

Research Article

A Self-Adaptive Numerical Method to Solve Convection-Dominated Diffusion Problems

Zhi-Wei Cao, Zhi-Fan Liu, Zhi-Feng Liu, and Xiao-Hong Wang

Department of Thermal Science and Energy Engineering, University of Science and Technology of China, Hefei, Anhui 230026, China

Correspondence should be addressed to Zhi-Fan Liu; zfliu13@ustc.edu.cn

Received 5 April 2017; Accepted 11 June 2017; Published 17 July 2017

Academic Editor: Mohsen Sheikholeslami

Copyright © 2017 Zhi-Wei Cao et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Convection-dominated diffusion problems usually develop multiscaled solutions and adaptive mesh is popular to approach high resolution numerical solutions. Most adaptive mesh methods involve complex adaptive operations that not only increase algorithmic complexity but also may introduce numerical dissipation. Hence, it is motivated in this paper to develop an adaptive mesh method which is free from complex adaptive operations. The method is developed based on a range-discrete mesh, which is uniformly distributed in the value domain and has a desirable property of self-adaptivity in the spatial domain. To solve the time-dependent problem, movement of mesh points is tracked according to the governing equation, while their values are fixed. Adaptivity of the mesh points is automatically achieved during the course of solving the discretized equation. Moreover, a singular point resulting from a nonlinear diffusive term can be maintained by treating it as a special boundary condition. Several numerical tests are performed. Residual errors are found to be independent of the magnitude of diffusive term. The proposed method can serve as a fast and accuracy tool for assessment of propagation of steep fronts in various flow problems.

1. Introduction

It is well known that convection-diffusion equations arise in a variety of important science and engineering fields, for example, thermodynamics, fluid mechanics, and chemical reactions [1, 2]. In many applications recognized as convection-dominated problems, diffusion may be quite small in contrast to convection, so that the solution will develop steep moving fronts that are nearly shocks. Efficient and accurate assessment of propagations of moving fronts is important. However, standard Finite Difference Methods (FDM) tend to introduce spurious oscillations and stabilized methods often resort to upwind schemes, which introduce excessive numerical dissipations that will oversmooth the fronts [3, 4]. Consequently, an extremely fine mesh is needed to increase the resolution, making the convection-diffusion equation a most challenging one to be solved numerically.

There are mainly two lines of research in numerical methods for achieving higher resolution with limited computation cost. One is the development of higher-order methods trying to reduce the numerical diffusion, such as high-order FDM

with flux limiters [5–10] and the method of characteristics [3, 11–14]. The other important approach is using adaptive meshes. That is, mesh points are concentrated in the region where the solution varies steeply and is less concentrated in slowly varying regions.

Different adaptive strategies, for example, Adaptive Mesh Refinement method and moving mesh methods, have shown success in solving convection-diffusion problems [15–22]. As an adaptive scheme is designed to yield highly nonuniform mesh, discretization of the governing equation on general meshes should be settled first [23]. As for how to locate mesh points adaptively, such methods often appeal to equidistributing a monitor function or solving mesh equations [19]. As a whole, adaptive methods generally consist of two parts: one is to locate mesh points adaptively and the other is to solve the governing equation on the nonuniform mesh. As a result, computation complexity is certainly increased. Moreover, a local interpolation is usually needed to project the information to new meshes. Such a remeshing operation often brings numerical diffusion. So, it is desirable to develop a numerical

method that solves the problem adaptively but without complex adaptive operations.

In this paper, we propose a self-adaptive numerical method to capture the steep fronts accurately. We first consider the following convection-diffusion equation to illustrate the main ideas of our method [1, 2]:

$$\frac{\partial S}{\partial t} + \frac{\partial f(S)}{\partial x} = \frac{\partial}{\partial x} \left(\epsilon(S) \frac{\partial S}{\partial x} \right), \quad (1)$$

where the diffusive coefficient $\epsilon(S)$ is positive. Instead of a traditional spatial-discretized mesh, the proposed method is based on a “range-discrete mesh” which is uniformly distributed in the value domain [24]. It is found that the range-discrete mesh has an adaptive nature because the density of mesh points is proportionate to the spatial gradient of the solution. During the computation process, positions of the mesh points are determined by the governing equation, while their values are fixed. In this way, movement of the mesh points not only solves the equation but also locates mesh points adaptively. In other words, the proposed method combines the two parts of an adaptive method, the adaptive strategy and solving the equation, in one procedure. Therefore, neither complex adaptive operations nor any interpolations are involved in the proposed method, leading to a simple and efficient method for convection-diffusion problems.

It must be emphasized that, with a linear diffusive coefficient, the solution of (1) is smooth, though there exist steep gradients, which is the most concerned situation in the literature. However, with a nonlinear diffusive coefficient, though small, spatial gradient of the solution can be discontinuous or even infinite [25–27]. This singularity phenomenon is common in many applications but often smeared out by traditional methods. By simply treating it as a boundary condition, such a singular point can be maintained in our method.

It should be noted that, for monotone solutions, the proposed method is similar to the method of Fayers and Sheldon [25]. Unfortunately, as the unknown is no longer single valued in their Lagrangian form, their method does not work for nonmonotone solutions. The proposed method overcomes this disadvantage by discretizing the governing equation over dynamic control volumes. Furthermore, how to deal with different kinds of boundary conditions with the proposed method is discussed.

This paper is organized as follows. Section 2 introduces the range-discrete strategy to obtain an adaptive mesh. Next, the numerical scheme based on the range-discrete mesh, including boundary conditions, will be illustrated in detail in Sections 3, 4, and 5. Several numerical tests are conducted to validate the numerical method in Section 6 and conclusions are drawn in Section 7.

2. Range-Discrete Mesh

Considering that the unknown in (1) has bounded value, that is, $S(x, t) \in [m, M]$, a series of discrete values can be obtained as $m \leq S_0 < S_1 < \dots < S_j < \dots < S_k \leq M$, where $S_{j+1} = S_j + \Delta S_j$. Then, one can get a series of intersection points of the line sets $S = S_j$ ($j = 0, 1, \dots, k$) and the curve $S = S(x, t)$,

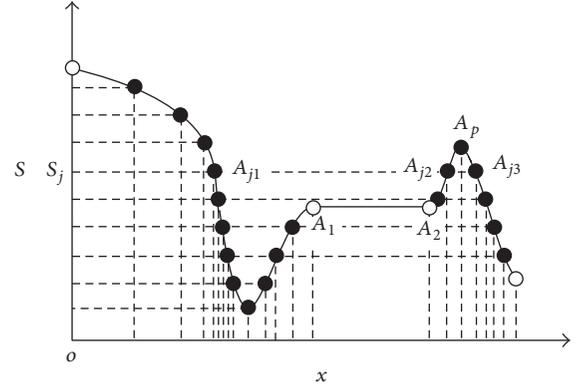


FIGURE 1: Description of the range-discrete mesh.

namely, A_i ($i = 0, 1, \dots$) as graphed in Figure 1. Then, with the location of each intersection known, approximation of $S(x, t)$ is obtained. Thus, a mesh can be formed by treating each intersection as a mesh point.

For a constant piece in the solution curve, the governing equation is satisfied trivially. The solutions on both sides of this constant piece do not interact with each other until the constant piece disappears. Consequently, each endpoint of the constant piece can be treated as a boundary condition for each side, respectively. With two adjacent mesh points assigned to the same value and marked as boundary points, for example, A_1 and A_2 shown in Figure 1, a constant piece is sufficiently described.

If $S = S(x)$ is nonmonotone, several intersections of the line $S = S_j$ and the curve $S = S(x)$ exist, for example, points A_{j1}, A_{j2}, A_{j3} shown in Figure 1. It is intuitive to employ all these intersections, assigned with the same discrete value S_j , as different mesh points to describe the nonmonotone feature. Moreover, a mesh point should also be set at the position of an extremum of $S = S(x)$, for example, point A_p in Figure 1, because, without it, the solution between A_{j2} and A_{j3} will be identified as a constant piece.

In contrast to spatial-discrete mesh used by FDM and many other methods, the discrete mesh obtained using the rules above is named as a “range-discrete” mesh. We should state that, though ΔS_j can be different from each other, we focus our attention on the equidistance situation for the sake of simplicity.

