the domain $\Omega = [-4, 4]^d$ with parameters $C = 1, D = 1$. Consequently, the threshold $m_c = (2\pi)^{d/2}$ in (4.5).

To verify the convergence order, we begin with a uniform initial data $\rho_0 \equiv 0.1$ in one dimension and $\rho_0 \equiv 0.09375$ in two dimensions. This ensures that $\|\rho_0\|_{L^1} \le m_c$ and leads to a smooth steady state according to (4.5). We then calculate the numerical solution at $T = 15$ as the numerical steady state. Figure 1 and Figure 2 illustrate the numerical evolution to the steady state in one and two dimensions, respectively. Furthermore, Table 3 and Table 4 demonstrate the anticipated second-order accuracy in both one and two dimensions. In Figure 3, we present the behavior of the relative entropy $E(t|\infty) = E(\rho_h(t)) - E(\rho_\infty)$ and the associated bound of our solution. The results indicate that our scheme preserves the bound, aligning with our theoretical analysis, and also exhibits energy dissipation during this test.

TABLE 3
Accuracy test in one dimension for computing the steady state to the equation (4.4) *with a uniform initial density* $\rho_0 \equiv 0.1$*. The error is calculated at* $T = 15$*.*

N	L^1 error	Order	L^2 error	Order	L^∞ error	Order
20	8.573e-03		4.186e-03		3.199e-03	
40	2.179e-03	1.976	1.023e-03	2.033	7.885e-04	2.020
80	5.436e-04	2.003	2.475e-04	2.047	1.811e-04	2.122
160	1.383e-04	1.975	6.153e-05	2.008	4.329e-05	2.065

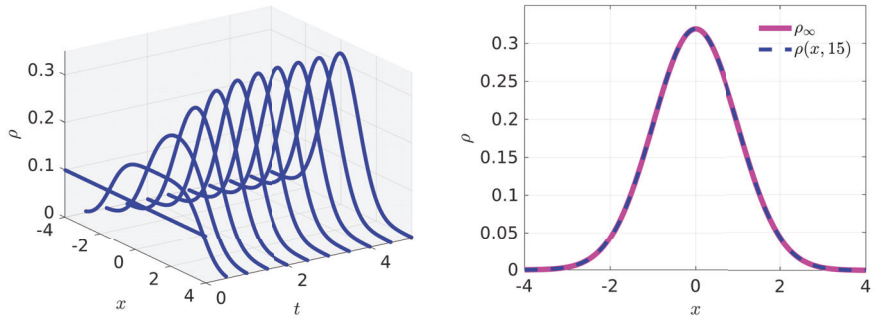

FIG. 1. *Computation of a smooth solution to the one-dimensional saturation experiment* (4.4) *with* $\rho_0 \equiv 0.1$. ***Left:*** *Evolution of* $\rho_h(x,t)$; ***Right:*** *Comparison between* ρ_∞ *and the numerical solution calculated at* $T = 15$;

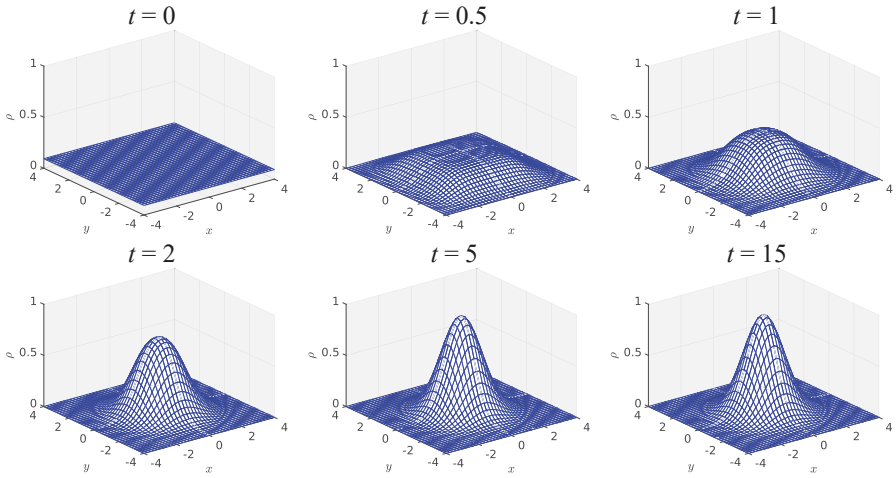

FIG. 2. *Evolution of a solution* $\rho(\boldsymbol{x},t)$ *with* $\rho_0 \equiv 0.09375$ *in a two-dimensional saturation experiment* (4.4).

