Electronic Research Archive

http://www.aimspress.com/journal/era

ERA, 30(6): 2165–2182.
DOI: 10.3934/era.2022109
Received: 31 December 2021
Revised: 07 April 2022
Accepted: 12 April 2022
Published: 18 April 2022

Research article

A fast method for solving time-dependent nonlinear convection diffusion problems

Qian He[1], **Wenxin Du**[1], **Feng Shi**[1,*] **and Jiaping Yu**[2]

[1] College of Science, Harbin Institute of Technology, Shenzhen 518055, China
[2] College of Science, Donghua University, Shanghai 201620, China

* **Correspondence:** Email: shi.feng@hit.edu.cn.

Abstract: In this paper, a fast scheme for solving unsteady nonlinear convection diffusion problems is proposed and analyzed. At each step, we firstly isolate a nonlinear convection subproblem and a linear diffusion subproblem from the original problem by utilizing operator splitting. By Taylor expansion, we explicitly transform the nonlinear convection one into a linear problem with artificial inflow boundary conditions associated with the nonlinear flux. Then a multistep technique is provided to relax the possible stability requirement, which is due to the explicit processing of the convection problem. Since the self-adjointness and coerciveness of diffusion subproblems, there are so many preconditioned iterative solvers to get them solved with high efficiency at each time step. When using the finite element method to discretize all the resulting subproblems, the major stiffness matrices are same at each step, that is the reason why the unsteady nonlinear systems can be computed extremely fast with the present method. Finally, in order to validate the effectiveness of the present scheme, several numerical examples including the Burgers type and Buckley-Leverett type equations, are chosen as the numerical study.

Keywords: nonlinear convection diffusion problems; operator splitting; multistep technique; finite element method; Burgers type equation; Buckley-Leverett type equation

1. Introduction

In this paper, we introduce a fast IMEX (implicit-explicit) scheme for the 2D unsteady nonlinear convection-diffusion problems. Our model problem is as follows

$$u_t + (P(u))_x + (Q(u))_y - \varepsilon(u_{xx} + u_{yy}) = f \quad \text{in} \quad \Omega \times (0, T), \tag{1.1}$$

with the Dirichlet boundary condition

$$u(x, y, t) = b(x, y, t) \quad \text{on} \quad \partial\Omega \times (0, T), \tag{1.2}$$

and the initial value condition

$$u(x, y, 0) = a(x, y) \quad \text{in} \quad \Omega\,, \tag{1.3}$$

where $\Omega \subset R^2$ is a bounded polygonal domain with Lipschitz continuity, $P(u)$ and $Q(u)$ are the convective field functions, $\varepsilon > 0$ is a given diffusion coefficient, and T is the final time.

Note that we can rewrite the system of Eqs (1.1)–(1.3) into the vector form as

$$u_t + \nabla \cdot (F(u)) - \varepsilon \Delta u = f \quad \text{in} \quad \Omega \times (0, T)\,, \tag{1.4}$$

with the Dirichlet boundary and initial value conditions

$$u(\mathbf{x}, t) = u_b(\mathbf{x}, t) \quad \text{on} \quad \partial\Omega \times (0, T)\,, \quad u(\mathbf{x}, 0) = u_0(\mathbf{x}) \quad \text{in} \quad \Omega\,, \tag{1.5}$$

where $F(u) = (P(u), Q(u))^T$, $\mathbf{x} = (x, y)^T$, $u = u(\mathbf{x}, t)$.

Generally, some physical phenomenon, where the energy or other physical substances are transferred in physical systems as a result of diffusion and convection processes, can be described by nonlinear convection-diffusion Eqs (1.1)–(1.3). These systems are of reasonably practical significance since they can describe varieties of physical phenomenon, which appear in a plenty of applications and symbolize mathematical models for a great deal of physical processes in mechanics of fluids, astrophysics, groundwater flow, meteorology, semiconductors and reactive flows. In particular, for the certain case when $f = 0$, (1.1)–(1.3) are scalar convection-diffusion equations, which usually arise in various applications, with the range including from turbulence model [1, 2], through traffic flow [3], to two-phase flow in a porous medium [4].

For the general nonlinear convection-diffusion Eqs (1.1)–(1.3), a group of high order finite difference numerical schemes were raised based on Euler implicit scheme by Cecchi and Pirozzi [5], a discontinuous Galerkin finite element discretization was introduced to study the effect of numerical integration by Sobotikova et al. [6], and a Lattice Boltzmann model was shown by Shi and Guo [7]. For the certain case $F = 0$, namely the nonlinear conservative laws, Karlsen et al. [8] used the idea of operator splitting to decompose the convection part and the diffusion part of the problem, and provided a dimensional splitting method to simplify the m-dimensional convection equation to a series of one-dimensional equations. After that, Karlsen et al. [9] presented a corrected operator splitting method to deal with much more complicated problems with general flux functions and initial value. Many central schemes were also constructed to achieve high resolution of the numerical solutions; See for instance Nessyahu and Tadmor [10], Jin and Xin [11], Kurganov and Tadmor [12]. Recently, some high order Maximal principle-Preserving finite difference Rungeutta WENO (weighted essentially nonoscillatory) or DG (discontinuous Galerkin) schemes were constructed; See Jiang and Xu [13], and Xiong et al. [14] and the references therein.

As explained by Chertock and Kurganov [15], such problems are parabolic and that's why even for discontinuous initial value, they can be admitted global smooth solutions. In this paper, we combine the continuous finite element scheme and operator splitting approach to propose a fast method for solving these problems. It is obvious that the treatment of nonlinear convection term in partial differential equations like (1.1)–(1.3) is one of the key procedures, which affects the efficiency of the numerical methods. An important class of numerical methods about this issue is referred as the IMEX method; See for instance Akrivis [16] for general nonlinear parabolic problems, Long and Chen [17] for nonlinear convection diffusion equations, as well as Zhang et al. [18] for the slight simple non-stationary thermal convection problems.

In this paper, first of all, at each time step, the initial coupled problem is decomposed into convection and diffusion subproblems by employing the same idea of operator splitting with [8] (see also [19] for much general parabolic problems), and much importantly, make use of a specific Taylor expansion to deal with the nonlinear term in convection subproblems, and eventually provide an IMEX scheme for solving the nonlinear convection diffusion problems. Besides, for the sake of relaxing the possible poor stability condition that results from the explicit operation of the convection problem, a multistep technique is proposed. There are some major superiorities of the present method. Firstly, at each time step, since the self-adjointness and coerciveness of diffusion subproblems, there are a lot of forceful preconditioned iterative solvers, like preconditioned conjugate gradient solver, which can be used to solve them efficiently. Secondly, it is easy for us to utilize the mass lumping method to explicitly solve the mass matrices for spatial linear elements because of the linearity of convection subsystems. When we use the finite element method to discretize the equations, all the major stiffness matrices keep invariable in the time stepping process, that is the reason why the present method is notably fast for solving the unsteady nonlinear systems.

The paper is formed as follows. We propose the idea of construction of the present semi-discrete IMEX method via operator splitting, which is shown in Section 2. After that, by means of the spatial finite element discretization, a further representation of the algorithm of full discretization is illustrated. Meanwhile, the superiorities of the proposed method are illuminated by some remarks as well as the multistep scheme. As the numerical study, several numerical experiments on Burgers type and Buckley-Leverett type equations and discussions are presented in Section 3.

2. The IMEX scheme

In this section, we propose the semi-discrete and its corresponding fully discrete IMEX algorithms. Without loss of generality, we uniformly split the time interval to make temporal discretization. Fix positive integer $N \geq 1$, set $0 = t^0 < t^1 \cdots < t^{N-1} < t^N = T$, with $t^n = n\Delta t$, $t^{n+\frac{1}{2}} = t^n + \Delta t/2$, and the step size $\Delta t = T/N$.