If there exist n mesh points in the spatial area $[x_a, x_b]$, the density of mesh points can be defined as $\rho = n/|x_b - x_a|$. For a range-discrete mesh, variation of the solution over the area $[x_a, x_b]$ can be defined as $TV_{[x_a, x_b]}(S) = \sum_{j=1}^{n-1} |S(x_{j+1}) - S(x_j)| = (n-1)\Delta S$. Thus, we can obtain

$$\rho \propto \frac{TV(S)_{[x_a, x_b]}}{|x_a - x_b|}, \quad (2)$$

where $TV(S)_{[x_a, x_b]}/|x_a - x_b|$ can be interpreted as the spatial variation speed of the solution. Thus, we can conclude from (2) that ρ is proportionate to the spatial variation speed of the solution, so that the density of mesh points is large where the solution varies quickly and vice versa, indicating an adaptive nature, which can also be seen from Figure 1. As a special case,

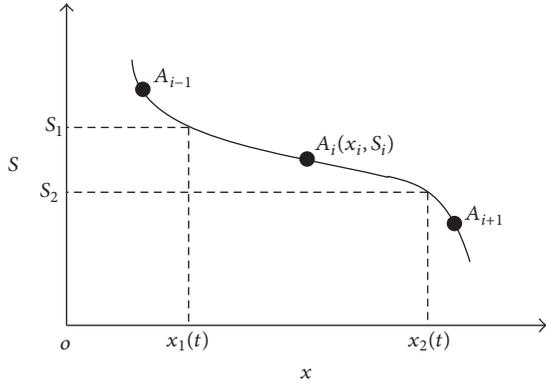


FIGURE 2: A dynamic control volume around a mesh point.

the density of mesh points for a constant piece is actually zero, being quite computational effort saving.

In fact, as $\lim_{|x_a - x_b| \rightarrow 0} TV(S)_{[x_a, x_b]} / |x_a - x_b| = |S_x(x)|$, we can observe an equidistribution of the solution gradient $|S_x(x)|$ in the range-discrete grid. This is an obvious adaptive criterion widely used in many adaptive mesh methods [19, 28, 29]. While complex adaptive operations have to be involved in these methods, it is done by the range-discrete mesh simply via an equidistance discretization in value domain.

With the technique illustrated above, a range-discrete mesh can be obtained for general solutions. Next, the numerical scheme based on the range-discrete mesh for solving (1) will be present.

3. Numerical Scheme Based on the Range-Discrete Mesh

For the initial-boundary value problem of (1), one can get an initial range-discrete mesh using the technique illustrated in Section 2. Once a range-discrete mesh is obtained, a dynamic control volume can be formed around each mesh point. Take $[x_1(t), x_2(t)]$ in Figure 2 as an example, and let $S(x_1(t), t) \equiv S_1 = (S_i + S_{i-1})/2$ and $S(x_2(t), t) \equiv S_2 = (S_{i+1} + S_i)/2$, making the control volume moving with the flow. Integrating (1) on this control volume and a small time interval $[t_0, t_1]$ where $t_1 - t_0 = \Delta t$, one can get

$$\begin{aligned} & \int_{t_0}^{t_1} \int_{x_1(t)}^{x_2(t)} \frac{\partial S}{\partial t} dx dt + \int_{t_0}^{t_1} \int_{x_1(t)}^{x_2(t)} \frac{\partial f(S)}{\partial x} dx dt \\ & = \int_{t_0}^{t_1} \int_{x_1(t)}^{x_2(t)} \frac{\partial}{\partial x} \left(\epsilon(S) \frac{\partial S}{\partial x} \right) dx dt, \end{aligned} \quad (3)$$

where the integration of the convective and diffusive term can be done directly as

$$\begin{aligned} & \int_{t_0}^{t_1} \int_{x_1(t)}^{x_2(t)} \frac{\partial f(S)}{\partial x} dx dt \\ & = \int_{t_0}^{t_1} [f(S(x_2(t), t)) - f(S(x_1(t), t))] dt \\ & = \int_{t_0}^{t_1} [f(S_2) - f(S_1)] dt, \end{aligned}$$

$$\begin{aligned} & \int_{t_0}^{t_1} \int_{x_1(t)}^{x_2(t)} \frac{\partial}{\partial x} \left(\epsilon(S) \frac{\partial S}{\partial x} \right) dx dt \\ & = \int_{t_0}^{t_1} [(\epsilon(S) S_x)|_{x_2(t)} - (\epsilon(S) S_x)|_{x_1(t)}] dt \\ & = \int_{t_0}^{t_1} [\epsilon(S_2) S_{x_2} - \epsilon(S_1) S_{x_1}] dt. \end{aligned} \quad (4)$$

As for the transient term, according to Appendix A, if the solution curve $S(x, t)$ is monotone in $[x_1(t), x_2(t)]$, we have

$$\begin{aligned} \int_{t_0}^{t_1} \int_{x_1(t)}^{x_2(t)} \frac{\partial S}{\partial t} dx dt & = - \int_{S_1}^{S_2} x(S, t_1) dS \\ & \quad + \int_{S_1}^{S_2} x(S, t_0) dS. \end{aligned} \quad (5)$$

Consequently, the integration form of (1) is

$$\begin{aligned} & \int_{S_1}^{S_2} x(S, t_1) dS - \int_{S_1}^{S_2} x(S, t_0) dS \\ & = \int_{t_0}^{t_1} [f(S_2) - \epsilon(S_2) S_{x_2}] - [f(S_1) - \epsilon(S_1) S_{x_1}] dt. \end{aligned} \quad (6)$$

The item on the left-hand side of (6) is the variation of the total quantity of the control volume over the time interval $[t_0, t_1]$; $f(S_1) - \epsilon(S_1) S_{x_1}$ and $f(S_2) - \epsilon(S_2) S_{x_2}$ can be interpreted as the left and right boundary flux, respectively. Thus, (6) is actually a natural result of mass conservation principle. It should be emphasized that (6) is exact as it is derived from (1) without any approximation.

Using linear approximation in the FDM manner, we can obtain first-order discrete form of (6) as

$$\begin{aligned} & (x_i^{t_0 + \Delta t} - x_i^{t_0}) (S_2 - S_1) \\ & = (f(S_2) - \epsilon(S_2) S_{x_2}) \Delta t \\ & \quad - (f(S_1) - \epsilon(S_1) S_{x_1}) \Delta t, \end{aligned} \quad (7)$$

where

$$\begin{aligned} S_2 & = \frac{S_{i+1} + S_i}{2}, \\ S_1 & = \frac{S_i + S_{i-1}}{2}, \\ S_{x_2} & = \frac{S_{i+1} - S_i}{x_{i+1} - x_i}, \\ S_{x_1} & = \frac{S_i - S_{i-1}}{x_i - x_{i-1}}. \end{aligned} \quad (8)$$

If the items on the right-hand side of (7) are the values at t , the discrete equation is explicit; and if they are values at $t + \Delta t$, it is implicit. One may notice that (7) is quite similar to the discrete equations of FDM and can be solved in the same way

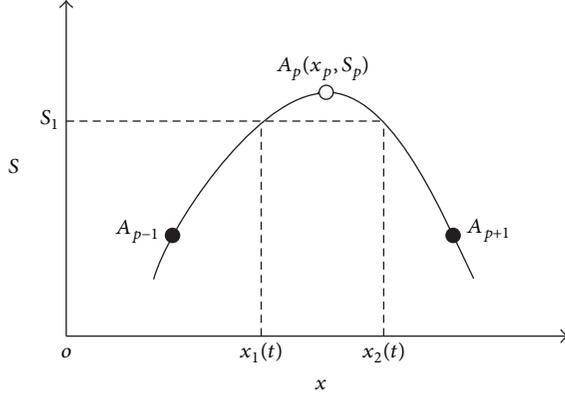


FIGURE 3: The dynamic control volume around an extremum point.

of FDM, for example, Newton's method. Generally speaking, other kinds of discrete equations and higher-order forms can also be obtained in the same manner as FDM.

However, we should emphasize that the unknown in (7) is x_i , not S_i , which is the essential difference between the proposed method and FDM. It means that the movements of the mesh points are tracked while their values are fixed throughout the computation process. When new positions of the mesh points are located, the approximation of the solution is obtained. Moreover, adaptivity is obtained simultaneously as a benefit of the range-discrete strategy.

We should note that though derived in different manners, (7) is similar to that in the method of Fayers and Sheldon [25]. And also, it only fits for monotone solutions in $[x_1(t), x_2(t)]$. To make the range-discrete grid work for nonmonotone solutions, adjustments are needed.

4. Adjustments for the Extremum Point

As has been mentioned in Section 2, a mesh point, for example, A_p in Figure 1 or Figure 3, is set at the extremum if the solution is nonmonotone. For the control volume of such an extremum point, (6) does not fit and adjustments are needed.

