A Hybrid Interpolating Meshless Method for 3D Advection–Diffusion Problems

Zhijuan Meng, Xiaofei Chi, and Lidong Ma

Abstract: A hybrid interpolating meshless (HIM) method is established for dealing with three-dimensional (3D) advection–diffusion equations. To improve computational efficiency, a 3D equation is changed into correlative two-dimensional (2D) equations. The improved interpolating moving least-squares (IIMLS) method is applied in 2D subdomains to obtain the required approximation function with interpolation property. The finite difference method (FDM) is utilized in time domain and the splitting direction. Setting diagonal elements to one in the coefficient matrix is chosen to directly impose Dirichlet boundary conditions. Using the HIM method, difficulties created by the singularity of the weight functions, such as truncation error and calculation inconvenience, are overcome. To prove the advantages of the new method, some advection–diffusion equations are selected and solved by HIM, dimension splitting element-free Galerkin (DSEFG), and improved element-free Galerkin (IEFG) methods. Comparing and analyzing the calculation results of the three methods, it can be shown that the HIM method effectively improves computation speed and precision. In addition, the effectiveness of the HIM method in the nonlinear problem is verified by solving a 3D Richards’ equation.

Keywords: hybrid interpolating meshless method; nonsingular weight function; advection–diffusion equation; finite difference method

MSC: 65N30; 76R99

1. Introduction

In practical engineering problems, many partial differential equations are hard to solve using analytic methods, so numerical methods have become an important tool to obtain approximate solutions. The finite element method (FEM) [1–3] divides the continuum into finite elements, and the junction nodes of the elements are taken as the discrete points. The boundary element method (BEM) [4] reduces the dimension of the problem. Under the condition of the same discretization accuracy, the accuracy of the BEM is higher than that of the FEM. However, when dealing with elastic-plastic problems or large finite deformation problems, the advantage of BEM disappears. The meshless method [5–7] is a typical method. Since it is implemented on the basis of multiple groups of discrete nodes, the construction of a trial function is not constrained by elements or grids, and there is no need for mesh reconstruction. When simulating complex problems such as large deformation, the meshless method has great adaptivity [8,9].

Advective–diffusion equations are mainly used to describe the variation law of some physical quantities carried by fluid particles, such as the distribution of pollutants in nuclear waste pollution, heat conduction in fluid, and other physical phenomena [10,11]. The numerical method for advection–diffusion problems can be widely applied in the fields of hydromechanics, aerodynamics, environmental science and energy development. Although the FEM and BEM can be used in solving advection–diffusion problems, the numerical
solutions obtained by these methods usually cause numerical oscillation for advection-dominated problems. The advantages of the meshless method include its great precision and its ability to overcome the mesh-dependence phenomenon. Therefore, utilizing the meshless method to calculate solutions of advection–diffusion equations has important research significance.

Moving least square (MLS) approximation [12–14] is a common way to fit discrete data using the meshless method. When exploring the meshless method, the element-free Galerkin (EFG) method [15], which is extensively researched and applied in various fields, provides a new direction for development. Adopting the improvement of MLS approximation [16], an improved EFG (IEFG) method that reduces the amount of computation is presented to solve diffusion equations, etc. [17–20]. For the convenience of applying boundary conditions, an interpolating EFG method [21,22] is proposed according to the interpolating MLS method [23]. By modifying the basic functions, an improved interpolating MLS (IIMLS) method [24] with a simpler format is proposed. From this, the improved interpolating EFG (IEFG) method [25–28] is established for some classical problems.

To overcome the disadvantages of the IEFG method, such as its slow speed when calculating 3D equations, the dimension splitting method (DSM) [29,30] is introduced to further develop the meshless method. A dimension splitting EFG (DSEFG) method [31–33], which greatly improves the computation speed, is proposed and applied to calculate the numerical solutions of 3D equations. The hybrid complex variable meshless method [34–36] is presented according to the improved complex variable EFG method [37] and DSM. Moreover, a reducing dimension interpolating EFG method [38,39], which is highly accurate, is proposed. However, this method has difficulties, such as truncation error and calculation inconvenience, created by singular weight functions. To solve the above difficulties, an efficient interpolating EFG method [40] is further proposed and successfully used to solve 3D wave equations.

In this paper, a hybrid interpolating meshless (HIM) method is proposed for dealing with 3D advection–diffusion equations. To improve computational efficiency, a three-dimensional equation is changed into correlative two-dimensional equations on the basis of DSM. The IIEFG method is applied on two-dimensional planes. Utilizing the IIMLS method, the approximation function with interpolation property is constructed. The FDM (finite difference method) is utilized in the time domain and axis after splitting. Dirichlet boundary conditions are introduced directly by setting diagonal elements in the coefficient matrix to one. Then, dealing with the solvable algebraic equations, the numerical solutions can be obtained. To prove the advantages of the HIM method, some examples are selected and solved by the IEFG, HIM and DSEFG methods. The influence of time step ∆t, node distributions, scaling parameters of influence domain dmax, temporal interval t0 and splitting directions on the numerical results is discussed. Comparing results obtained by the above three methods, it is demonstrated that the HIM method significantly increases calculation efficiency and accuracy. Moreover, by solving a 3D Richards’ equation [41,42], the effectiveness of the HIM method in the nonlinear problem is verified.

2. Splitting Process of 3D Advection–Diffusion Problems

Consider a 3D advection–diffusion equation as follows.

\[
\frac{\partial u}{\partial t} - \beta_x \frac{\partial^2 u}{\partial x^2} - \beta_y \frac{\partial^2 u}{\partial y^2} - \beta_z \frac{\partial^2 u}{\partial z^2} + \gamma_x \frac{\partial u}{\partial x} + \gamma_y \frac{\partial u}{\partial y} + \gamma_z \frac{\partial u}{\partial z} = f(a, t), \quad a \in \Omega
\]  

(1)

When \(a \in \Gamma_u\), it has the essential boundary condition

\[
u - \bar{u} = 0
\]  

(2)
When \( \alpha \in \Gamma_q \), it has the natural boundary condition

\[
\beta_x \frac{\partial u}{\partial \xi} + \beta_y \frac{\partial u}{\partial \eta} + \beta_z \frac{\partial u}{\partial \zeta} = \bar{f}(x, t)
\]  

(3)

The initial condition is

\[
u(u, 0) = u_0
\]  

(4)

where \( u = (x, y, z) \), \( u(x, t) \) is the field function used to describe physical quantities carried per unit volume of a fluid, \( t \) is time, \( \bar{f}(x, t) \) is a source term representing the unsteady term that cannot be included in the governing equation, \( \beta_x, \beta_y, \beta_z \) are diffusion coefficients which can be constants, \( \gamma_x, \gamma_y, \gamma_z \) are advection coefficients in direction \( x, y, z \), respectively. The 3D solution domain \( \Omega \) gives the boundary \( \Gamma = \Gamma_u \cup \Gamma_q \) with \( \Gamma_u \cap \Gamma_q = \emptyset \). \( \pi \) and \( \bar{f}(x, t) \) are known functions on \( \Gamma_u \) and \( \Gamma_q \), respectively. \( \eta_x, \eta_y, \eta_z \) are the projections on the direction \( x, y, z \) of \( n = (\eta_x, \eta_y, \eta_z) \), which is a unit outer normal vector of \( \Gamma_q \). \( u_0 \) is a prescribed function at \( t = 0 \).

The DSM was adopted to reduce the dimension of the 3D problem. Generally, a splitting direction that is easy to calculate and program is selected according to the form of control equation or boundary condition. For complex problems, we tend to choose the direction with a uniform cross section as the splitting direction. Assume \( \Omega \) is split along direction \( x \), and \( L = 1{}^{2}D \) planes are inserted in \( \Omega \) with equal spacing, then a group of subdomains \( \Omega^{(k)}(k = 0, 1, \cdots, L) \) can be obtained. The distance between two planes is denoted by \( \Delta x \), and the 3D solution domain \( \Omega \) can be given by

\[
\Omega = \bigcup_{k=0}^{L-1} \left\{ \Omega^{(k)} \times \left[ x^{(k)}, x^{(k+1)} \right] \right\} \cup \Omega^L.
\]  

(5)

here \( x \in [a, b] \),

\[
a = x^{(0)}, \quad b = x^{(L)}, \quad x^{(k)} < x^{(k+1)},
\]  

(6)

\[
\Delta x = (b - a) / L
\]  

(7)

If the values of \( x^{(k)} \) and \( t \) are fixed, \( u, \frac{\partial u}{\partial x} \) and \( \frac{\partial^2 u}{\partial x^2} \) can be thought of as functions of \( y \) and \( z \). Similarly, \( u = u(x, y, z) \) can be considered as a function in subdomains with the new boundary \( \Gamma^{(k)} \), where \( \Gamma^{(k)} = \Gamma_{u}^{(k)} \cup \Gamma_{q}^{(k)} \), \( \Gamma_{u}^{(k)} \cap \Gamma_{q}^{(k)} = \emptyset \). Hence, Equations (1)–(3) can be changed to correlative 2D equations, i.e.,

\[
-x \frac{\partial^2 u}{\partial x^2} - \beta_x \frac{\partial u}{\partial x} + \gamma_x \frac{\partial u}{\partial y} + \gamma_y \frac{\partial u}{\partial z} = \bar{f} + \beta_x \frac{\partial^2 u}{\partial x^2} - \gamma_x \frac{\partial u}{\partial x} - \gamma_y \frac{\partial u}{\partial y}, \quad x = x^{(k)}, \quad (y, z) \in \Omega^{(k)},
\]  