Firstly, the nonlinear convection term $\nabla \cdot (F(u))$ and the linear diffusion term $\varepsilon \Delta u$ are separated. We can write it in the form of operators described below

$$A\,u = A_1\,u + A_2\,u, \quad \text{with} \quad A_1\,u = \nabla \cdot (F(u)), \quad A_2\,u = -\varepsilon \Delta\,u.$$

Where, $A_1 u$ is the nonlinear part of the convection diffusion equations. In order to compute the Eqs (1.4) and (1.5), we can directly utilize numerous operator splitting methods, see, e.g., [8] by Karlsen et al. and [19, Chapter II] by R. Glowinski and among the references. In this paper, we employ the following time splitting technique

$$v_t + \nabla \cdot (F(v)) = 0, \quad v(\mathbf{x}, t^n) = u(\mathbf{x}, t^n), \tag{2.1}$$

$$w_t - \varepsilon \Delta w = f, \quad w(\mathbf{x}, t^n) = v(\mathbf{x}, t^{n+1}), \tag{2.2}$$

on the time interval $[t^n, t^{n+1}]$ to solve the original nonlinear system (1.4)–(1.5). When it comes to the time splitting, the certain Euler central difference and backward schemes are utilized to discretize (2.1) and (2.2), seperately,

$$\frac{v^{n+1} - v^n}{\Delta t} + \nabla \cdot (F^{n+\frac{1}{2}}) = 0, \quad v^n = u^n, \tag{2.3}$$

$$\frac{w^{n+1}-w^n}{\Delta t}-\varepsilon\triangle w^{n+1} \;=\; f^{n+1},\quad w^n=v^{n+1}, \tag{2.4}$$

where $F(v(\mathbf{x},t^{n+\frac{1}{2}}))$ is approximated by $F^{n+\frac{1}{2}}$, and the solution $u(\mathbf{x},t)$ to the system (1.4)–(1.5) at time instances $t=t^n$ and t^{n+1} are approximated by u^n and u^{n+1}, correspondingly.

It worth noting that the performance of the resulting algorithm, such as the stability, convergence and efficiency, would be significantly affected by the numerical treatment of $F^{n+\frac{1}{2}}$ in the iterative scheme (2.3)–(2.4). For instance, seting $F^{n+\frac{1}{2}}=(F(v^{n+1})+F(v^n))/2$ in (2.3) results in a Crank-Nicolson scheme, and generally speaking, we still get a nonlinear resulting system. If we take $F^{n+\frac{1}{2}}$ as an explicit linearization, i.e., $F^{n+\frac{1}{2}}=F(v^n)=F(u^n)$, Eq (2.3) will generate a Forward Euler scheme with strong instability generally. Without choosing $F^{n+\frac{1}{2}}$ as these two ways, we linearize $F^{n+\frac{1}{2}}$ at the time instance $t=t^n$ by proposing a new unique Taylor expansion method. More accurately, as a result of the fact that Eq (2.1) is discretized into (2.3), then we can access through Taylor expansion:

$$F^{n+\frac{1}{2}}\approx F(v(\mathbf{x},t^n+\frac{\Delta t}{2})) \;=\; F(v(\mathbf{x},t^n))+F_u(v(\mathbf{x},t^n))\,v_t(\mathbf{x},t^n)\frac{\Delta t}{2}+O(\Delta t^2),$$

here we use the definition $F_u=\frac{dF(u)}{du}$. It is noted that throughout this paper, we assume that $F(u)\in C^1(R)$ in order to construct our present numerical algorithm; See Remark 1 in Reference [6] for more details.

In term of (2.1) and (2.3) we obtain that

$$v(\mathbf{x},t^n)\approx v^n=u^n,\quad F(v(\mathbf{x},t^n))\approx F(u^n),\quad F_u(v(\mathbf{x},t^n))\approx F_u(u^n)$$

and

$$v_t(\mathbf{x},t^n)\approx-\nabla\cdot(F(u^n)),$$

then a linearization of $F^{n+\frac{1}{2}}$ is derived as

$$F^{n+\frac{1}{2}}\approx F(u^n)-\frac{1}{2}\Delta tF_u(u^n)\nabla\cdot(F(u^n)).$$

Denoting $\eta^n=F(u^n)-\frac{1}{2}\Delta tF_u(u^n)\nabla\cdot(F(u^n))$, the following time splitting scheme is the newly proposed one that we now obtain:

$$\frac{u_*^{n+1}-u^n}{\Delta t}+\nabla\cdot\eta^n \;=\; 0\,, \tag{2.5}$$

$$\frac{u^{n+1}-u_*^{n+1}}{\Delta t}-\varepsilon\triangle u^{n+1} \;=\; f^{n+1}\,. \tag{2.6}$$

Meanwhile, the boundary condition for these Eqs (2.5) and (2.6) is need to be fulfilled. When solving u^{n+1} in the subsystem (2.6), further a temporal approximation of diffusion problem (2.2), we can naturally inherit a Dirichlet boundary condition on $\Gamma=\partial\Omega$ for u^{n+1} from (1.5), namely

$$u^{n+1}|_\Gamma=u_b(\cdot,t^{n+1}). \tag{2.7}$$

Since u_*^{n+1} in the subsystem (2.5) is a nontrivial temporal approximation of convective Eq (2.1), we should only impose partial boundary condition. Here, a similar artificial inflow boundary condition is introduced

$$u_*^{n+1}|_{\Gamma_{n+1}^-}=u_b(\cdot,t^{n+1}), \tag{2.8}$$

where Γ_t^- is the inflow boundary named by

$$\Gamma_t^- = \{\mathbf{x} \in \Gamma : \ F_u(u_b(\mathbf{x}, t)) \cdot \mathbf{n}(\mathbf{x}) < 0\},$$

and the outward normal vector to Γ is expressed by $\mathbf{n}$.

For simplicity in the rest of the paper, let us denote by $u_b^{n+1} = u_b(\cdot, t^{n+1})$, and functional space is defined as $X = H^1(\Omega)$. Besides, terms (2.5) and (2.6) are respectively multiplied by a testing functions $v \in X$ with $v|_{\Gamma_{n+1}^-} = 0$ and $v|_\Gamma = 0$, (as the options of boundary conditions (2.7) and (2.8)) correspondingly, and integrated by parts to obtain the variational formulations. Finally, we can describe the time splitting method as the following:

Temporal Semi-approximation Scheme:

1. Initialize value: Let $u^0 = u_0$.
2. For any $n \geq 0$, supposing u^n is already known, find u_*^{n+1} and u^{n+1} to solve
   Convection subsystem: Compute $u_*^{n+1} \in X$, $u_*^{n+1}|_{\Gamma_{n+1}^-} = u_b^{n+1}$, solve

$$(u_*^{n+1}, v) = (u^n, v) - \Delta t \langle \eta^n \cdot \mathbf{n}, v \rangle_{\Gamma \setminus \Gamma_{n+1}^-} + \Delta t (\eta^n, \nabla v) \tag{2.9}$$

   for any $v \in X$, with $v = 0$ on Γ_{n+1}^- ;

and

   Diffusion subsystem: Compute $u^{n+1} \in X$, $u^{n+1}|_\Gamma = u_b^{n+1}$, such that

$$(\Delta t)^{-1}(u^{n+1}, v) + \varepsilon(\nabla u^{n+1}, \nabla v) = (f^{n+1} + (\Delta t)^{-1} u_*^{n+1}, v) \tag{2.10}$$

   for any $v \in X$, with $v = 0$ on Γ .
3. Let $n+ = 1$, until $n > N$, else retrun to Step 2.

