


Article

# High-Order Finite Difference Hermite Weighted Essentially Nonoscillatory Method for Convection–Diffusion Equations

Yabo Wang<sup>1</sup> and Hongxia Liu<sup>2,\*</sup> <sup>1</sup> School of Mathematics, Taiyuan University of Technology, Taiyuan 030600, China; wyb2044388699@163.com<sup>2</sup> School of Artificial Intelligence, Taiyuan University of Technology, Taiyuan 030600, China

\* Correspondence: liuhongxia@tyut.edu.cn

**Abstract:** A kind of finite difference Hermite WENO (HWENO) method is presented in this paper to deal with convection-dominated convection-diffusion equations in uniform grids. The benefit of the HWENO method is its compactness, allowing great accuracy to be attained in the solution's smooth regions and maintaining the essential nonoscillation in the solution's discontinuities. We discretize the convection term using the HWENO method and the diffusion term using the Hermite central interpolation schemes. However, it is difficult to deal with mixed derivative terms when solving two-dimensional problems using the HWENO method mentioned. To address this problem, we also employ the Hermite interpolation approach, which can keep the compactness. Lastly, we apply this method to two-dimensional Navier-Stokes problems that are incompressible. The efficiency and stability of the presented method are illustrated through numerous numerical experiments.

**Keywords:** high-order; accuracy; convection-diffusion equations; HWENO method; finite difference method



Academic Editors: Ruo Li and Gianluigi Rozza

Received: 25 October 2024  
Revised: 12 December 2024  
Accepted: 17 December 2024  
Published: 3 January 2025

**Citation:** Wang, Y.; Liu, H. High-Order Finite Difference Hermite Weighted Essentially Nonoscillatory Method for Convection–Diffusion Equations. *Math. Comput. Appl.* **2025**, *30*, 3. <https://doi.org/10.3390/mca30010003>

**Copyright:** © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Convection-dominated diffusion problems of the form

$$\mathbf{u}_t + \nabla \cdot F(\mathbf{u}) = \nabla \cdot (\epsilon \nabla \mathbf{u}) \quad (1)$$

are addressed in this paper using a type of HWENO method within the framework of finite difference schemes, where  $\epsilon$  is the diffusion coefficient. We take into account one- and two-dimensional cases here. Convection-diffusion problems are significant in many fields and have a wide range of applications in real-world issues [1], including reservoir simulation, the dispersion and diffusion of water in rivers and the ocean, environmental research, and even energy development. And, these realistic models can construct corresponding convection-diffusion equations. But, unfortunately, the mathematical models for most of these problems are complex nonlinear partial differential equations, and finding analytical solutions to these equations is difficult. Therefore, it is crucial to conduct research on partial differential equations' numerical solutions [2].

Such physical problems have been solved by researchers in recent decades using a variety of numerical approaches [3–8], and a rapid development has taken place with a variety of systematic numerical methods. A key problem of the numerical method is preventing nonphysical oscillations when the convective term is dominant. Due to the discontinuous characteristics of equations and the complexity of their corresponding mathematical models, it is urgent that we establish simple and effective numerical schemes. In many references, the numerical solutions of these equations are solved using the finite

difference method because of its versatility, simplicity, and diversity. In the traditional finite difference method, low-order schemes cannot approach the exact solution well, while high-order schemes often imply the use of more stencil points, which leads to lots of numerical difficulties in the calculations [9]. Therefore, in this paper, we construct a high-order and compact finite difference numerical method to solve the convection-dominated convection-diffusion equation. Since convection-dominated convection-diffusion equations have hyperbolic properties, we can solve them using various numerical methods for solving hyperbolic conservation laws without making any major adjustments. Among all of these numerical methods, we would like to mention the essentially nonoscillating (ENO), weighted ENO (WENO), and Hermite WENO (HWENO) schemes.

The ENO scheme was first presented within a one-dimensional finite volume framework by Harten et al. [10] in 1987 based on a total variation diminishing (TVD) scheme [11]. Subsequently, it was expanded to multidimensional cases [12]. Shu et al.'s [13] finite difference ENO numerical techniques for one-dimensional problem resolution were developed in 1988, and they were expanded to the multidimensional case in [14]. The fundamental idea of the ENO method is to attain high-order accuracy by using the smoothest candidate stencil, which results in the waste of stencils but also allows the schemes to maintain nonoscillating qualities at discontinuities. The first WENO method was introduced by Liu et al. [15] in 1994 within the one-dimensional finite volume framework, utilizing all the information of the stencils to circumvent the deficiency of the ENO schemes. A general structure for the reconstruction process was provided by Shu and Jiang [16] in 1996 when they developed the WENO method for a multidimensional finite difference framework. The core principle of the WENO method is to use a combination of all candidate stencils and assign a nonlinear weight to each one in accordance with how smooth the stencils are. Since then, there has been rapid development in WENO schemes in higher-dimensional cases, with more types of grids, higher-order accuracy, and more solution fields (see, e.g., [17–20]). WENO schemes exhibit superior resilience, flux smoothness, and steady-state convergence compared to ENO schemes.

However, WENO schemes with higher accuracy require wider stencils for reconstruction. In order to solve this problem, Qiu and Shu [21,22] established the HWENO method, a novel method within the finite-volume framework constructed from the principles of the WENO method and Hermite interpolation, with the goal of creating a more compact form. The HWENO method in the finite difference framework was suggested in 2015 by Liu and Qiu [23]. In one-dimensional problems, the methods can attain fifth-order accuracy, but they can only attain fourth-order accuracy in two-dimensional problems due to the introduction of derivative equations. Later, Zhao et al. [24] modified the method to achieve fifth-order accuracy even in the two-dimensional case. However, the cost is that the derivative equation does not use the HWENO scheme. The crucial distinction with regard to WENO and HWENO methods is that the former just requires the original function value, and the latter also needs its first derivative value in addition to the original function value in order to guarantee that the reconstruction process can achieve higher accuracy. Because of this, HWENO schemes can achieve the same accuracy level with smaller stencils, comparable to those of WENO schemes. This means that the three-point HWENO method can attain the accuracy of the five-point WENO method because it has more conditions that can be used. As the derivative equation does not use the HWENO scheme, sometimes the robustness and stability of the HWENO method are not as good as those of the WENO method because the derivatives may become large near discontinuities. In order to avoid this problem and enhance the robustness of the HWENO method, Zhao et al. [25] proposed an alternative method to effectively overcome oscillations. Later, a modified HWENO scheme with artificial linear weights [26], the positivity-preserving HWENO scheme [27],

and the multiresolution HWENO scheme [28] were also developed to solve hyperbolic conservation laws. In addition, fifth- or sixth-order accuracy is obtained through more compact stencils in the HWENO schemes [26,29,30], providing higher resolution and smaller numerical errors. For more HWENO schemes for solving hyperbolic conservation laws, see [28,31].

Due to the drawbacks of true two-dimensional reconstruction on a structured grid [32] and the deficiency of robustness and nonstability resulting from the lack of the HWENO scheme for the derivative equation, this paper adopts a dimension-by-dimension idea to generalize the HWENO method for solving convective dominance problems. For implementing the solution to the convection-dominated problems within the finite difference framework, we present an HWENO method, which uses the Hermite interpolation method to approximate the diffusion terms and the mixed derivative terms. Because the discretization of mixed derivative terms only has fourth-order accuracy under specific conditions, the proposed method can achieve fifth-order accuracy in the one-dimensional situation, but it only has fourth-order accuracy in the two-dimensional case.

This paper is structured as follows: The finite-difference HWENO reconstruction process of convection-diffusion problems in the one-dimensional case is covered in Section 2. A workable solution for mixed derivatives is addressed in Section 3, where we generalize the schemes to two-dimensional problems. In Section 4, numerical examples, including Navier-Stokes equations, are tested, and the results are shown to confirm the viability of the method. Conclusions are discussed in Section 5.

