Analyzing 3D Advection-Diffusion Problems by Using the Improved Element-Free Galerkin Method

Heng Cheng and Guodong Zheng

1School of Applied Science, Taiyuan University of Science and Technology, Taiyuan 030024, China
2Shanghai Key Laboratory of Mechanics in Energy Engineering, Shanghai Institute of Applied Mathematics and Mechanics, School of Mechanics and Engineering Science, Shanghai University, Shanghai 200072, China

Correspondence should be addressed to Heng Cheng; chengheng@shu.edu.cn

Received 14 July 2020; Accepted 25 July 2020; Published 14 August 2020

1. Introduction

Currently, meshless method has been applied successfully to deal with various kinds of problems in the fields of science and engineering. When solving large deformation problems and dynamic propagation of cracks, meshless method can obtain greater precision than finite element method [1].

As an important meshless method, the EFG method has been applied to different kinds of engineering problems. In the EFG method, the moving least-squares (MLS) approximation is employed to construct the shape function. Because the MLS approximation is based on the least-squares method [2, 3], the disadvantages of the least-squares method also exist in the MLS approximation, in which sometimes ill-conditioned or singular matrices occur in the final equations.

Cheng and Chen studied the IMLS approximation [4] which can make up for the deficiency of the MLS approximation; then the IEFG method is applied for transient heat conduction [5], wave equation [6], fracture [7], elastoplasticity [8], and viscoelasticity [9] problems. The improved complex variable EFG method is presented for wave equation problem [10] and bending problem of thin plate on elastic foundations [11]. The IEFG and the improved complex variable EFG method can enhance the computational speed of the EFG method.

Based on the MLS approximation with a singular weight function, the interpolating MLS method was proposed by Lancaster et al. [12], and the corresponding meshless method can be applied with the essential boundary condition directly. Based on the interpolating MLS method, Kaljevic et al. proposed the improved formulation of EFG method [13]. Based on the concept of an inner production, Ren et al. improved the interpolating MLS method by using singular weight function in interpolating points and orthogonalizing some of basis functions [14]. And the interpolating EFG method is presented for potential [15, 16], transient heat conduction [17], elasticity [18], viscoelasticity [19], elastoplasticity [20], and elastic large deformation [21] problems.

In order to overcome the difficulties caused by singular weight function in the interpolating MLS method, an improved interpolating least-squares (IIMLS) method based on nonsingular weight function was presented by Wang et al. Based on the IIMLS method, the improved interpolating
EFG method is presented for potential [22], elasticity [23], and some complex mechanical problems [24–26].

By combining the dimension splitting method and meshless methods, the hybrid complex variable EFG method [27–31], the dimension split EFG method [32–35], and the dimension splitting reproducing kernel particle method [36] for 3D problems are proposed, respectively, and those new methods can improve the computational speed of traditional meshless methods for solving 3D problems greatly.

In this paper, combining the IMLS approximation and the Galerkin weak form, the IEFG method is used for solving 3D advection-diffusion problems. The IMLS approximation is used to form the trial function, the penalty method is applied to introduce the essential boundary conditions, the Galerkin weak form and the difference method are used to obtain the final discretized equations, and then the formulae of the IEFG method for 3D advection-diffusion problems are presented.

In the section of numerical examples, the weight functions, the scale parameter, the penalty factor, the node distribution, and the time step are discussed, respectively. The numerical results show that the IEFG method in this paper has a higher computational speed. And the advantage that the IEFG method can avoid the singular matrix is given.

2. The IMLS Approximation

The approximation of a function \( u(x) \) is defined as

\[
u_h(x) = \sum_{i=1}^{m} p_i(x) \cdot a_i(x) = \mathbf{p}^T(x) \cdot \mathbf{a}(x), \quad (x \in \Omega),
\]

(1)

where \( \mathbf{p}^T(x) \) is the basis function vector, \( m \) is the basis function number, and

\[
\mathbf{a}^T(x) = (a_1(x), a_2(x), \cdots, a_m(x)),
\]

(2)

is the corresponding coefficient vector of \( \mathbf{p}^T(x) \).

In general,

\[
\mathbf{p}^T(x) = (1, x_1, x_2, x_3),
\]

(3)

\[
\mathbf{p}^T(x) = (1, x_1, x_2, x_3, x_1^2, x_2^2, x_3^2, x_1x_2, x_1x_3, x_2x_3).
\]

(3)

The local approximation of equation (1) is

\[
u_h(x, \bar{x}) = \sum_{i=1}^{m} p_i(\bar{x}) \cdot a_i(x) = \mathbf{p}^T(\bar{x}) \cdot \mathbf{a}(x).
\]

(4)

Define

\[
J = \sum_{i=1}^{n} w(x-x_i) \left[ u_h(x, x_i) - u_i \right]^2 = \sum_{i=1}^{n} w(x-x_i) \left[ \sum_{i=1}^{m} p_i(x_i) \cdot a_i(x) - u_i \right]^2,
\]

(5)

where \( w(x-x_i) \) is a weighting function which contains compact support, and \( x_i (i = 1, 2, \cdots, n) \) are the nodes with influence domains covering the point \( x \).