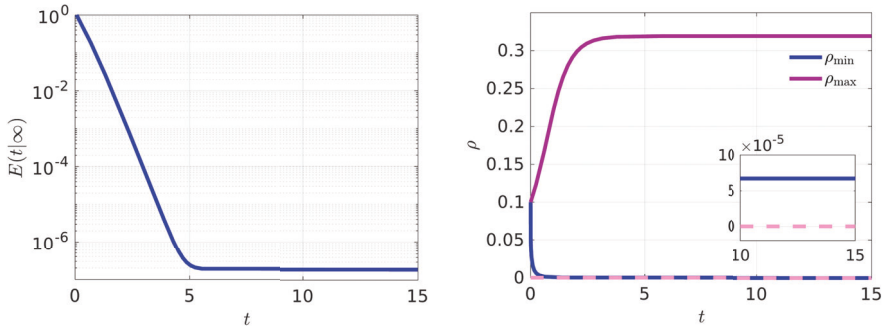

FIG. 3. *Computation of a smooth solution to the one-dimensional saturation experiment* (4.4) *with* $\rho_0 = 0.1$. ***Left:*** *Behaviour of the relative energy* $E(t;\infty)$; ***Right:*** *Behaviour of bound.*

TABLE 4
Accuracy test in two dimensions for computing a smooth steady state to the equation (4.4) *with uniform initial density* $\rho_0 \equiv 0.09375$. *The error is calculated at* $T = 15$

N	L^1 error	Order	L^2 error	Order	L^∞ error	Order
10×10	9.706e-02		2.206e-02		1.143e-02	
20×20	2.351e-02	2.046	5.166e-03	2.094	2.981e-03	1.939
40×40	5.841e-03	2.009	1.252e-03	2.045	6.315e-04	2.239

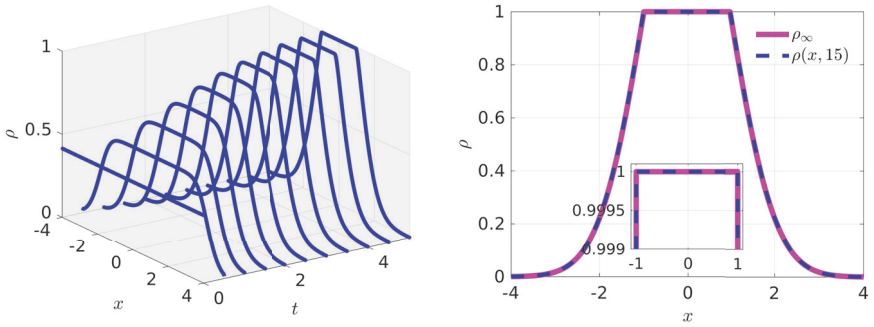

FIG. 4. *Computation of a non-smooth steady state to the one-dimensional saturation experiment* (4.4) *with* $\rho_0 = 0.415$. ***Left:*** *Evolution of* $\rho(x,t)$; ***Right:*** *Comparison between* ρ_∞ *and the numerical solution calculated at* $T = 15$.