## 2. Construction of One-Dimensional HWENO Method

The implementation process of the fifth-order finite difference HWENO method for solving convection-diffusion equations is covered in this section. First, we examine the one-dimensional scalar situation

$$\begin{cases} \frac{\partial u}{\partial t} + \frac{\partial f(u)}{\partial x} = \varepsilon \frac{\partial^2 u}{\partial x^2}, \\ u(x, 0) = u_0(x), \end{cases} \tag{2}$$

where  $t$  is the time variable and  $x$  is the space variable in the one-dimensional case. Furthermore, the conserved variable  $u$  is related to both time and space, while  $f(u)$  represents the flux function, and  $\varepsilon$  is the diffusion coefficient. In the finite-difference framework, uniform grid points  $\{x_i\}$  divide the spatial domain, with  $x_i$  denoting the center of cell  $I_i$ , where  $x_i = x_{i-\frac{1}{2}} + \frac{\Delta x}{2}$  and  $I_i = [x_i - \frac{\Delta x}{2}, x_i + \frac{\Delta x}{2}]$  ( $i = 1, \dots, N$ ). The notation  $\Delta x$  is the cell size, which equals  $x_{i+1} - x_i$ . To create the HWENO method, we calculate the derivative equation of (2) relating to the space variable  $x$ , and  $v(x)$  is the presentation for the spatial first derivative of  $u(x, t)$ . Then, the following equations are obtained by

$$\begin{cases} \frac{\partial u}{\partial t} + \frac{\partial f(u)}{\partial x} = \varepsilon \frac{\partial^2 u}{\partial x^2}, & u(x, 0) = u_0(x), \\ \frac{\partial v}{\partial t} + \frac{\partial h(u, v)}{\partial x} = \varepsilon \frac{\partial^2 v}{\partial x^2}, & v(x, 0) = v_0(x), \end{cases} \tag{3}$$

where  $v_0(x) = (u_x)_0$  and  $h(u, v) = f'(u)v$ . The semi-discrete form of (3) is

$$\begin{cases} \frac{\partial u_i}{\partial t} = -\frac{1}{\Delta x}(\hat{f}_{i+\frac{1}{2}} - \hat{f}_{i-\frac{1}{2}}) + \varepsilon \frac{\partial^2 u}{\partial x^2} \Big|_{x_i}, \\ \frac{\partial v_i}{\partial t} = -\frac{1}{\Delta x}(\hat{h}_{i+\frac{1}{2}} - \hat{h}_{i-\frac{1}{2}}) + \varepsilon \frac{\partial^2 v}{\partial x^2} \Big|_{x_i}, \end{cases} \tag{4}$$

where  $u_i = u(x_i, t)$ ,  $v_i = v(x_i, t)$ .  $\hat{f}_{i+\frac{1}{2}}$  and  $\hat{h}_{i+\frac{1}{2}}$  are the numerical fluxes of the original equation and the derivative equation, respectively. Here, the numerical fluxes  $\hat{f}_{i+\frac{1}{2}}$  and  $\hat{h}_{i+\frac{1}{2}}$  are monotone numerical fluxes that are Lipschitz continuous and satisfy the consistency

condition. We utilize fifth-order approximations for these numerical fluxes, specifically  $\hat{f}_{i+\frac{1}{2}} = \phi(x_{i+\frac{1}{2}})$  and  $\hat{h}_{i+\frac{1}{2}} = \psi(x_{i+\frac{1}{2}})$ . The definitions are detailed in references [16,23,33], namely,

$$f(u) = \frac{1}{\Delta x} \int_{x-\frac{\Delta x}{2}}^{x+\frac{\Delta x}{2}} \varphi(x) dx, \quad h(u, v) = \frac{1}{\Delta x} \int_{x-\frac{\Delta x}{2}}^{x+\frac{\Delta x}{2}} \psi(x) dx. \tag{5}$$

To maintain the stability of the finite difference HWENO method, the flux function in both the derivative equation and the original equation must be expressed as the sum of its positive and negative parts. These two parts satisfy  $\frac{df^+(u)}{du} \geq 0$ ,  $\frac{df^-(u)}{du} \leq 0$  and  $\frac{\partial h^+(u, v)}{\partial v} \geq 0$ ,  $\frac{\partial h^-(u, v)}{\partial v} \leq 0$ , respectively. The fluxes  $f^\pm(u)$  and  $h^\pm(u, v)$  are associated with the numerical fluxes  $\hat{f}_{i+\frac{1}{2}}^\pm$  and  $\hat{h}_{i+\frac{1}{2}}^\pm$ , respectively. As a result, two components can be derived from the numerical fluxes  $\hat{f}_{i+\frac{1}{2}} = \hat{f}_{+i+\frac{1}{2}}^+ + \hat{f}_{i+\frac{1}{2}}^-$  and  $\hat{h}_{i+\frac{1}{2}} = \hat{h}_{+i+\frac{1}{2}}^+ + \hat{h}_{i+\frac{1}{2}}^-$ . This clarifies the relationship between the fluxes and the numerical components derived from them.

In this paper, we present a construction algorithm for the three-point HWENO method, with the possibility of obtaining higher-order HWENO reconstructions using a similar approach. We then introduce an algorithm to construct the numerical fluxes  $\hat{f}_{i+\frac{1}{2}}^-$  and  $\hat{h}_{i+\frac{1}{2}}^-$  on the left-biased stencil. An analogous procedure is employed to derive the constructs  $\hat{f}_{i+\frac{1}{2}}^+$  and  $\hat{h}_{i+\frac{1}{2}}^+$  on the right-biased stencil. Further details can be found in the following two sections.

Step I. First, we pay attention to the reconstruction of  $\hat{f}_{i+\frac{1}{2}}^-$ . The steps of our algorithm are as follows.

1.1 Here, we utilize the following Lax-Friedrichs flux splitting to compute the point values  $\{f^\pm(u_i), h^\pm(u_i, v_i)\}$  at all points based on the given nodal values  $\{u_i, v_i\}$ . The equations for these calculations are defined as follows:

$$\begin{aligned} f^\pm(u_i) &= \frac{1}{2}(f(u_i) \pm \alpha u_i), \\ h^\pm(u_i, v_i) &= \frac{1}{2}(h(u_i, v_i) \pm \alpha v_i), \end{aligned} \tag{6}$$

where  $\alpha = \max_u |f'(u)|$ . We abbreviate  $f^\pm(u_i)$  and  $h^\pm(u_i, v_i)$  to  $f_i^\pm$  and  $h_i^\pm$ , respectively, to simplify notation.

1.2 Given three small stencils  $S_0 = \{x_{i-1}, x_i\}$ ,  $S_1 = \{x_i, x_{i+1}\}$ ,  $S_2 = \{x_{i-1}, x_i, x_{i+1}\}$  and a large stencil  $S = \{S_0, S_1, S_2\}$ , respectively, we first generate quadratic polynomials  $p_j(x)$  on  $S_j$  using Hermite interpolation. They fulfill the requirements listed below.

$$\begin{aligned} \frac{1}{\Delta x} \int_{I_k} p_0(x) dx &= f_k^+, \quad k = i-1, i, \\ \frac{1}{\Delta x} \int_{I_k} p'_0(x) dx &= h_k^+, \quad k = i-1, \\ \frac{1}{\Delta x} \int_{I_k} p_1(x) dx &= f_k^+, \quad k = i, i+1, \\ \frac{1}{\Delta x} \int_{I_k} p'_1(x) dx &= h_k^+, \quad k = i+1, \\ \frac{1}{\Delta x} \int_{I_k} p_2(x) dx &= f_k^+, \quad k = i-1, i, i+1. \end{aligned}$$

Similarly, the same method is applied to the large stencil  $S$  to create a quartic polynomial  $q(x)$  that satisfies

$$\begin{aligned} \frac{1}{\Delta x} \int_{I_k} q(x) dx &= f_k^+, \quad k = i - 1, i, i + 1, \\ \frac{1}{\Delta x} \int_{I_k} q'(x) dx &= h_k^+, \quad k = i - 1, i + 1. \end{aligned}$$