Equation (5) can be written as

\[
J = (\mathbf{P} \mathbf{a} - \mathbf{u})^T \mathbf{W}(x) (\mathbf{P} \mathbf{a} - \mathbf{u}),
\]

(6)

where

\[
\mathbf{u}^T = (u_1, u_2, \cdots, u_n),
\]

\[
\mathbf{p} = \begin{bmatrix} p_1(x_1) & p_2(x_1) & \cdots & p_m(x_1) \\ p_1(x_2) & p_2(x_2) & \cdots & p_m(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ p_1(x_n) & p_2(x_n) & \cdots & p_m(x_n) \end{bmatrix},
\]

(7)

\[
\mathbf{w}(x-x_i) = \begin{bmatrix} 0 & \cdots & 0 \\ w(x-x_2) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & w(x-x_n) \end{bmatrix}.
\]

(7)

From

\[
\frac{\partial J}{\partial \mathbf{a}} = \mathbf{A}(x) \mathbf{a}(x) - \mathbf{B}(x) \mathbf{u} = 0,
\]

(8)

we have

\[
\mathbf{A}(x) \mathbf{a}(x) = \mathbf{B}(x) \mathbf{u},
\]

(9)

where

\[
\mathbf{A}(x) = \mathbf{P}^T \mathbf{W}(x) \mathbf{P},
\]

(10)

\[
\mathbf{B}(x) = \mathbf{P}^T \mathbf{W}(x).
\]

(10)

Equation (9) sometimes forms singular or ill-conditional matrix. In order to make up for this deficiency, for basis functions

\[
\mathbf{q} = (q_i) = (1, x_1, x_2, x_3, x_1^2, x_2^2, x_3^2, x_1x_2, x_1x_3, x_2x_3, \cdots),
\]

(11)

using Gram-Schmidt process, we can make the orthogonal basis functions as

\[
\mathbf{p}_i = \mathbf{q}_i - \sum_{k=1}^{i-1} \frac{(\mathbf{q}_i, \mathbf{p}_k)}{(\mathbf{p}_k, \mathbf{p}_k)} \mathbf{p}_k, \quad (i = 1, 2, 3, \cdots),
\]

(12)

\[
(\mathbf{p}_i, \mathbf{p}_j) = 0, \quad (i \neq j).
\]

(12)

Then from equation (9), \( \mathbf{a}(x) \) can be obtained directly as

\[
\mathbf{a}(x) = \mathbf{A}^*(x) \mathbf{B}(x) \mathbf{u},
\]

(13)

where

\[
\mathbf{A}^*(x) = \begin{bmatrix} \frac{1}{(\mathbf{p}_1, \mathbf{p}_1)} & 0 & \cdots & 0 \\ 0 & \frac{1}{(\mathbf{p}_2, \mathbf{p}_2)} & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{1}{(\mathbf{p}_m, \mathbf{p}_m)} \end{bmatrix}.
\]

(14)
Substituting equation (13) into equation (4), we have
\[ u^h(x) = \Phi^*(x)u = \sum_{i=1}^{n} \Phi^*_i(x)u_i, \]  
(15)

where
\[ \Phi^*(x) = (\Phi^*_1(x), \Phi^*_2(x), \ldots, \Phi^*_n(x)) = p^T(x)A^*(x)B(x) \]
(16)
is the shape function.

This is the IMLS approximation, in which \( a(x) \) can be obtained simply and directly because it is unnecessary to obtain the inverse matrix of \( A^*(x) \). Therefore, the IMLS approximation can avoid ill-conditioned or singular matrix; then it can improve the computational precision and efficiency of the MLS approximation [4].

3. The IEFG Method for 3D Advection-Diffusion Problems

For 3D advection-diffusion problems, the governing equation is
\[ \frac{\partial u}{\partial t} = \frac{\partial u}{\partial x_i} \left( k_i \frac{\partial u}{\partial x_j} \right) - \frac{\partial q}{\partial x_j} + \frac{\partial v(x_i)}{\partial x_j} \frac{\partial u}{\partial x_j} + \frac{\partial v(x_j)}{\partial x_j} \frac{\partial u}{\partial x_i} + f(x,t), \quad (x = (x_1, x_2, x_3) \in \Omega), \]
(17)

with the boundary conditions
\[ u(x,t) = \Pi(x,t), \quad (x \in \Gamma_u), \]
\[ q(x,t) = k_1 \frac{\partial u(x,t)}{\partial x_1} n_1 + k_2 \frac{\partial u(x,t)}{\partial x_2} n_2 + k_3 \frac{\partial u(x,t)}{\partial x_3} n_3 = \overline{q}(x,t), \quad (x \in \Gamma_q), \]
(18)

and the initial condition
\[ u(x,t) = u_0, \]
(19)
where \( u(x,t) \) is the field function, \( \Pi(x,t) \) is the given field function on the essential boundary \( \Gamma_u \), \( \overline{q}(x,t) \) is the given value on the natural boundary \( \Gamma_q \), \( \Gamma = \Gamma_u \cup \Gamma_q \) is the boundary of the problem domain \( \Omega \), and \( \Gamma_u \cap \Gamma_q = \emptyset \); \( f(x,t) \) is the source term; \( k_i \) is the diffusion efficient in the direction \( x_i \); \( v(x_i) \) is the advection velocity in the direction \( x_i \); \( u_0 \) is known function; \( n_i \) is the unit outward normal to the boundary \( \Gamma \) in the direction \( x_i \).

The equivalent functional of 3D advection-diffusion problem can be obtained as
\[ \Pi = \int_\Omega u \left( \frac{\partial u}{\partial t} - f \right) d\Omega \]
\[ + \int_\Omega \frac{1}{\Omega^2} \left[ k_1 \left( \frac{\partial u}{\partial x_1} \right)^2 + k_2 \left( \frac{\partial u}{\partial x_2} \right)^2 + k_3 \left( \frac{\partial u}{\partial x_3} \right)^2 \right] d\Omega \]
\[ + \int_\Omega u \left( v(x_1) \frac{\partial u}{\partial x_1} + v(x_2) \frac{\partial u}{\partial x_2} + v(x_3) \frac{\partial u}{\partial x_3} \right) d\Omega \]
\[ - \int_{\Gamma_q} u \overline{q} d\Gamma. \]
(20)

Suppose that \( \alpha \) is the penalty factor; by introducing the penalty method, we can obtain
\[ \Pi^* = \Pi + \frac{\alpha}{2} \int_{\Gamma_q} (u - \Pi)(u - \Pi) d\Gamma. \]
(21)