To observe saturation, we begin with a uniform initial density of $\rho_0 \equiv 0.415$ in one dimension and $\rho_0 \equiv 0.147$ in two dimensions such that $\|\rho_0\|_{L^1} > m_c$. According to (4.5), this initial density results in a non-smooth steady state that is bounded between 0 and 1. A numerical solution obtained at $T = 15$ is considered as the numerical steady state. The evolution in both one and two dimensions can be found in Figure 4 and Figure 5. As depicted in Figure 4, in contrast to the results from [14, Section 4.1], our DG discretization method yields a superior approximation devoid of oscillations near the upper bound $\rho_{\max} = 1$. Additionally, we present the behavior of the relative entropy and the bound in Figure 6, which shows that the numerical solution generated by our scheme demonstrates bound preservation within the interval $[0, 1]$ and also exhibits entropy dissipation. In order to test the accuracy of this example, we calculated the error at $T = 1$, employing reference solutions calculated at a finer mesh size with $N = 320$ for one-dimensional computations and $N = 80 \times 80$ for two-dimensional computations. Table 5 and Table 6 present the second-order accuracy of our approach in both one and two dimensions.

To demonstrate the efficiency of our proposed scheme, we compare its performance with three second-order methods: the explicit SSP-RK scheme [34, Eq (2.10)], the IMEX-RK scheme [8, Scheme H-DIRK2(2,2,2)], and the Lagrange approach [15, Eq (3.5)-(3.6)]. To ensure a fair comparison, an appropriate fixed timestep that guarantees bound-preservation is chosen for each scheme. The explicit SSP-RK scheme has been proven to preserve positivity under a CFL condition of $\tau \sim h^2$. While the IMEX-RK scheme shows efficient performance under a CFL condition of $\tau \sim h$ [32], the left panel in Figure 7 reveals that when $\tau \sim h$, the numerical solution exceeds the upper bound. Therefore, bound preservation for the IMEX-RK scheme is expected under the more restrictive condition $\tau \sim h^2$. The Lagrange scheme, which employs a predictor-corrector approach, is proposed as an unconditionally bound-preserving

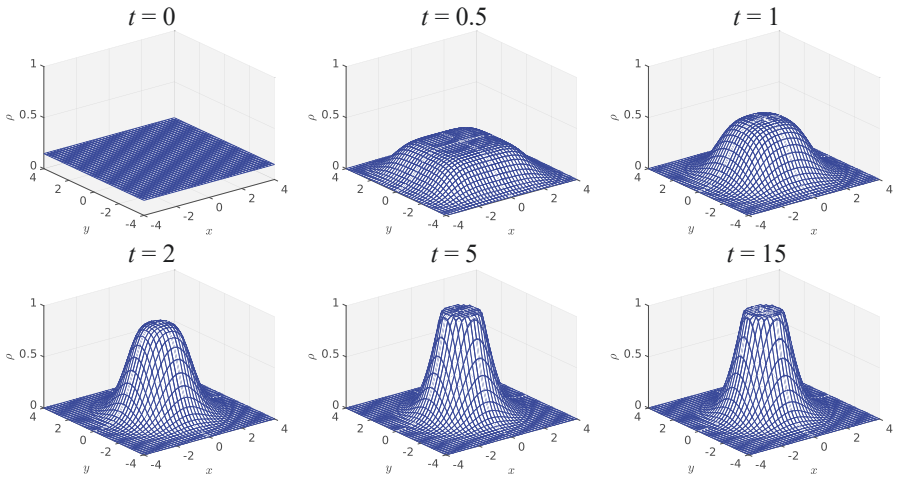

FIG. 5. *Evolution of a solution* $\rho(\boldsymbol{x}, t)$ *with* $\rho_0 = 0.147$ *in a two-dimensional saturation experiment* (4.4).