Next, the finite element discretization of the above time splitting method will be discussed with a shape-regular triangulation of Ω, which is denoted by $\mathcal{T}_h$ and has mesh scale $h = \max_{K \in \mathcal{T}_h} \mathrm{diam}(K)$. Denote the finite element space with order $k \geq 1$ as

$$X_h = \{v_h \in C^0(\overline{\Omega}_h) \cap X ; \quad v_h|_K \in P_k(K), \forall K \in \mathcal{T}_h\},$$

that is a subspace of X used to approximate the velocity. In the next segment, we simply offer a detailed discussion under the certain circumstance $k = 1$, which is the continuous piece-wise linear functions over $\mathcal{T}_h$. Apart from that, the standard nodal interpolation operator denoted by $I_h : X \to X_h$ is needed.

Next, the following scheme is our full (space-time) approximation approach.

Algorithm A: Space-time full approximation Scheme

1. Initialize value: Let $u_h^0 = I_h u_0$.
2. For any $n \geq 0$, supposing u_h^n is already known, find $u_{h,*}^{n+1}$ and u_h^{n+1} by solving
   Convection subsystem: Denote $\eta_h^n = F(u_h^n) - \frac{1}{2} F_u(u_h^n)\nabla \cdot (F(u_h^n))$; Compute $u_{h,*}^{n+1} \in X_h$, with $u_{h,*}^{n+1}|_{\Gamma_{n+1}^-} = I_h u_b^{n+1}$, solve

$$(u_{h,*}^{n+1}, v_h) = (u_h^n, v_h) - \Delta t \langle \eta_h^n \cdot \mathbf{n}, v_h \rangle_{\Gamma \setminus \Gamma_{n+1}^-} + \Delta t (\eta_h^n, \nabla v_h) \tag{2.11}$$

for any $v_h \in X_h$, with $v_h = 0$ on Γ^-_{n+1};

and

Diffusion subsystem: Find $u_h^{n+1} \in X_h$ with $u_h^{n+1}|_\Gamma = I_h u_b^{n+1}$, such that

$$(\Delta t)^{-1}(u_h^{n+1}, v_h) + \varepsilon(\nabla u_h^{n+1}, \nabla v_h) = (f^{n+1} + (\Delta t)^{-1} u_{h,*}^{n+1}, v_h) \tag{2.12}$$

for any $v_h \in X_h$, with $v_h = 0$ on Γ.

3. Let $n+ = 1$, until $n > N$, else retrun to Step 2.

Remark 1. *For any given time step Δt, and for all $n \le N$, we can get some positive definite and permanently invariant stiffness matrices for (2.12).*

Remark 2. *Since the stiffness matrices of (2.11) are mass matrices as well as maintain invariable for fixed $\Gamma_n^- = \Gamma^-$ ($n \le N$), the subsystem can be explicitly solved. And particularly, when using linear finite element space ($k = 1$), the mass-lumping technique works efficiently (see details in reference [20]).*

Remark 3. *Compared with the majority of existing techniques for computing (1.4)–(1.5), the distinct operation of nonlinear convection term would produce some unique superiorities of the present approach, as depicted in Remark 2. Nevertheless, as smaller time step size is required to realize convergence in this explicit linearization, the practical applications of our present approach would slightly decline. As a matter of fact, it is a good way to apply the multistep pattern of the approach to remedy this deficiency. Practically, when we deal with each diffusion subsystem (2.12) using time step size Δt, we can utilize a smaller $\delta t = \Delta t/m$, to complete the convection subsystem (2.11) for m times.*

More accurately, the multistep approach is summarized as follows.

Algorithm B: Multistep Scheme

1. Initialize value: Let $u_h^0 = I_h u_0$.

2. For any $n \ge 0$, supposing u_h^n is already known, find $u_{h,*}^{n+1}$ and u_h^{n+1} by solving
Convection subsystem: Let $u_{h,*}^n = u_h^n$; For any $i = 1, 2, \cdots, m$, solve

$$\eta_h^{n+\frac{i-1}{m}} = F(u_{h,*}^{n+\frac{i-1}{m}}) - \frac{1}{2} F_u(u_{h,*}^{n+\frac{i-1}{m}}) \nabla \cdot (F(u_{h,*}^{n+\frac{i-1}{m}})),$$

and compute $u_{h,*}^{n+\frac{i}{m}} \in X_h$, with $u_{h,*}^{n+\frac{i}{m}}|_{\Gamma^-_{n+i/m}} = I_h u_b^{n+\frac{i}{m}}$, such that

$$(u_{h,*}^{n+\frac{i}{m}}, v_h) = (u_h^{n+\frac{i-1}{m}}, v_h) \quad - \quad \delta t \left\langle \eta_h^{n+\frac{i-1}{m}} \cdot \mathbf{n}, v_h \right\rangle_{\Gamma \setminus \Gamma^-_{n+i/m}} + \delta t \left(\eta_h^{n+\frac{i-1}{m}}, \nabla v_h \right) \tag{2.13}$$

for any $v_h \in X_h$, with $v_h = 0$ on $\Gamma^-_{n+i/m}$;

and

Diffusion subsystem: Find $u_h^{n+1} \in X_h$ with $u_h^{n+1}|_\Gamma = I_h u_b^{n+1}$, and compute

$$(\Delta t)^{-1}(u_h^{n+1}, v_h) + \varepsilon(\nabla u_h^{n+1}, \nabla v_h) = (f^{n+1} + (\Delta t)^{-1} u_{h,*}^{n+1}, v_h) \tag{2.14}$$

for any $v_h \in X_h$, with $v_h = 0$ on Γ.

3. Let $n = n + 1$, until $n > N$, else return to Step 2.

Remark 4. *In this multistep algorithm, the inflow boundary* $\Gamma^-_{n+\frac{i}{m}}$ *for subsystem (2.13) is modified as* $\Gamma^-_{n+\frac{i}{m}} = \left\{\mathbf{x} \in \Gamma :\ F_u(u_b(\mathbf{x}, t^{n+\frac{i}{m}})) \cdot \mathbf{n}(\mathbf{x}) < 0\right\}$, *which is consistent with the previous definition* Γ^-_n.

Remark 5. *Actually, our multistep method is similar with the one used in wave-like equation strategy to compute convection advection equations (see e.g., Section 31.5 in [19] and the references therein). Likely, considering solving convection-dominated diffusion equations, the same multistep method was adopted and generally carried out, see for instance, References [21,22]. Moreover, we note that a multirate iterative scheme was proposed for solving poroelasticity model in a similar way of dealing with the Stokes and diffusion systems [23], more recently the multirate finite steps methods were provided to deal with the phenomena of different scales [24].*

3. Numerical experiments

This section aims to test the performance of the present scheme, including validness, stability and convergence. We use finite element package Freefem++ [25] to implement our scheme and carry out numerical experiments in the following.

3.1. Burgers-type equations

This subsection focuses on the particular case $F(u) = (P(u), Q(u))^T = (\frac{1}{2}u^2, \frac{1}{2}u^2)^T$, and $f = 0$. Actually, these choices give rise to the equations of Burgers type.

Problem 1. *The first example is a smooth problem on the unit square domain* $\Omega = [0,1]^2$, *and the exact solution is given by* $u(x,y,t) = 16x(1-x)y(1-y)(e^{10t}-1)/(e^{10}-1)$. *The original and boundary conditions depend on (1.1)–(1.3) and the analytical solution. We choose* $T = 1$, $\varepsilon = 10^{-8}$.