In actuality, as we are reconstructing the numerical flux at the half point  $x_{i+\frac{1}{2}}$ , we only need to calculate the function values of the polynomials at  $x_{i+\frac{1}{2}}$ . This eliminates the need to derive the complete expressions for these polynomials. Their final formulas are as follows:

$$\begin{aligned} p_0(x_{i+\frac{1}{2}}) &= \frac{1}{6}(-7f_{i-1}^+ + 13f_i^+ - 4\Delta x h_{i-1}^+), \\ p_1(x_{i+\frac{1}{2}}) &= \frac{1}{6}(f_i^+ + 5f_{i+1}^+ - 2\Delta x h_{i+1}^+), \\ p_2(x_{i+\frac{1}{2}}) &= \frac{1}{6}(-f_{i-1}^+ + 5f_i^+ + 2f_{i+1}^+), \\ q(x_{i+\frac{1}{2}}) &= \frac{1}{120}(-23f_{i-1}^+ + 76f_i^+ + 67f_{i+1}^+ - 9\Delta x h_{i-1}^+ - 21\Delta x h_{i+1}^+). \end{aligned} \tag{7}$$

1.3 We represent  $q(x_{i+\frac{1}{2}})$  as a convex combination of  $p_0(x_{i+\frac{1}{2}})$ ,  $p_1(x_{i+\frac{1}{2}})$ , and  $p_2(x_{i+\frac{1}{2}})$ . These combination coefficients, denoted as  $\gamma_0$ ,  $\gamma_1$ , and  $\gamma_2$ , are also known as linear weights, and they must satisfy the following equation:

$$q(x_{i+\frac{1}{2}}) = \sum_{j=0}^2 \gamma_j p_j(x_{i+\frac{1}{2}}),$$

with  $\sum_{j=0}^2 \gamma_j = 1$ . This results in

$$\gamma_0 = \frac{9}{80}, \gamma_1 = \frac{21}{40}, \gamma_2 = \frac{29}{80}. \tag{8}$$

1.4 Next, the smoothness indicators  $\beta_j$  are determined, which are essential for non-linear weights. For the target cell  $I_i$ , these indicators measure the smoothness of the corresponding polynomials  $p_j(x)$ . The smoother the polynomial associated with a particular cell, the smaller the smoothness indicator for that cell. Here, we employ the smoothness indicator defined in reference [16,19]:

$$\beta_j = \sum_{l=1}^r \int_{I_i} \Delta x^{2l-1} \left( \frac{\partial^l}{\partial x^l} p_j(x) \right)^2 dx. \tag{9}$$

The summation upper limit  $r$  represents the degree of polynomial  $p_j(x)$ . Here,  $r = 2$ , and  $\beta_j$  has the following form:

$$\begin{aligned} \beta_0 &= (-2f_{i-1}^+ + 2f_i^+ - \Delta x h_{i-1}^+)^2 + \frac{13}{3}(-f_{i-1}^+ + f_i^+ - \Delta x h_{i-1}^+)^2, \\ \beta_1 &= (-2f_i^+ + 2f_{i+1}^+ - \Delta x h_{i+1}^+)^2 + \frac{13}{3}(-f_i^+ + f_{i+1}^+ - \Delta x h_{i+1}^+)^2, \\ \beta_2 &= \frac{1}{4}(-f_{i-1}^+ + f_{i+1}^+)^2 + \frac{13}{12}(-f_{i-1}^+ + 2f_i^+ - f_{i+1}^+)^2. \end{aligned} \tag{10}$$

1.5 Using the following formula, we obtain the nonlinear weights  $\omega_j$  on the corresponding stencils from  $\gamma_j$  in (8) and  $\beta_j$  in (10).

$$\omega_j = \frac{\bar{\omega}_j}{\sum_k \bar{\omega}_k}, \quad \text{with } \bar{\omega}_k = \frac{\gamma_k}{(\beta_k + \varepsilon)^2}, \quad j, k = 0, 1, 2, \tag{11}$$

where  $\varepsilon$  is taken as  $10^{-6}$  in our numerical experiments. The  $\varepsilon$  is a tiny positive value that keeps the denominator from going to zero. One can compute the HWENO reconstruction expression of  $\hat{f}_{i+\frac{1}{2}}^-$  by

$$\hat{f}_{i+\frac{1}{2}}^- = \sum_{j=0}^2 \omega_j p_j(x_{i+\frac{1}{2}}). \tag{12}$$

1.6 Finally, we can derive our numerical flux as follows:

$$\hat{f}_{i+\frac{1}{2}} = \hat{f}_{i+\frac{1}{2}}^+ + \hat{f}_{i+\frac{1}{2}}^-. \tag{13}$$

Step II. We consider reconstructing  $\hat{h}_{i+\frac{1}{2}}^-$  using HWENO from the function values  $\{f^\pm(u_i), h^\pm(u_i, v_i)\}$ .

2.1 As in (6), we compute  $\{f^\pm(u_i), h^\pm(u_i, v_i)\}$ .

2.2 Given three small stencils  $S_0, S_1, S_2$ , and one large stencil  $S$ , we proceed as in step 1.2. Next, using Hermite interpolation, we construct three cubic polynomials  $p_j(x)$  on the small stencils  $S_j$ . These polynomials will satisfy the requirements outlined below:

$$\begin{aligned} \frac{1}{\Delta x} \int_{I_k} p_0(x) dx &= f_k^+, & k = i-1, i, \\ \frac{1}{\Delta x} \int_{I_k} p_0'(x) dx &= h_k^+, & k = i-1, i, \\ \frac{1}{\Delta x} \int_{I_k} p_1(x) dx &= f_k^+, & k = i, i+1, \\ \frac{1}{\Delta x} \int_{I_k} p_1'(x) dx &= h_k^+, & k = i, i+1, \\ \frac{1}{\Delta x} \int_{I_k} p_2(x) dx &= f_k^+, & k = i-1, i, i+1, \\ \frac{1}{\Delta x} \int_{I_k} p_2'(x) dx &= h_k^+, & k = i. \end{aligned}$$

Similarly, Hermite interpolation is utilized to construct a quintic polynomial  $q(x)$  on the set  $S$ , which satisfies the condition

$$\begin{aligned} \frac{1}{\Delta x} \int_{I_k} q(x) dx &= f_k^+, & k = i-1, i, i+1, \\ \frac{1}{\Delta x} \int_{I_k} q'(x) dx &= h_k^+, & k = i-1, i, i+1. \end{aligned}$$

Next, we calculate the approximate value of the first derivative of the stated polynomials at the half point  $x_{i+\frac{1}{2}}$ , as we are reconstructing the numerical flux of the derivative equation. The expression is shown below.

$$\begin{aligned} p_0'(x_{i+\frac{1}{2}}) &= \frac{1}{2\Delta x} (8f_{i-1}^+ - 8f_i^+ + 6\Delta x h_{i-1}^+ + 14\Delta x h_i^+), \\ p_1'(x_{i+\frac{1}{2}}) &= \frac{1}{2\Delta x} (-4f_i^+ + 4f_{i+1}^+ - \Delta x h_i^+ - \Delta x h_{i+1}^+), \\ p_2'(x_{i+\frac{1}{2}}) &= \frac{1}{4\Delta x} (f_{i-1}^+ - 4f_i^+ + 3f_{i+1}^+ + 2\Delta x h_i^+), \\ q'(x_{i+\frac{1}{2}}) &= \frac{1}{12\Delta x} (3f_{i-1}^+ - 24f_i^+ + 21f_{i+1}^+ + \Delta x h_{i-1}^+ - 2\Delta x h_i^+ - 5\Delta x h_{i+1}^+). \end{aligned} \tag{14}$$

2.3 We compute linear weights  $\gamma'_0, \gamma'_1$  and  $\gamma'_2$  that satisfy

$$q'(x_{i+\frac{1}{2}}) = \sum_{j=0}^2 \gamma'_j p'_j(x_{i+\frac{1}{2}}),$$

with  $\sum_{j=0}^2 \gamma'_j = 1$ . This leads to a result:

$$\gamma'_0 = \frac{1}{18}, \gamma'_1 = \frac{5}{6}, \gamma'_2 = \frac{1}{9}. \tag{15}$$

2.4 To reconstruct derivatives, we define the smoothness indicator as

$$\beta_j = \sum_{l=2}^3 \int_{I_i} \Delta x^{2l-1} \left( \frac{\partial^l}{\partial x^l} p_j(x) \right)^2 dx. \tag{16}$$