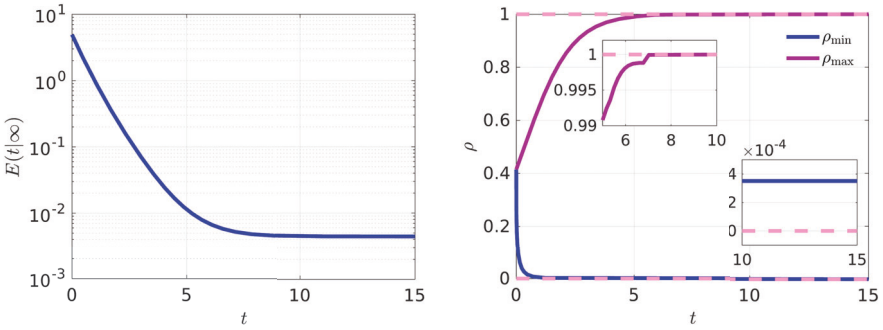

FIG. 6. *Computation of a non-smooth steady state to the one-dimensional saturation experiment* (4.4) *with* $\rho_0 = 0.415$. ***Left:*** *Behaviour of the relative entropy* $E(t; \infty)$*;* ***Right:*** *Behaviour of bound.*

scheme. However, the error analysis has only been posed for specific cases. In this numerical example, we observe that with $\tau = 0.01$, the numerical solution introduces significant errors, although the bound is preserved. In contrast, we find that this error can be mitigated by using a CFL condition of $\tau \sim h$. The right panel of Figure 7 compares the computational time for these methods. Our results show that our scheme outperforms both the explicit SSP-RK and IMEX-RK methods due to its milder timestep restriction. The Lagrange approach demonstrates the best performance in terms of computational time, as it only requires solving a single linear equation and a one-dimensional zero-point problem. However, a rigorous analysis of the Lagrange approach is still needed.

4.4. Aggregation–diffusion equation. We proceed with our study of the scheme (3.22) on the aggregation-diffusion equation

$$\rho_t = \nabla \cdot \left(\rho \nabla \left(\frac{\nu m}{m-1} \rho^{m-1} + W * \rho \right) \right) = \nu \Delta \rho^m + \nabla \cdot (\rho \nabla (W * \rho)) \tag{4.6}$$

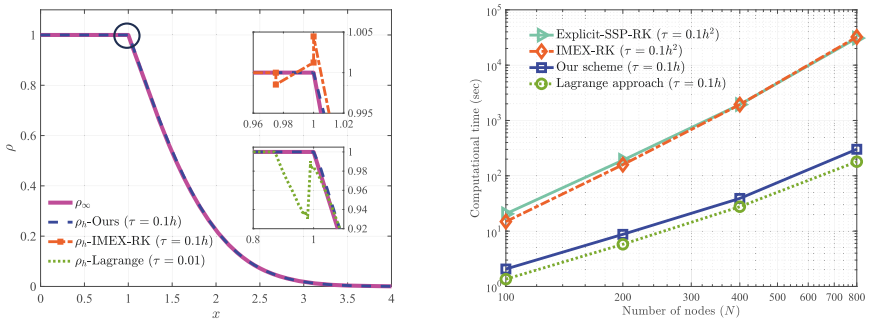

FIG. 7. *Comparison of performance for various second-order numerical schemes to compute the one-dimensional saturation experiment* (4.4) *with* $\rho_0 = 0.415$. ***Left:*** *Comparison between* ρ_∞ *and the numerical solution calculated at* $T = 15$*;* ***Right:*** *Computation time for computing the numerical solution at* $T = 15$.

TABLE 5
Accuracy test in one dimension for computing a solution to the equation (4.4) *with uniform initial density* $\rho_0 \equiv 0.415$. *The error is calculated at* $T = 1$.

N	L^1 error	Order	L^2 error	Order	L^∞ error	Order
20	3.320e-02		1.652e-02		1.553e-02	
40	8.476e-03	1.970	4.172e-03	1.986	4.382e-03	1.825
80	1.960e-03	2.113	9.682e-04	2.107	1.031e-03	2.088
160	3.786e-04	2.372	1.822e-04	2.410	1.852e-04	2.476

TABLE 6
Accuracy test in two dimensions for computing a solution to the equation (4.4) *with uniform initial density* $\rho_0 = 0.147$. *The error is calculated at* $T = 1$.