From
\[ \delta \Pi^* = 0, \]
(22)
we can obtain that the equivalent integral weak form is
\[ \delta \Pi^* = \int_\Omega \delta u \cdot \frac{\partial u}{\partial t} d\Omega - \int_\Omega \delta u \cdot f d\Omega + \int_\Omega \delta u \cdot v(x_i) \frac{\partial u}{\partial x_i} d\Omega \]
\[ + \int_\Omega \delta u \cdot v(x_j) \frac{\partial u}{\partial x_j} d\Omega + \int_\Omega \delta u \cdot v(x_k) \frac{\partial u}{\partial x_k} d\Omega \]
\[ - \int_{\Gamma_q} \delta u \cdot \overline{q} d\Gamma + \alpha \int_{\Gamma_q} \delta u \cdot u d\Gamma - \alpha \int_{\Gamma_u} \delta u \cdot \Pi d\Gamma, \]
(23)
where
\[ \overline{k} = \begin{bmatrix} k_1 & 0 & 0 \\ 0 & k_2 & 0 \\ 0 & 0 & k_3 \end{bmatrix}, \]
(24)
\[ L(\cdot) = \begin{bmatrix} \frac{\partial}{\partial x_1} \\ \frac{\partial}{\partial x_2} \\ \frac{\partial}{\partial x_3} \end{bmatrix}(\cdot). \]
(25)

In the cubic domain \( \Omega \), we employ \( M \) nodes \( x_l \) \( (I = 1, 2, \ldots, M) \). The function \( u(x_l) \) at the node \( x_l \) is
\[ u_l = u(x_l, t). \]
(26)
From the IMLS approximation, at the time of $t$, the function at an arbitrary node $x$ is

$$u(x, t) = \Phi^* (x) \mathbf{u} = \sum_{i=1}^{n} \Phi^*_i (x) u_i,$$

(27)

where

$$\mathbf{u} = (u_1, u_2, \ldots, u_n)^T.$$

(28)

From equations (25) and (27), we have

$$\frac{\partial u(x, t)}{\partial t} = \frac{\partial}{\partial t} \sum_{i=1}^{n} \Phi^*_i (x) u_i = \sum_{i=1}^{n} \Phi^*_i (x) \frac{\partial u_i}{\partial t} = \Phi^* (x) \mathbf{u},$$

(29)

$$L \mathbf{u}(x, t) = \sum_{i=1}^{n} \frac{\partial}{\partial x_i} \left( \Phi^*_i (x) u_i \right) = \sum_{i=1}^{n} B_i (x) u_i = \mathbf{B}(x) \mathbf{u},$$

(30)

where

$$\mathbf{u} = \left( \frac{\partial u(x_1, t)}{\partial t}, \frac{\partial u(x_2, t)}{\partial t}, \ldots, \frac{\partial u(x_n, t)}{\partial t} \right)^T,$$

$$\mathbf{B}(x) = \begin{bmatrix} B_1 (x) \\ B_2 (x) \\ \vdots \\ B_n (x) \end{bmatrix},$$

(31)

Substituting equations (27), (29), and (30) into equation (23) yields

$$\int_{\Omega} \delta [\Phi^* (x) \mathbf{u}] \cdot \mathbf{f} d\Omega = \int_{\Omega} \delta [\Phi^* (x) \mathbf{u}] \cdot f d\Omega$$

$$+ \int_{\Omega} \delta [\mathbf{B}(x) \mathbf{u}]^T \cdot \mathbf{k} \cdot [\mathbf{B}(x) \mathbf{u}] d\Omega + \int_{\Omega} \delta [\Phi^* (x) \mathbf{u}] \cdot \mathbf{v}(x_1) \frac{\partial}{\partial x_1} [\Phi^* (x) \mathbf{u}] d\Omega$$

$$+ \int_{\Omega} \delta [\Phi^* (x) \mathbf{u}] \cdot \mathbf{v}(x_2) \frac{\partial}{\partial x_2} [\Phi^* (x) \mathbf{u}] d\Omega$$

$$+ \int_{\Omega} \delta [\Phi^* (x) \mathbf{u}] \cdot \mathbf{v}(x_3) \frac{\partial}{\partial x_3} [\Phi^* (x) \mathbf{u}] d\Omega$$

$$- \int_{\Gamma_y} \delta [\Phi^* (x) \mathbf{u}] \cdot \mathbf{n} d\Gamma + \alpha \int_{\Gamma_y} \delta [\Phi^* (x) \mathbf{u}] \cdot [\Phi^* (x) \mathbf{u}] d\Gamma$$

$$- \alpha \int_{\Gamma_y} \delta [\Phi^* (x) \mathbf{u}] \cdot \mathbf{n} d\Gamma = 0.$$  

(32)

Analyzing the integral terms in equation (32), respectively, we have

$$\int_{\Omega} \delta [\Phi^* (x) \mathbf{u}] \cdot [\Phi^* (x) \mathbf{u}] d\Omega = \delta \mathbf{u}^T \cdot \left( \int_{\Omega} \Phi^{*T} (x) \Phi^* (x) d\Omega \right) \mathbf{u} = \delta \mathbf{u}^T \cdot C \cdot \mathbf{u},$$

(33)

$$\int_{\Omega} \delta [\Phi^* (x) \mathbf{u}] \cdot f d\Omega = \delta \mathbf{u}^T \cdot \left( \int_{\Omega} \Phi^{*T} (x) f d\Omega \right) \mathbf{u} = \delta \mathbf{u}^T \cdot \mathbf{F}_1,$$