Unlike (9), the summation in (16) starts with the second derivative since we are now reconstructing the first derivative value. The expressions are

$$\begin{aligned} \beta_0 &= 4(3f_{i-1}^+ - 3f_i^+ + \Delta x h_{i-1}^+ + 2\Delta x h_i^+)^2 \\ &\quad + 39(2f_{i-1}^+ - 2f_i^+ + \Delta x h_{i-1}^+ + \Delta x h_i^+)^2, \\ \beta_1 &= 4(-3f_i^+ + 3f_{i+1}^+ - \Delta x h_{i+1}^+ - 2\Delta x h_i^+)^2 \\ &\quad + 39(2f_i^+ - 2f_{i+1}^+ + \Delta x h_{i+1}^+ + \Delta x h_i^+)^2, \\ \beta_2 &= (f_{i-1}^+ - 2f_i^+ + f_{i+1}^+)^2 + \frac{39}{4}(-f_{i-1}^+ + f_{i+1}^+ - 2\Delta x h_i^+)^2. \end{aligned} \tag{17}$$

2.5 Now, we will apply the following formula to obtain the nonlinear weights  $\omega'_j$  by  $\gamma'_j$  in (15) and  $\beta_j$  in (17):

$$\omega'_j = \frac{\bar{\omega}'_j}{\sum_k \bar{\omega}'_k}, \quad \bar{\omega}'_k = \frac{\gamma'_k}{(\beta_k + \varepsilon)^2}, \quad j, k = 0, 1, 2, \tag{18}$$

where  $\varepsilon$  is still taken as  $10^{-6}$ . The HWENO reconstruction of  $\hat{h}_{i+\frac{1}{2}}^-$  can be calculated by

$$\hat{h}_{i+\frac{1}{2}}^- = \sum_{j=0}^2 \omega'_j p'_j(x_{i+\frac{1}{2}}). \tag{19}$$

2.6 Finally, our numerical flux can be obtained by

$$\hat{h}_{i+\frac{1}{2}} = \hat{h}_{i+\frac{1}{2}}^+ + \hat{h}_{i+\frac{1}{2}}^-. \tag{20}$$

When considering convection-dominated convection-diffusion equations, the diffusion constant is typically small. As a result, the characteristics of these equations resemble those of hyperbolic conservation laws. In Step I and Step II, we complete the discretization of the convection term in the semi-discrete schemes (4). Next, we interpolate the diffusion term. In this study, the diffusion terms are computed using the central difference method, which is based on three-point Hermite interpolation:

$$\begin{cases} \frac{\partial^2 u}{\partial x^2} \Big|_{x_i} = \frac{1}{2\Delta x^2} (4u_{i-1} - 8u_i + 4u_{i+1} + \Delta x v_{i-1} - \Delta x v_{i+1}), \\ \frac{\partial^2 v}{\partial x^2} \Big|_{x_i} = \frac{1}{2\Delta x^3} (-15u_{i-1} + 15u_{i+1} - 3\Delta x v_{i-1} - 24\Delta x v_i - 3\Delta x v_{i+1}), \end{cases} \tag{21}$$

where these methods can attain fifth-order accuracy.

The convection and diffusion terms in the equation have been fully discretized. We will now begin to consider the discretization of the time term. In this paper, we utilize the TVD Runge-Kutta method, which consists of three stages [13,14]:



$$\begin{cases} U^{(1)} = U^{(n)} + \Delta t L(U^{(n)}), \\ U^{(2)} = \frac{3}{4}U^{(n)} + \frac{1}{4}U^{(1)} + \frac{1}{4}\Delta t L(U^{(1)}), \\ U^{(n+1)} = \frac{1}{3}U^{(n)} + \frac{2}{3}U^{(2)} + \frac{2}{3}\Delta t L(U^{(2)}), \end{cases} \tag{22}$$

with  $U = (u, v)^T$  and the time step  $\Delta t = cfl \frac{\Delta x}{\alpha}$ , where  $cfl$  is the CFL number.

### 3. Construction of Two-Dimensional HWENO Method

This part generalizes the one-dimensional scalar equation from Section 2 to two dimensions,

$$\begin{cases} \frac{\partial u}{\partial t} + \frac{\partial f(u)}{\partial x} + \frac{\partial g(u)}{\partial y} = \varepsilon(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}), \\ u(x, y, 0) = u_0(x, y), \end{cases} \tag{23}$$

where  $t$  denotes the time variable and  $x$  and  $y$  denote the two spatial directional variables. In addition,  $u$  is a conserved variable with respect to time and space, and  $f(u)$  and  $g(u)$  are flux functions in two directions in space, respectively. The spatial computation region  $[a, b] \times [c, d]$  is divided equally by  $N_x \times N_y$ , i.e.,  $a = x_{\frac{1}{2}} < x_{\frac{3}{2}} < \dots < x_{N_x+\frac{1}{2}} = b$ ,  $c = y_{\frac{1}{2}} < y_{\frac{3}{2}} < \dots < y_{N_y+\frac{1}{2}} = d$ .  $x_{i+\frac{1}{2}}$  and  $y_{j+\frac{1}{2}}$  represent the half point in both directions, respectively. Then, the cell and cell center are  $I_{i,j} = [x_i - \frac{\Delta x}{2}, x_i + \frac{\Delta x}{2}] \times [y_j - \frac{\Delta y}{2}, y_j + \frac{\Delta y}{2}]$  and  $(x_i, y_j) = (\frac{1}{2}(x_{i-\frac{1}{2}} + x_{i+\frac{1}{2}}), \frac{1}{2}(y_{j-\frac{1}{2}} + y_{j+\frac{1}{2}}))$ , respectively, where  $\Delta x = \frac{b-a}{N_x}$  and  $\Delta y = \frac{d-c}{N_y}$  are cell sizes. To develop the Hermite WENO method, we first compute the first derivatives of Equation (23) with respect to the spatial variables  $x$  and  $y$  separately. The first derivatives of conserved quantities  $u$  with respect to  $x$  and  $y$  are denoted by  $v$  and  $w$ , respectively. Following this, we derive the corresponding formulas:

$$\begin{cases} \frac{\partial u}{\partial t} + \frac{\partial f(u)}{\partial x} + \frac{\partial g(u)}{\partial y} = \varepsilon(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}), & u(x, y, 0) = u_0(x, y), \\ \frac{\partial v}{\partial t} + \frac{\partial h(u,v)}{\partial x} + \frac{\partial r(u,v)}{\partial y} = \varepsilon(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}), & v(x, y, 0) = v_0(x, y), \\ \frac{\partial w}{\partial t} + \frac{\partial q(u,w)}{\partial x} + \frac{\partial s(u,w)}{\partial y} = \varepsilon(\frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2}), & w(x, y, 0) = w_0(x, y), \end{cases} \tag{24}$$

where

$$\begin{cases} h(u, v) = f'(u)v, r(u, v) = g'(u)v, v_0(x, y) = (u_0)_x, \\ q(u, w) = f'(u)w, s(u, w) = g'(u)w, w_0(x, y) = (u_0)_y. \end{cases}$$