N	L^1 error	Order	L^2 error	Order	L^∞ error	Order
10×10	4.825e-01		8.143e-02		3.071e-02	
20×20	1.343e-01	1.845	2.300e-02	1.824	8.874e-03	1.791
40×40	3.123e-02	2.104	5.192e-03	2.147	1.753e-03	2.340

with interaction kernel $W = e^{|\boldsymbol{x}|^2}/(2\pi)^{d/2}$. The parameters $\nu > 0$ and $m > 1$ are set to $\nu = 0.05$ and $m = 3$ in our computation. In this example, $f = \rho$, $F = \nu m \rho^{m-1}$, $V = 0$, and the density ρ remains positive. For our tests, we utilize periodic boundary conditions and evaluate the solution over the domain $[-6, 6]$ in one dimension and $[-4, 4]^2$ in two dimensions.

For the one-dimensional test, we compute the solution up to time $T = 200$, using a smooth initial datum $\rho_0 = \frac{1}{\sqrt{2\pi}}(e^{-(x-2)^2/2} + e^{-(x+2)^2/2})$. The evolution and behavior of entropy are plotted in Figure 8. As time progresses, the density starts at a smooth initial state with two peaks and then converges to a non-smooth steady state with only one peak and two discontinuity points. The entropy also shows dissipation throughout the computation. To ascertain the accuracy of this example, we compute the error at $T = 1$ and refer to a benchmark solution computed with a mesh of $N = 320$ points. Table 7 demonstrates the desired second-order accuracy of our scheme (3.22).

In two dimensions, we consider a test with a discontinuous initial state given by $\rho_0(\boldsymbol{x}) = 1_{[-2,2]\times[-2,2]}(\boldsymbol{x})$. Figure 9 shows the dynamic evolution used to compute a solution at $T = 20$ with $N = 40 \times 40$ cells. In this setting, ρ exhibits a transition from

TABLE 7

Accuracy test in one dimension for computing a solution to the equation (4.6) *with* $\rho_0 = \frac{1}{\sqrt{2\pi}}(e^{-(x-2)^2/2} + e^{-(x+2)^2/2})$. *The error is calculated at* $T = 1$.

N	L^1 error	Order	L^2 error	Order	L^∞ error	Order
40	1.634e-02		5.839e-03		3.147e-03	
80	5.205e-03	1.650	1.842e-03	1.664	9.728e-04	1.694
160	1.301e-03	2.000	4.670e-04	1.980	2.610e-04	1.898
320	2.419e-04	2.427	8.624e-05	2.437	5.122e-05	2.349

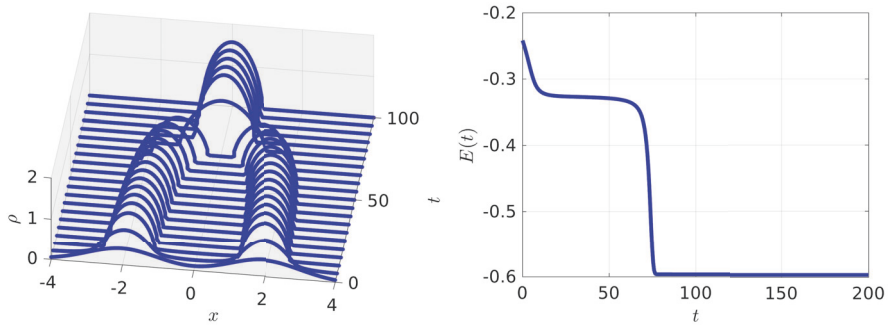

FIG. 8. *Computation of a steady state to the one-dimensional aggregation–diffusion equation* (4.6) *with* $\rho_0 = \frac{1}{\sqrt{2\pi}}(e^{-(x-2)^2/2} + e^{-(x+2)^2/2})$. ***Left:*** *Evolution of* $\rho(x,t)$; ***Right:*** *Behaviour of the discrete entropy.*

a discontinuous distribution to a concentrated central peak. To test the accuracy in this two-dimensional case, we computed the error at $T = 1$ with a reference solution calculated using a mesh of $N = 80 \times 80$ cells. The error table and convergence rate are provided in Table 8. In this example, the L^1 error shows second-order accuracy, while the observed degeneracy in L^2 and L^∞ accuracy is due to the discontinuity in the initial state and the non-smoothness of the solution.