(8)

with the new form of boundary conditions

\[
u(y, z) = \pi(y, z) = \pi(x^{(k)}, y, z, t), \quad (y, z) \in \Gamma_{u}^{(k)},
\]  

(9)

\[
\beta_y \frac{\partial u}{\partial y} + \beta_z \frac{\partial u}{\partial z} \eta_z = \bar{q}(y, z, t) = \bar{q}(x^{(k)}, y, z, t), \quad (y, z) \in \Gamma_{q}^{(k)}.
\]  

(10)

To obtain its Galerkin weak form after splitting, applying the variational principle to Equation (8), we can obtain

\[
\int_{\Omega^{(k)}} \delta u \left( \frac{\partial u}{\partial x} - \bar{f} + \beta_x \frac{\partial^2 u}{\partial x^2} + \gamma_x \frac{\partial u}{\partial y} + \gamma_y \frac{\partial u}{\partial z} \right) d\Omega^{(k)} + \int_{\Omega^{(k)}} \delta u \cdot \gamma_z \frac{\partial u}{\partial z} d\Omega^{(k)} = \int_{\Omega^{(k)}} \delta u \cdot \left( \beta_y \frac{\partial^2 u}{\partial y^2} + \beta_z \frac{\partial^2 u}{\partial z^2} \right) d\Omega^{(k)} = 0
\]  

(11)
Using the integration aspect of the last integral term in Equation (11), we can obtain

\[ \int_{\Omega(\delta)} \delta u \cdot \left( \frac{\partial u}{\partial n} - \mathbf{f} - \beta z \frac{\partial^2 u}{\partial z^2} + \gamma x \frac{\partial u}{\partial x} \right) d\Omega(k) + \int_{\Omega(\delta)} \delta u \cdot \gamma y \frac{\partial u}{\partial y} d\Omega(k) \]

\[ + \int_{\Omega(\delta)} \delta u \cdot \gamma \frac{\partial u}{\partial e} d\Omega(k) - \int_{\Gamma_y^k} \delta u \cdot \mathbf{n} \cdot \nabla u d\Gamma_y^k \]

\[ + \int_{\Omega(\delta)} \delta (\nabla u)^T \cdot \mathbf{\bar{b}} \nabla u d\Omega(k) = 0 \]  

Equation (12) becomes

\[ \int_{\Omega(i)} \delta u \cdot \left( \frac{\partial u}{\partial n} - \mathbf{f} - \beta z \frac{\partial^2 u}{\partial z^2} + \gamma x \frac{\partial u}{\partial x} \right) d\Omega(k) + \int_{\Omega(i)} \delta u \cdot \gamma y \frac{\partial u}{\partial y} d\Omega(k) \]

\[ + \int_{\Omega(i)} \delta u \cdot \gamma \frac{\partial u}{\partial e} d\Omega(k) - \int_{\Gamma_y^k} \delta u \cdot \mathbf{n} \cdot \nabla u d\Gamma_y^k \]

\[ + \int_{\Omega(i)} \delta (\nabla u)^T \cdot \mathbf{\bar{b}} \nabla u d\Omega(k) = 0 \]  

Equation (15) is the Galerkin weak form for the original 3D problem described by Equations (1)–(3) under the DSM.

3. The HIM Method for 3D Advection–Diffusion Equations

3.1. The IIMLS Method on the Basis of Nonsingular Weight Functions

The IIMLS approximation [24] is utilized in subdomains \( \Omega(k) \) to acquire the shape function, and the approximating function with interpolation property can be further formed.

Assume there are \( M \) discrete nodes \( \mathbf{a}_i^{(k)} \) in subdomains \( \Omega(k) \), where \( I = 1, 2, \ldots, M \), \( k = 0, 1, \ldots, L \). Unknown function \( u(\mathbf{a}, t) \) of point \( \mathbf{a}_i^{(k)} \) when time takes \( t \) is expressed as follows.

\[ u(\mathbf{a}_i^{(k)}) = u(x, \mathbf{a}_i^{(k)}, t) \]

Equation (16) is the IIMLS method, and then the approximating function with interpolation property can be further formed.

The value of \( u(\mathbf{a}_i^{(k)}) \) is affected by points \( \mathbf{a}_j^{(k)} \) whose influence domains cover node \( \mathbf{a}_i^{(k)} \), where \( I = 1, 2, \ldots, n, \mathbf{a}_j^{(k)} = (y, z) \in \Omega(k) \).

To obtain the desired shape function, we choose the following basis function.

\[ p^T = (p_0(\mathbf{a}_i^{(k)}), p_1(\mathbf{a}_i^{(k)}), p_2(\mathbf{a}_i^{(k)})) = (1, y, z) \]  

Let \( i = 0, 1, 2 \) and transform \( p_i(\mathbf{a}_i^{(k)}) \) as

\[ \tilde{p}_i(\mathbf{a}_i^{(k)}) = p_i(\mathbf{a}_i^{(k)}) - \sum_{j=1}^{n} \tilde{v}(\mathbf{a}_i^{(k)}, \mathbf{a}_j^{(k)}) p_i(\mathbf{a}_j^{(k)}) \]  

\[ \tilde{v}(\mathbf{a}_i^{(k)}, \mathbf{a}_j^{(k)}) \] is expressed by

\[ \tilde{v}(\mathbf{a}_i^{(k)}, \mathbf{a}_j^{(k)}) = \frac{\tilde{\zeta}(\mathbf{a}_i^{(k)}, \mathbf{a}_j^{(k)})}{\sum_{j=1}^{n} \tilde{\zeta}(\mathbf{a}_i^{(k)}, \mathbf{a}_j^{(k)})}, \tilde{\zeta}(\mathbf{a}_i^{(k)}, \mathbf{a}_j^{(k)}) = \prod_{l \neq j} \frac{||\mathbf{a}_i^{(k)} - \mathbf{a}_j^{(k)}||^2}{||\mathbf{a}_i^{(k)} - \mathbf{a}_j^{(k)}||^2} \]  

\( n \) denotes the total amount of points with influence domains covering node \( \mathbf{a}_i^{(k)} \).

\[ \tilde{\zeta}(\mathbf{a}_i^{(k)}, \mathbf{a}_j^{(k)}) = \prod_{l \neq j} \frac{||\mathbf{a}_i^{(k)} - \mathbf{a}_j^{(k)}||^2}{||\mathbf{a}_i^{(k)} - \mathbf{a}_j^{(k)}||^2} \]
Apply similar transformation to \( u(x^{(k)}, a^{(k)}, t) \), then a new form is obtained as
\[
\tilde{u}^{(k)}(a^{(k)}) = u(a^{(k)}) - \sum_{i=1}^{n} \tilde{v}(a^{(k)}, a_i^{(k)}) u(a_i^{(k)})
\] (20)

That is
\[
u(a^{(k)}) = \tilde{u}^{(k)}(a^{(k)}) + \sum_{i=1}^{n} \tilde{v}(a^{(k)}, a_i^{(k)}) u(a_i^{(k)})
\] (21)

In subdomain \( \Omega^{(k)} \), the approximation of \( \tilde{u}^{(k)}(a^{(k)}) \) is represented by \( \tilde{n}^{h}(x^{(k)}, a^{(k)}, t) \) as
\[
\tilde{n}^{h}(x^{(k)}, a^{(k)}, t) = \sum_{i=1}^{m} \tilde{p}_i(a^{(k)}) \tilde{g}_i(a^{(k)}) = \tilde{p}^T(a^{(k)}) \tilde{g}(a^{(k)})
\] (22)

where
\[
\tilde{p}^T(a^{(k)}) = (\tilde{p}_1(a^{(k)}), \tilde{p}_2(a^{(k)}), \ldots, \tilde{p}_m(a^{(k)}))
\] (23)
\[
\tilde{g}(a^{(k)}) = (\tilde{g}_1(a^{(k)}), \tilde{g}_2(a^{(k)}), \ldots, \tilde{g}_m(a^{(k)}))^T
\] (24)

\( m \) denotes the total number of \( \tilde{p}_i(a^{(k)}) \) other than constant 1. \( \tilde{g}_i(a^{(k)}) \) are undetermined coefficients.