(34)

$$\int_{\Omega} \delta [\mathbf{B}(x) \mathbf{u}]^T \cdot \mathbf{k} \cdot [\mathbf{B}(x) \mathbf{u}] d\Omega = \delta \mathbf{u}^T \cdot \left( \int_{\Omega} \mathbf{B}^T (x) \mathbf{k} \mathbf{B}(x) d\Omega \right) \mathbf{u} = \delta \mathbf{u}^T \cdot \mathbf{K} \mathbf{u},$$

(35)

$$\int_{\Omega} \delta [\Phi^* (x) \mathbf{u}] \cdot \mathbf{v}(x_1) \frac{\partial}{\partial x_1} [\Phi^* (x) \mathbf{u}] d\Omega$$

$$= \delta \mathbf{u}^T \cdot \left( \int_{\Omega} \Phi^{*T} (x) \mathbf{v}(x_1) \frac{\partial}{\partial x_1} \Phi^* (x) d\Omega \right) \mathbf{u} = \delta \mathbf{u}^T \cdot \mathbf{G}_1 \mathbf{u},$$

(36)

$$\int_{\Omega} \delta [\Phi^* (x) \mathbf{u}] \cdot \mathbf{v}(x_2) \frac{\partial}{\partial x_2} [\Phi^* (x) \mathbf{u}] d\Omega$$

$$= \delta \mathbf{u}^T \cdot \left( \int_{\Omega} \Phi^{*T} (x) \mathbf{v}(x_2) \frac{\partial}{\partial x_2} \Phi^* (x) d\Omega \right) \mathbf{u} = \delta \mathbf{u}^T \cdot \mathbf{G}_2 \mathbf{u},$$

(37)

$$\int_{\Omega} \delta [\Phi^* (x) \mathbf{u}] \cdot \mathbf{v}(x_3) \frac{\partial}{\partial x_3} [\Phi^* (x) \mathbf{u}] d\Omega$$

$$= \delta \mathbf{u}^T \cdot \left( \int_{\Omega} \Phi^{*T} (x) \mathbf{v}(x_3) \frac{\partial}{\partial x_3} \Phi^* (x) d\Omega \right) \mathbf{u} = \delta \mathbf{u}^T \cdot \mathbf{G}_3 \mathbf{u},$$

(38)

$$\int_{\Gamma_y} \delta [\Phi^* (x) \mathbf{u}] \cdot \mathbf{a} d\Gamma = \delta \mathbf{u}^T \cdot \left( \int_{\Gamma_y} \Phi^{*T} (x) \mathbf{a} d\Gamma \right) \mathbf{u} = \delta \mathbf{u}^T \cdot \mathbf{F}_2,$$

(39)

$$\int_{\Gamma_u} \delta [\Phi^* (x) \mathbf{u}] \cdot \mathbf{n} d\Gamma = \delta \mathbf{u}^T \cdot \left( \int_{\Gamma_u} \Phi^{*T} (x) \mathbf{n} d\Gamma \right) \mathbf{u} = \delta \mathbf{u}^T \cdot \mathbf{F}_3,$$

(40)

$$\int_{\Gamma_u} \delta [\Phi^* (x) \mathbf{u}] \cdot \mathbf{n} d\Gamma = \delta \mathbf{u}^T \cdot \left( \int_{\Gamma_u} \Phi^{*T} (x) \mathbf{n} d\Gamma \right) \mathbf{u} = \delta \mathbf{u}^T \cdot \mathbf{F}_4,$$

(41)
\[ C = \int_{\Omega} \Phi^T(x) \Phi(x) \, d\Omega, \]
\[ F_1 = \int_{\Omega} \Phi^T(x) f \, d\Omega, \]
\[ K = \int_{\Omega} B^T(x) k B(x) \, d\Omega, \]
\[ G_1 = \int_{\Omega} \Phi^T(x) v(x_1) \frac{\partial}{\partial x_1} \Phi^*(x) \, d\Omega, \]
\[ G_2 = \int_{\Omega} \Phi^T(x) v(x_2) \frac{\partial}{\partial x_2} \Phi^*(x) \, d\Omega, \]
\[ G_3 = \int_{\Omega} \Phi^T(x) v(x_3) \frac{\partial}{\partial x_3} \Phi^*(x) \, d\Omega, \]
\[ F_2 = \int_{\Gamma_1} \Phi^T(x) \eta \, d\Gamma, \]
\[ K_n = a \int_{\Gamma_n} \Phi^T(x) \Phi^*(x) \, d\Gamma, \]
\[ F_n = a \int_{\Gamma_n} \Phi^T(x) \eta \, d\Gamma. \]

Substituting equations (33)-(41) into equation (32), we can obtain
\[ \delta u^T \cdot (C u + K u + G_1 u + G_2 u + G_3 u + G_4 u + G_5 u - F_1 - F_2 - F_n) = 0. \]
(43)

\( \delta u^T \) is arbitrary in equation (43); then the second-order ordinary differential equations can be obtained as follows:
\[ C \dot{u} + K u = F, \]
(44)
where
\[ \dot{K} = K + K_n + G_1 + G_2 + G_3, \]
\[ \dot{F} = F_1 + F_2 + F_n. \]
(45)

Then, using the finite difference method to separate the time, the relation of \( u_{t+\Delta t} \) and \( u_t \) can be established as
\[ \theta \left( \frac{\partial u}{\partial t} \bigg|_{t+\Delta t} \right) + (1 - \theta) \left( \frac{\partial u}{\partial t} \bigg|_t \right) = \frac{u_{t+\Delta t} - u_t}{\Delta t}, \]
(46)
where \( \Delta t \) is the time step; solving equation (44) for \( (\partial u/\partial t)_{t+\Delta t} \) and \( (\partial u/\partial t)_t \), respectively, and substituting the results into equation (44), we have
\[ \left( \frac{C}{\Delta t} + \theta \dot{K}_{n+1} \right) u_{n+1} = \left[ \frac{C}{\Delta t} - (1 - \theta) \dot{K}_n \right] u_n + \theta \dot{F}_{n+1} + (1 - \theta) \dot{F}_n, \]
(47)
where \( \theta \) is a time weighted coefficient; we select \( \theta = 1 \) in numerical examples.