TABLE 8

Accuracy test in two dimensions for computing a solution to the equation (4.6) *with* $\rho_0 = 1_{[-2,2]\times[-2,2]}$. *The error is calculated at* $T = 1$.

N	L^1 error	Order	L^2 error	Order	L^∞ error	Order
10×10	6.686e-01		2.983e-01		1.836e-01	
20×20	1.750e-01	1.934	9.611e-02	1.634	8.452e-02	1.119
40×40	4.405e-02	1.990	2.610e-02	1.881	3.317e-02	1.349

Conclusions. This paper presented a fully discrete scheme for solving a class of degenerate parabolic equations. Spatially, we applied a discontinuous Galerkin method to an appropriate reformulation of equations, resulting in an ODE with a splitting structure. Our numerical examples demonstrated that this spatial discretization induces fewer oscillations in the solution. Temporally, we proposed a second-order exponential scheme, which involves only linear solvers, by introducing integrating factors into the ODE and utilizing SSP-RK methods. Notably, our approach consistently evidences second-order accuracy, bound preservation and mass conservation with a

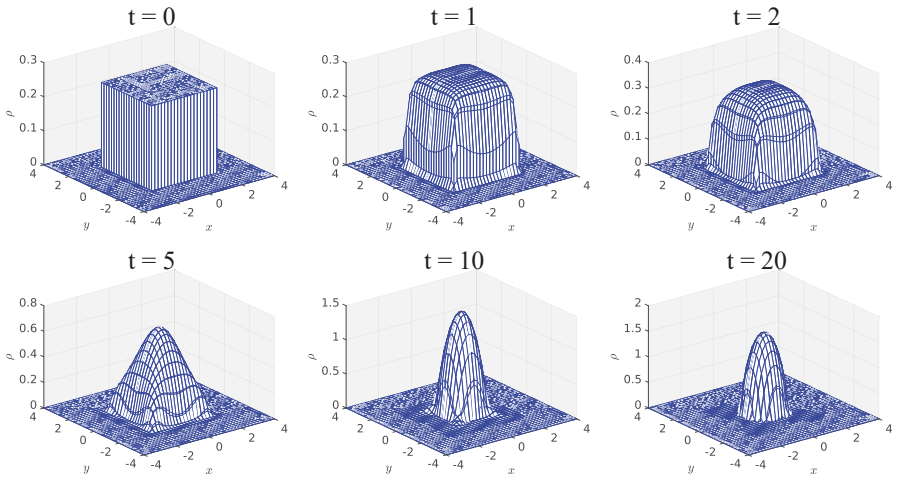

FIG. 9. *Evolution of a solution $\rho(\boldsymbol{x}, t)$ to a two-dimensional aggregation–diffusion equation* (4.6) *with* $\rho_0 = 1_{[-2,2]\times[-2,2]}$.

favorable CFL condition $\tau \sim h$ both in theory and in practice. Nevertheless, the calculation of the exponential matrix partly limited our efficiency, and energy dissipation was only shown in our numerical tests but not proved theoretically. Indeed, devising an efficient linear scheme that simultaneously provides high-order accuracy in both space and time, while guaranteeing bound preservation, mass conservation, and energy dissipation, continues to be a major challenge.

Acknowledgments. The first author would like to acknowledge the Tsinghua Scholarship for Overseas Graduate Students (No. 2022118) for financial support and advisors Prof. Chi-wang Shu and Chenglong Bao for their invaluable guidance.

REFERENCES

[1] A. H. Al-Mohy and N. J. Higham, *Computing the action of the matrix exponential, with an application to exponential integrators*, SIAM Journal on Scientific Computing, 33 (2011), pp. 488–511.

[2] R. Bailo, J. A. Carrillo, and J. Hu, *Bound-preserving finite-volume schemes for systems of continuity equations with saturation*, SIAM Journal on Applied Mathematics, 83 (2023), pp. 1315–1339.