Using the weighted least squares method to determine the expression for \( \tilde{g}(a^{(k)}) \), we define
\[
J = \sum_{l=1}^{n} \omega(l^{(k)} - a_i^{(k)}) \left[ \tilde{n}^{h}(a_i^{(k)}) - \tilde{u}^{(k)}(a_i^{(k)}) \right]^2,
\] (25)

where weight function \( \omega(l^{(k)} - a_i^{(k)}) \) is arbitrary and non-singular. Then, let the partial derivative of \( J \) with respect to \( \tilde{g}_i(a^{(k)}) \) be zero, i.e.,
\[
\frac{\partial J}{\partial \tilde{g}_i(a^{(k)})} = 0,
\] (26)

we can obtain the expression for \( \tilde{g}(a^{(k)}) \) as
\[
\tilde{g}(a^{(k)}) = S^{-1}(a^{(k)}) T(a^{(k)}) u,
\] (27)

where
\[
S(a^{(k)}) = P^T(a^{(k)}) W(a^{(k)}) P(a^{(k)}),
\] (28)
\[
T(a^{(k)}) = P^T(a^{(k)}) W(a^{(k)}) (E - \tilde{V}(a^{(k)})),
\] (29)
\[
u = (u(a_1^{(k)}), u(a_2^{(k)}), \ldots, u(a_n^{(k)}))^T,
\] (30)
\[
P(a^{(k)}) = \begin{bmatrix}
\tilde{p}_1(a^{(k)}) & \tilde{p}_2(a^{(k)}) & \cdots & \tilde{p}_m(a^{(k)}) \\
\tilde{p}_1(a_2^{(k)}) & \tilde{p}_2(a_2^{(k)}) & \cdots & \tilde{p}_m(a_2^{(k)}) \\
\vdots & \vdots & \ddots & \vdots \\
\tilde{p}_1(a_n^{(k)}) & \tilde{p}_2(a_n^{(k)}) & \cdots & \tilde{p}_m(a_n^{(k)})
\end{bmatrix},
\] (31)
\[
W(a^{(k)}) = \begin{bmatrix}
\omega(a^{(k)} - a_1^{(k)}) & 0 & \cdots & 0 \\
0 & \omega(a^{(k)} - a_2^{(k)}) & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & \omega(a^{(k)} - a_n^{(k)})
\end{bmatrix},
\] (32)
The following approximation of \( u(k) \) can be obtained according to the combination of Equations (21), (22) and (27).

\[
\begin{align*}
\n^h u(k) = \tilde{v}(k)^T u + \tilde{p}^T(k) S^{-1}(k) I(k) u,
\end{align*}
\]

with

\[
\tilde{v}(k)^T = (\tilde{v}(k, a_1), \tilde{v}(k, a_2), \ldots, \tilde{v}(k, a_n)).
\]

The following formula can also express function \( u(k) \) in Equation (34).

\[
\begin{align*}
\n^h u(k) = \Phi(k) u = \sum_{i=1}^{n} \Phi_i(k) u(a_i(k)),
\end{align*}
\]

where \( \Phi(k) \) is the vector constructed by shape functions. By Equations (34) and (36), we can obtain the following expression for \( \Phi(k) \).

\[
\Phi(k) = (\Phi_1(k), \ldots, \Phi_n(k)) = \tilde{v}(k)^T + \tilde{p}^T(k) S^{-1}(k) I(k).
\]

Compared with singular weight functions given by the fixed format in the IMLS method, a variety of weight functions can be tried for different problems in the HIMLS method. Moreover, difficulties created by singular weight functions, such as truncation error and calculation inconvenience, can be overcome.

The weight function in the calculation process of this paper was chosen as follows.

\[
\omega(r) = \begin{cases} 
\frac{3}{2} - 4r^2 + 4r^3, & r \leq \frac{1}{2}, \\
\frac{4}{3} - 4r + 4r^2 - \frac{4}{3}r^3, & \frac{1}{2} < r \leq 1,
0, & r > 1.
\end{cases}
\]

where

\[
r = \frac{d_{ij}}{\rho_l} = \frac{|a_{i}(k) - a_{j}(k)|}{\rho_l},
\]

\[
\rho_l = c_l \times d_{max}, c_l \text{ denotes the distance between } a_{i}(k) \text{ and the point closest to it. } d_{max} \text{ is a custom scaling parameter that determines the size of the influence domain.}
\]

### 3.2. The HIM Method for 3D Advection–Diffusion Equations

According to Equation (36), \( \nabla u, \frac{\partial u}{\partial t}, \frac{\partial u}{\partial x}, \text{ and } \frac{\partial^2 u}{\partial x^2} \) are written as

\[
\nabla u = \nabla \Phi(k) u = \sum_{i=1}^{n} \nabla \Phi_i(k) u(a_i(k)) = B(k) u,
\]

\[
\frac{\partial u}{\partial t} = \frac{\partial}{\partial t} \sum_{i=1}^{n} \Phi_i(k) u(a_i(k)) = \Phi(k) \cdot u,
\]

\[
\frac{\partial u}{\partial x} = \frac{\partial}{\partial x} \sum_{i=1}^{n} \Phi_i(k) u(a_i(k)) = \Phi(k) \cdot u',
\]

\[
\frac{\partial u}{\partial x^2} = \frac{\partial}{\partial x^2} \sum_{i=1}^{n} \Phi_i(k) u(a_i(k)) = \Phi(k) \cdot u''.
\]
where
\[
\begin{align*}
\mathbf{u} &= \left( \frac{\partial u(a_1^{(k)})}{\partial t}, \ldots, \frac{\partial u(a_n^{(k)})}{\partial t} \right)^T, \\
\mathbf{u}' &= \left( \frac{\partial u(a_1^{(k)})}{\partial x}, \ldots, \frac{\partial u(a_n^{(k)})}{\partial x} \right)^T, \\
\mathbf{u}'' &= \left( \frac{\partial^2 u(a_1^{(k)})}{\partial x^2}, \ldots, \frac{\partial^2 u(a_n^{(k)})}{\partial x^2} \right)^T, \\
\end{align*}
\]

Substitute (36) and (40)–(43) into (15), then we obtain
\[
\delta \int_{\Omega} \left[ \mathbf{B}^{(k)}(\mathbf{a}^{(k)}) \right]^T \mathbf{u} d\Omega = \left[ \Phi_{1,J}(\mathbf{a}^{(k)}) \right] \mathbf{u} d\Omega + \left[ \Phi_{1,J}(\mathbf{a}^{(k)}) \right] \mathbf{u} d\Omega.
\]

Substitute (36) and (40)–(43) into (15), then we obtain
\[
\begin{align*}
\int_{\Omega} \delta \left[ \Phi(\mathbf{a}^{(k)}) \right]^T \mathbf{u} d\Omega &= \delta u^T \cdot \mathcal{C} \cdot \mathbf{u}, \\
\int_{\Omega} \delta \left[ \Phi(\mathbf{a}^{(k)}) \right]^T \left[ \mathcal{T} + \beta_x \Phi(\mathbf{a}^{(k)}) \right] \mathbf{u}' + \gamma_x \Phi(\mathbf{a}^{(k)}) \mathbf{u}' d\Omega &= \delta u^T \cdot \mathcal{T} \cdot \mathbf{u}', \\
\int_{\Omega} \delta \left[ \Phi(\mathbf{a}^{(k)}) \right]^T \gamma_y \frac{\partial}{\partial y} \left[ \Phi(\mathbf{a}^{(k)}) \right] d\Omega &= \delta u^T \cdot \mathcal{G}^{(1)} \cdot \mathbf{u}, \\
\int_{\Omega} \delta \left[ \Phi(\mathbf{a}^{(k)}) \right]^T \gamma_z \frac{\partial}{\partial z} \left[ \Phi(\mathbf{a}^{(k)}) \right] d\Omega &= \delta u^T \cdot \mathcal{G}^{(2)} \cdot \mathbf{u}, \\
\int_{\Gamma} \delta \left[ \Phi(\mathbf{a}^{(k)}) \right]^T \mathbf{v} d\Gamma &= \delta u^T \cdot \mathcal{H} \cdot \mathbf{u},
\end{align*}
\]

where
\[
\mathcal{C} = \int_{\Omega} \left[ \Phi^T(\mathbf{a}^{(k)}) \right] \mathbf{B}(\mathbf{a}^{(k)}) d\Omega, \\
\mathcal{T}^{(1)} = \int_{\Omega} \left[ \Phi^T(\mathbf{a}^{(k)}) \right] \mathcal{B} d\Omega, \\
\mathcal{G}^{(1)} = \int_{\Omega} \left[ \Phi^T(\mathbf{a}^{(k)}) \right] \gamma_y \frac{\partial}{\partial y} \Phi(\mathbf{a}^{(k)}) d\Omega, \\
\mathcal{G}^{(2)} = \int_{\Omega} \left[ \Phi^T(\mathbf{a}^{(k)}) \right] \gamma_z \frac{\partial}{\partial z} \Phi(\mathbf{a}^{(k)}) d\Omega, \\
\mathcal{H} = \int_{\Gamma} \left[ \Phi^T(\mathbf{a}^{(k)}) \right] \mathbf{B}(\mathbf{a}^{(k)}) d\Gamma.
\]

Substitute (48)–(53) into (47), then we obtain
\[
\delta u^T \left[ \mathcal{C} \mathbf{u} - \mathcal{T}^{(1)} \right] - \beta_x \mathcal{C} \mathbf{u}' + \gamma_x \mathcal{C} \mathbf{u}' + G^{(1)} \mathbf{u} + G^{(2)} \mathbf{u} - \mathcal{H} \mathbf{u} = 0.
\]

Because \( \delta u^T \) is arbitrary, we obtain
\[
\mathcal{C} \mathbf{u} - \beta_x \mathcal{C} \mathbf{u}' + \gamma_x \mathcal{C} \mathbf{u}' + \mathbf{K} \mathbf{u} = \mathcal{T},
\]
where
\[ \mathbf{F} = \tilde{f}^{(1)} + \tilde{f}^{(2)}, \quad \mathbf{K} = G^{(1)} + G^{(2)} + H. \] (60)

To obtain the discrete form of (59), the approximation of \( u \) on planes \( x = x^{(k)}, (k = 0, 1, \ldots, L) \) when the temporal interval takes \( \Delta t \) is denoted as
\[ u(x^{(k)}, t) = u^{(k)}(t). \] (61)