It is important to note that the elements in Equations (3) and (24) exhibit certain mathematical similarities, particularly in that the reconstructions are symmetric in both directions. Consequently, we can gradually extend the algorithm from a one-dimensional problem to a two-dimensional problem, addressing it dimension by dimension. However, a challenge arises due to the mixed derivatives  $q(u, w)_x, \frac{\partial^2 w}{\partial x^2}$ , and  $r(u, v)_y, \frac{\partial^2 v}{\partial y^2}$ , which cannot be treated dimensionally in the same way as in the one-dimensional case. In this context, the mixed derivatives do not require flux splitting for simplicity. In this research, we utilize a three-point Hermite interpolation approach to calculate the mixed derivative terms while preserving the method’s compactness. Nevertheless, the numerical accuracy of the two-dimensional scheme is limited to fourth-order due to the derivative equation. We present the semi-discrete finite difference schemes for Equation (24) by drawing parallels with the analogous one-dimensional case:

$$\begin{cases} \frac{\partial u_{i,j}}{\partial t} = -\frac{1}{\Delta x}(\hat{f}_{i+\frac{1}{2},j} - \hat{f}_{i-\frac{1}{2},j}) - \frac{1}{\Delta y}(\hat{g}_{i,j+\frac{1}{2}} - \hat{g}_{i,j-\frac{1}{2}}) + \varepsilon(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2})|_{(x_i,y_j)}, \\ \frac{\partial v_{i,j}}{\partial t} = -\frac{1}{\Delta x}(\hat{h}_{i+\frac{1}{2},j} - \hat{h}_{i-\frac{1}{2},j}) - \frac{1}{\Delta y}(\hat{r}_{i,j+\frac{1}{2}} - \hat{r}_{i,j-\frac{1}{2}}) + \varepsilon(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2})|_{(x_i,y_j)}, \\ \frac{\partial w_{i,j}}{\partial t} = -\frac{1}{\Delta x}(\hat{q}_{i+\frac{1}{2},j} - \hat{q}_{i-\frac{1}{2},j}) - \frac{1}{\Delta y}(\hat{s}_{i,j+\frac{1}{2}} - \hat{s}_{i,j-\frac{1}{2}}) + \varepsilon(\frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2})|_{(x_i,y_j)}, \end{cases} \tag{25}$$



where  $\hat{f}_{i\pm\frac{1}{2},j}$ ,  $\hat{h}_{i\pm\frac{1}{2},j}$ ,  $\hat{g}_{i,j\pm\frac{1}{2}}$  and  $\hat{s}_{i,j\pm\frac{1}{2}}$  can be directly reconstructed by HWENO scheme along their respective directions as in the one-dimensional case. The higher-order derivative terms or the diffusion terms in the equations, specifically  $\frac{\partial^2 u}{\partial x^2}$ ,  $\frac{\partial^2 v}{\partial x^2}$  and  $\frac{\partial^2 u}{\partial y^2}$ ,  $\frac{\partial^2 w}{\partial y^2}$ , are discretized using the central difference method in their respective directions, as shown in (21). The remaining mixed derivative terms are discretized using three-point Hermite interpolation.

The formulas for calculating the mixed derivative terms  $\hat{q}_{i+\frac{1}{2},j}$  and  $\hat{r}_{i,j+\frac{1}{2}}$  are given by

$$\begin{cases} \hat{q}_{i+\frac{1}{2},j} = \frac{1}{64\Delta x}(3f_{i-1,j} - 96f_{i,j} + 93f_{i+1,j} + \Delta x q_{i-1,j} - 12\Delta x q_{i,j} - 15\Delta x q_{i+1,j}), \\ \hat{r}_{i,j+\frac{1}{2}} = \frac{1}{64\Delta y}(3g_{i,j-1} - 96g_{i,j} + 93g_{i,j+1} + \Delta y r_{i,j-1} - 12\Delta y r_{i,j} - 15\Delta y r_{i,j+1}). \end{cases} \quad (26)$$

The higher mixed derivatives  $\frac{\partial^2 v}{\partial y^2}$ ,  $\frac{\partial^2 w}{\partial x^2}$  in the diffusion term are calculated by the following formulas:

$$\begin{cases} \left. \frac{\partial^2 v}{\partial y^2} \right|_{(x_i,y_j)} = \frac{3}{2\Delta y^3}(-5u_{i,j-1} + 5u_{i,j+1} - \Delta y v_{i,j-1} - 8\Delta y v_{i,j} - \Delta y v_{i,j+1}), \\ \left. \frac{\partial^2 w}{\partial x^2} \right|_{(x_i,y_j)} = \frac{3}{2\Delta x^3}(-5u_{i-1,j} + 5u_{i+1,j} - \Delta x w_{i-1,j} - 8\Delta x w_{i,j} - \Delta x w_{i+1,j}). \end{cases} \quad (27)$$

Time is discretized using the Runge-Kutta method, where the time step  $\Delta t = \frac{cfl}{\frac{\alpha_1}{\Delta x} + \frac{\alpha_2}{\Delta y}}$  for the two-dimensional cases, and  $\alpha_1 = \max_u |f'(u)|$  and  $\alpha_2 = \max_u |g'(u)|$ .

### 4. Numerical Results

In this section, we focus on extensive numerical experiments about convection-diffusion equations to demonstrate the effectiveness of the proposed HWENO method within the finite difference framework. Additionally, we will compare the suggested HWENO method to the classical WENO method introduced by Shu et al. [16] in a one-dimensional finite difference context. In both numerical methods, the CFL number used in the calculation example is 0.6. While both methods yield similar results, it is evident that the HWENO method is significantly more compact than the WENO method.

#### 4.1. Numerical Tests in One-Dimensional Case

**Example 1.** We first consider the following linear convection-diffusion equation:

$$\begin{cases} \frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} = \varepsilon \frac{\partial^2 u}{\partial x^2}, \\ u(x, 0) = \sin(x), \quad 0 \leq x \leq 2\beta. \end{cases} \quad (28)$$

We apply periodic boundary conditions at both ends of the calculation region. The exact expression for the solution is given by  $u(x, t) = \exp(-\varepsilon t)\sin(x - t)$ , where the total computation time is set to  $T = 1$  and  $\varepsilon = 0.01$ . Table 1 presents the numerical orders and errors comparing the HWENO method to the WENO method. Both the WENO and HWENO methods achieve the desired fifth-order accuracy. However, based on the error comparisons shown in Table 1, we can see that the HWENO method demonstrates smaller errors than the WENO method in both the  $L_1$  and  $L_\infty$  norm when using the same mesh size. It is worth noting that the HWENO scheme includes one additional derivative equation compared to the WENO scheme, which makes it challenging to directly compare their CPU times. However, the HWENO scheme benefits from utilizing three-point stencils for reconstruction, while the WENO scheme relies on five-point stencils.

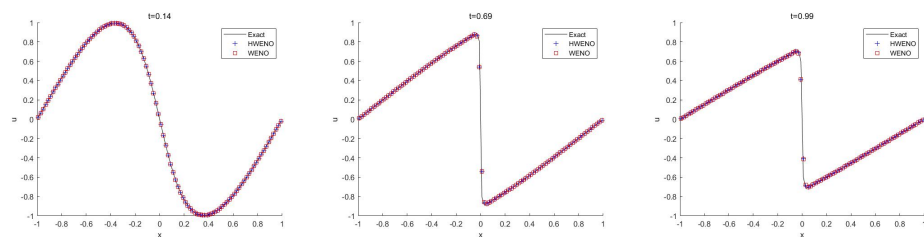
**Table 1.** Numerical results of HWENO and WENO solving one-dimensional linear convection-diffusion equation with initial value  $u(x,0) = \sin(x)$ .  $L_1$  and  $L_\infty$  errors and numerical order of accuracy.

N	HWENO		WENO					
	$L_1$ Error	Order	$L_\infty$ Error	Order	$L_1$ Error	Order	$L_\infty$ Error	Order
10	$3.78 \times 10^{-3}$		$5.76 \times 10^{-3}$		$7.67 \times 10^{-3}$		$1.05 \times 10^{-2}$	
20	$1.11 \times 10^{-4}$	5.09	$2.09 \times 10^{-4}$	4.79	$2.64 \times 10^{-4}$	4.86	$5.19 \times 10^{-4}$	4.34
40	$3.25 \times 10^{-6}$	5.09	$5.94 \times 10^{-6}$	5.14	$7.63 \times 10^{-6}$	5.11	$1.47 \times 10^{-5}$	5.14
80	$1.00 \times 10^{-7}$	5.02	$1.72 \times 10^{-7}$	5.11	$2.34 \times 10^{-7}$	5.03	$4.14 \times 10^{-7}$	5.15
160	$3.11 \times 10^{-9}$	5.01	$5.02 \times 10^{-9}$	5.10	$7.23 \times 10^{-9}$	5.02	$1.19 \times 10^{-8}$	5.13
320	$9.58 \times 10^{-11}$	5.02	$1.51 \times 10^{-10}$	5.06	$2.21 \times 10^{-10}$	5.03	$3.49 \times 10^{-10}$	5.09
640	$3.04 \times 10^{-12}$	4.98	$4.76 \times 10^{-12}$	4.99	$6.71 \times 10^{-12}$	5.04	$1.05 \times 10^{-12}$	5.06