The dynamic control volume $[x_1(t), x_2(t)]$ for an extremum mesh point A_p is shown in Figure 3, where $S_1 = S_2 = S_{p-1} + \Delta S/2$. According to Appendix A, The items inside the brackets in (3) for the control volume of an extremum point are

$$\int_{t_0}^{t_1} \int_{x_1(t)}^{x_2(t)} \frac{\partial S}{\partial t} dx dt = \int_{x_1(t_1)}^{x_2(t_1)} (S(x, t + \Delta t) - S_2) dx - \int_{x_1(t_0)}^{x_2(t_0)} (S(x, t) - S_2) dx. \quad (9)$$

So, we have

$$\begin{aligned} & \int_{x_1(t_1)}^{x_2(t_1)} (S(x, t_1) - S_2) dx - \int_{x_1(t_0)}^{x_2(t_0)} (S(x, t_0) - S_2) dx \\ &= \int_{t_0}^{t_1} [-\epsilon(S_2) S_{x_2}] - [-\epsilon(S_1) S_{x_1}] dt. \end{aligned} \quad (10)$$

With an parabolic approximation of the solution curve between $[x_1(t), x_2(t)]$, we obtain

$$S(x, t) = S_p - \frac{4(S_p - S_{p-1})}{(x_{p+1} - x_{p-1})^2} \left(x - \frac{x_{p+1} + x_{p-1}}{2} \right)^2. \quad (11)$$

Consequently, we have

$$\begin{aligned} & \int_{x_1}^{x_2} (S(x, t) - S_2) dx \\ &= \frac{2}{3} \sqrt{\frac{S_p - S_1}{S_{p-1} - S_1}} (S_p - S_1) (x_{p+1} - x_{p-1}), \end{aligned} \quad (12)$$

$$\epsilon(S_2) S_{x_2} - \epsilon(S_1) S_{x_1}$$

$$= -8\epsilon(S_1) \sqrt{\frac{S_p - S_1}{S_{p-1} - S_1}} \frac{S_p - S_{p-1}}{x_{p+1} - x_{p-1}}.$$

Thus, the discrete equation for an extremum point is obtained.

One can conclude from (10) that, with a positive $\epsilon(S)$, left-hand side of (10) always decreases as time goes on. It means that the value of the extremum S_p is approaching S_1 , making the extremum point a special case: its value and position are always changing and the control volume is shrinking.

To avoid too small a control volume, the mesh points A_{p-1} and A_{p+1} will be removed from the range-discrete mesh if $|S_p - S_1|$ is less than a threshold value, for example, $|S_p - S_1| < \Delta S/4$. Then, let $S_1 = S_{p-2} + \Delta S/2$ and a new control volume is formed immediately to continue the computation. In this way, the extremum point in the initial mesh is always an extremum point and represents the movement of the extremum in the solution.

5. Boundary Conditions

5.1. Dirichlet Boundary Condition. A Dirichlet boundary condition, that is, $S(x_d, t) = S_d$, can be used directly in the discrete equation in the same way as FDM. That is, one can consider a discrete point A_k , set at the boundary to provide the boundary information, with its position and value always fixed as $x_k = x_d$ and $S_k = S_d$ during the computing process.

5.2. Moving Boundary Condition. As has been mentioned in Section 2, either endpoint of a constant piece in the solution is treated as a boundary condition. The value of this boundary is fixed, which is quite similar to the Dirichlet boundary condition, but the position of the boundary varies. Thus, we name it a moving boundary condition.

At a moving boundary, we have $S(\lambda(t), t) = S_f$ and $(\epsilon(S)S_x)|_{\lambda(t)} = 0$ (see Appendix B). Thus, it can be involved in the range-discrete mesh by a boundary point A_k with $S_k = S_f$ and $(\epsilon(S)S_x)|_{A_k} = 0$. The boundary condition is not involved in a discrete manner as what was done in [25], but in a more precise manner, making it more promising for high precision.

One can see that the position of the moving boundary point is not really needed in the range-discrete equation.

However, a constant piece in the solution might disappear and the solution on two sides will then interact with each other. To illustrate this, we need to merge the two boundary points into a normal mesh point if their distance is less than a threshold d_{\min} . The position of the boundary point is only needed here and can be interpolated using $S(x) = a_0(x - \lambda)^n$ by two nearby mesh points, where n can be settled using the method illustrated in Appendix B.

As a benefit of such a special boundary condition, the singular point, where the gradient of the solution might be discontinuous or infinite, as indicated in Appendix B, can be maintained by the proposed method. Numerical tests in Section 6 also confirm this.

5.3. Neumann Boundary Condition. Given a Neumann boundary condition $S_x(x_b) = q_b$, a boundary mesh point A_k will also be set. The discrete value of A_k is determined dynamically both by the value of A_i , which is the point nearest the boundary, and by the sign of q_b , that is, $S_k = S_i - \text{sgn}(q_b)\Delta S$ for the left boundary and $S_k = S_i + \text{sgn}(q_b)\Delta S$ for the right boundary. The position of A_k can be calculated by an linear interpolation:

$$q_b = \frac{S_k - S_i}{x_k - x_i} \quad (13)$$

or other higher-order interpolations. As we can see, the position of a Neumann boundary point determined by (13) can be either in or out of the calculation area. So, we design the following rules:

- (1) If x_k is out the calculation area, computing continuous on.
- (2) If x_k is in the calculation area, this boundary point becomes a normal mesh point and a new boundary point will be inserted into the range-discrete mesh.
- (3) If x_i is out of the calculation area, A_i will be removed immediately and the information of A_k is determined by the new nearest boundary mesh point.

6. Numerical Examples

6.1. A Travelling Wave Solution. With $f(S) = S^2/2$ and $\epsilon(S)$ being a constant, we have Burgers' equation from (1):

$$S_t + \left(\frac{1}{2}S^2\right)_x = \epsilon S_{xx}. \quad (14)$$

It is easy to verify that this equation has the travelling wave solution:

$$S(x, t) = \frac{2c\epsilon^{c(ct-x)/\epsilon}}{1 + e^{c(ct-x)/\epsilon}}, \quad (15)$$

where c is the speed of the wave. With $c = 1$, (15) is the solution of (14) with an initial condition $S(x, 0) = 2e^{-x/\epsilon}/(1 + e^{-x/\epsilon})$. Figure 4 and Table 1 show the solutions and error tables at different time using the proposed method when $\epsilon = 10^{-1}, 10^{-2}, 10^{-3}$. It can be seen from Figure 4 that mesh points

are located adaptively according to the spatial gradient. Moreover, the adaptivity follows the movement of the travelling front. As a benefit, with only a few mesh points, the front is solved with good accuracy.

One should notice that, with the diffusive coefficient ϵ decreasing, the front becomes steeper, tending to form a moving shock. While it is hard to maintain the accuracy near the front with the same number of mesh points using FDM, the accuracy of the proposed method for different diffusive coefficients is uniform.

Furthermore, the calculating area of this problem is actually infinite, that is, $x \in [-\infty, +\infty]$. While most numerical methods approach this within a finite area, the proposed method solves this problem in an infinite area simply with two moving boundary conditions: $S(-\infty) = 1, S_x(-\infty) = 0$ and $S(+\infty) = 0, S_x(+\infty) = 0$.

6.2. A Nonmonotone Case. In this numerical example, the solution of Burgers' equation is designed to be nonmonotone to test the numerical method. The initial condition and boundary condition are set to be $S(x, 0) = \sin(\pi x), S(0, t) = 0$ and $S(1, t) = 0$, respectively. According to Caldwell and Smith [30], the solution is

$$S(x, t) = \frac{2\pi\epsilon \sum_{n=1}^{\infty} a_n e^{-n^2\pi^2\epsilon t} n \sin(n\pi x)}{a_0 + \sum_{n=1}^{\infty} a_n e^{-n^2\pi^2\epsilon t} \cos(n\pi x)}, \quad (16)$$

where

$$\begin{aligned} a_0 &= \int_0^1 \exp\{-(2\pi\epsilon)^{-1}[1 - \cos(\pi x)]\} dx, \\ a_n &= 2 \int_0^1 \exp\{-(2\pi\epsilon)^{-1}[1 - \cos(\pi x)]\} \cos(n\pi x) dx \quad (17) \\ &\quad (n = 1, 2, 3, \dots). \end{aligned}$$

The numerical solutions at different times with diffusive coefficients $\epsilon = 10^{-1}, 10^{-2}, 10^{-3}$ are given by Figure 5 and the errors are Table 2. The results also show good performance of the proposed method for nonmonotone solutions. Again, the adaptivity functions well and tracks the front in an accurate manner.