For this example, we can prove easily that for any $(x,y) \in \partial\Omega$ and $t \geq 0$, $F_u(u(x,y,t)) = (u,u)^T = (0,0)^T$, as well as $F_u(u(x,y,t)) \cdot \mathbf{n} = 0$, so the inflow boundary Γ^-_n is always empty for the convective subsystem (2.11) at each time step $t = t^n$ for any positive integer n satisfied the condition $t^n \leq T$.

Firstly, we test the order of convergence with fixing time step $\Delta t = 1/2^{16}$ and varying the mesh scale from $1/4$ to $1/64$ step by step. In Table 1, the computational results are presented and we can observe an optimal convergence of rate 2 for the numerical solutions under L^2-norm error.

Table 1. Rates of convergence and errors for Problem 1 ($\Delta t = 1/2^{16}$ and $\varepsilon = 10^{-8}$).

h	$\|u - u_h^N\|_{\mathrm{L}^2(\Omega)}$	order
1/4	4.15880e-2	-
1/8	8.97586e-3	2.2120
1/16	2.03951e-3	2.1378
1/32	4.83367e-4	2.0770
1/64	1.23786e-4	1.9652

Next, we settle the spatial mesh with $h = 1/128$ and along with the time step size varying from 0.1 to $0.1/2^6$. As it can be seen from Table 2, the Algorithm A (Single step scheme) can always derive

the convergent results up to terminal time T. According to this table, we can clearly obtain an optimal convergence of rate 1 under the L^2-norm of the numerical solutions with regard to the time step size.

Table 2. Rates of convergence and errors for Problem 1 ($h = 1/128$ and $\varepsilon = 10^{-8}$).

Δt	$\|u - u_h^N\|_{\mathrm{L}^2(\Omega)}$	order
0.1	3.18725e-1	-
$0.1/2$	1.47547e-1	1.1112
$0.1/2^2$	7.07840e-2	1.0597
$0.1/2^3$	3.46460e-2	1.0308
$0.1/2^4$	1.71370e-2	1.0156
$0.1/2^5$	8.52210e-3	1.0078
$0.1/2^6$	4.24952e-3	1.0039

Problem 2. *The second example is defined on the square* $\Omega = [-1, 1]^2$*, with the exact solution* $u(x, y, t) = -(1 - x^2)^2(1 - y^2)^2 \arctan\left(\frac{x+y+2-2t}{\nu}\right)$*. The initial and boundary conditions depend on (1.1)–(1.3) and the analytic solution. Besides, the parameters are selected as* $\varepsilon = 0.002$*,* $T = 1$ *and* $\nu = 0.5$*.*

Similarly with previous problem, for the Problem 2 we can check that, the inflow boundary Γ_n^- for the convective subsystem (2.11) is awayls null.

In order to check the convergence order in spatial scale, a smaller fixed time step $\Delta t = 1/2^{16}$ is utilized for the present algorithm. As to this time step, Algorithm A can constantly get convergent solutions up to $T = 1.0$, and obtain an optimal convergence of rate 2 with respect to mesh scales by observing the numerical simulations under the L^2-norm error, which are shown in Table 3.

Table 3. Rates of convergence of Algorithm A for Problem 2 ($\Delta t = 1/2^{16}$).

h	$\|u - u_h^N\|_{\mathrm{L}^2(\Omega)}$	order
1/4	3.62638e-1	-
1/8	8.17792e-2	2.1487
1/16	1.96814e-2	2.0549
1/32	4.93634e-3	1.9953
1/64	1.23852e-3	1.9948
1/128	3.12747e-4	1.9855

Now, we show numerical results in Table 4 with fixed spatial mesh ($h = 1/64$), time step size from 0.1 to $0.1/2^5$, and the same computational parameters. It displays the performance of the Algorithm B. Our observations for this testing case focus on the following two-folds: 1) An optimal convergence of order 1 under the convergence region with respect to the time step size is examined for the L^2-norm of the numerical simulations in time; 2) Algorithm A can get convergent solution for relatively large time step, for instance $\Delta t = 0.05$. Noting that for time step $\Delta t = 0.025$, the computational solution by Algorithm A shows apparently oscillation near the top region, which can be examined in the left contour plot of Figure 1. Compared with that, when using Algorithm B with multistep steps of $m = 4$

for convection subproblem, the smooth solution can be derived. This phenomena is observed in the right contour plot of Figure 1.

Table 4. Rates of convergence of Algorithm B for Problem 2 ($h = 1/64$).

Δt	$\|u - u_h^N\|_{L^2(\Omega)}$	order
0.1	divergence	-
0.1/2	1.21862e-1	-
$0.1/2^2$	4.55624e-2	1.4193
$0.1/2^3$	2.23644e-2	1.0267
$0.1/2^4$	1.11191e-2	1.0081
$0.1/2^5$	5.56048e-3	0.9998

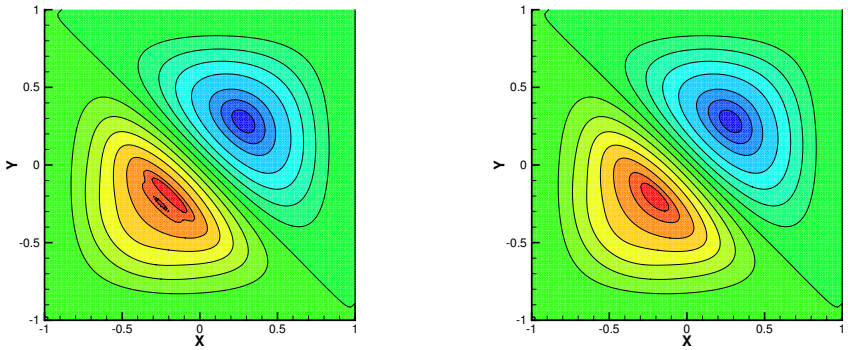

Figure 1. Numerical contors of u for Problem 2 with $\Delta t = 0.025$ at T=1. Left: Algorithm B with $m = 1$, Right: Algorithm B with $m = 4$.

Next, we apply the Algorithm B for different time step sizes to compute the numerical solutions with fixing the multistep size $m = 64$, which are displayed in Table 5. We can notice that Algorithm B always get convergent and smooth solutions for reasonable large multisteps, namely $m = 64$ in this testing case. In sum, this multistep scheme behaves like an unconditional stable scheme in this aspect. Apart from that, we also observe the optimal convergence order with regard to time step size.

Table 5. Rates of convergence of Algorithm B for Problem 2 ($m = 64$ and $h = 1/128$).

Δt	$\|u - u_h^N\|_{L^2(\Omega)}$	order
0.1	1.79769	-
0.1/2	8.97229e-1	1.0026
$0.1/2^2$	4.47486e-2	1.0036
$0.1/2^3$	2.23207e-2	1.0035
$0.1/2^4$	1.11494e-2	1.0014
$0.1/2^5$	5.58507e-3	0.9973

Moreover, at $\Delta t = 0.001$, the profiles of the solutions are shown in Figure 2, from which the smooth evolution of the numerical solutions at different time can be observed. Here, only the 3D plot of the numerical solutions at initial time and $t = 0.6, 0.7, 0.8, 0.9$ and terminal time $t = 1.0$ are shown.