**Example 2.** The expression of the viscous Burgers equation in the one-dimensional case is as follows:

$$\begin{cases} \frac{\partial u}{\partial t} + \frac{\partial(u^2/2)}{\partial x} = \varepsilon \frac{\partial^2 u}{\partial x^2}, \\ u(x,0) = -\sin(\beta x), \quad -1 \leq x \leq 1. \end{cases} \tag{29}$$

In the computation region  $[-1, 1]$ , Dirichlet boundary conditions are applied, specifically  $u(-1, t) = u(1, t) = 0$ . The diffusion constant  $\varepsilon$  is assumed to be  $0.01/\pi$  in this test. The Hopf-Cole transformation can be employed to derive the analytical solution; for further information, see Basdevant et al. [34]. Figure 1 presents the numerical outcomes of the experiment with  $N = 200$  and contrasts them with the findings obtained at the same uniform points using the WENO method. Numerical results generated using the finite difference HWENO and WENO methods align with the exact results, even at discontinuities. Additionally, we can observe that the performance of HWENO is quantitatively comparable to that of the WENO method.



**Figure 1.** The one-dimensional viscous Burgers equation solutions via HWENO and WENO methods at  $t = 0.14, 0.69,$  and  $0.99$ . Red squares represent the outcome of the classical WENO method; blue plus signs represent the result of the finite difference HWENO method; black solid line represents the exact answer.

**Example 3.** The Buckley-Leverett equation is a class of equations that is often used in reservoir simulations; let us now address this one-dimensional equation:

$$\frac{\partial u}{\partial t} + \frac{\partial f(u)}{\partial x} = \varepsilon \frac{\partial(v(u)u_x)}{\partial x}, \quad \varepsilon v(u) \geq 0. \tag{30}$$

In this example, the ultimate calculation time of the numerical method is  $T = 0.2$  and the diffusion constant  $\varepsilon$  is 0.01. In addition,

$$v(u) = \begin{cases} 4u(1 - u), & 0 \leq u \leq 1, \\ 0, & \text{otherwise.} \end{cases} \tag{31}$$

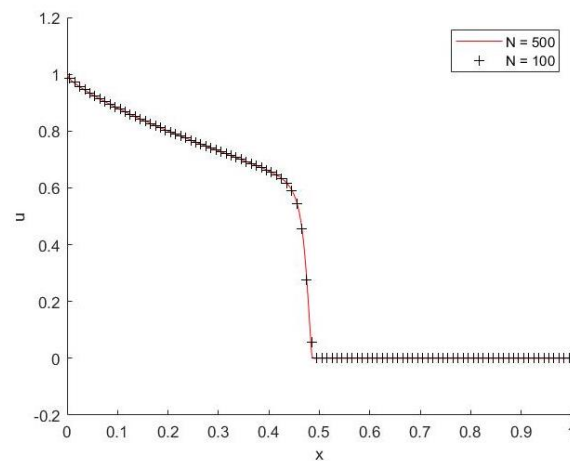
In this test, there is no gravity, and the flux function can be expressed as

$$f(u) = \frac{u^2}{u^2 + (1 - u)^2} \tag{32}$$

and the conserved quantity  $u$  has the following continuous initial conditions:

$$u(x, 0) = \begin{cases} 1 - 3x, & 0 \leq x \leq 1/3, \\ 0, & 1/3 \leq x \leq 1. \end{cases} \tag{33}$$

The calculation region has a left boundary condition defined as  $u(0, t) = 1$ , while the right boundary condition is determined through extrapolation. There is no exact solution expression available for this problem. We use the numerical solution obtained from the HWENO method with  $N = 500$  as a reference exact solution. A comparison between the numerical solution with 100 grid points and the reference exact solution is shown in Figure 2. There is no oscillation at the discontinuity, and the numerical solution aligns well with the reference solution derived from the method described in the study.



**Figure 2.** The Buckley-Leverett equation for one-dimensional case.  $T = 0.2$ . Comparison of reference and numerical solutions. The red solid line is the reference solution; the black plus sign is the numerical solution.

#### 4.2. Numerical Tests in Two-Dimensional Case

**Example 4.** We now test the accuracy and error of the two-dimensional linear convection-diffusion equation:

$$\begin{cases} \frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} + \frac{\partial u}{\partial y} = \varepsilon \left( \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right), \\ u(x, y, 0) = \sin(2\pi(x + y)). \end{cases} \tag{34}$$

Periodic boundaries are utilized in this case at all the edges of the calculating region  $(x, y) \in [0, 1] \times [0, 1]$ .  $u(x, y, t) = \exp(-8\pi^2\varepsilon t)\sin(2\pi(x + y - t))$  is the exact solution. The ultimate calculation time is  $T = 0.1$ . The diffusion coefficient  $\varepsilon = 0.001$ . The numerical results of the HWENO method under the  $L_1$ ,  $L_2$ , and  $L_\infty$  are displayed in Table 2. The results illustrate that the suggested two-dimensional numerical method’s accuracy meets the required fourth order.

**Table 2.**  $L_1$ ,  $L_2$ , and  $L_\infty$  errors and numerical order of accuracy of HWENO for solving two-dimensional linear convection-diffusion equation with initial value  $u(x, y, 0) = \sin(2\pi(x + y))$ .

$N_x * N_y$	$L_1$ Error	Order	$L_2$ Error	Order	$L_\infty$ Error	Order
$10 \times 10$	$4.61 \times 10^{-3}$		$1.46 \times 10^{-2}$		$4.61 \times 10^{-2}$	
$20 \times 20$	$1.19 \times 10^{-4}$	5.28	$5.32 \times 10^{-4}$	4.78	$2.38 \times 10^{-3}$	4.28
$40 \times 40$	$4.05 \times 10^{-6}$	4.88	$2.56 \times 10^{-5}$	4.38	$1.62 \times 10^{-4}$	3.88
$80 \times 80$	$1.62 \times 10^{-7}$	4.64	$1.45 \times 10^{-6}$	4.14	$1.30 \times 10^{-5}$	3.64
$160 \times 160$	$6.09 \times 10^{-9}$	4.73	$7.71 \times 10^{-8}$	4.23	$9.75 \times 10^{-7}$	3.73
$320 \times 320$	$2.73 \times 10^{-10}$	4.48	$4.89 \times 10^{-9}$	3.98	$8.74 \times 10^{-8}$	3.48

**Example 5.** The Buckley-Leverett equation in two dimensions is examined in this example:

$$\frac{\partial u}{\partial t} + \frac{\partial f(u)}{\partial x} + \frac{\partial g(u)}{\partial y} = \varepsilon \left( \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right). \tag{35}$$

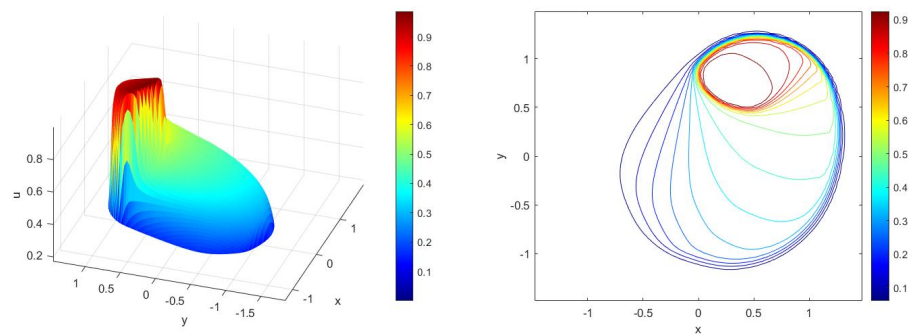
The diffusion coefficient  $\varepsilon$  takes 0.01. The flux function in both directions can be expressed by

$$\begin{aligned} f(u) &= \frac{u^2}{(u^2 + (1 - u)^2)}, \\ g(u) &= f(u)(1 - 5(1 - u)^2). \end{aligned} \tag{36}$$

The initial conditions are

$$u(x, y, 0) = \begin{cases} 1, & x^2 + y^2 < 0.5, \\ 0, & \text{otherwise.} \end{cases} \tag{37}$$

Note that in this example, there is a gravitational effect in the y-direction. This question is more complex and does not have an analytical answer. The rectangular domain  $[-1.5, 1.5] \times [-1.5, 1.5]$  is equally divided by an  $80 \times 80$  grid whose boundaries are constrained by periodic boundary conditions. Figure 3 shows the numerical results and two images for  $T = 0.5$ . The surface and contour plots are consistent with the results listed in reference [35].