6.3. Nonlinear Diffusive Term. How a nonlinear diffusive term effects the distribution of the front is tested here. Consider Burgers' equation with a nonlinear diffusive term:

$$S_t + \left(\frac{1}{2}S^2\right)_x = (\epsilon(S) S_x)_x \quad (18)$$

with two Dirichlet boundary conditions $S(0, t) = 1, S(1, t) = 0$ and a lamp initial condition, which is $S(0 \leq x \leq 0.2, t = 0) = 1 - 5x$ and $S(0.2 < x \leq 1, t = 0) = 0$. Diffusive coefficients set as $\epsilon(S) = 0.1S^\alpha$ with $\alpha = 0.5, 1, 2$ are tested. Solutions at $t = 0.5$ are shown in Figure 6.

It is shown by Figure 6 that the gradient of the solution with $\alpha = 0.5$ tends to be zero at the front and thus a smooth solution. However, the gradient with $\alpha = 2$ tends to be infinite at the front, resulting in a continuous but not

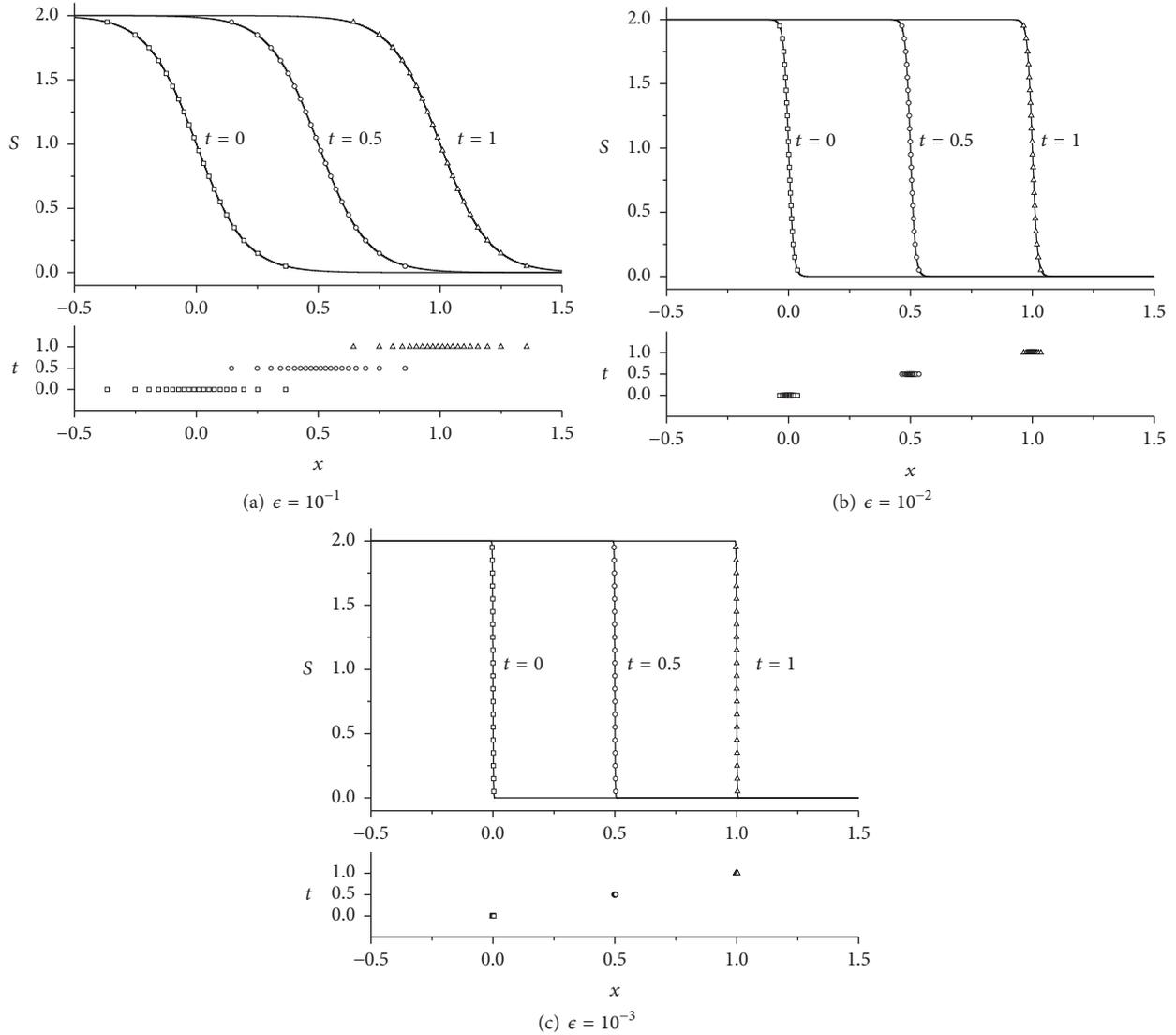


FIGURE 4: Numerical solutions at $t = 0.5$ and $t = 1$ of the travelling wave with different diffusive coefficients. Solid line represents exact solutions and squares, circles, and triangles in S - x figures are numerical solutions. The t - x figures below each S - x figure show the spatial distribution of the mesh points at different time.

TABLE 1: Error table of the numerical solution of Burgers' equation with different diffusive coefficients at different time. N is the number of mesh points.

ϵ	t	errors	$N = 10$	20	40	80	160
10^{-1}	0.5	L^∞ error	$1.00E-02$	$5.23E-03$	$2.70E-03$	$1.37E-03$	$6.95E-04$
		L^1 error	$4.43E-03$	$2.39E-03$	$1.35E-03$	$7.61E-04$	$4.21E-04$
	1.0	L^∞ error	$1.13E-02$	$5.83E-03$	$2.96E-03$	$1.50E-03$	$7.52E-04$
		L^1 error	$5.43E-03$	$2.77E-03$	$1.39E-03$	$7.17E-04$	$3.74E-04$
10^{-2}	0.5	L^∞ error	$1.16E-02$	$5.97E-03$	$3.03E-03$	$1.53E-03$	$7.66E-04$
		L^1 error	$5.64E-04$	$2.94E-04$	$1.50E-04$	$7.59E-05$	$3.82E-05$
	1.0	L^∞ error	$1.16E-02$	$5.97E-03$	$3.03E-03$	$1.53E-03$	$7.66E-04$
		L^1 error	$5.64E-04$	$2.94E-04$	$1.50E-04$	$7.59E-05$	$3.82E-05$
10^{-3}	0.5	L^∞ error	$1.16E-02$	$5.97E-03$	$3.03E-03$	$1.53E-03$	$7.66E-04$
		L^1 error	$5.64E-05$	$2.94E-05$	$1.50E-05$	$7.59E-06$	$3.82E-06$
	1.0	L^∞ error	$1.16E-02$	$5.97E-03$	$3.03E-03$	$1.53E-03$	$7.66E-04$
		L^1 error	$5.64E-05$	$2.94E-05$	$1.50E-05$	$7.59E-06$	$3.82E-06$

TABLE 2: Error table of the numerical solution of the burgers' equation with different diffusive coefficients at different time. N is the number of mesh points.

ϵ	t	errors	$N = 10$	20	40	80	160
10^{-1}	0.5	L^∞ error	$7.95E - 03$	$2.94E - 03$	$1.18E - 03$	$3.64E - 04$	$1.72E - 04$
		L^1 error	$2.83E - 03$	$8.55E - 04$	$2.60E - 04$	$1.04E - 04$	$7.38E - 05$
	1.0	L^∞ error	$1.46E - 02$	$5.16E - 03$	$1.50E - 03$	$5.35E - 04$	$1.26E - 04$
		L^1 error	$5.41E - 03$	$1.60E - 03$	$4.46E - 04$	$1.11E - 04$	$6.53E - 05$
10^{-2}	0.5	L^∞ error	$3.65E - 02$	$7.29E - 03$	$2.39E - 03$	$6.99E - 04$	$4.64E - 04$
		L^1 error	$5.83E - 03$	$1.18E - 03$	$3.28E - 04$	$9.05E - 05$	$4.18E - 05$
	1.0	L^∞ error	$6.16E - 02$	$1.97E - 02$	$9.19E - 03$	$2.93E - 03$	$9.49E - 04$
		L^1 error	$7.38E - 03$	$1.66E - 03$	$6.63E - 04$	$1.95E - 04$	$5.95E - 05$
10^{-3}	0.5	L^∞ error	$2.42E - 02$	$1.10E - 02$	$4.52E - 03$	$1.52E - 03$	$1.49E - 03$
		L^1 error	$5.00E - 03$	$1.46E - 03$	$4.02E - 04$	$9.82E - 05$	$9.71E - 05$
	1.0	L^∞ error	$9.05E - 02$	$4.77E - 02$	$1.68E - 02$	$7.01E - 03$	$2.94E - 03$
		L^1 error	$3.31E - 03$	$1.70E - 03$	$2.07E - 04$	$1.13E - 04$	$6.32E - 05$