[3] R. Bailo, J. A. Carrillo, H. Murakawa, and M. Schmidtchen, *Convergence of a fully discrete and energy-dissipating finite-volume scheme for aggregation-diffusion equations*, Mathematical Models and Methods in Applied Sciences, 30 (2020), pp. 2487–2522.

[4] D. Benedetto, E. Caglioti, J. A. Carrillo, and M. Pulvirenti, *A non-maxwellian steady distribution for one-dimensional granular media*, Journal of Statistical Physics, 91 (1998), pp. 979–990.

[5] M. Bessemoulin-Chatard and F. Filbet, *A finite volume scheme for nonlinear degenerate parabolic equations*, SIAM Journal on Scientific Computing, 34 (2012), pp. B559–B583.

[6] S. Blanes, A. Iserles, and S. Macnamara, *Positivity-preserving methods for ordinary differential equations*, ESAIM: Mathematical Modelling and Numerical Analysis, 56 (2022), pp. 1843–1870.

[7] S. Boi, V. Capasso, and D. Morale, *Modeling the aggregative behavior of ants of the species polyergus rufescens*, Nonlinear Analysis: Real World Applications, 1 (2000), pp. 163–176.

[8] S. Boscarino, R. Bürger, P. Mulet, G. Russo, and L. M. Villada, *Linearly implicit IMEX Runge–Kutta methods for a class of degenerate convection-diffusion problems*, SIAM Journal on Scientific Computing, 37 (2015), pp. B305–B331.