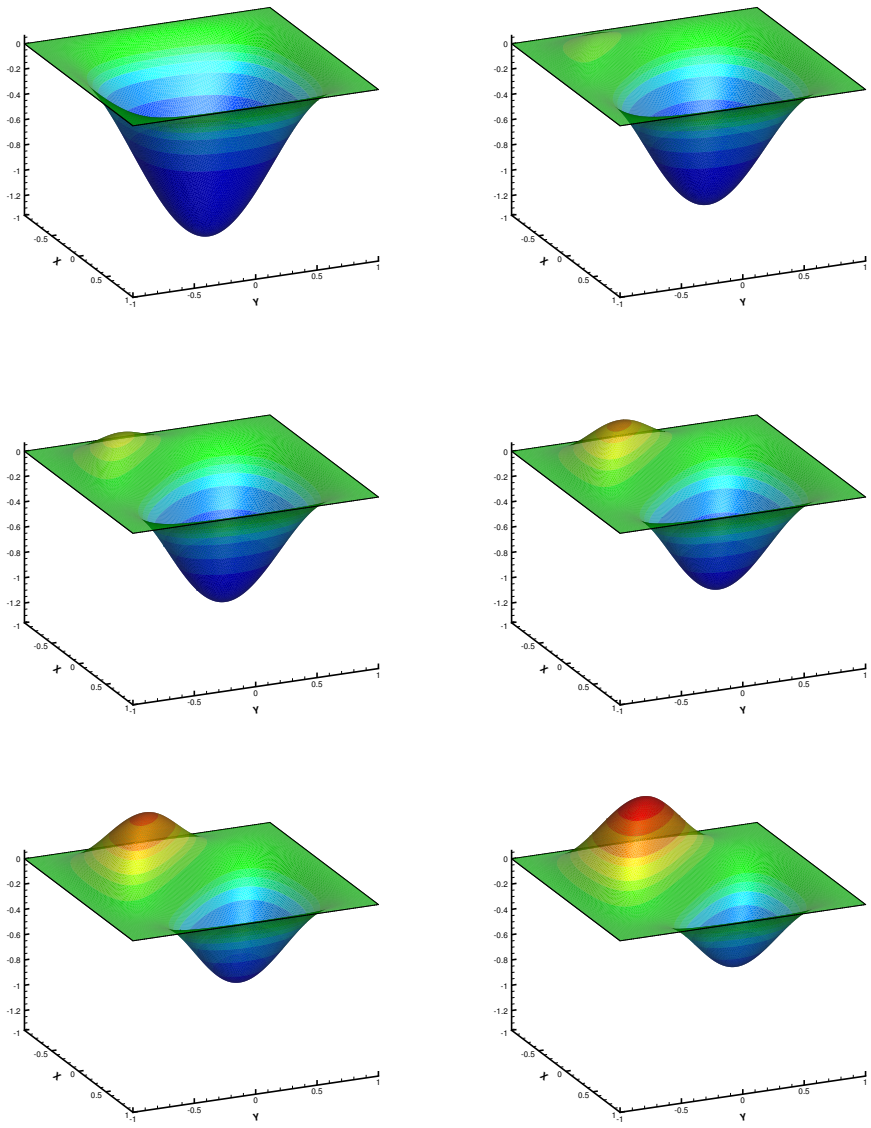

Figure 2. Computed solution of u for $\Delta t = 0.001$ at different times, using Algorithm A for Problem 2. Top: $t = 0, 0.6$; Middle: $t = 0.7, 0.8$; Bottom: $t = 0.9, 1.0$.

3.2. Buckley-leverett type equation

In this subsection, we will engage in the numerical simulations for two-dimensional Buckley-Leverett problem, which is a conservation system used to simulate two-phase flow in porous medium. The Buckley-Leverett problem or the Buckley-Leverett displacement describes an immiscible displace-

ment process, such as the displacement of oil by water.

Problem 3. *Consider the flux function in the form of* $P(u) = u + (u - 0.25)^3, Q(u) = -(u + u^2)$. *The computational domain is* $\Omega = [-0.5, 0.5]^2$, *and the exact solution is* $u(x, y, t) = 1/4 + xy$. *The source is* $f(x, y) = y + 3x^2y^3 - x(1.5 + 2xy)$, *and the parameters are selected as* $\varepsilon = 0.1, 0.01$, $T = 1$.

In this problem, we choose the same flux $F(u) = (P(u), Q(u))$ as the one from Example 2 of Reference [8]. For this flux, an exact solution with non-homogeneous boundary condition is constructed. Then, from these settings, since the exact solution is independent of time t, the present method can be interpreted as a pseudo-time method for solving nonlinear elliptic problems. In the present method (Algorithm A or B), we can easily check that $F_u \cdot \mathbf{n} < 0$ for the right and top boundaries, namely, $\{(x, y) : x = 0.5, -0.5 \le y \le 0.5; or - 0.5 \le x \le 0.5, y = 0.5\}$.

We apply Algorithm A to capture the numerical solutions up to $T = 1.0$ for two different viscosity parameters $\varepsilon = 0.1$ and 0.01 with fixing a much smaller time step size $\Delta t = 1/2^{16}$. As it is shown in Table 6, the convergence results of the second and fourth columns correspond to the viscosity $\varepsilon = 0.1$ and $\varepsilon = 0.01$, respectively. From this table, an optimal convergence of rate 2 with respect to mesh scales is derived for the numerical simulations under L^2-norm for both viscosity parameters.

Table 6. Rates of convergence of Algorithm A for Problem 3 ($\Delta t = 1/2^{16}$ and $\varepsilon = 0.1, 0.01$).

h	$\|u - u_h^N\|_{\mathrm{L}^2(\Omega)}$ ($\varepsilon = 0.1$)	order	$\|u - u_h^N\|_{\mathrm{L}^2(\Omega)}$ ($\varepsilon = 0.01$)	order
1/4	6.52497e-3	-	6.54057e-3	-
1/8	1.62734e-3	2.0035	1.61568e-3	2.0173
1/16	4.04613e-4	2.0079	4.00571e-4	2.0120
1/32	9.90670e-5	2.0301	9.72728e-5	2.0419
1/64	2.28654e-5	2.1152	2.18647e-5	2.1534
1/128	4.85729e-6	2.2349	5.36036e-6	2.0282

Next, the spatial mesh with $h = 1/128$ is fixed, and the time step size $0.1/2^k$ ($k = 1, 2, \cdots, 8$) are selected to check the performance of the Algorithm A or B. Firstly, for the case $\varepsilon = 0.1$, Table 7 shows that the single step scheme (Algorithm A) can always get the convergence results. From this table, the optimal first order convergence in time is verified under L^2-norm in time. Secondly, for the case $\varepsilon = 0.01$, as shown in the second column of Table 8, Algorithm A is convergent for $k \le 4$, and Algorithm B is then used by doubly increasing the multistep index m each time until deriving the convergence numerical solution at the terminal time for each $k > 4$. Then for these selected time step sizes, we apply Algorithm B with fixed multistep index $m = 32$ to capture the numerical solutions, of which the approximate errors under L^2-norm at terminal time are listed in the three column of Table 8. We can give some observation based on this testing case: 1) An optimal convergence of rate 1 with regard to the time step size is derived for the numerical simulations under L^2-norm in time for Algorithm B with fixed index m; 2) For fixed time step size, Algorithm B with different multistep indices (noting that Algorithm B with $m = 1$ reduces to Algorithm A) can get very close approximate accuracy, which shows that the multistep scheme for convection subproblem in Algorithm B mainly plays a significant role in improving the stability of the algorithm, not the computational accuracy.

Table 7. Rates of convergence of Algorithm A for Problem 3 ($h = 1/128$ and $\varepsilon = 0.1$).

Δt	$\|u - u_h^N\|_{\mathrm{L}^2(\Omega)}$	order
0.1	2.77557e-2	-
$0.1/2$	1.29973e-2	1.0946
$0.1/2^2$	6.83726e-3	0.9267
$0.1/2^3$	3.50402e-3	0.9644
$0.1/2^4$	1.77204e-3	0.9836
$0.1/2^5$	8.89513e-4	0.9943
$0.1/2^6$	4.44104e-4	1.0021
$0.1/2^7$	2.20378e-4	1.0109
$0.1/2^8$	1.08297e-4	1.0250

Table 8. Rates of convergence of Algorithm B for Problem 3 ($h = 1/128$ and $\varepsilon = 0.01$).