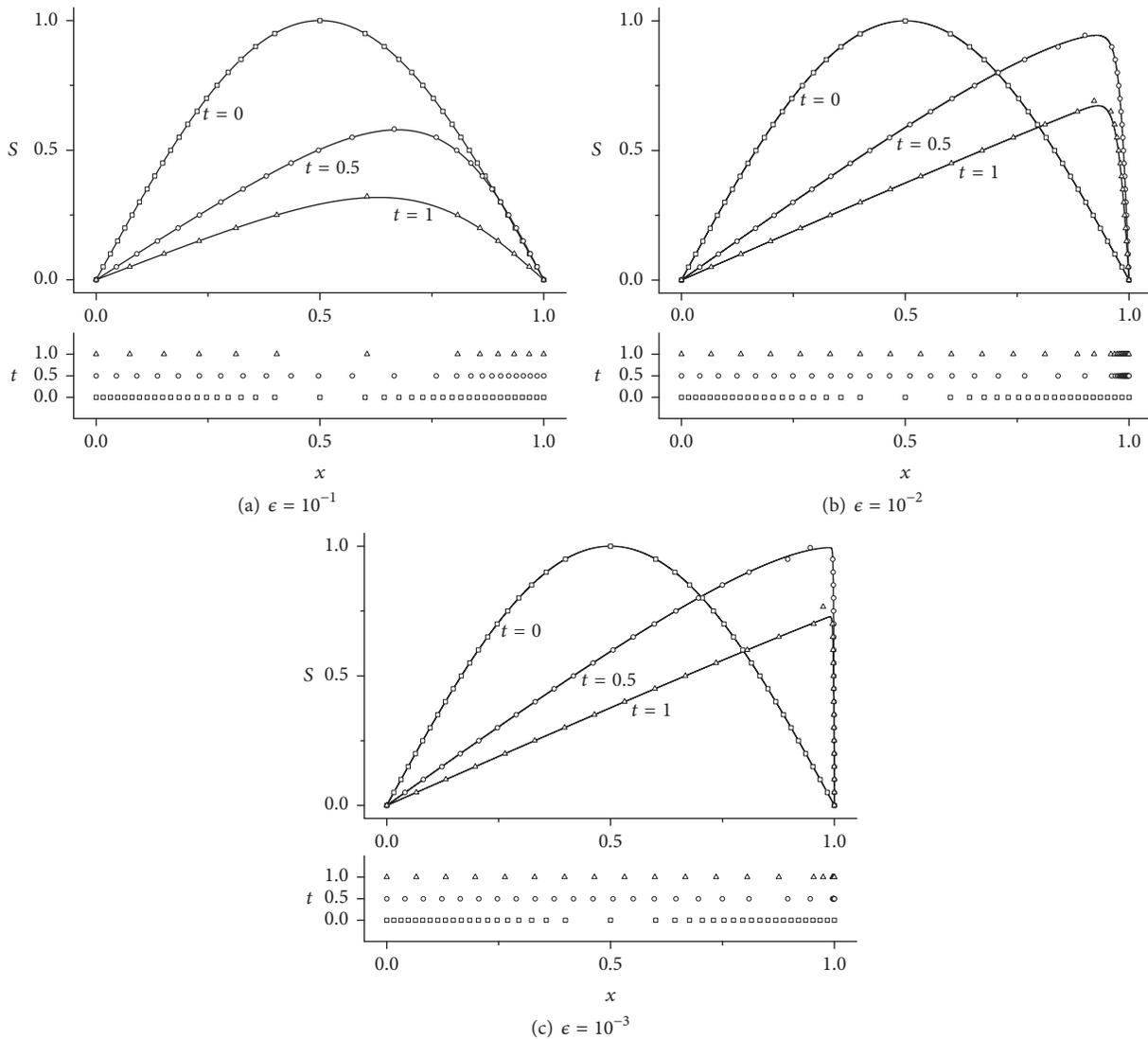


FIGURE 5: Numerical solutions obtained by the proposed method at $t = 0.5$ and $t = 1$ of Burgers' equation with different diffusive coefficients. Solid line represents exact solutions and squares, circles, and triangles in S - x figures are numerical solutions. The t - x figures below each S - x figure show the spatial distribution of the mesh points at different time.

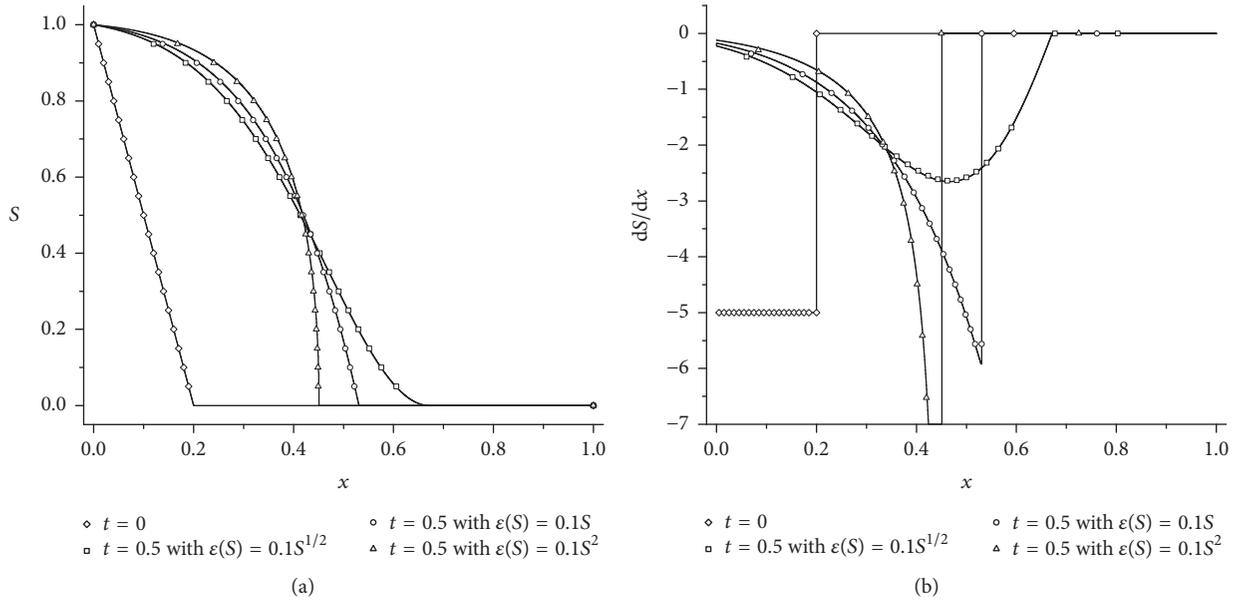


FIGURE 6: Numerical solutions of different nonlinear diffusive coefficients: (a) $S(x)$ and (b) dS/dx .

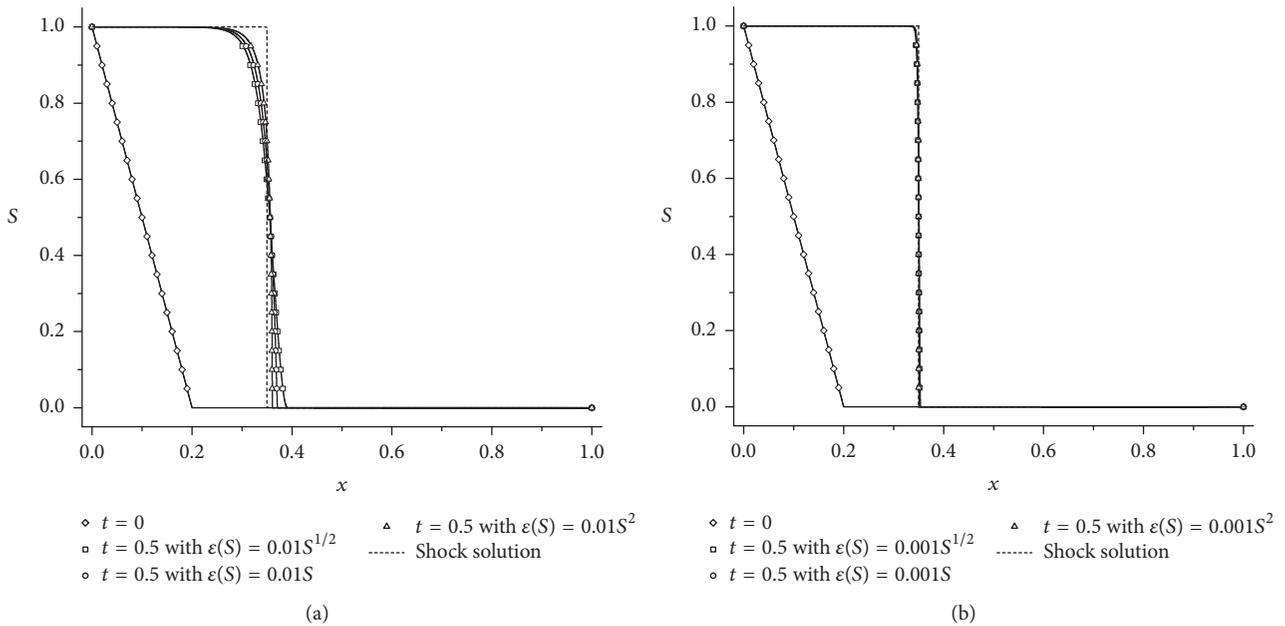


FIGURE 7: Numerical solutions of different nonlinear diffusive coefficients: (a) $\epsilon(S) = 0.01S^\alpha$ and (b) $\epsilon(S) = 0.001S^\alpha$.