**Figure 3.** Two-dimensional surface mapping of the Buckley-Leverett equation, shown on the left. Two-dimensional Buckley-Leverett equation contour map, as shown on the right.  $80 \times 80$  grid points at  $T = 0.5$ .

**Example 6.** By applying the vorticity transport equation and the stream-function equation, one can solve the two-dimensional incompressible high Reynolds number ( $Re \gg 1$ ) Navier-Stokes equation:

$$\frac{\partial \omega}{\partial t} + u \frac{\partial \omega}{\partial x} + v \frac{\partial \omega}{\partial y} = \frac{1}{Re} \left( \frac{\partial^2 \omega}{\partial x^2} + \frac{\partial^2 \omega}{\partial y^2} \right), \quad (38)$$

$$\frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial y^2} = -\omega, \quad (39)$$

where  $u$  and  $v$  are the velocity components,  $\psi$  is the stream-function, and  $Re$  is the Reynolds number. Let  $\omega = v_x - u_y$  be the vorticity. The following is the relationship between velocity and the stream-function:

$$u = \frac{\partial \psi}{\partial y}, \quad v = -\frac{\partial \psi}{\partial x}. \quad (40)$$

For simplicity, we take the no-slip boundary conditions  $\psi|_{\partial\Omega} = 0$ , where  $\Omega$  is the space domain in the actual issue. Equation (39) is an elliptical equation, and (38) is a parabolic-type vorticity transport equation. To maintain high accuracy, Equations (39) and (40) are solved discretely by the compact finite difference schemes described by Lele [36]. The article uses this fourth-order compact difference to solve the first and second derivatives of the stream function  $\psi$  in the  $x$  and  $y$  directions:

$$\begin{cases} \frac{1}{10}f''_{i-1} + f''_i + \frac{1}{10}f''_{i+1} = \frac{6}{5} \frac{f_{i+1} - 2f_i + f_{i-1}}{\Delta x^2}, \\ \frac{1}{4}f'_{i-1} + f'_i + \frac{1}{4}f'_{i+1} = \frac{3}{4\Delta x}(f_{i+1} - f_{i-1}), \end{cases} \quad (41)$$

where  $f'$  and  $f''$  represent the first and second derivatives of the stream function with respect to  $x$  or  $y$ , respectively. Finally, Equation (38) is solved using the numerical method introduced in this paper.

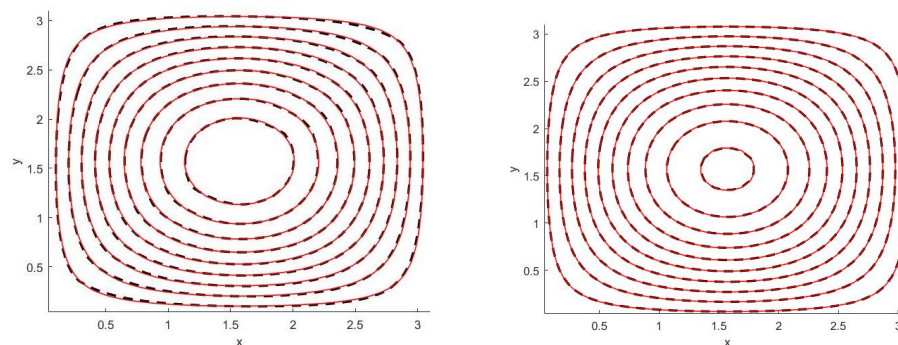
In the square domain  $\Omega = [0, \pi] \times [0, \pi]$ , the vorticity stream-function formula for the two-dimensional Navier-Stokes problem is solved. The following are the initial conditions of the vorticity:

$$\omega(x, y, 0) = -2\sin(x)\sin(y). \quad (42)$$

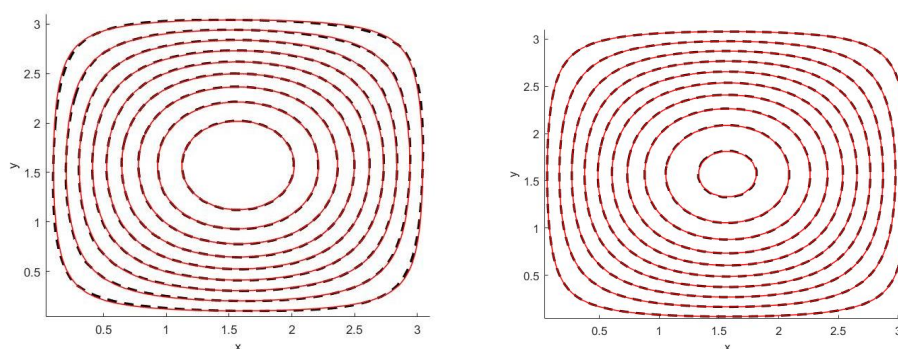
The following are exact solutions that are known to exist:

$$\omega(x, y, t) = -2\sin(x)\sin(y)e^{-2t/Re}, \quad (43)$$

used in [37]. We take the Dirichlet boundary for the space domain derived from the exact solution. The Reynolds numbers in the uniform grid are 200 points and 1000 points, respectively, and the final calculation time of the numerical solution is  $T = 0.5$ . To demonstrate the effectiveness of the proposed method, the exact solution images of the vorticity flow function are compared with the numerical solution images. The contours of the vorticity and stream function are displayed in Figure 4 at the final time for  $Re = 200$ . According to the results, we take  $40 \times 40$  uniform meshes, and the numerical solution images of vorticity and stream function match up well with the exact solution images. For  $Re = 1000$ , similar results are given in Figure 5.



**Figure 4.** The red solid line represents the exact solutions, while the dashed black line represents the numerical results. The left is the vorticity contour image, and the right is the stream-function contour image.  $Re = 200$ ,  $40 \times 40$  grid.



**Figure 5.**  $Re = 1000$ , which is a similar representation to  $Re = 200$ .

## 5. Concluding Remarks

In this paper, we presented an HWENO method for solving one- and two-dimensional convection-dominated problems within the framework of finite difference techniques. The efficiency of the finite difference method outperformed the finite volume method when solving multidimensional equations. Another significant advantage of the HWENO method is its compactness over the classical WENO method. This improvement arises from the fact that WENO reconstruction relies solely on function values, while HWENO reconstruction incorporates both function values and their derivatives. Numerical tests show that the HWENO method often yields lower numerical errors than the WENO method. However, compared with the WENO method, the HWENO method requires more computation time and storage space for the same set of grid points. One of the challenges addressed in this paper is the treatment of mixed derivatives in spatial terms and higher derivatives in diffusion terms for two-dimensional cases. Given the properties of the equations and the high-order accuracy and compactness offered by the HWENO numerical method, we propose an appropriate processing procedure to ensure both the stability and conservation of the method. Our results indicate that when solving convection-dominated diffusion problems, the finite difference HWENO method consistently outperforms the WENO method in various numerical examples.

**Author Contributions:** Conceptualization, H.L.; methodology, H.L.; software, Y.W.; validation, H.L.; investigation, H.L.; writing—original draft preparation, Y.W.; writing—review and editing, H.L.; visualization, Y.W.; supervision, H.L.; funding acquisition, H.L. All authors have read and agreed to the published version of the manuscript.

**Funding:** The work was partially supported by Natural Science Foundation of Shanxi Province, China (No. 202103021224041) and National Natural Science Foundation of China, grant 11601364.



**Data Availability Statement:** Some or all data, models, or code generated or used during the study are available from the corresponding author by request.