### 4. Numerical Examples

Three numerical examples of 3D advection-diffusion problems are solved with the EFG and the IEFG method, and the computational accuracy and efficiency are compared, respectively.

Define the relative error as
\[ \| u - u^h \|_{L^2(\Omega)} = \| u - u^h \|_{L^2(\Omega)}, \]
(48)
where
\[ \| u - u^h \|_{L^2(\Omega)} = \left( \int_{\Omega} (u - u^h)^2 \, d\Omega \right)^{1/2}. \]
(49)

In this section, the linear basis function is employed and the node distribution is regular. Moreover, \( 3 \times 3 \times 3 \) Gaussian points are used in an integral cell.

The first example we considered is a 3D advection-diffusion problem:
\[ \frac{\partial u(x, t)}{\partial t} = \frac{1}{2} \left( \frac{\partial^2 u(x, t)}{\partial x_1^2} + \frac{\partial^2 u(x, t)}{\partial x_2^2} + \frac{\partial^2 u(x, t)}{\partial x_3^2} \right) - \frac{1}{2} \left( \frac{\partial u(x, t)}{\partial x_1} + \frac{\partial u(x, t)}{\partial x_2} + \frac{\partial u(x, t)}{\partial x_3} \right), \]
(50)
with the initial condition
\[ u(x, 0) = e^{-x_1} + e^{-x_2} + e^{-x_3}, \]
(51)
and the boundary conditions
\[ u|_{x_1=0} = (1 + e^{-x_2} + e^{-x_3})e^t, \]
\[ u|_{x_1=1} = (e^{-1} + e^{-x_2} + e^{-x_3})e^t, \]
\[ u|_{x_2=0} = (e^{-x_1} + 1 + e^{-x_3})e^t, \]
\[ u|_{x_2=1} = (e^{-x_1} + e^{-1} + e^{-x_3})e^t, \]
\[ u|_{x_3=0} = (e^{-x_1} + e^{-x_2} + 1)e^t, \]
\[ u|_{x_3=1} = (e^{-x_1} + e^{-x_2} + e^{-1})e^t. \]
(52)

The problem domain is \( \Omega = [0, 1] \times [0, 1] \times [0, 1] \), and \( T \) is total time.

The analytical solution is
\[ u(x, t) = (e^{-x_1} + e^{-x_2} + e^{-x_3})e^t. \]
(53)

In this example, we discuss the effects of the weight functions, the scale parameter, the penalty factor, and the node distribution on the solution of the IEFG method, respectively.

#### 4.1. Weight Functions
The time step \( \Delta t \) is selected as 0.01, \( 11 \times 11 \times 11 \) nodes are distributed regularly, and \( 10 \times 10 \times 10 \) integral cells are selected. Then the accuracy of the IEFG method is discussed.
When the cubic spline function is used, \( \Delta t = 0.1 \), \( \alpha = 8.7 \times 10^6 \), and \( \Delta t = 0.1 \) then the smaller relative errors of the IEFG method are consistent, and the cubic spline function is used as the weight function. Figure 1 shows the relative errors of the IEFG method with the increase of \( \Delta t \). It is shown that the numerical solution has greater computational precision when \( \Delta t = 0.1 \).

Thus, when we use the cubic spline function as the weight function, higher accuracy can be obtained.

4.2. Scale Parameter. \( \Delta t = 0.01 \), \( 11 \times 11 \times 11 \) regular nodes and \( 10 \times 10 \times 10 \) integral cells are selected, respectively, \( \alpha = 8.7 \times 10^6 \), and the cubic spline function is used as the weight function. Figure 1 shows the relative errors of the IEFG method with the increase of \( \Delta t \). It is shown that the numerical solution has greater computational precision when \( \Delta t = 0.1 \).

4.3. Penalty Factor. The same time step, weight function, node distribution, and background integral grid are used, respectively; \( \Delta t = 1.44 \). Figure 2 shows the relative errors of the IEFG method with the change of \( \alpha \) when \( \Delta t = 0.1 \). It is shown that the numerical solution has greater computational precision when \( \alpha = 5.0 \times 10^6 \sim 1.0 \times 10^7 \).

4.4. Node Distribution. The same weight function and time step are used, respectively; \( \Delta t = 1.44 \) and \( \alpha = 8.7 \times 10^6 \). Figure 3 shows the relative errors of the IEFG method with the increase of nodes when \( \Delta t = 0.1 \). Therefore, the numerical solution of the IEFG method for 3D advection-diffusion problems is convergent.

4.5. Time Step. The same weight function is used; \( 11 \times 11 \times 11 \) regular nodes and \( 10 \times 10 \times 10 \) integral cells are selected, respectively; \( \Delta t = 1.44 \) and \( \alpha = 8.7 \times 10^6 \). Figure 4 shows the relative errors of the IEFG method with the change of \( \Delta t \) when \( \Delta t = 0.1 \). We can see that when \( \Delta t = 0.01 \), not only is the greater precision obtained, but also the corresponding CPU time is saved.

The EFG method is selected to solve this example, \( 11 \times 11 \times 11 \) regular nodes and \( 10 \times 10 \times 10 \) integral cells are selected, respectively, \( \Delta t = 0.01 \), \( \Delta t = 1.44 \), and \( \alpha = 8.7 \times 10^6 \), and the cubic spline function is selected; then the great precision can be obtained. When the IEFG method is selected to solve it, the same parameters and the weight function are selected, respectively, then the great precision can also be obtained, and the relative errors are similar.