Δt	$\|u - u_h^N\|_{\mathrm{L}^2(\Omega)}$	$\|u - u_h^N\|_{\mathrm{L}^2(\Omega)}$ $(m = 32)$	order
$0.1/2$	1.55939e-2 (m = 32)	1.55939e-2	-
$0.1/2^2$	8.22204e-3 (m = 16)	8.20885e-3	0.9257
$0.1/2^3$	4.33042e-3 (m = 8)	4.30033e-3	0.9327
$0.1/2^4$	2.26302e-3 (m = 4)	2.21436e-3	0.9576
$0.1/2^5$	1.26284e-3 (m = 1)	1.12487e-3	0.9771
$0.1/2^6$	6.31144e-4 (m = 1)	5.65956e-4	0.9910
$0.1/2^7$	3.14027e-4 (m = 1)	2.82513e-4	1.0024
$0.1/2^8$	1.55175e-4 (m = 1)	1.39755e-4	1.0154

Problem 4. *Consider the same flux function as in Problem 3 with the computed domain selected as* $\Omega = [-2, 5]^2$. *The initial value is chosen as* $u(x, y, 0) = 1$ *in* $(x - 0.25)^2 + (y - 2.25)^2 < 0.5$ *and otherwise* $u(x, y, 0) = 0$. *The boundary condition is* $u(x, y, t) = 0$, *for any* $(x, y) \in \partial\Omega$ *and the right source is* $f = 0$. *The parameters are selected as* $\varepsilon = 0.01$, $T = 1.0$.

This problem with discontinuous initial data was first given in [8] by Karlsen and Risebro, where only the convergence rates were plotted for different CFL numbers. As to this problem, similarly with the previous problem, we can deduce that the inflow boundaries for Eq (2.5) with condition $F_u \cdot \mathbf{n} < 0$ are the right and top boundaries, namely, $\{(x, y) : x = -2, -2 \le y \le 5; or - 2 \le x \le 5, y = 5\}$.

We execute Algorithm A under the discretization parameters $\Delta t = 0.001$ and $h = 1/64$ up to the terminal time $T = 1.0$. The contour plot of the numerical solution at $T = 1.0$ is displayed in Figure 3, where the contour values with Labels 1–18 correspond to the lines located from outside to inside in order. Besides, we displayed the three dimensional plot of the computational numerical approximations at different times in Figure 4. From this figure, we can observe the smooth evolution of the numerical solutions by Algorithm A, which verifies the stability of the proposed algorithm.

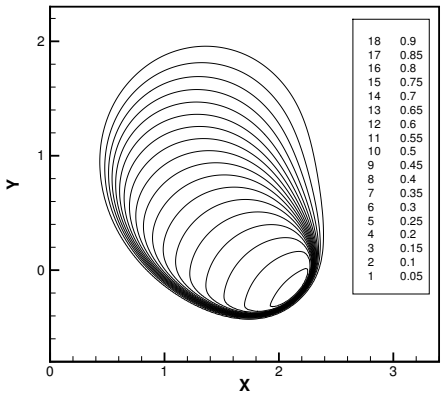

Figure 3. Contour plot of computed solution at $T = 0.1$ of Problem 4, using Algorithm A with $\Delta t = 0.001$ and $h = 1/64$.

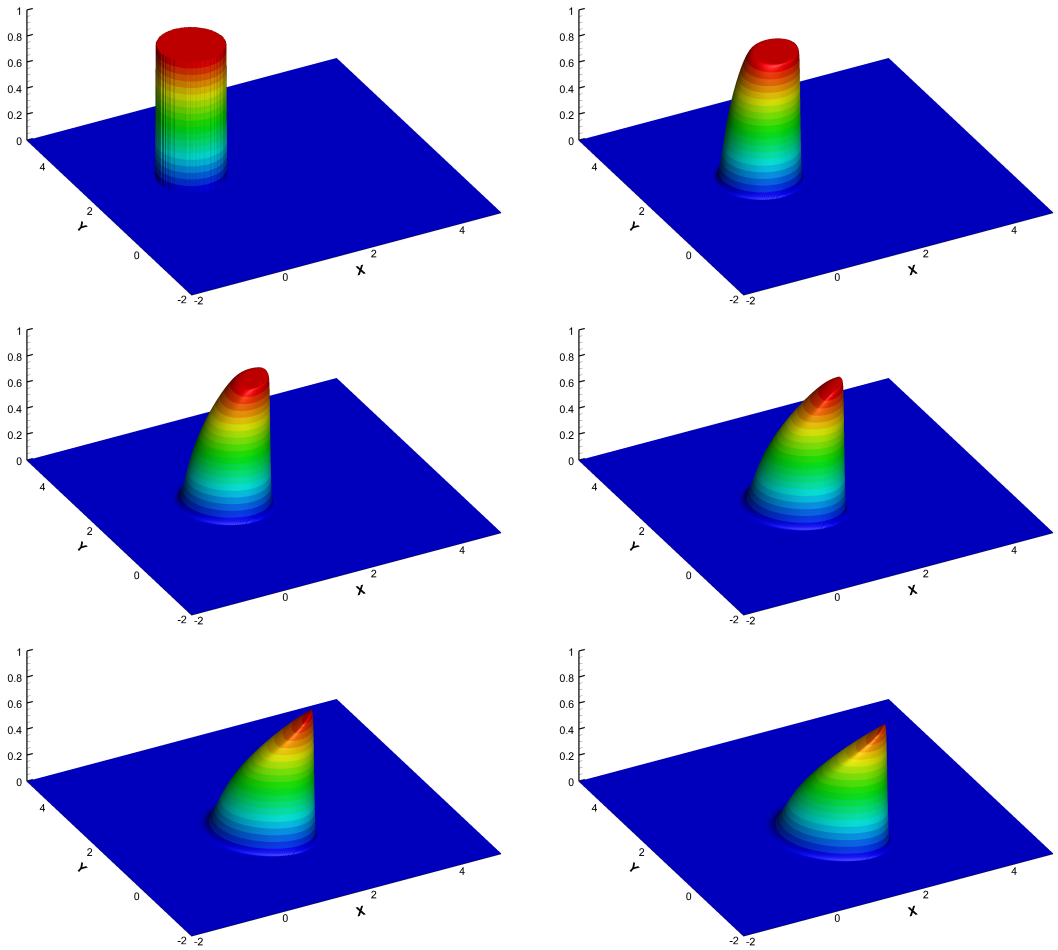

Figure 4. 3D plot of the computed solutions at different times of Problem 4, using Algorithm A with $\Delta t = 0.001$ and $h = 1/64$. Top: $t = 0, 0.2$; Middle: $t = 0.4, 0.6$; Bottom: $t = 0.8, 1.0$.

Problem 5. *Consider the flux function of the form*

$$P(u) = \frac{u^2}{u^2 + (1-u)^2}, \quad Q(u) = P(u)(1 - 5(1-u)^2).$$

The computed domain is $\Omega = [-1.5, 1.5]^2$*, the initial value is* $u(x, y, 0) = 1$ *in* $x^2 + y^2 < 0.5$*, and otherwise* $u(x, y, 0) = 0$*. The source component is* $f = 0$*. The homogeneous boundary condition is also assumed. The parameters are selected as* $\varepsilon = 0.01$*,* $T = 0.5$*.*

This problem also permitted discontinuous initial data and was first given in [8] by Karlsen and Risebroand, and since then was widely used for experimenting several numerical methods for nonlinear convection diffusion equations, see [7,9,12–14] and also [26,27] for slightly different problem domain or terminal time setting.