Utilizing the FDM, the approximate values of \( u, u' \) and \( u'' \) on intermediate planes \( x = x^{(k)}, (k = 1, 2, \ldots, L - 1) \) are written as
\[ u^{(k)} \approx \frac{u^{(k)}(t) - u^{(k)}(t - \Delta t)}{\Delta t}, \quad u' \approx \frac{u^{(k)}(t) - u^{(k-1)}(t)}{\Delta x}, \quad u'' \approx \frac{u^{(k)}(t) - 2u^{(k)}(t) + u^{(k+1)}(t)}{(\Delta x)^2}. \] (62)

From (62), Equation (59) is expressed as
\[ \begin{align*}
& \mathcal{C} \frac{u^{(1)}(t) - u^{(0)}(t)}{\Delta t} - \beta \gamma \mathcal{C} \frac{u^{(0)}(t) - 2u^{(1)}(t) + u^{(2)}(t)}{(\Delta x)^2} \\
& + \mathcal{K} u^{(1)}(t) = f^{(1)}(t + \Delta t), \\
& \mathcal{C} \frac{u^{(2)}(t) - u^{(1)}(t)}{\Delta t} - \beta \gamma \mathcal{C} \frac{u^{(1)}(t) - 2u^{(2)}(t) + u^{(3)}(t)}{(\Delta x)^2} \\
& + \mathcal{K} u^{(2)}(t) = f^{(2)}(t + \Delta t), \\
& \vdots \\
& \mathcal{C} \frac{u^{(L-2)}(t) - u^{(L-3)}(t)}{\Delta t} - \beta \gamma \mathcal{C} \frac{u^{(L-3)}(t) - 2u^{(L-2)}(t) + u^{(L-1)}(t)}{(\Delta x)^2} \\
& + \mathcal{K} u^{(L-2)}(t) = f^{(L-2)}(t + \Delta t), \\
& \mathcal{C} \frac{u^{(L-1)}(t) - u^{(L-2)}(t)}{\Delta t} - \beta \gamma \mathcal{C} \frac{u^{(L-2)}(t) - 2u^{(L-1)}(t) + u^{(L)}(t)}{(\Delta x)^2} \\
& + \mathcal{K} u^{(L-1)}(t) = f^{(L-1)}(t + \Delta t).
\end{align*} \] (63–66)

The matrix form of Equations (63–66) is
\[ \begin{bmatrix}
Q & M & D \\
M & Q & M \\
D & M & Q
\end{bmatrix}
\begin{bmatrix}
\bar{u}^{(1)}(t + \Delta t) \\
\bar{u}^{(2)}(t + \Delta t) \\
\vdots
\end{bmatrix}
= \begin{bmatrix}
-\bar{D} u^{(1)}(t) + \bar{N} u^{(1)}(t) + \bar{F}^{(1)}(t + \Delta t) \\
\vdots \\
\bar{D} u^{(L-1)}(t) + \bar{N} u^{(L-1)}(t) + \bar{F}^{(L-1)}(t + \Delta t)
\end{bmatrix}, \] (67)

where
\[ Q = 2\beta \gamma \mathcal{C} \frac{(\Delta t)^2}{(\Delta x)^2}, \quad \mathcal{C} \frac{\Delta t}{\Delta x}, \quad \bar{M} = -\beta \gamma \mathcal{C} \frac{(\Delta t)^2}{(\Delta x)^2}, \quad \bar{D} = -\beta \gamma \mathcal{C} \frac{(\Delta t)^2}{(\Delta x)^2} - \beta \gamma \mathcal{C} \frac{\Delta t}{\Delta x}, \quad N = \mathcal{C} \frac{\Delta t}{\Delta t}. \] (68)
Let
\[
\begin{bmatrix}
Q & M \\
D & Q & M \\
& & \ddots & \ddots & \ddots \\
& & & D & Q & M \\
& & & & & \ddots & \ddots & \ddots \\
& & & & & & \ddots & D & Q & M \\
\end{bmatrix}, \quad \begin{bmatrix}
u^{(1)}_{(t+\Delta t)} \\
u^{(2)}_{(t+\Delta t)} \\
\vdots \\
u^{(L-2)}_{(t+\Delta t)} \\
u^{(L-1)}_{(t+\Delta t)} \\
\end{bmatrix}, \quad \begin{bmatrix}
- \mathbf{D} \mathbf{u}^{(0)}_{(t+\Delta t)} + \mathbf{N}^{(1)}_{(t)} + \mathbf{T}^{(1)}_{(t+\Delta t)} \\
- \mathbf{M}^{(2)}_{(t)} + \mathbf{T}^{(2)}_{(t+\Delta t)} \\
\vdots \\
- \mathbf{M}^{(L-2)}_{(t)} + \mathbf{T}^{(L-2)}_{(t+\Delta t)} \\
- \mathbf{M}^{(L-1)}_{(t)} + \mathbf{T}^{(L-1)}_{(t+\Delta t)} \\
\end{bmatrix}.
\]
then Equation (67) becomes
\[
\bar{E} \mathbf{R} = \mathbf{\bar{Z}}.
\]

The method of setting diagonal elements in the coefficient matrix to one was chosen to impose Dirichlet boundary conditions.

For a discrete system of equations as
\[
\begin{bmatrix}
a_{11} & a_{12} & \cdots & a_{1r} & \cdots & a_{1n} \\
a_{21} & a_{22} & \cdots & a_{2r} & \cdots & a_{2n} \\
\vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\
a_{r1} & a_{r2} & \cdots & a_{rr} & \cdots & a_{rn} \\
\vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\
a_{n1} & a_{n2} & \cdots & a_{nr} & \cdots & a_{nn} \\
\end{bmatrix}
\begin{bmatrix}
u_1 \\
u_2 \\
\vdots \\
u_r \\
\vdots \\
u_n \\
\end{bmatrix}
\begin{bmatrix}
u_1 \\
u_2 \\
\vdots \\
u_r \\
\vdots \\
u_n \\
\end{bmatrix}.
\]
if the unsolved displacement \(u_r\) in row \(r\) lies on the boundary \(\Gamma_u\), it means \(u_r\) can be known by the essential boundary condition \(u = \bar{u}\). Replace the principal diagonal element \(a_{rr}\) of corresponding coefficient matrix with 1. Change the other elements and corresponding right-hand vector elements in row \(r\) to 0 and \(\pi_r\), respectively. Then, (71) becomes
\[
\begin{bmatrix}
a_{11} & a_{12} & \cdots & a_{1r} & \cdots & a_{1n} \\
a_{21} & a_{22} & \cdots & a_{2r} & \cdots & a_{2n} \\
\vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\
a_{r1} & a_{r2} & \cdots & a_{rr} & \cdots & a_{rn} \\
\vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\
a_{n1} & a_{n2} & \cdots & a_{nr} & \cdots & a_{nn} \\
\end{bmatrix}
\begin{bmatrix}
u_1 \\
u_2 \\
\vdots \\
u_r \\
\vdots \\
u_n \\
\end{bmatrix}
\begin{bmatrix}
u_1 \\
u_2 \\
\vdots \\
u_r \\
\vdots \\
u_n \\
\end{bmatrix}.
\]
Considering the row \(r\) of (72), we can obtain \(u_r = \pi_r\).
In particular, if the essential boundary condition is \(u = \bar{u} = 0\), the elements in column \(r\) other than the main diagonal element should also be changed to 0, i.e.,
\[
\begin{bmatrix}
a_{11} & a_{12} & \cdots & 0 & \cdots & a_{1n} \\
a_{21} & a_{22} & \cdots & 0 & \cdots & a_{2n} \\
\vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & 1 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\
a_{n1} & a_{n2} & \cdots & 0 & \cdots & a_{nn} \\
\end{bmatrix}
\begin{bmatrix}
u_1 \\
u_2 \\
\vdots \\
u_r \\
\vdots \\
u_n \\
\end{bmatrix}.
\]
Considering the row \(i, (i = 1, 2, \cdots, n, i \neq r)\) of (73), we can obtain
\[
a_{i1}u_1 + a_{i2}u_2 + \cdots + a_{i(r-1)}u_{r-1} + a_{i(r+1)}u_{r+1} + \cdots + a_{in}u_n = v_i.
\]
Solving Equation (70), we can acquire solutions on planes \( x = x^{(k)} \), \( (k = 0, 1, \cdots, L) \) at time \( \Delta t \). Using the linear interpolation method, we can obtain the approximate value of \( u(\alpha, t) \) at any time \( t \). Hence, \( u(x, y, z, t) \) in the 3D solving domain \( \Omega \) is expressed as
\[
u(x, y, z, t) = \frac{x - x^{(k)}}{\Delta x} u(x^{(k)} - 1, y, z, t) + \frac{x^{(k+1)} - x}{\Delta x} u(x^{(k)}, y, z, t). \tag{75}
\]

4. Numerical Examples

To demonstrate the advantages of the HIM method, some specific examples were selected. Each problem was solved using three methods: the DSEFG, HIM, and IEFG methods. The calculated results were analyzed with different time steps, scaling parameters of influence domain, splitting directions, node distributions and time interval. Moreover, a 3D Richards’ equation was chosen to verify the application of the HIM method in the nonlinear problem.