The comparison of the CPU time of the EFG and the IEFG methods under the same relative errors is shown in Table 1. And the numerical results are given to be compared with the analytical one when \( \Delta t = 0.1 \), 0.5 s, and 0.9 s, respectively (see Figures 5–7). Comparing with the EFG method, the IEFG method can obtain the solutions with similar computational accuracy, but it has greater computational efficiency.
Moreover, the IEFG method which can avoid singular matrix is another advantage when constructing the shape functions. If $d_{\text{max}} = 1.0$, when using the EFG method to solve it, the computational result cannot be obtained, and the error "Warning: Matrix is singular to working precision" appears in MATLAB code because of the singular matrix formed. But when the IEFG method is used, $d_{\text{max}} = 1.0$, and other parameters are the same, then we can obtain the computational solution, and the relative error is 1.4187% when $T$ is 0.1 s. The numerical solutions are compared with the analytical one which is shown in Figure 8; it is shown that the solutions of the IEFG method are in agreement with the analytic ones.

In the second example, a 3D advection-diffusion problem with source term is given as
\[ \frac{\partial u(x,t)}{\partial t} = \frac{\partial^2 u(x,t)}{\partial x_1^2} + \frac{\partial^2 u(x,t)}{\partial x_2^2} + \frac{\partial^2 u(x,t)}{\partial x_3^2} - \partial u(x,t) \]

\[ - \frac{\partial u(x,t)}{\partial x_2} - \frac{\partial u(x,t)}{\partial x_3} - e^{x_1+x_2+x_3-t}, \quad (x \in \Omega, t \in [0,T]), \]

with the initial condition

\[ u(x,0) = e^{x_1+x_2+x_3}, \]

and the boundary conditions

\[ u|_{x_3=0} = e^{x_1+x_3-t}, \]
\[ u|_{x_3=1} = e^{x_1+x_3-t}, \]
\[ u|_{x_3=0} = e^{x_1+x_3-t}, \]
\[ u|_{x_3=1} = e^{x_1+x_3-t}, \]
\[ u|_{x_3=0} = e^{x_1+x_3-t}, \]
\[ u|_{x_3=1} = e^{x_1+x_3-t}. \]

The problem domain is \( \Omega = [0,1] \times [0,1] \times [0,1] \).

The analytical solution is

\[ u(x,t) = e^{x_1+x_2+x_3-t}. \]

The EFG method is used to solve this example, 11 \( \times \) 11 regular nodes and 10 \( \times \) 10 \( \times \) 10 integral cells are selected, respectively, \( \Delta t = 0.01, \) \( d_{\text{max}} = 1.22, \) and \( a = 8.4 \times 10^{-7}, \) and the cubic spline function is used; then the great precision can be obtained. When using the IEFG method to solve it, the same parameters and the weight function are selected, respectively, the great precision can also be obtained, and the relative errors of two methods are similar.

The comparison of the CPU time of the EFG and the IEFG methods under the same relative errors is shown in Table 2. And the numerical solutions are compared with the analytical one when \( T = 0.1 \text{s}, 0.5 \text{s}, \) and 0.9 s, respectively.
(see Figures 9–11). We can see again that the computational speed of the IEFG method is faster.

Similarly, if $d_{\text{max}} = 1.0$, when using the EFG method to solve it, the computational result cannot be obtained, and the same error appears in MATLAB code. But when the IEFG method is used, $d_{\text{max}} = 1.0$, and other parameters are the same, then the relative error can be obtained which is $1.4096\%$ when $T$ is $0.1\,s$. The numerical solutions are compared with the analytical one which is shown in Figure 12; the solutions of the IEFG method are in agreement with the analytic ones which are shown.

In the third example, a 3D advection-diffusion equation is given as

$$
\frac{\partial u(x,t)}{\partial t} = k_1 \frac{\partial^2 u(x,t)}{\partial x_1^2} + k_2 \frac{\partial^2 u(x,t)}{\partial x_2^2} + k_3 \frac{\partial^2 u(x,t)}{\partial x_3^2} - v_1 \frac{\partial u(x,t)}{\partial x_1} - v_2 \frac{\partial u(x,t)}{\partial x_2} - v_3 \frac{\partial u(x,t)}{\partial x_3},
$$

with the initial condition

$$u(x,0) = a(e^{-c_1 x_1} + e^{-c_2 x_2} + e^{-c_3 x_3}),$$

and the boundary conditions

$$u(0,x_2,x_3,t) = ae^{bt}(1 + e^{-c_2 x_2} + e^{-c_3 x_3}),$$

$$u(1,x_2,x_3,t) = ae^{bt}(e^{-c_1} + e^{-c_2 x_2} + e^{-c_3 x_3}),$$

$$u(x_1,0,x_3,t) = ae^{bt}(e^{-c_2 x_2} + 1 + e^{-c_3 x_3}),$$

$$u(x_1,1,x_3,t) = ae^{bt}(e^{-c_1 x_1} + e^{-c_3 x_3} + e^{-c_2 x_2}),$$

$$u(x_1,x_2,0,t) = ae^{bt}(e^{-c_1 x_1} + e^{-c_2 x_2} + 1),$$

$$u(x_1,x_2,1,t) = ae^{bt}(e^{-c_1 x_1} + e^{-c_2 x_2} + e^{-c_3}).$$

The problem domain is $\Omega = [0,1] \times [0,1] \times [0,1]$.

The analytical solution is

$$u(x,t) = ae^{bt}(e^{-c_1 x_1} + e^{-c_2 x_2} + e^{-c_3 x_3}),$$

where

$$c_1 = -v_1 - \sqrt{v_1^2 + 4bk_1},$$

$$c_2 = -v_2 - \sqrt{v_2^2 + 4bk_2},$$

$$c_3 = -v_3 - \sqrt{v_3^2 + 4bk_3}.$$

We set $k_1 = 1.4, k_2 = 1.7, k_3 = 1.5, v_1 = v_2 = v_3 = 1$, and $a = 1, b = -0.1$, for simplicity. The EFG method is used to solve this example, $11 \times 11 \times 11$ regular nodes and $10 \times 10 \times 10$ integral cells are selected, respectively, $\Delta t = 0.01$, $d_{\text{max}} = 2.15$, and $\alpha = 1.1 \times 10^4$, and the cubic spline function is selected; then the great precision can be obtained. When using the IEFG method to solve it, the same parameters and the weight function are selected, respectively, the numerical results can also be obtained with great precision, and the relative errors of two methods are similar.