smooth solution. As a critical condition, the gradient tends to be a limited value at the front with $\alpha = 1$, which is also a conclusion of Appendix B.

The solutions with diffusive coefficients set as $\epsilon(S) = 0.01S^\alpha$ and $\epsilon(S) = 0.001S^\alpha$ with $\alpha = 0.5, 1, 2$ are shown in Figure 7. With a degenerating diffusive coefficient, the nonlinear term has less impact on the solution, approaching the shock solution of the corresponding hyperbolic equation. With such

a small diffusive coefficient, not only its nonlinearity but also the whole diffusive term can be neglected. If the diffusive term is not so small in contrast to the convective term, its nonlinearity does have a significant influence on the solution.

6.4. Buckley-Leverett Equation. In this example, a two-phase flow problem in porous medium is solved. Considering an infinite horizontal layer, the region $x < 0$ is full of the wetting

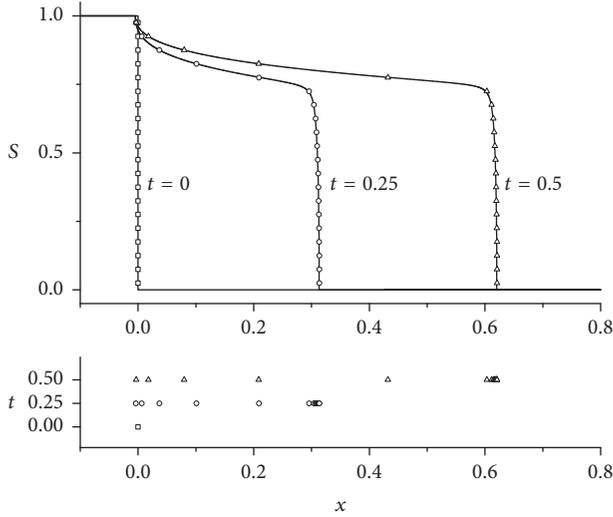


FIGURE 8: Numerical solutions of the Buckley-Leveret equation at $t = 0.25$ and $t = 0.5$. The exact solution shown in solid is obtained by FDM with a fine mesh.

phase, for example, water, and the region $x > 0$ is occupied by a nonwetting phase, for example, oil, initially. The governing equation can be obtained by Buckley-Leverett theory [31]:

$$\phi \frac{\partial S}{\partial t} + \frac{\partial}{\partial x} \left[V(t) f(S) + \frac{KK_{rn}}{\mu_n} f(S) \frac{dP_c}{dS} \frac{\partial S}{\partial x} \right] = 0, \quad (19)$$

where S is the saturation of the wetting phase, $V(t)$ is the total infiltration rate, $P_c(S)$ represents the capillary pressure, and $f(S) = \mu_o K_{rw} / (\mu_n K_{rw} + \mu_w K_{rn})$ is the flux coefficient of the wetting phase. In this example, $V(t) = 1$ and porosity and absolute permeability of the layer are nondimensionalized to be $\phi = 1$ and $K = 1$, respectively. Relative permeability of the wetting phase (w) and the nonwetting phase (n) are $K_{rw} = S^4$ and $K_{rn} = (1 + S)(1 - S)^3$, respectively, and their viscosity is $\mu_w = \mu_n = 1$. The capillary pressure is the J -function $P_c = 0.1S^{-0.5}$.

The convective term is typically nonconvex. Another problem is that the diffusive term is nonlinear, which will cause infinite saturation gradients at the front of each phase. Most methods will smear this singularity out because of their numerical diffusion, no matter how small. With the range-discrete mesh method, this singular feature can be maintained by dealing the front as a moving boundary condition.

The numerical solutions at different time with $\Delta S = 0.05$ are shown in Figure 8. It is obvious that there exists not only a steep front where saturation varies quickly but also two singular points at $S \rightarrow 0$ and $S \rightarrow 1$, where the saturation gradient tends to be infinite. As a benefit of the range-discrete mesh and the moving boundary condition, the steep front is described precisely and two singular points are maintained intactly with only 20 mesh points.

7. Conclusion

Achievements of this paper are summarized as follows:

- (i) Traditional adaptive mesh methods involve complex adaptive operations and introduce numerical dissipation by interpolation.
- (ii) We have developed a self-adaptive numerical method based on uniformly distributed range-discrete mesh for convection-diffusion equation. The governing equation is discretized by integration on the dynamic control volume of each mesh point. The goal of adaptivity and solving the equation are achieved simultaneously by solving the discretized equation, which moves the mesh points but does not change their values.
- (iii) The method was verified against the analytical solution of several numerical examples. Steep fronts were tracked accurately with only a few mesh points. The residual error is found to be independent of the magnitude of diffusion.
- (iv) With a moving boundary condition, singular points at the fronts were maintained by the proposed method. It is found that, despite its magnitude, a nonlinear diffusion coefficient $\epsilon(S) = S^\alpha$ ($\alpha \geq 1$) can produce singularities where the gradient of the solution can be discontinuous or infinite.

In all, the proposed method has shown advantages in solving convection-diffusion equations with high efficiency. The method can serve as a fast numerical tool for a wide range of front propagation problems, such as underground water flow, chemical floods of oil fields, and air pollution problems. On the other hand, it is obvious that, for higher dimensions, the elements in the range-discrete mesh become contour lines or contour planes and the solution can be achieved by tracking their movements. This work is in our next research scope.

Appendix

A. Integration of the Transient Term in (3)

In (3), we have the integration of the transient term over the dynamic control volume $[x_1(t), x_2(t)]$. To understand this term, first, we know

$$\begin{aligned} \frac{\partial}{\partial t} \int_{x_1(t)}^{x_2(t)} S dx &= \int_{x_1(t)}^{x_2(t)} \frac{\partial S}{\partial t} dx + S(x_2(t), t) \frac{dx_2(t)}{dt} \\ &\quad - S(x_1(t), t) \frac{dx_1(t)}{dt} \\ &= \int_{x_1(t)}^{x_2(t)} \frac{\partial S}{\partial t} dx + S_2 \frac{dx_2(t)}{dt} \\ &\quad - S_1 \frac{dx_1(t)}{dt}. \end{aligned} \quad (A.1)$$

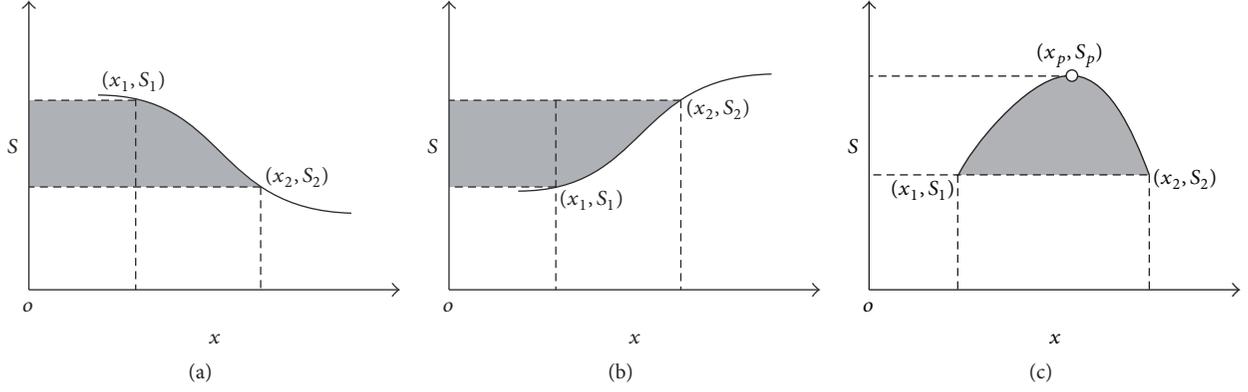


FIGURE 9: Geometric description of the integration.