For this problem, we know that $u = 0$ for any $(x, y) \in \partial\Omega$ and $t \geq 0$, and by direct calculation

$$P_u(u) = \frac{2u\,(1 - u(2u-1))}{(u^2 + (1-u)^2)^2}, \quad Q_u(u) = P_u(u)(1 - 5(1-u)^2) + 10P(u)(1-u),$$

hence, $F_u(u) = (P_u(u), Q_u(u))^T = (0, 0)^T$, and $F_u(u(x, y, t)) \cdot \mathbf{n} = 0$, so the inflow boundary Γ_n^- is always empty for the convection subsystem (2.11) at each time step $t = t^n < T$.

Now we execute Algorithm A under the discretization parameters $\Delta t = 0.001$ and $h = 1/128$ up to the final time $T = 0.5$. Firstly, the contour plot of the numerical solution at $T = 0.5$ is displayed in Figure 5, where the contour values with Labels 1–19 correspond to the lines located from outside to inside in order.

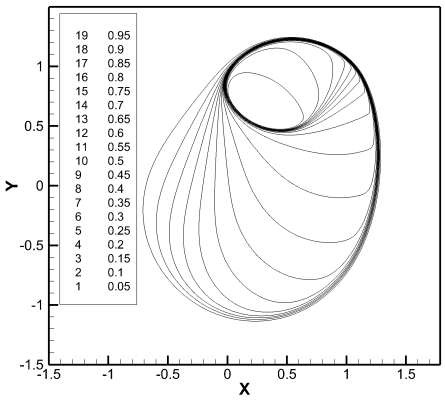

Figure 5. Contour plot of computed solution at $T = 0.5$ of Problem 5, using Algorithm A with $\Delta t = 0.001$ and $h = 1/128$.

The three dimensional plot of the computational numerical approximations at different times are drawn in Figure 6. From these figures, we can observe the smooth evolution of the numerical solutions by Algorithm A, which verifies the stability of the proposed algorithm. In summary, the effectivity of the present method is verified by good coincide of this contour with those ones in [7, 9, 12–14].

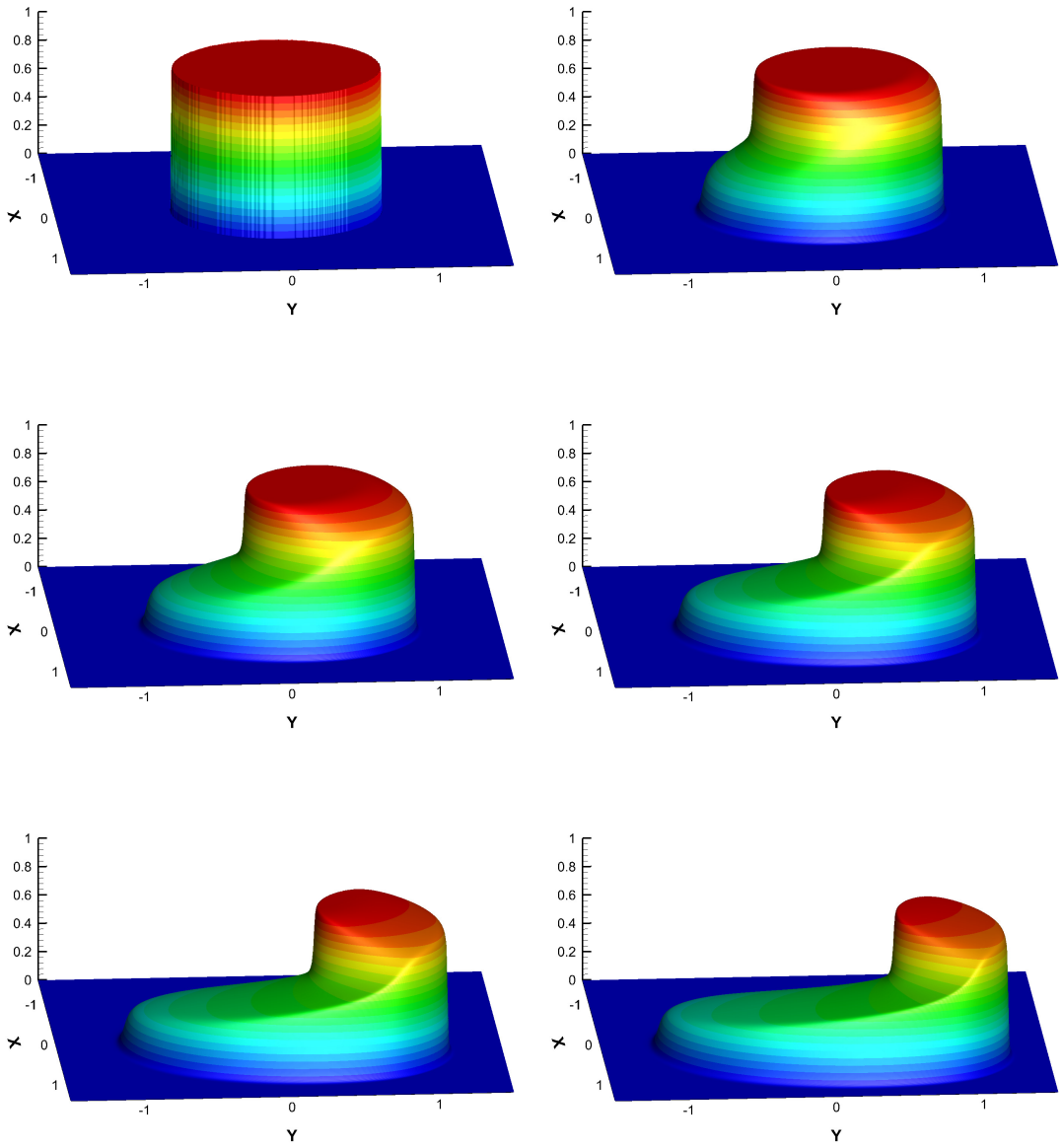

Figure 6. 3D plot of the computed solution at different times of Problem 5, using Algorithm A with $\Delta t = 0.001$ and $h = 1/128$. Top: $t = 0, 0.1$; Middle: $t = 0.2, 0.3$; Bottom: $t = 0.4, 0.5$.

4. Conclusions

We have introduced a new IMEX approach for computing the unsteady nonlinear convection-diffusion problems. Convection and diffusion subsystems can be both dealt with independently, and moreover, the stiffness matrix of the subsystems keep invariant along time marching. We can use many efficient preconditioned iterative solvers to calculate the diffusion subsystems because of its self-adjointness and coerciveness, like preconditioned Conjugate Gradient solver; at the same time, the

mass matrices are permitted in convection subsystems, as well as can be explicitly and fast computed when using spatial linear elements. We also provide a multistep technique to relax the instability, that mostly results from the explicit operation of the convection equations. All the above make the present approach a fast scheme for computing the unsteady nonlinear systems. In the end, we carry out a number of numerical simulations to evaluate the stability and validation of the algorithm. As for the spatial finite element approximation, some special triangulations like the Shishkin-type meshes [28] and Bakhvalov meshes [29,30], or adaptive grid method [31] along time marching can be incorporated with the present time-splitting scheme for the small diffusion coefficient in the future. Also the extensions of the present method to the three-dimensional problems (with complex domains) are possible and will be our further study.

Acknowledgments

The second and third authors are supported by Shenzhen Technology Project grant GXWD20201230155427003-20200822102539001. The fourth author is supported by Fundamental Research Funds for the Central Universities grant 2232019D3-39.

Conflict of interest

The authors declare there is no conflicts of interest.