The comparison of the CPU time of the EFG and the IEFG methods under the same relative errors is shown in Table 3. And the numerical solutions are compared with the analytical one when $T$ are $0.1\,s$, $0.5\,s$, and $0.9\,s$, respectively (see Figures 13–15).

Similarly, if $d_{\text{max}} = 1.0$, when using the EFG method to solve it, the computational result cannot be obtained, and the same error appears in MATLAB code. But when the IEFG method is used, $d_{\text{max}} = 1.0$, and other parameters are the same, then the relative error can be obtained which is $0.1358\%$ when $T$ is $0.1\,s$. The numerical solutions are
compared with the analytical one which is shown in Figure 16. Again, the solutions of the IEFG method are in agreement with the analytical ones which are shown.

For this example, we can select different parameters, $k_1 = 1.4$, $k_2 = 1.7$, $k_3 = 1.5$, $v_1 = v_2 = v_3 = 5$, and $a = 1$, $b = 1$. Using the EFG method to solve it, $11 \times 11 \times 11$ regular nodes and $10 \times 10 \times 10$ integral cells are selected,

| Comparison | $0.1$ s | $0.3$ s | $0.5$ s | $0.7$ s | $0.9$ s |
|------------|---------|---------|---------|---------|---------|
| Relative error (%) | 0.1220 | 0.1402 | 0.1402 | 0.1402 | 0.1402 |
| CPU time of EFG (s) | 366.2 | 588.2 | 781.4 | 987.1 | 1184.2 |
| CPU time of IEFG (s) | 336.1 | 517.3 | 702.4 | 888.6 | 1064 |

Table 3: Comparison of the CPU time of two methods under the same relative error.
respectively, $\Delta t = 0.01$, $d_{\text{max}} = 1.39$, and $\alpha = 4.7 \times 10^3$, and cubic spline function is selected; then the great precision can be obtained. When the IEFG method is selected to solve it, the same parameters and the weight function are selected, respectively, then the great precision can also be obtained, and the relative errors of two methods are similar.

The comparison of the CPU time of the EFG and the IEFG methods under the same relative errors is shown in Table 4. And the numerical solutions are compared with the analytical one when $T$ are $0.1\ s$, $0.5\ s$, and $0.9\ s$, respectively (see Figures 17–19).

Similarly, if $d_{\text{max}} = 1.0$, when using the EFG method to solve it, the computational result cannot be obtained, and the same error appears in MATLAB code. But when the IEFG method is used, $d_{\text{max}} = 1.0$, and other parameters are the same, then the relative error can be obtained which is $0.6367\%$ when $T$ is $0.3\ s$. The numerical solutions are compared with the analytical one which is shown in Figure 20. The solutions of the IEFG method are in agreement with the analytic ones which are shown.

From this example, we can see that both diffusion dominated case and advection dominated one can be solved by using the EFG and the IEFG methods. Although two methods can obtain the solutions with similar computational accuracy, the computational speed of the IEFG is faster than the EFG method.
5. Conclusions

On the basis of the IMLS approximation, the IEFG method for 3D advection-diffusion problems is proposed in this paper. In Section 4, the influences of the weight functions, the scale parameter, the penalty factor, the node distribution, and the time step on the computational precision of the solutions of the IEFG method are discussed, respectively. We can see that the IEFG method in this paper is convergent.

From three examples, we can conclude that, compared with the EFG method for 3D advection-diffusion problems, the IEFG method has a higher computational efficiency. The numerical solutions show that the IEFG method can avoid singular matrix when constructing the shape functions; then the deficiency of the EFG method is overcome.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (no. 11571223) and the Science and Technology Innovation Project of Shanxi Colleges and Universities (Grant no. 2020L0344).