In this section, \( 4 \times 4 \) points Gaussian integration is adopted to obtain numerical integration on the background grids. The calculation program is implemented using MATLAB. For comparison, the relative error \( \delta \) is an important indicator to describe the accuracy. \( \delta \) is defined by
\[
\delta = \frac{\tilde{\delta}}{\|u\|_{L^2(\Omega)}}, \quad \tilde{\delta} = \left( \int_{\Omega} \left( u - u^k \right)^2 \,d\Omega \right)^{1/2}.
\tag{76}
\]

4.1. An Advection–Diffusion Equation with the Isotropic Diffusion

The governing equation is
\[
\frac{\partial u}{\partial t} = \frac{1}{2} \left( \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} \right) - \frac{1}{2} \left( \frac{\partial u}{\partial x} + \frac{\partial u}{\partial y} + \frac{\partial u}{\partial z} \right),
\tag{77}
\]
\( \alpha \in \Omega, \Omega = [0, 1] \times [0, 1] \times [0, 1], t \in [0, t_0]. \)

The boundary conditions are
\[
u(0, y, z, t) = (1 + e^{-y} + e^{-z})e^t, \quad u(1, y, z, t) = (e^{-1} + e^{-y} + e^{-z})e^t,
\tag{78}
\]
\[
u(x, 0, z, t) = (e^{-x} + 1 + e^{-z})e^t, \quad u(x, 1, z, t) = (e^{-x} + e^{-1} + e^{-z})e^t,
\tag{79}
\]
\[
u(x, y, 0, t) = (e^{-x} + e^{-y} + 1)e^t, \quad u(x, y, 1, t) = (e^{-x} + e^{-y} + e^{-1})e^t.
\tag{80}
\]

The initial condition is
\[
u(x, y, z, 0) = e^{-x} + e^{-y} + e^{-z}.
\tag{81}
\]

The expression of the corresponding exact solution is given by
\[
u(\alpha, t) = (e^{-x} + e^{-y} + e^{-z})e^t.
\tag{82}
\]
Solving this problem using the HIM method, the splitting direction is \( z \), \( d_{\text{max}} = 1.35 \), \( t_0 = 0.1 \text{ s}, \Delta t = 0.001 \text{ s} \). The node distribution takes \( 11 \times 11 \times 11 \). Finally, the relative error \( \delta = 1.9780 \times 10^{-4} \). Computing time takes 2.90 s.

Solving this problem using the DSEFG method, the splitting direction is \( z \), \( d_{\text{max}} = 1.3 \), \( t_0 = 0.1 \text{ s}, \Delta t = 0.001 \text{ s} \). The distribution of discrete nodes takes \( 11 \times 11 \times 11 \), penalty factor \( \alpha = 1 \times 10^8 \). Finally, the relative error \( \delta = 0.0015 \). Computing time takes 9.28 s.

Solving this problem by the IEFG method, \( d_{\text{max}} = 1.5, t_0 = 0.1 \text{ s}, \Delta t = 0.001 \text{ s} \). Penalty factor \( \alpha = 8.7 \times 10^6 \). The distribution of discrete nodes takes \( 11 \times 11 \times 11 \). Finally, the relative error \( \delta = 0.0146 \). Computing time takes 320.82 s.
Comparing data given by these three methods, the HIM method is about three-times faster than the DSEFG method. Additionally, it is about 100-times faster than the IEFG method. Moreover, the precision of the HIM method is one order of magnitude higher than that of the DSEFG method. Additionally, it is two orders of magnitude higher than that of the IEFG method.

Table 1 presents a comparison of relative error and computing time when the three methods take different scaling parameters of influence domain \( d_{\text{max}} \). All methods have the same \( \Delta t = 0.001 \) s, \( t_0 = 0.1 \) s, and node distribution \( 11 \times 11 \times 11 \). We can see that it is necessary to find appropriate \( d_{\text{max}} \) for different methods. To obtain the high precision results under current parameters, we let \( d_{\text{max}} = 1.35 \) in the HIM method, while the IEFG method takes \( d_{\text{max}} = 1.5 \) and the DSEFG method takes \( d_{\text{max}} = 1.3 \).

Table 1. Comparison of relative error and computing time when \( d_{\text{max}} \) is different.

| \( d_{\text{max}} \) | HIM \( \delta \) | DSEFG \( \delta \) | IEFG \( \delta \) | Computing Time (s) |
|---------------------|-----------------|-----------------|------------------|------------------|
| 1.3                 | 2.0380 \( \times 10^{-4} \) | 0.0015          | 0.0202           | 2.83             | 9.28             | 258.88          |
| 1.35                | 1.9780 \( \times 10^{-4} \) | 0.0016          | 0.0196           | 2.90             | 9.28             | 242.98          |
| 1.4                 | 2.0056 \( \times 10^{-4} \) | 0.0035          | 0.0178           | 2.74             | 9.52             | 262.31          |
| 1.45                | 2.0175 \( \times 10^{-4} \) | 0.0064          | 0.0151           | 2.80             | 9.27             | 263.31          |
| 1.5                 | 2.0251 \( \times 10^{-4} \) | 0.0102          | 0.0146           | 2.89             | 9.24             | 302.80          |

Table 2 shows the comparison of relative error and computing time when all three methods choose different \( \Delta t \). The distribution of discrete nodes takes \( 11 \times 11 \times 11 \), \( t_0 = 0.1 \) s, \( d_{\text{max}} = 1.3 \). Compared with the other two methods, the HIM method extremely improves computing speed and accuracy. We can also draw a conclusion that the HIM method is convergent with \( \Delta t \).

Table 2. Comparison of relative error and computing time when \( \Delta t \) is different.

| \( \Delta t \) (s) | HIM \( \delta \) | DSEFG \( \delta \) | IEFG \( \delta \) | Computing Time (s) |
|-------------------|-----------------|-----------------|------------------|------------------|
| 0.01              | 0.0024          | 0.0021          | 0.0149           | 1.01             | 1.61             | 49.21           |
| 0.005             | 0.0011          | 0.0016          | 0.0173           | 1.28             | 2.59             | 74.29           |
| 0.001             | 2.0380 \( \times 10^{-4} \) | 0.0015          | 0.0202           | 2.83             | 9.28             | 258.88          |
| 0.0005            | 2.0251 \( \times 10^{-4} \) | 0.0015          | 0.0206           | 4.60             | 20.48            | 496.89          |

Figures 1–3 are the comparison of calculation results along axis \( x, y \) and \( z \) when the IEFG method, HIM method and DSEFG method take different \( t_0 \). The node distribution takes \( 11 \times 11 \times 11 \), \( \Delta t = 0.001 \) s. The figures indicate that the calculation results of the above methods agree with analytical solutions well.

Figure 4 provides a 3D diagram of the calculation results given by the HIM method on layers \( z = k/10 \), \( (k = 0, 1, \ldots, 10) \). \( \Delta t = 0.001 \) s, the node distribution takes \( 11 \times 11 \times 11 \), \( t_0 = 0.1 \) s.
Table 2. Comparison of relative error and computing time when $t_{\Delta}$ is different.

| $t_{\Delta}$ (s) | Relative Error $\delta$ | Computing Time (s) |
|------------------|--------------------------|---------------------|
| 0.01             | 0.0024                   | 1.01 1.61           |
| 0.005            | 0.0011                   | 1.28 2.59           |
| 0.001            | 0.0015 42.0380 $10^{-3}$ | 0.0202 2.83 9.28    |
| 0.0005           | 0.0015 42.0251 $10^{-3}$ | 0.0206 4.60 20.48   |

Figures 1–3 are the comparison of calculation results along axis $x$, $y$ and $z$ when the IEFG method, HIM method and DSEFG method take different $t_{0}$. The node distribution takes $11 \times 11 \times 11$, $0.001 \text{s} = t_{\Delta}$. The figures indicate that the calculation results of the above methods agree with analytical solutions well.

Figure 4 provides a 3D diagram of the calculation results given by the HIM method on layers $z_{k} = 0, 1, \ldots, 10$, $0.001 \text{s} = t_{\Delta}$, the node distribution takes $11 \times 11 \times 11$, $0.1 \text{s} = t_{0}$.

4.2. An Advection–Diffusion Equation with the Source Term

The governing equation is

$$
\begin{align*}
\frac{\partial^2 (u_{x})}{\partial x^2} + \frac{\partial^2 (u_{y})}{\partial y^2} + \frac{\partial^2 (u_{z})}{\partial z^2} + \frac{\partial(u_{x})}{\partial t} &= f(x, y, z, t), \\
\end{align*}
$$

where $\Omega = [0,1] \times [0,1] \times [0,1]$.

The boundary conditions are

$$
\begin{align*}
u &= 0, & x = 0, 1, & \forall y, z, & t > 0, \\
v &= 0, & y = 0, 1, & \forall x, z, & t > 0, \\
u &= 0, & z = 0, 1, & \forall x, y, & t > 0, \\
u &= 0, & x = 0, 1, & y = 0, 1, & t = 0.
\end{align*}
$$

The initial condition is

$$
\begin{align*}
u(x, y, z, 0) &= \phi(x, y, z),
\end{align*}
$$

where $\phi(x, y, z)$ is a given function.

The expression of the corresponding exact solution is given by $u(x, y, z, t) = \phi(x, y, z)$.
The data in Table 3 show that the HIM method takes less running time and has higher precision than the other two methods.