So, integration of the transient term in (3) is

$$\begin{aligned}
 & \int_{t_0}^{t_1} \int_{x_1(t)}^{x_2(t)} \frac{\partial S}{\partial t} dx dt \\
 &= \int_{t_0}^{t_1} \frac{\partial}{\partial t} \int_{x_1(t)}^{x_2(t)} S dx dt - \int_{t_0}^{t_1} S_2 \frac{dx_2(t)}{dt} dt \\
 & \quad + \int_{t_0}^{t_1} S_1 \frac{dx_1(t)}{dt} dt \\
 &= \int_{x_1(t_1)}^{x_2(t_1)} S dx - \int_{x_1(t_0)}^{x_2(t_0)} S dx \\
 & \quad - S_2 [x_2(t_1) - x_2(t_0)] + S_1 [x_1(t_1) - x_1(t_0)] \\
 &= \left[\int_{x_1(t)}^{x_2(t)} S dx - S_2 x_2(t) + S_1 x_1(t) \right]^{t_1} \\
 & \quad - \left[\int_{x_1(t)}^{x_2(t)} S dx - S_2 x_2(t) + S_1 x_1(t) \right]^{t_0}.
 \end{aligned} \tag{A.2}$$

The items inside the brackets in (A.2) are

$$\begin{aligned}
 & \int_{x_1(t)}^{x_2(t)} S dx - S_2 x_2(t) + S_1 x_1(t) \\
 &= \int_{x_1(t)}^{x_2(t)} S dx - \int_0^{x_2(t)} S_2 dx + \int_0^{x_1(t)} S_1 dx \\
 &= \int_{x_1(t)}^{x_2(t)} S dx - \left(\int_0^{x_1(t)} S_2 dx + \int_{x_1(t)}^{x_2(t)} S_2 dx \right) \\
 & \quad + \int_0^{x_1(t)} S_1 dx \\
 &= \int_0^{x_1(t)} (S_1 - S_2) dx + \int_{x_1(t)}^{x_2(t)} (S(x) - S_2) dx.
 \end{aligned} \tag{A.3}$$

which can be understood according to Figure 9. Consider the following three situations:

- (1) $S = S(x, t)$ is monotone decreasing in $[x_1(t), x_2(t)]$, as shown in Figure 9(a). Then (A.3) is the shaded area shown in Figure 9(a). So, we have

$$\begin{aligned}
 & \int_0^{x_1(t)} (S_1 - S_2) dx + \int_{x_1(t)}^{x_2(t)} (S(x) - S_2) dx \\
 &= - \int_{S_1}^{S_2} x(S, t) dS.
 \end{aligned} \tag{A.4}$$

- (2) $S = S(x, t)$ is monotone increasing in $[x_1(t), x_2(t)]$, as shown in Figure 9(b). Then (A.3) is the shaded negative area shown in Figure 9(b). So, we have

$$\begin{aligned}
 & \int_0^{x_1(t)} (S_1 - S_2) dx + \int_{x_1(t)}^{x_2(t)} (S(x) - S_2) dx \\
 &= - \int_0^{x_1(t)} (S_2 - S_1) dx - \int_{x_1(t)}^{x_2(t)} (S_2 - S(x)) dx \\
 &= - \int_{S_1}^{S_2} x(S, t) dS.
 \end{aligned} \tag{A.5}$$

- (3) $S = S(x, t)$ is nonmonotone in $[x_1(t), x_2(t)]$, as shown in Figure 9(c). As we have $S_1 = S_2$, so

$$\begin{aligned}
 & \int_0^{x_1(t)} (S_1 - S_2) dx + \int_{x_1(t)}^{x_2(t)} (S(x) - S_2) dx \\
 &= \int_{x_1(t)}^{x_2(t)} (S(x) - S_2) dx
 \end{aligned} \tag{A.6}$$

is the shaded area or negative area according to the sign of $S(x) - S_2$.

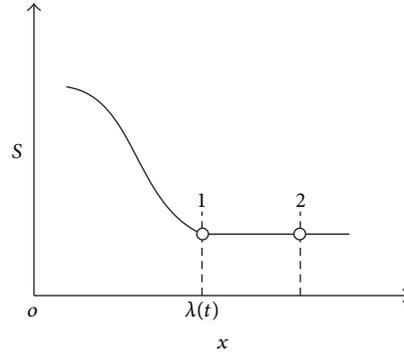


FIGURE 10: The description of a moving boundary condition.

In short, the integration of the transient item is

$$\int_{t_0}^{t_1} \int_{x_1(t)}^{x_2(t)} \frac{\partial S}{\partial t} dx dt = \begin{cases} - \int_{S_1}^{S_2} x(S, t_1) dS + \int_{S_1}^{S_2} x(S, t_0) dS, & \text{for monotone } S(x, t) \\ \int_{x_1(t_1)}^{x_2(t_1)} (S(x, t_1) - S_2) dx - \int_{x_1(t_0)}^{x_2(t_0)} (S(x, t_0) - S_2) dx, & \text{for non-monotone } S(x, t). \end{cases} \quad (\text{A.7})$$

B. The Moving Boundary Condition

Assume that there is a piece of constant solution at time t , a moving boundary condition is formed at either end of the constant piece. In this section, without loss of generality, the constant piece is $x > \lambda(t)$, which means that $S(x \geq \lambda(t), t) = S_C$, and $x = \lambda(t)$ is the so-called moving boundary.

A dynamic control volume $[x_1, x_2]$, where $x_1(t) = \lambda(t)$ and x_2 is an arbitrary point in the constant piece, is chosen as shown in Figure 10. For $S(x \geq \lambda(t), t) = S_C$, we have

$$S(x, t) = S_C, \quad x \in [x_1(t), x_2(t)] \quad (\text{B.1})$$

and also $S_1 = S_2 = S_C$. For this control volume, the integration equation becomes

$$0 = \int_t^{t+\Delta t} \left[\epsilon(S_2) \left(\frac{\partial S}{\partial x} \right) \Big|_{x_2(t)} - \epsilon(S_1) \left(\frac{\partial S}{\partial x} \right) \Big|_{x_1(t)} \right] dt. \quad (\text{B.2})$$

For $x = x_2(t)$, we have $S_x(x_2(t)) = 0$; thus (B.2) becomes

$$\left(\epsilon(S) \frac{\partial S}{\partial x} \right) \Big|_{\lambda(t)} = 0. \quad (\text{B.3})$$

Therefore, (B.3) and $S(\lambda(t)) = S_C$ both hold for the front point $x = \lambda(t)$ and will serve as the boundary condition for range-discrete method.

However, $\lambda(t)$ is not a given condition but rather a result of the solution. It can be calculated by an interpolation as one of the two following cases:

- (1) If $\epsilon(S) \rightarrow \epsilon_0 \neq 0$ when $x \rightarrow \lambda^-$ or $S \rightarrow S_C$: from (B.3), we can conclude that $S_x(\lambda(t)) = 0$. Thus, using the Taylor series, the solution when $x \rightarrow \lambda^-$ is

$$S(x) = S(\lambda) + S_x(\lambda)(\lambda - x) + \frac{1}{2} S_{xx}(\lambda)(\lambda - x)^2 + \frac{1}{6} S_{xxx}(\lambda)(\lambda - x)^3 + \dots \quad (\text{B.4})$$

Dropping the items of order more than 2, we have an approximation as

$$S(x) = S_C + a_0(\lambda - x)^2. \quad (\text{B.5})$$

- (2) If $\epsilon(S) \rightarrow 0$ when $x \rightarrow \lambda^-$ or $S \rightarrow S_C$: an approximation behavior of the diffusive coefficient can be obtained as $\epsilon(S) \rightarrow C_\epsilon(S - S_C)^\alpha$ when $S \rightarrow S_C$. And we assume that the solution when $x \rightarrow \lambda^-$ can be approximated by $S(x) = S_C + a_0(\lambda - x)^n$. n can be calculated using the limitation that the quality flux cannot be infinite, that is, $Q(x)/S(x) < \infty$ when $x \rightarrow \lambda^-$, where

$$Q(x) = f(S(x)) - \epsilon(S(x)) \frac{\partial S(x)}{\partial x}. \quad (\text{B.6})$$

Thus, $\epsilon(S)S_x$ should approach zero at the same speed as $S(x) - S_C$ when $x \rightarrow \lambda^-$. So, we have $S_x(x) = -n(\lambda(t) - x)^{n-1}$, $\epsilon(S) = C_\epsilon a_0^\alpha (\lambda - x)^{\alpha n}$ and finally

$$\epsilon(S)S_x = -nC_\epsilon a_0^\alpha (\lambda - x)^{\alpha n + n - 1}. \quad (\text{B.7})$$

Thus, $\alpha n + n - 1 = n$, and so we have $\alpha = 1/n$. This means that, when $x \rightarrow \lambda^-$,

$$S(x) = S_C + a_0 (\lambda - x)^{1/\alpha}. \quad (\text{B.8})$$

It can be seen from (B.5) that when $\alpha > 1$, the gradient of the solution tends to be infinite when $x \rightarrow \lambda(t)$. The situation when $\alpha = 2$ graphed in Figure 7 shows us an example.