References

[1] R. Huang, S. J. Zheng, Z. S. Liu, and T. Y. Ng, “Recent advances of the constitutive models of smart materials-hydrogels and shape memory polymers,” International Journal of Applied Mechanics, vol. 12, no. 2, Article ID 2050014, 2020.
[2] J. Cheng, “Analyzing the factors influencing the choice of the government on leasing different types of land uses: evidence from Shanghai of China,” Land Use Policy, vol. 90, Article ID 104303, 2020.
[3] J. Cheng, “Data analysis of the factors influencing the industrial land leasing in Shanghai based on mathematical models,” Mathematical Problems in Engineering, vol. 2020, Article ID 9346863, 11 pages, 2020.
[4] Y. M. Cheng and M. J. Chen, “A boundary element-free method for linear elasticity,” Acta Mechanica Sinica, vol. 35, no. 2, pp. 181–186, 2003.
[5] Z. Zhang, J. F. Wang, Y. M. Cheng, and K. M. Liew, “The improved element-free Galerkin method for three-dimensional transient heat conduction problems,” Science China Physics, Mechanics and Astronomy, vol. 56, no. 8, pp. 1568–1580, 2013.
[6] Z. Zhang, D. M. Li, Y. M. Cheng, and K. M. Liew, “The improved element-free Galerkin method for three-dimensional wave equation,” Acta Mechanica Sinica, vol. 28, no. 3, pp. 808–818, 2012.
[7] Z. Zhang, K. M. Liew, Y. Cheng, and Y. Y. Lee, “Analyzing 2D fracture problems with the improved element-free Galerkin method,” Engineering Analysis with Boundary Elements, vol. 32, no. 3, pp. 241–250, 2008.
[8] S. Y. Yu, M. J. Peng, H. Cheng, and Y. M. Cheng, “The improved element-free Galerkin method for three-dimensional elastoplasticity problems,” Engineering Analysis with Boundary Elements, vol. 104, pp. 215–224, 2019.
[9] M. J. Peng, R. X. Li, and Y. M. Cheng, “Analyzing three-dimensional viscoelasticity problems via the improved element-free Galerkin (IEFG) method,” Engineering Analysis with Boundary Elements, vol. 40, pp. 104–113, 2014.
[10] H. Cheng, M. J. Peng, and Y. M. Cheng, “Analyzing wave propagation problems with the improved complex variable element-free Galerkin method,” Engineering Analysis with Boundary Elements, vol. 100, pp. 80–87, 2019.
[11] B. H. Wang, Y. Q. Ma, and Y. M. Cheng, “The improved complex variable element-free Galerkin method for bending problem of thin plate on elastic foundations,” International Journal of Applied Mechanics, vol. 11, no. 10, Article ID 1950105, 2019.
[12] P. Lancaster and K. Salkauskas, “Surfaces generated by moving least squares methods,” Mathematics of Computation, vol. 37, no. 155, pp. 141–158, 1981.
[13] I. Kaljevic and S. Saigal, “An improved element free Galerkin formulation,” International Journal for Numerical Methods in Engineering, vol. 40, no. 16, pp. 2953–2974, 1997.
[14] H. P. Ren, Y. M. Cheng, and W. Zhang, “Researches on the improved interpolating moving least-squares method,” Chinese Journal of Engineering Mathematics, vol. 27, no. 6, pp. 1021–1029, 2010.
[15] H. P. Ren and Y. M. Cheng, “The interpolating element-free Galerkin (IEFG) method for two-dimensional potential problems,” Engineering Analysis with Boundary Elements, vol. 36, no. 5, pp. 873–880, 2012.
[16] D. Liu and Y. M. Cheng, “The interpolating element-free Galerkin (IEFG) method for three-dimensional potential problems,” Engineering Analysis with Boundary Elements, vol. 108, pp. 115–123, 2019.
H. P. Ren and Y. M. Cheng, “The interpolating element-free Galerkin (IEFG) method for two-dimensional elasticity problems,” *International Journal of Applied Mechanics*, vol. 3, no. 4, pp. 735–758, 2011.

P. X. Zhang and M. J. Peng, “Interpolating element-free Galerkin method for viscoelasticity problems,” *Acta Physica Sinica*, vol. 68, no. 17, Article ID 170203, 2019.

Q. Wu, F. B. Liu, and Y. M. Cheng, “The interpolating element-free Galerkin method for three-dimensional elastoplasticity problems,” *Engineering Analysis with Boundary Elements*, vol. 115, pp. 156–167, 2020.

Q. Wu, P. P. Peng, and Y. M. Cheng, “The interpolating element-free Galerkin method for elastic large deformation problems,” *Science China Technological Sciences*, vol. 63, 2020.

J. F. Wang, F. X. Sun, and Y. M. Cheng, “An improved interpolating element-free Galerkin method with nonsingular weight function for two-dimensional potential problems,” *Chinese Physics B*, vol. 21, no. 9, Article ID 090204, 2012.

F. X. Sun, J. F. Wang, and Y. M. Cheng, “An improved interpolating element-free Galerkin method for elasticity,” *Chinese Physics B*, vol. 22, no. 12, Article ID 120203, 2013.

F. B. Liu and Y. M. Cheng, “The improved element-free Galerkin method based on the nonsingular weight functions for elastic large deformation problems,” *International Journal of Computational Materials Science and Engineering*, vol. 7, Article ID 1850023, 2018.

F. B. Liu, Q. Wu, and Y. M. Cheng, “A meshless method based on the nonsingular weight functions for elastoplastic large deformation problems,” *International Journal of Applied Mechanics*, vol. 11, no. 1, Article ID 1950006, 2019.

F. B. Liu and Y. M. Cheng, “The improved element-free Galerkin method based on the nonsingular weight functions for inhomogeneous swelling of polymer gels,” *International Journal of Applied Mechanics*, vol. 10, no. 4, Article ID 1850047, 2018.

H. Cheng, M. J. Peng, and Y. M. Cheng, “A hybrid improved complex variable element-free Galerkin method for three-dimensional potential problems,” *Engineering Analysis with Boundary Elements*, vol. 84, pp. 52–62, 2017.

H. Cheng, M. J. Peng, and Y. M. Cheng, “A fast complex variable element-free Galerkin method for three-dimensional wave propagation problems,” *International Journal of Applied Mechanics*, vol. 9, no. 6, Article ID 1750090, 2017.

H. Cheng, M. J. Peng, and Y. M. Cheng, “The dimension splitting and improved complex variable element-free Galerkin method for 3-dimensional transient heat conduction problems,” *International Journal for Numerical Methods in Engineering*, vol. 114, no. 3, pp. 321–345, 2018.

H. Cheng, M. J. Peng, and Y. M. Cheng, “A hybrid improved complex variable element-free Galerkin method for three-dimensional advection-diffusion problems,” *Engineering Analysis with Boundary Elements*, vol. 97, pp. 39–54, 2018.

H. Cheng, M. J. Peng, Y. M. Cheng, and Z. J. Meng, “A hybrid complex variable element-free Galerkin method for 3D elasticity problems,” *Engineering Structures*, vol. 219, Article ID 110835, 2020.

Z. J. Meng, H. Cheng, L. D. Ma, and Y. M. Cheng, “The dimension split element-free Galerkin method for three-dimensional potential problems,” *Acta Mechanica Sinica*, vol. 34, no. 3, pp. 462–474, 2018.