4.2. An Advection–Diffusion Equation with the Source Term

The governing equation is

\[
\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} - \frac{\partial u}{\partial x} - \frac{\partial u}{\partial y} - e^{x+y-z-t},
\]

\(\alpha \in \Omega, \Omega = [0,1] \times [0,1] \times [0,1], t \in [0,t_0].\) (83)

The boundary conditions are

\[
\begin{align*}
u(0,y,z,t) &= e^{y+z-t}, \\
u(1,y,z,t) &= e^{1+y+z-t}, \\
u(x,0,z,t) &= e^{x+z-t}, \\
u(x,1,z,t) &= e^{x+1+z-t}, \\
u(x,y,0,t) &= e^{x+y-t}, \\
u(x,y,1,t) &= e^{x+y+1-t}.
\end{align*}
\]

The initial condition is

\[
u(x,y,z,0) = e^{x+y+z}.
\] (87)

The expression of the corresponding exact solution is given by

\[
u(\alpha, t) = e^{x+y+z-t}.
\] (88)

Solving this problem using the HIM method, the splitting direction is \(z, d_{\text{max}} = 1.4, \Delta t = 0.001 s, t_0 = 0.1 s.\) The distribution of discrete nodes takes \(11 \times 11 \times 11.\) Finally, the relative error \(\delta = 6.0322 \times 10^{-4}.\) Computing time takes 20.27 s.

Solving this problem by the DSEFG method, the splitting direction is \(z, d_{\text{max}} = 1.3, \Delta t = 0.001 s, t_0 = 0.1 s.\) Penalty factor \(\alpha = 1 \times 10^6.\) The distribution of discrete nodes takes \(11 \times 11 \times 11.\) Finally, the relative error \(\delta = 7.0654 \times 10^{-4}.\) Computing time takes 24.81 s.

Solving this problem using the IEFG method, the distribution of discrete nodes takes \(11 \times 11 \times 11.\) \(d_{\text{max}} = 1.22, \Delta t = 0.001 s, t_0 = 0.1 s.\) Penalty factor \(\alpha = 8.4 \times 10^7.\) Finally, the relative error \(\delta = 0.0068.\) Computing time takes 429.75 s.

Compared to the results of all three methods, the computing speed and precision of the HIM method are obviously improved.

Table 3 is the comparison of relative error and computing time when the above three methods take different \(t_0\) with the distribution of nodes \(11 \times 11 \times 11.\) Time step \(\Delta t = 0.001 s.\) The data in Table 3 show that the HIM method takes less running time and has higher precision than the other two methods.
Table 3. Comparison of relative error and computing time when \( t_0 \) is different.

| \( t_0 \) (s) | Relative Error \( \delta \) | Computing Time (s) |
|--------------|--------------------------|------------------|
|              | HIM          | DSEFG       | IEFG | HIM | DSEFG | IEFG |   |
| 0.01         | 2.6113 \times 10^{-4} | 5.4060 \times 10^{-4} | 0.0052 | 3.02 | 3.16 | 70.59 |
| 0.05         | 5.1110 \times 10^{-4} | 6.5775 \times 10^{-4} | 0.0050 | 9.92 | 12.76 | 229.67 |
| 0.1          | 6.0322 \times 10^{-4} | 7.0654 \times 10^{-4} | 0.0068 | 20.27 | 24.81 | 429.75 |
| 0.3          | 6.3098 \times 10^{-4} | 7.2198 \times 10^{-4} | 0.0081 | 56.57 | 76.54 | 1416.50 |
| 0.5          | 6.3106 \times 10^{-4} | 7.2202 \times 10^{-4} | 0.0081 | 89.88 | 120.68 | 2081.20 |

Table 4 shows the comparison of relative error and computing time when the above three methods take different node distributions. The HIM and DSEFG methods choose different splitting layers, and the IEFG method distributes different nodes along direction \( z \). All methods have the same \( \Delta t = 0.001 \) s, \( t_0 = 0.05 \) s. From the data, when there are a few nodes distributed in the discrete area, the HIM method still has high accuracy while maintaining fast calculation speed. With the increase in discrete nodes, the calculation accuracy of all three methods is improved gradually. Compared with the other two methods, the HIM method improves calculation precision and reduces running time. Moreover, the HIM method is convergent with the number of splitting layers.

Table 4. Comparison of relative error and computing time when node distribution is different.

| Node Distribution | Relative Error \( \delta \) | Computing Time (s) |
|-------------------|--------------------------|------------------|
|                   | HIM          | DSEFG       | IEFG | HIM | DSEFG | IEFG |   |
| 11 \times 11 \times 7 | 8.7684 \times 10^{-4} | 8.8917 \times 10^{-4} | 0.0563 | 6.98 | 7.15 | 144.24 |
| 11 \times 11 \times 9 | 6.5045 \times 10^{-4} | 7.3120 \times 10^{-4} | 0.0154 | 8.24 | 10.87 | 201.41 |
| 11 \times 11 \times 11 | 5.1110 \times 10^{-4} | 6.5775 \times 10^{-4} | 0.0050 | 9.92 | 12.76 | 229.67 |
| 11 \times 11 \times 13 | 4.2114 \times 10^{-4} | 6.2541 \times 10^{-4} | 0.0068 | 12.11 | 15.40 | 300.01 |

Figures 5–7 show the calculation results along axis \( x, \) \( y, \) \( z \) when the node distribution of these three methods takes \( 11 \times 11 \times 11, \) \( \Delta t = 0.001 \) s. It demonstrates that the calculation results given by all methods fit analytical solutions quite well according to the figures.

Figure 8 provides a 3D diagram of the calculation results on planes \( z = k/10, \) \( (k = 0, 1, \ldots, 10) \) given by the HIM method with \( d_{\text{max}} = 1.4, \) \( 11 \times 11 \times 11 \) nodes, \( \Delta t = 0.001 \) s, \( t_0 = 0.1 \) s.
4.3. An Advection–Diffusion Equation with Dirichlet Boundary Conditions

The equation is

$$\frac{\partial u}{\partial t} = \beta_x \frac{\partial^2 u}{\partial x^2} + \beta_y \frac{\partial^2 u}{\partial y^2} + \beta_z \frac{\partial^2 u}{\partial z^2} - \gamma_x \frac{\partial u}{\partial x} - \gamma_y \frac{\partial u}{\partial y} - \gamma_z \frac{\partial u}{\partial z},$$  

(89)

The boundary conditions are

$$u(0, y, z, t) = \lambda e^{t(1 + e^{-ax} + e^{-ay})}, \quad u(1, y, z, t) = \lambda e^{t(1 + e^{-ax} + e^{-ay})},$$  

(90)

Figure 6. Calculation results along y-axis when $t_0$ is different.

Figure 7. Calculation results along z-axis when $t_0$ is different.

Figure 8. Calculation results on layers $z = k/10$, ($k = 0, 1, \ldots, 10$) given by the HIM method.
where
\[ u(x, 0, z, t) = \lambda e^{\tau t}(e^{-\alpha_1 x} + 1 + e^{-\alpha_3 z}), \quad u(x, 1, z, t) = \lambda e^{\tau t}(e^{-\alpha_1 x} + e^{-\alpha_2 z} + e^{-\alpha_3 z}), \]
\[ u(x, y, 0, t) = \lambda e^{\tau t}(e^{-\alpha_1 y} + e^{-\alpha_2 y} + 1), \quad u(x, y, 1, t) = \lambda e^{\tau t}(e^{-\alpha_1 y} + e^{-\alpha_2 y} + e^{-\alpha_3 y}). \]

The initial condition is
\[ u(x, y, z, 0) = \lambda(e^{-\alpha_1 x} + e^{-\alpha_2 y} + e^{-\alpha_3 z}). \]

The expression of the corresponding exact solution is given by
\[ u(\alpha, t) = \lambda e^{\tau t}(e^{-\alpha_1 x} + e^{-\alpha_2 y} + e^{-\alpha_3 z}), \]
where
\[ a_1 = -\gamma_x - \sqrt{\frac{\gamma_x^2}{2\beta_x} + 4\tau \beta_x}, \quad a_2 = \frac{-\gamma_y - \sqrt{\gamma_y^2 + 4\tau \beta_y}}{2\beta_y}, \quad a_3 = \frac{-\gamma_z - \sqrt{\gamma_z^2 + 4\tau \beta_z}}{2\beta_z}. \]

For the sake of simplicity, parameters in the above equations are \( \beta_x = 1.4, \beta_y = 1.7, \beta_z = 1.5, \gamma_x = \gamma_y = \gamma_z = 1, \lambda = 1, \tau = -0.1. \)

Solving this problem by the IEFG method, \( d_{\text{max}} = 1.8, \Delta t = 0.01 \text{ s}, t_0 = 0.1 \text{ s}. \) Penalty factor \( \alpha = 1.1 \times 10^4. \) The distribution of discrete nodes takes \( 11 \times 11 \times 11. \) Finally, the relative error \( \delta = 0.0012. \) The computing time is 64.46 s.