With the information of the two neighboring mesh points known, $\lambda(t)$ can be obtained approximately using (B.5) or (B.8).

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (nos. 11572315 and 11202205) and CNPC-CAS Science and Technology Cooperation Project (no. 2015A-4812).

References

- [1] B. A. Finlayson, *Numerical Methods for Problems with Moving Fronts*, Ravenna Park Publishing, Incorporated, Seattle, WA, USA, 1992.
- [2] L. Debnath, *Nonlinear Partial Differential Equations for Scientists and Engineers*, Birkhäuser Boston, Inc, Boston, MA, USA, 3rd edition, 2011.
- [3] R. E. Ewing and H. Wang, "A summary of numerical methods for time-dependent advection-dominated partial differential equations," *Journal of Computational and Applied Mathematics*, vol. 128, no. 1-2, pp. 423–445, 2001.
- [4] R. J. LeVeque, *Finite Volume Methods for Hyperbolic Problems*, Cambridge Texts in Applied Mathematics, Cambridge University Press, Cambridge, UK, 2002.
- [5] A. Kurganov and E. Tadmor, "New high-resolution central schemes for nonlinear conservation laws and convection-diffusion equations," *Journal of Computational Physics*, vol. 160, no. 1, pp. 241–282, 2000.
- [6] T. Nazir, M. Abbas, A. I. Ismail, A. A. Majid, and A. Rashid, "The numerical solution of advection-diffusion problems using new cubic trigonometric B-splines approach," *Applied Mathematical Modelling. Simulation and Computation for Engineering and Environmental Systems*, vol. 40, no. 7-8, pp. 4586–4611, 2016.
- [7] P. K. Sweby, "High resolution schemes using flux limiters for hyperbolic conservation laws," *SIAM Journal on Numerical Analysis*, vol. 21, no. 5, pp. 995–1011, 1984.
- [8] K. Alhumaizi, "Comparison of finite difference methods for the numerical simulation of reacting flow," *Computers and Chemical Engineering*, vol. 28, no. 9, pp. 1759–1769, 2004.
- [9] C.-W. Shu, "High order weighted essentially nonoscillatory schemes for convection dominated problems," *SIAM Review*, vol. 51, no. 1, pp. 82–126, 2009.
- [10] H. Ding and Y. Zhang, "A new difference scheme with high accuracy and absolute stability for solving convection-diffusion equations," *Journal of Computational and Applied Mathematics*, vol. 230, no. 2, pp. 600–606, 2009.
- [11] H. K. Dahle, M. S. Espedal, and R. E. Ewing, "Characteristic Petrov-Galerkin subdomain methods for convection-diffusion problems," *Computer Methods in Applied Mechanics and Engineering*, vol. 64, pp. 77–86, 1988.
- [12] T. F. Russell and M. A. Celia, "An overview of research on Eulerian-Lagrangian localized adjoint methods (ELLAM)," *Advances in Water Resources*, vol. 25, no. 8-12, pp. 1215–1231, 2002.
- [13] Z. Chen, "Characteristic mixed discontinuous finite element methods for advection-dominated diffusion problems," *Computer Methods in Applied Mechanics and Engineering*, vol. 191, no. 23-24, pp. 2509–2538, 2002.
- [14] M. Seaïd, "An Eulerian-Lagrangian method for coupled parabolic-hyperbolic equations," *Applied Numerical Mathematics*, vol. 59, no. 3-4, pp. 754–768, 2009.
- [15] J. Caldwell, P. Wanless, and A. E. Cook, "Solution of Burgers' equation for large Reynolds number using finite elements with moving nodes," *Applied Mathematical Modelling*, vol. 11, no. 3, pp. 211–214, 1987.
- [16] J. Behrens, A. Iske, and M. Käser, "Adaptive Meshfree Method of Backward Characteristics for Nonlinear Transport Equations," in *Meshfree Methods for Partial Differential Equations SE - 2*, M. Griebel and M. Schweitzer, Eds., vol. 26 of *Lecture Notes in Computational Science and Engineering*, pp. 21–36, Springer Berlin Heidelberg, Berlin, Germany, 2003.
- [17] T. Plewa, T. Linde, and V. G. Weirs, "Adaptive mesh refinement-theory and applications," in *Proceedings of the Chicago Workshop on Adaptive Mesh Refinement Methods*, vol. 41, Chicago, Ill, USA, 2003.
- [18] X.-H. Wang, M. Quintard, and G. Darche, "Adaptive mesh refinement for one-dimensional three-phase flow with phase change in porous media," *Numerical Heat Transfer, Part B: Fundamentals*, vol. 50, no. 3, pp. 231–268, 2006.
- [19] W. Huang and R. D. Russell, *Adaptive Moving Mesh Methods*, Springer Science & Business Media, New York, NY, USA, 2011.
- [20] H. Dong, Z. Qiao, S. Sun, and T. Tang, "Adaptive moving grid methods for two-phase flow in porous media," *Journal of Computational and Applied Mathematics*, vol. 265, pp. 139–150, 2014.
- [21] S. Reboux, B. Schrader, and I. F. Sbalzarini, "A self-organizing Lagrangian particle method for adaptive-resolution advection-diffusion simulations," *Journal of Computational Physics*, vol. 231, no. 9, pp. 3623–3646, 2012.
- [22] C.-C. Chueh, N. Djilali, and W. Bangerth, "An h -adaptive operator splitting method for two-phase flow in 3D heterogeneous porous media," *SIAM Journal on Scientific Computing*, vol. 35, no. 1, pp. B149–B175, 2013.
- [23] O. Angelini, K. Brenner, and D. Hilhorst, "A finite volume method on general meshes for a degenerate parabolic convection-reaction-diffusion equation," *Numerische Mathematik*, vol. 123, no. 2, pp. 219–257, 2013.
- [24] Z. Cao, Z. Liu, X. Wang, A. Shi, H. Luo, and B. Noettinger, "A dissipation-free numerical method to solve one-dimensional hyperbolic flow equations," *International Journal for Numerical Methods in Fluids*.

- [25] F. J. Fayers and J. W. Sheldon, "The effect of capillary pressure and gravity on two-phase fluid flow in a porous medium, petroleum transactions," *AIME*, vol. 216, pp. 147–155, 1959.
- [26] F. Hussain, Y. Cinar, and P. Bedrikovetsky, "A semi-analytical model for two phase immiscible flow in porous media honouring capillary pressure," *Transport in Porous Media*, vol. 92, no. 1, pp. 187–212, 2012.
- [27] I. R. Goumiri, J. H. Prévost, and M. Preisig, "The effect of capillary pressure on the saturation equation of two-phase flow in porous media," *International Journal for Numerical and Analytical Methods in Geomechanics*, vol. 36, no. 3, pp. 352–361, 2012.
- [28] J. Mackenzie, "Uniform convergence analysis of an upwind finite-difference approximation of a convection-diffusion boundary value problem on an adaptive grid," *IMA Journal of Numerical Analysis*, vol. 19, no. 2, pp. 233–249, 1999.
- [29] N. Kopteva and M. Stynes, "A robust adaptive method for a quasi-linear one-dimensional convection-diffusion problem," *SIAM Journal on Numerical Analysis*, vol. 39, no. 4, pp. 1446–1467, 2001.
- [30] J. Caldwell and P. Smith, "Solution of Burgers' equation with a large Reynolds number," *Applied Mathematical Modelling*, vol. 6, no. 5, pp. 381–385, 1982.
- [31] S. E. Buckley and M. C. Leverett, "Mechanism of fluid displacement in sands," *Transactions of the AIME*, vol. 146, no. 1, pp. 107–116, 2013.



Hindawi

Submit your manuscripts at
<https://www.hindawi.com>