References

1. J. M. Burgers, Mathematical examples illustrating relations occurring in the theory of turbulent fluid motion, *North-Holland Pub. Co. Amsterdam*, (1939), 1–53. https://doi.org/10.1007/978-3-319-11080-6-4
2. J. M. Burgers, A mathematical model illustrating the theory of turbulence, *Adv. Appl. Mech.*, (1948), 171–199. https://doi.org/10.1016/S0065-2156(08)70100-5
3. M. J. Lighthill, G. B. Whitham, On kinematic waves. II. Theory of traffic flow on long crowded roads, *Proc. Roy. Soc. 229A*, (1955), 317–345. https://doi.org/10.1098/rspa.1955.0089
4. D. W. Peaceman, Fundamentals of numerical reservoir simulation, *Elsevier, Amsterdam*, (1977), 1–190.
5. M. M. Cecchi, M. A. Pirozzi, High-order finite difference numerical methods for time-dependent convection-dominated problems, *Appl. Numer. Math.*, **55** (2005), 334–356. https://doi.org/10.5555/1133931.1133939
6. V. Sobotikova, M. Feistauer, Effect of numerical integration in the DGFEM for nonlinear convection-diffusion problems, *Numer. Meth. PDEs*, **23** (2007), 1368–1395. https://doi.org/10.1002/num.20225
7. B. C. Shi, Z. L. Guo, Lattice boltzmann simulation of some nonlinear convection-diffusion equations, *Comput. Math. Appl.*, **61** (2011), 3443–3452. https://doi.org/10.1016/j.camwa.2011.01.041

8. K. H. Karlsen, N. H. Risbro, An operator splitting method for nonlinear convection-diffusion equations, *Numer. Math.*, **77** (1997), 365–382. https://doi.org/10.1007/s002110050291

9. K. H. Karlsen, K. Brusdal, H. K. Dahle, S. Evje, K. A. Lie, The corrected operator splitting approach applied to a nonlinear advection-diffusion problem, *Comput. Methods Appl. Mech. Engrg.*, **167** (1998), 239–260. https://doi.org/10.1016/S0045-7825(98)00122-4

10. H. Nessyahu, E. Tadmor, Non-oscillatory central differencing for hyperbolic conservation laws, *J. Comput. Phys.*, **87** (1990), 408–463. https://doi.org/10.1016/0021-9991(90)90260-8

11. S. Jin, Z. Xin, The relaxation schemes for hyperbolic systems of conservation laws in arbitrary space dimensions, *Commun. Pure Appl. Math.*, **48** (1995), 235–276.

12. A. Kurganov, E. Tadmor, New high-resolution central schemes for nonlinear conservation laws and convection-diffusion equations, *J. Comput. Phys.*, **160** (2000), 241–282. https://doi.org/10.1006/jcph.2000.6459

13. Y. Jiang, Z. F. Xu, Parametrized maximum principle preserving limiter for finite difference WENO schemes solving convection-dominated diffusion equations, *SIAM J. Sci. Comput.*, **35** (2013), A2524–A2553. https://doi.org/10.1137/130924937

14. T. Xiong, J. M. Qiu, Z. F. Xu, High order maximum-principle-preserving discontinuous galerkin method for convection-diffusion equations, *SIAM J. Sci. Comput.*, **37** (2015), A583–A608. https://doi.org/10.1137/140965326

15. A. Chertock, A. Kurganov, On splitting-based numerical methods for convection-diffusion equations, http://www4.ncsu.edu/ acherto/papers/Chertock-Kurganov.pdf.

16. G. Akrivis, M. Crouzeix, C. Makridakis, Implicit-explicit multistep finite element methods for nonlinear parabolic problems, *Math. Comput.*, **67** (1998), 457–477.

17. X. H. Long, C. J. Chen, Implicit-explicit multistep characteristic finite element methods for nonlinear convection-diffusion equations, *Numer. Meth. PDEs*, **23** (2007), 1321–1342. https://doi.org/10.1002/num.20222

18. T. Zhang, X. L. Feng, J. Y. Yuan, Implicit-explicit schemes of finite element method for the non-stationary thermal convection problems with temperature-dependent coefficients, *Int. Commun. Heat Mass Transf.*, **76** (2016), 325–336. https://doi.org/10.1016/j.icheatmasstransfer.2016.06.011

19. R. Glowinski, Numerical methods for incompressible viscous flow, in *Handbook of Numerical Analysis*, (eds. P. G. Ciarlet and J. L. Lions), North-Holland, Amsterdam, (2003), 3–1176.

20. C. M. Chen, V. Thomée, The lumped mass finite element method for a parabolic problem, *J. Austral. Math. Soc. Ser. B*, **26** (1985), 329–354. https://doi.org/10.1017/S0334270000004549

21. F. Shi, G. P. Liang, Y. B. Zhao, J. Zou, New splitting methods for time-dependent convection-dominated diffusion problems, *Commun. Comput. Phys.*, **16** (2014), 1239–1262. https://www.researchgate.net/publication/235638692

22. F. Shi, H. B. Zheng, Y. Cao, J. Z. Li, R. Zhao, A fast numerical method for solving coupled burgers' equations, *Numer. Meth. PDEs*, **33** (2017), 1823–1838. https://doi.org/10.1002/num.22160

23. Z. Ge, M. Ma, Multirate iterative scheme based on multiphysics discontinuous Galerkin method for a poroelasticity model, *Appl. Numer. Math.*, **128** (2018), 125–138. https://doi.org/10.1016/j.apnum.2018.02.003
24. A. Naumann, J. Wensch, Multirate finite step methods, *Numer. Algor.*, **81** (2019), 1547–1571.
25. F. Hecht, O. Pironneau, K. Ohtsuka, FreeFEM++ version 3.43, 2016.
26. C. D. Acosta, C. E. Mejia, A mollification based operator splitting method for convection diffusion equations, *Comput. Math. Appl.*, **59** (2010), 1397–1408. https://doi.org/10.1016/j.camwa.2009.11.011
27. C. D. Acosta, R. Burger, Difference schemes stabilized by discrete mollification for degenerate parabolic equations in two space dimensions, *IMA J. Numer. Anal.*, **32** (2012), 1509–1540. https://doi.org/10.1093/imanum/drr049
28. G. I. Shishkin, Grid approximation of singularly perturbed elliptic and parabolic equations, *Comput. Math. Math. Phys.*, (2006), 388–401.
29. N. S. Bakhvalov, The optimization of methods of solving boundary value problems with a boundary layer, *USSR Comp. Math. Math. Phys.*, **9** (1969), 139–166. https://doi.org/10.1016/0041-5553(69)90038-X
30. Y. Chen, H. Leng, L. Liu, Error analysis for a non-monotone FEM for a singularly perturbed problem with two small parameters, *Adv. Appl. Math. Mech.*, **7** (2015), 196–206. https://doi.org/10.4208/aamm.2013.m399
31. Y. Chen, L. Liu, An adaptive grid method for singularly perturbed time-dependent convection-diffusion problems, *Commun. Comput. Phys.*, **20** (2016), 1340–1358. https://doi.org/10.4208/cicp.240315.301215a
32. F. J. Wang, C. Wang, Z. T. Chen, Local knot method for 2D and 3D convection-diffusion-reaction equations in arbitrary domains, *Appl. Math. Lett.*, **105** (2020), 0893–9659. https://doi.org/10.1016/j.aml.2020.106308
33. F. J. Wang, C. M. Fan, C. Z. Zhang, A localized space-time method of fundamental solutions for diffusion and convection-diffusion problems, *Adv. Appl. Math. Mech.*, **12** (2020), 940–958. https://doi.org/10.4208/aamm.OA-2019-0269

AIMS Press