Solving this problem using the DSEFG method or the HIM method, we discuss three cases under different splitting directions when the scaling parameter of influence domain \( d_{\text{max}} = 1.8, \) the node distribution takes \( 11 \times 11 \times 11, \Delta t = 0.01 \text{ s}, t_0 = 0.1 \text{ s}. \)

Table 5 shows the comparison of relative error and computing time under splitting directions \( x, y \) and \( z \) given by the DSEFG method and HIM method. It demonstrates that these two methods are more accurate when the splitting direction is \( z. \) In addition, the HIM method has less computing time and higher precision than the DSEFG method according to the data.

| Splitting Direction | Relative Error \( \delta \) | Computing Time (s) |
|---------------------|----------------------------|---------------------|
|                     | HIM | DSEFG | HIM | DSEFG |
| \( x \)             | 4.2108 \times 10^{-4} | 4.2258 \times 10^{-4} | 1.83 | 2.11   |
| \( y \)             | 3.1079 \times 10^{-4} | 3.1411 \times 10^{-4} | 1.70 | 2.14   |
| \( z \)             | 2.7551 \times 10^{-4} | 2.7867 \times 10^{-4} | 1.72 | 2.17   |

Table 6 shows the comparison of relative error and computing time at time interval \( t_0 = 1.0 \text{ s} \) when the above three numerical methods take different \( \Delta t. \) With the decrease in time step \( \Delta t, \) the calculation precision of the HIM method and DSEFG method is improved, which indicates that the two methods are convergent with the time step. Furthermore, although the HIM method has slight improvement in accuracy compared with the DSEFG method, it takes much less running time than the other two methods.

| \( \Delta t \) (s) | Relative Error \( \delta \) | Computing Time (s) |
|-------------------|----------------------------|---------------------|
|                   | HIM | DSEFG | IEFG | HIM | DSEFG | IEFG |
| 0.1               | 0.0030 | 0.0030 | 0.0014 | 1.86 | 2.10 | 63.20 |
| 0.01              | 2.7831 \times 10^{-4} | 2.8101 \times 10^{-4} | 0.0014 | 4.21 | 10.36 | 309.28 |
| 0.001             | 1.0506 \times 10^{-4} | 1.1965 \times 10^{-4} | 0.0014 | 29.01 | 91.76 | 2667.7 |
Figures 9–11 are calculation results along axis $x$, $y$ and $z$ when the IEFG method, HIM method and DSEFG method take different $t_0$ with the node distribution $11 \times 11 \times 11$, $\Delta t = 0.01$ s. From the figures, the calculation results of all three methods agree with well with the exact ones.

Figure 12 provides a 3D diagram of the calculation results at $(x, y, k/10, 0.1$ s), $(k = 0, 1, \cdots, 10)$ given by the HIM method. The distribution of discrete nodes is $11 \times 11 \times 11$, $\Delta t = 0.01$ s.
Consider the 3D Richards’ equation

$$\nabla \cdot (K_s \cdot k_r \nabla (h(a, t) - z)) = \frac{\partial \theta}{\partial t}, \ a \in \Omega, \ \Omega = [0,a] \times [0,b] \times [0,L],$$

(96)

where the direction of gravity is $(0,0,1)$ (the vertical coordinate is positive down), the pressure head $h < 0$, the relative permeability $k_r = e^{\lambda h}$, the water content $\theta$ is given by $\theta = \theta_r + (\theta_s - \theta_r)e^{\lambda h}$ with saturated water content parameters $\theta_r$, $\theta_s$ and $\lambda$, $K_s$ is the saturated hydraulic conductivity related to the physical property of the medium. The boundary conditions are

$$h_{|z=0} = \frac{1}{\lambda} \ln \left[ e^{\lambda h} + (1 - e^{\lambda h}) \sin \left( \frac{\pi x}{a} \right) \sin \left( \frac{\pi y}{b} \right) \right], \ t > 0,$$

(97)

$$h_{|x=0} = h_{|x=a} = h_{|y=0} = h_{|y=b} = h_{|z=L} = h_r.$$  

(98)

The initial condition is

$$h(x, y, z, 0) = h_r.$$  

(99)

The expression of corresponding exact solution is given by

$$h(x, t) = \frac{1}{\lambda} \ln \left[ e^{\lambda h_r} + T_1 (T_2 + T_3) \right],$$

(100)

where

$$T_1 = h_0 \sin \left( \frac{\pi x}{a} \right) \sin \left( \frac{\pi y}{b} \right) e^{\frac{1}{\lambda} (L-z)}, \ h_0 = (1 - e^{\lambda h_r}),$$

(101)

$$T_2 = \frac{\sinh(\beta z)}{\sinh(\beta L)}, \ \beta = \sqrt{\frac{\lambda^2}{4} + \frac{\pi^2}{a^2} + \frac{\pi^2}{b^2}},$$

(102)

$$T_3 = \frac{2}{\cosh L} \left( \sum_{k=1}^{\infty} (-1)^k \frac{\lambda_k}{\gamma} \sin(\lambda_k z)e^{-\gamma t}, \ c = \frac{\lambda_0}{\lambda}, \ \lambda_k = \frac{k\pi}{L}, \ \gamma = \frac{\beta^2 + \lambda_k^2}{a}. \right.$$  

(103)

According to [43,44], (96) can be transformed into a linear equation by using the Kirchhoff integral transformation

$$\phi(h) = \int_{-\infty}^{h} k_r(s) ds,$$

(104)

and the Gardner relation

$$\phi(h) = \frac{1}{\lambda} e^{\lambda h} = \frac{1}{\lambda} k_r.$$  

(105)
Substituting (104) and (105) into (96), a linear format with the transform variable \( \phi(t) \) can be expressed as

\[
\nabla \cdot (K \nabla \phi) - \nabla \cdot (\phi \lambda K_z \nabla z) = \lambda (\theta_s - \theta_r) \frac{\partial \phi}{\partial t}
\]

Let \( a = b = L = 15.24 \text{ m}, h_r = -15.24 \text{ m}, K_z = 1 \text{ h/m}, \theta_r = 0.15, \theta_s = 0.45, \lambda = 0.5 \).

Solving this problem by the HIM method, the splitting direction is \( z \), \( \Delta_{\text{max}} = 1.1 \), \( \Delta t = 0.05 \text{ h}, t_0 = 5 \text{ h} \). The distribution of discrete nodes takes \( 11 \times 11 \times 11 \). Finally, the relative error \( \delta = 0.0239 \). Computing time takes 3.19 s.

Table 7 is the comparison of relative error and computing time of the HIM method when \( \Delta t \) is different. It shows that the method is convergent with time step \( \Delta t \) when dealing with the 3D Richards’ equation.

**Table 7.** Relative error and computing time of the HIM method when \( \Delta t \) is different.

| \( \Delta t \) (h) | Relative Error \( \delta \) | Computing Time (s) |
|------------------|--------------------------|-------------------|
| 0.5              | 0.2737                   | 1.00              |
| 0.1              | 0.0328                   | 1.90              |
| 0.05             | 0.0239                   | 3.19              |
| 0.025            | 0.0200                   | 4.18              |

Figure 13 provides a 3D diagram of the calculation results at \((x,y,7.62,5 \text{ h})\) given by the HIM method.

**Figure 13.** Calculation results at \((x,y,7.62,5 \text{ h})\) given by the HIM method.

### 5. Conclusions

A hybrid interpolating meshless method is proposed for solving 3D advection–diffusion equations. To improve computational efficiency, a three-dimensional equation is changed into correlative two-dimensional equations. After the dimension reduction by the DSM, the IIEFG method can be utilized in 2D subdomains. On the basis of the IIMLS method, the approximation function with interpolation property can be formed. For obtaining solvable algebraic equations, FDM is utilized in the time domain and axis after splitting. Setting diagonal elements in the coefficient matrix to one is chosen to impose Dirichlet boundary conditions directly. Then, the discrete system of equations can be solved, and the numerical solutions can be acquired.

To prove the advantages of the HIM method, some advection–diffusion equations are selected and solved by the IIEFG method, HIM method, and DSEFG method. The influence of time steps \( \Delta t \), node distributions, scaling parameters of influence domain \( d_{\text{max}} \), temporal interval \( t_0 \) and splitting directions on the calculation results is discussed and analyzed. Collecting the data and comparing the numerical results obtained by the above
three methods, it can be shown that the HIM method significantly increases the computing speed and precision. Moreover, by solving a 3D Richards’ equation, the effectiveness of the HIM method in the nonlinear problem is verified.

In addition, it can be found that parameters need to be selected in an appropriate range for obtaining numerical solutions with high precision. In order to explain the variation in error and convergence speed when the HIM method takes different parameter settings, further theoretical research on error estimation and stability analysis will be carried out in the future.