[9] M. Burger, M. Di Francesco, and Y. Dolak-Struss, *The Keller–Segel model for chemotaxis with prevention of overcrowding: Linear vs. nonlinear diffusion*, SIAM Journal on Mathematical Analysis, 38 (2006), pp. 1288–1315.
[10] R. Bürger, D. Inzunza, P. Mulet, and L. M. Villada, *Implicit-explicit methods for a class of nonlinear nonlocal gradient flow equations modelling collective behaviour*, Applied Numerical Mathematics, 144 (2019), pp. 234–252.
[11] M. Caliari, F. Cassini, L. Einkemmer, and A. Ostermann, *Accelerating exponential integrators to efficiently solve semilinear advection-diffusion-reaction equations*, SIAM Journal on Scientific Computing, 46 (2024), pp. A906–A928.
[12] J. A. Carrillo, R. J. McCann, and C. Villani, *Kinetic equilibration rates for granular media and related equations: entropy dissipation and mass transportation estimates*, Revista Matematica Iberoamericana, 19 (2003), pp. 971–1018.
[13] J. A. Carrillo and G. Toscani, *Asymptotic l 1-decay of solutions of the porous medium equation to self-similarity*, Indiana University Mathematics Journal, (2000), pp. 113–142.
[14] J. A. Carrillo, L. Wang, and C. Wei, *Structure preserving primal dual methods for gradient flows with nonlinear mobility transport distances*, SIAM Journal on Numerical Analysis, accepted (2023).
[15] Q. Cheng and J. Shen, *A new Lagrange multiplier approach for constructing structure preserving schemes, II. Bound preserving*, SIAM Journal on Numerical Analysis, 60 (2022), pp. 970–998.
[16] A. Chertock, S. Cui, A. Kurganov, and T. Wu, *Steady state and sign preserving semi-implicit Runge–Kutta methods for ODEs with stiff damping term*, SIAM Journal on Numerical Analysis, 53 (2015), pp. 2008–2029.
[17] Q. Du, L. Ju, X. Li, and Z. Qiao, *Maximum bound principles for a class of semilinear parabolic equations and exponential time-differencing schemes*, SIAM Review, 63 (2021), pp. 317–359.
[18] C. Duan, W. Chen, C. Liu, X. Yue, and S. Zhou, *Structure-preserving numerical methods for nonlinear Fokker–Planck equations with nonlocal interactions by an energetic variational approach*, SIAM Journal on Scientific Computing, 43 (2021), pp. B82–B107.
[19] C. M. Elliott and H. Garcke, *On the Cahn–Hilliard equation with degenerate mobility*, SIAM Journal on Mathematical Analysis, 27 (1996), pp. 404–423.
[20] S. Gottlieb, Z. J. Grant, J. Hu, and R. Shu, *High order strong stability preserving multiderivative implicit and IMEX Runge–Kutta methods with asymptotic preserving properties*, SIAM Journal on Numerical Analysis, 60 (2022), pp. 423–449.
[21] S. Gottlieb, C.-W. Shu, and E. Tadmor, *Strong stability-preserving high-order time discretization methods*, SIAM Review, 43 (2001), pp. 89–112.
[22] M. Hochbruck and A. Ostermann, *Exponential integrators*, Acta Numerica, 19 (2010), pp. 209–286.
[23] J. Hu and R. Shu, *A second-order asymptotic-preserving and positivity-preserving exponential Runge–Kutta method for a class of stiff kinetic equations*, Multiscale Modeling & Simulation, 17 (2019), pp. 1123–1146.
[24] J. Huang and C.-W. Shu, *A second-order asymptotic-preserving and positivity-preserving discontinuous Galerkin scheme for the Kerr–Debye model*, Mathematical Models and Methods in Applied Sciences, 27 (2017), pp. 549–579.
[25] J. Huang and C.-W. Shu, *Bound-preserving modified exponential Runge–Kutta discontinuous Galerkin methods for scalar hyperbolic equations with stiff source terms*, Journal of Computational Physics, 361 (2018), pp. 111–135.
[26] L. Isherwood, Z. J. Grant, and S. Gottlieb, *Strong stability preserving integrating factor Runge–Kutta methods*, SIAM Journal on Numerical Analysis, 56 (2018), pp. 3276–3307.
[27] L. Ju, X. Li, Z. Qiao, and J. Yang, *Maximum bound principle preserving integrating factor Runge–Kutta methods for semilinear parabolic equations*, Journal of Computational Physics, 439 (2021), p. 110405.
[28] H. Li, S. Xie, and X. Zhang, *A high order accurate bound-preserving compact finite difference scheme for scalar convection diffusion equations*, SIAM Journal on Numerical Analysis, 56 (2018), pp. 3308–3345.
[29] H. Li and X. Zhang, *On the monotonicity and discrete maximum principle of the finite difference implementation of C^0-Q^2 finite element method*, Numerische Mathematik, 145 (2020), pp. 437–472.
[30] C. Moler and C. Van Loan, *Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later*, SIAM review, 45 (2003), pp. 3–49.
[31] R. J. Plemmons, *M-matrix characterizations. I—nonsingular M-matrices*, Linear Algebra and its Applications, 18 (1977), pp. 175–188.

[32] B. Sebastiano, *High-order semi-implicit schemes for evolutionary partial differential equations with higher order derivatives*, Journal of Scientific Computing, 96 (2023), p. 11.
[33] M. Spijker, *Contractivity in the numerical solution of initial value problems*, Numerische Mathematik, 42 (1983), pp. 271–290.
[34] Z. Sun, J. A. Carrillo, and C.-W. Shu, *A discontinuous Galerkin method for nonlinear parabolic equations and gradient flow problems with interaction potentials*, Journal of Computational Physics, 352 (2018), pp. 76–104.
[35] K. Wu and C.-W. Shu, *Provably positive high-order schemes for ideal magnetohydrodynamics: analysis on general meshes*, Numerische Mathematik, 142 (2019), pp. 995–1047.
[36] F. Yan, J. Van der Vegt, Y. Xia, and Y. Xu, *Entropy dissipative higher order accurate positivity preserving time-implicit discretizations for nonlinear degenerate parabolic equations*, Journal of Computational and Applied Mathematics, 441 (2024), p. 115674.
[37] X. Zhang and C.-W. Shu, *On maximum-principle-satisfying high order schemes for scalar conservation laws*, Journal of Computational Physics, 229 (2010), pp. 3091–3120.