**Acknowledgments:** The authors wish to thank the anonymous reviewers for their suggestions.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

1. Auerbach, R.; Globerson, A. *Numerical Solution of Convection-Diffusion Problems*; Chapman & Hall: London, UK, 1996.
2. Li, L.; Yin, Z. Numerical simulation of groundwater pollution problems based on convection diffusion equation. *Am. J. Comput. Math.* **2017**, *7*, 350–370. [[CrossRef](#)]
3. Ammi, M.R.S.; Jamiai, I. Finite difference and legendre spectral method for a time-fractional diffusion-convection equation for image restoration. *Discret. Contin. Dyn. Syst. Ser. S* **2017**, *11*, 117.
4. Baliga, B.R.; Patankar, S.V. A new finite-element formulation for convection-diffusion problems. *Numer. Heat Transf.* **1980**, *3*, 393–409. [[CrossRef](#)]
5. Brenner, S.C.; Scott, L.R. *The Mathematical Theory of Finite Element Methods*; Springer: Berlin/Heidelberg, Germany, 1994.
6. Douglas, J.; Russell, T.F. Numerical methods for convection-dominated diffusion problems based on combining the method of characteristics with finite element or finite difference procedures. *SIAM J. Numer. Anal.* **1982**, *19*, 871–885. [[CrossRef](#)]
7. Shu, C.-W. Bound-preserving high order finite volume schemes for conservation laws and convection-diffusion equations. In Proceedings of the 8th Conference on Finite Volumes for Complex Applications (FVCA 8), Lille, France, 12–16 June 2017; pp. 3–14.
8. Saqib, M.; Hasnain, S.; Mashat, D.S. Computational solutions of two dimensional convection diffusion equation using crank-nicolson and time efficient adi. *Am. J. Comput. Math.* **2017**, *7*, 208–227. [[CrossRef](#)]
9. Li, L.; Jiang, Z.; Yin, Z. Fourth-order compact finite difference method for solving two-dimensional convection-diffusion equation. *Adv. Differ. Equations* **2018**, *2018*, 234.
10. Harten, A.; Engquist, B.; Osher, S.; Chakravarthy, S.R. Uniformly high order accuracy essentially non-oscillatory schemes, iii. *J. Comput. Phys.* **1987**, *71*, 231–303. [[CrossRef](#)]
11. Harten, A. High resolution schemes for hyperbolic conservation laws. *J. Comput. Phys.* **1983**, *49*, 357–393. [[CrossRef](#)]
12. Harten, A. *Preliminary Results on the Extension of Eno Schemes to Two-Dimensional Problems*; Springer: Berlin/Heidelberg, Germany, 1987.
13. Shu, C.-W.; Osher, S. Efficient implementation of essentially non-oscillatory shock-capturing schemes. *J. Comput. Phys.* **1988**, *77*, 439–471. [[CrossRef](#)]
14. Shu, C.-W.; Osher, S. Efficient implementation of essentially non-oscillatory shock-capturing schemes, ii. *J. Comput. Phys.* **1989**, *83*, 32–78. [[CrossRef](#)]
15. Liu, X.-D.; Osher, S.; Chan, T. Weighted essentially non-oscillatory schemes. *J. Comput. Phys.* **1994**, *115*, 200–212. [[CrossRef](#)]
16. Jiang, G.-S.; Shu, C.-W. Efficient implementation of weighted eno schemes. *J. Comput. Phys.* **1996**, *126*, 202–228. [[CrossRef](#)]
17. Balsara, D.S.; Shu, C.-W. Monotonicity preserving weighted essentially non-oscillatory schemes with increasingly high order of accuracy. *J. Comput. Phys.* **2000**, *160*, 405–452. [[CrossRef](#)]
18. Friedrich, O. Weighted essentially non-oscillatory schemes for the interpolation of mean values on unstructured grids. *J. Comput. Phys.* **1998**, *144*, 194–212. [[CrossRef](#)]
19. Shu, C.-W. High order weighted essentially nonoscillatory schemes for convection dominated problems. *SIAM Rev.* **2009**, *51*, 82–126. [[CrossRef](#)]
20. Shi, J.; Hu, C.; Shu, C.-W. A technique of treating negative weights in weno schemes. *J. Comput. Phys.* **2002**, *175*, 108–127. [[CrossRef](#)]
21. Qiu, J.; Shu, C.-W. Hermite weno schemes and their application as limiters for runge-kutta discontinuous galerkin method: One-dimensional case. *J. Comput. Phys.* **2004**, *193*, 115–135. [[CrossRef](#)]
22. Qiu, J.; Shu, C.-W. Hermite weno schemes and their application as limiters for runge-kutta discontinuous galerkin method ii: Two dimensional case. *Comput. Fluids* **2005**, *34*, 642–663. [[CrossRef](#)]
23. Liu, H.; Qiu, J. Finite difference hermite weno schemes for hyperbolic conservation laws. *J. Sci. Comput.* **2014**, *63*, 548–572. [[CrossRef](#)]
24. Zhao, Z.; Zhang, Y.-T.; Qiu, J. A modified fifth order finite difference Hermite WENO scheme for hyperbolic conservation laws. *J. Sci. Comput.* **2020**, *85*, 29. [[CrossRef](#)]
25. Zhao, Z.; Chen, Y.; Qiu, J. A hybrid Hermite WENO scheme for hyperbolic conservation laws. *J. Comput. Phys.* **2020**, *405*, 109175. [[CrossRef](#)]
26. Zhao, Z.; Qiu, J. A Hermite WENO scheme with artificial linear weights for hyperbolic conservation laws. *J. Comput. Phys.* **2020**, *417*, 109583. [[CrossRef](#)]



27. Fan, C.; Zhang, X.; Qiu, J. Positivity-preserving high order finite volume hybrid Hermite WENO schemes for compressible Navier-Stokes equations. *J. Comput. Phys.* **2021**, *445*, 110596. [[CrossRef](#)]
28. Li, J.; Shu, C.-W.; Qiu, J. Moment-Based Multi-Resolution HWENO Scheme for Hyperbolic Conservation Laws. *J. Comput. Phys.* **2022**, *32*, 364–400.
29. Zhao, Z.; Qiu, J. An oscillation-free Hermite WENO scheme for hyperbolic conservation laws. *Sci. China Math.* **2024**, *67*, 431–454. [[CrossRef](#)]
30. Zhang, M.; Zhao, Z. A fifth-order finite difference HWENO scheme combined with limiter for hyperbolic conservation laws. *J. Comput. Phys.* **2023**, *472*, 111676. [[CrossRef](#)]
31. Wibisono, I.; Yanuar; Kosasih, E.A. Fifth-Order Hermite Targeted Essentially Non-oscillatory Schemes for Hyperbolic Conservation Laws. *J. Sci. Comput.* **2021**, *87*, 69. [[CrossRef](#)]
32. Zheng, F.; Qiu, J. Dimension by Dimension Finite Volume HWENO Method for Hyperbolic Conservation Laws. *Commun. Appl. Math. Comput.* **2024**, *6*, 605–624. [[CrossRef](#)]
33. Liu, H.; Qiu, J. Finite difference hermite weno schemes for conservation laws, ii: An alternative approach. *J. Sci. Comput.* **2015**, *66*, 598–624. [[CrossRef](#)]
34. Basdevant, C.; Deville, M.; Haldenwang, P.; Lacroix, J.M.; Ouazzani, J.; Peyret, R.; Orlandi, P.; Patera, A.T. Spectral and finite difference solutions of the burgers equation. *Comput. Fluids* **1986**, *14*, 23–41. [[CrossRef](#)]
35. Kurganov, A.; Tadmor, E. New High-Resolution Central Schemes for Nonlinear Conservation Laws and Convection–Diffusion Equations. *J. Comput. Phys.* **2000**, *160*, 241–282. [[CrossRef](#)]
36. Lele, S.K. Compact finite difference schemes with spectral-like resolution. *J. Comput. Phys.* **1992**, *103*, 16–42. [[CrossRef](#)]
37. Liu, J.-G.; Shu, C.-W. A high-order discontinuous galerkin method for 2d incompressible flows. *J. Comput. Phys.* **2000**, *160*, 577–596. [[CrossRef](#)]

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.