Author Contributions: Conceptualization, Z.M. and L.M.; methodology, Z.M. and L.M.; software, X.C.; writing—original draft preparation, X.C.; writing—review and editing, Z.M. and X.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (Grant No. 52004169), Fund Program for the Scientific Activities of Selected Returned Overseas Professionals in Shanxi Province (Grant No. 20220030), and the Research Project Supported by Shanxi Scholarship Council of China (Grant No. 2021-132).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References
1. Lewis, R.W.; Nithiarasu, P.; Seetharamu, K.N. Fundamentals of the Finite Element Method for Heat and Fluid Flow; John Wiley & Sons: New York, NY, USA, 2004.
2. Oñate, E.; Manzán, M. A general procedure for deriving stabilized space-time finite element methods for advective-diffusive problems. Int. J. Numer. Meth. Fluids 2015, 31, 203–221. [CrossRef]
3. Nakshatrala, K.B.; Turner, D.Z.; Hjelmstad, K.D.; Masud, A. A stabilized mixed finite element method for Darcy flow based on a multiscale decomposition of the solution. Comput. Methods Appl. Mech. Eng. 2006, 195, 4036–4049. [CrossRef]
4. Carrer, J.; Cunha, C.; Mansur, W.J. The boundary element method applied to the solution of two-dimensional diffusion-advection problems for non-isotropic materials. J. Braz. Soc. Mech. Sci. 2017, 39, 4533–4545. [CrossRef]
5. Belytschko, T.; Krongauz, Y.; Organ, D.; Fleming, M.; Krysl, P. Meshless methods: An overview and recent developments. Comput. Methods Appl. Mech. Eng. 1996, 139, 3–47. [CrossRef]
6. Vikendra, S.; Siraj-ul-Islam; Mohanty, R.K. Local meshless method for convection dominated steady and unsteady partial differential equations. Eng. Comput.-Ger. 2019, 35, 803–812.
7. Benkhaldoun, F.; Halassi, A.H.; Ouazar, D.; Seaid, M.; Taik, A. A stabilized meshless method for time-dependent convection-dominated flow problems. Math. Comput. Simulat. 2017, 137, 159–176. [CrossRef]
8. Barbieri, E.; Petrinic, N. Three-dimensional crack propagation with distance-based discontinuous kernels in meshfree methods. Comput. Mech. 2014, 53, 325–342. [CrossRef]
9. Wu, Q.; Peng, P.P.; Cheng, Y.M. The interpolating element-free Galerkin method for elastic large deformation problems. Sci. China 2021, 64, 364–374. [CrossRef]
10. Auerbach, R.; Globerson, A. Numerical Solution of Convection-Diffusion Problems; Chapman & Hall: London, UK, 1996.
11. Franca, L.P.; Hauke, G.; Masud, A. Revisiting stabilized finite element methods for the advective-diffusive equation. Comput. Methods Appl. Mech. Eng. 2006, 195, 1560–1572. [CrossRef]
12. Lancaster, P.; Salkauskas, K. Surfaces generated by moving least square methods. Int. J. Numer. Methods Eng. 1994, 37, 229–256. [CrossRef]
13. Cheng, J. Analyzing the factors influencing the choice of the government on leasing different types of land uses: Evidence from Shanghai of China. Land Use Policy 2020, 90, 104303. [CrossRef]
14. Cheng, J. Data analysis of the factors influencing the industrial land leasing in Shanghai based on mathematical models. Math. Prob. Eng. 2020, 2020, 9346863. [CrossRef]
15. Belytschko, T.; Lu, Y.Y.; Gu, L. Element-free Galerkin methods. Int. J. Numer. Methods Eng. 1994, 37, 229–256. [CrossRef]
16. Li, X.; Chen, H.; Wang, Y. Error analysis in Sobolev spaces for the improved moving least-square approximation and the improved element-free Galerkin method. Appl. Math. Comput. 2015, 262, 56–78. [CrossRef]
17. Zhang, Z.; Li, D.M.; Cheng, Y.M.; Liew, K.M. The improved element-free Galerkin method for three-dimensional wave equation. Acta Mech. Sin. 2012, 28, 808–818. [CrossRef]
18. Zhang, Z.; Zhao, P.; Liew, K.M. Analyzing three-dimensional potential problems with the improved element-free Galerkin method. Comput. Mech. 2009, 44, 273–284. [CrossRef]
19. Zhang, Z.; Wang, J.F.; Cheng, Y.M.; Liew, K.M. The improved element-free Galerkin method for three-dimensional transient heat conduction problems. Sci. China 2013, 56, 1568–1580. [CrossRef]
20. Zheng, G.D.; Cheng, Y.M. The improved element-free Galerkin method for diffusional drug release problems. Int. J. Appl. Mech. 2020, 12, 2050096. [CrossRef]
21. Ren, H.P.; Cheng, Y.M. The interpolating element-free Galerkin (IEFG) method for two-dimensional elasticity problems. Int. J. Appl. Mech. 2011, 3, 735–758. [CrossRef]
22. Ren, H.P.; Cheng, Y.M. The interpolating element-free Galerkin (IEFG) method for two-dimensional potential problems. Eng. Anal. Bound. Elem. 2012, 36, 873–880. [CrossRef]
23. Maisuradze, G.G.; Thompson, D.L.; Wagner, A.F.; Minkoff, M. Interpolating moving least-squares methods for fitting potential energy surfaces: Detailed analysis of one-dimensional applications. J. Chem. Phys. 2003, 119, 10002–10014. [CrossRef]
24. Wang, J.F.; Sun, F.X.; Cheng, Y.M. An improved interpolating element-free Galerkin method with nonsingular weight function for two-dimensional potential problems. Chin. Phys. B 2012, 21, 53–59. [CrossRef]
25. Sun, F.X.; Wang, J.F.; Cheng, Y.M. An improved interpolating element-free Galerkin method for elasticity. Chin. Phys. B 2013, 22, 47–54. [CrossRef]
26. Sun, F.X.; Liu, C.; Cheng, Y.M.; Peng, M.J. An improved interpolating element-free Galerkin method based on nonsingular weight functions. Math. Probl. Eng. 2014, 2014, 323945. [CrossRef]
27. Sun, F.X.; Wang, J.F.; Cheng, Y.M. An improved interpolating element-free Galerkin method for elastoplasticity via nonsingular weight functions. Int. J. Appl. Mech. 2016, 8, 1650096. [CrossRef]
28. Wang, J.F.; Wang, J.; Sun, F.X.; Cheng, Y.M. An interpolating boundary element-free method with nonsingular weight function for two-dimensional potential problems. Int. J. Comput. Methods 2013, 10, 1350043. [CrossRef]
29. Hansen, E.; Ostermann, A. Dimension splitting for quasilinear parabolic equations. IMA J. Numer. Anal. 2010, 30, 857–869. [CrossRef]
30. Li, K.; Yu, J.; Shi, F.; Huang, A.X. Dimension splitting method for the three dimensional rotating Navier-Stokes equations. Acta Math. Appl. Sin. Engl. Ser. 2012, 28, 417–442. [CrossRef]
31. Meng, Z.J.; Cheng, H.; Ma, L.D.; Cheng, Y.M. The dimension splitting element-free Galerkin method for three-dimensional potential problems. Acta. Mech. Sin. 2018, 34, 462–474. [CrossRef]
32. Meng, Z.J.; Cheng, H.; Ma, L.D.; Cheng, Y.M. The dimension splitting element-free Galerkin method for 3D transient heat conduction problems. Sci. China Phys. Mech. Astron. 2019, 62, 49–60. [CrossRef]
33. Meng, Z.J.; Cheng, H.; Ma, L.D.; Cheng, Y.M. The hybrid element-free Galerkin method for three-dimensional wave propagation problem. Int. J. Numer. Methods Eng. 2019, 117, 15–37. [CrossRef]
34. Cheng, H.; Peng, M.J.; Cheng, Y.M. A hybrid improved complex variable element-free Galerkin method for three-dimensional potential problems. Eng. Anal. Bound. Elem. 2017, 84, 52–62. [CrossRef]
35. Cheng, H.; Peng, M.J.; Cheng, Y.M. The dimension splitting and improved complex variable element-free Galerkin method for 3-dimensional transient heat conduction problems. Int. J. Numer. Methods Eng. 2018, 114, 321–345. [CrossRef]
36. Cheng, H.; Peng, M.J.; Cheng, Y.M. A hybrid improved complex variable element-free Galerkin method for three-dimensional advection-diffusion problems. Eng. Anal. Bound. Elem. 2018, 97, 39–54. [CrossRef]
37. Wang, B.; Ma, Y.; Cheng, Y. The improved complex variable element-free Galerkin method for bending problem of thin plate on elastic foundations. Int. J. Appl. Mech. 2019, 11, 1950105. [CrossRef]
38. Wu, Q.; Peng, M.J.; Cheng, Y.M. The interpolating dimension splitting element-free Galerkin method for 3D potential problems. Eng. Comput. 2021, 128, 1–15. [CrossRef]
39. Wu, Q.; Peng, M.J.; Fu, Y.D.; Cheng, Y.M. The dimension splitting interpolating element-free Galerkin method for solving three-dimensional transient heat conduction problems. Eng. Anal. Bound. Elem. 2021, 128, 326–341. [CrossRef]
40. Meng, Z.J.; Xi, X.F. An improved interpolation dimension split element-free Galerkin method for 3D wave equations. Eng. Anal. Bound. Elem. 2022, 134, 96–106. [CrossRef]
41. Tracy, F.T. Accuracy and Performance Testing of three-dimensional unsaturated flow finite element groundwater programs on the Cray XT3 using analytical solutions. In Proceedings of the HPCMP Users Group Conference, Washington, DC, USA, 26 June 2006.
42. Gasiorowski, D.; Kolerski, T. Numerical solution of the two-dimensional Richards equation using alternate splitting methods for dimensional decomposition. Water 2020, 12, 1780. [CrossRef]
43. Berardi, M.; Difonzo, F.V. A quadrature-based scheme for numerical solutions to Kirchhoff transformed Richards’ equation. J. Comput. Dynam. 2022, 9, 69–84. [CrossRef]
44. Suk, H.; Park, E. Numerical solution of the Kirchhoff-transformed Richards equation for simulating variably saturated flow in heterogeneous layered porous media. J. Hydrof. 2019, 579, 124213. [CrossRef]