



The interpolating dimension splitting element-free Galerkin method for 3D potential problems

Qian Wu¹ · Miaojuan Peng¹ · Yumin Cheng²

Received: 17 December 2020 / Accepted: 20 April 2021 / Published online: 27 May 2021
© The Author(s), under exclusive licence to Springer-Verlag London Ltd., part of Springer Nature 2021

Abstract

In this paper, based on the improved interpolating moving least-squares (IMLS) method and the dimension splitting method, the interpolating dimension splitting element-free Galerkin (IDSEFG) method for three-dimensional (3D) potential problems is proposed. The key of the IDSEFG method is to split a 3D problem domain into many related two-dimensional (2D) subdomains. The shape function is constructed by the improved IMLS method on the 2D subdomains, and the Galerkin weak form based on the dimension splitting method is used to obtain the discretized equations. The discrete equations on these 2D subdomains are coupled by the finite difference method. Take the improved element-free Galerkin (IEFG) method as a comparison, the advantage of the IDSEFG method is that the essential boundary conditions can be enforced directly. The effects of the number of nodes, the direction of dimension splitting, and the parameters of the influence domain on the calculation accuracy are studied through four numerical examples, the numerical solutions of the IDSEFG method are compared with the numerical solutions of the IEFG method and the analytical solutions. It is verified that the numerical solutions of the IDSEFG method are highly consistent with the analytical solution, and the calculation efficiency of this method is significantly higher than that of the IEFG method.

Keywords Meshless method · Dimension splitting method · Improved interpolating moving least-squares method · Finite difference method · Interpolating dimension splitting element-free Galerkin method · Potential problem

1 Introduction

There are many complicated problems, such as crack propagation and large deformation, in science and engineering fields must be solved with numerical methods. The boundary element method and the finite element method are the major numerical methods which based on meshes or elements.

In recent years, the meshless method has made great progress [1, 2]. When meshless method is used to solve a

problem, only discrete nodes are distributed on the problem domain and its boundary without meshing. Without domain discretization, this method uses a node-based approximation to construct the approximation function or interpolation function. This fully ensures the calculation accuracy and efficiency. Moreover, the approach leads to a flexible choice of nodes in the domain for the specific characteristics. It is shown that the method has good adaptability and calculation accuracy. Therefore, as a new and efficient method in scientific and engineering computing, the meshless method has gradually become a research hotspot.

Currently, the element-free Galerkin method (EFG) [3] and the reproducing kernel particle method (RKPM) [4, 5] are major meshless methods. Based on the moving least-squares (MLS) approximation [6], the EFG method has high calculation accuracy, because the MLS approximation is obtained from the ordinary least squares method with the best approximation [7–10]. Because of the complicated shape function of the MLS approximation, the computational efficiency of the EFG method is low, especially for solving three-dimensional (3D) problems. In addition, the

✉ Miaojuan Peng
mjpeng@shu.edu.cn

✉ Yumin Cheng
ymcheng@shu.edu.cn

¹ Department of Civil Engineering, School of Mechanics and Engineering Science, Shanghai University, Shanghai 200444, China

² Shanghai 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

MLS approximation maybe causes an ill-conditioned or singular system of equations. To improve the computational efficiency of the EFG method, the improved MLS approximation [11, 12] and complex variable moving least-squares (CVMLS) approximation [13–15] were presented. In the improved MLS approximation, the basis functions of the approximation are orthogonal, which can avoid the ill-conditioned or singular equations. The improved MLS approximation improves the calculation efficiency under similar calculation accuracy because it does not require computing the inversion of matrices. By using the improved MLS approximation Zhang et al. [16–18] proposed the improved EFG method for wave equations, transient heat conduction and elastodynamics problems, and Peng et al. [19] and Cheng et al. [20, 21] presented the improved EFG method for viscoelasticity, elastoplasticity, and diffusional drug release problems. In the CVMLS approximation, the complex variable basis functions are used, which results in the fewer number of the coefficients of the basis functions in the trial function, then the calculation efficiency is improved. Based on the CVMLS approximation, the complex variable EFG method has been presented for solving temperature field [22], elastoplasticity [23] and plate bending problems [24].

In the above EFG method and the corresponding improved methods, the essential boundary conditions are enforced indirectly, and the Lagrange multiplier method or the penalty function method is used. To enforce the essential boundary conditions directly, based on the interpolating moving least-squares (IMLS) method [6], Ren et al. presented the improved IMLS method [25], and then the interpolating EFG method was proposed for elasticity and potential problems [26, 27]. Cheng et al. presented the interpolating EFG method for solving elastoplasticity [28] and large deformation problems [29, 30]. And Liu et al. solved the 3D problems by using the interpolating EFG method [31–33]. Based on nonsingular weight functions, the improved IMLS method is studied further [34, 35], and then the corresponding interpolating EFG method was presented for solving potential [36, 37], elasticity [38], elastoplasticity [39], two-point boundary value problems [40]. Liu et al. proposed the formulae of the interpolating EFG method for solving inhomogeneous swelling of polymer gels [41] and large deformation problems [42, 43].

The dimensional splitting method was introduced into meshless methods to improve the calculation efficiency of the EFG method for 3D problems. Cheng et al. presented the hybrid improved complex variable element-free Galerkin method for 3D potential [44], wave propagation [45], transient heat conduction [46], advection–diffusion [47] and elasticity problems [48]. Meng et al. combined the dimensional splitting method with the IIEFG method and proposed the hybrid element-free Galerkin method for the 3D potential [49], transient heat conduction [50] and wave equation

[51]. And Peng et al. presented the hybrid reproducing kernel particle method for solving 3D potential [52], transient heat conduction [53] and wave equation [54]. These methods can greatly improve the calculation efficiency for 3D problems greatly while ensuring calculation accuracy. In these methods, the essential boundary conditions are enforced indirectly.

In this paper, based on the improved IMLS method and dimension splitting method, the interpolating dimension splitting element-free Galerkin (IDSEFG) method for the 3D potential problems, is proposed. The key of the IDSEFG method is to split a 3D problem domain into a series of related two-dimensional (2D) subdomains. The shape function is constructed with the improved IMLS method on the 2D subdomain, and the Galerkin weak form based on the dimension splitting method is used to obtain the discretized equations. The discretized equations are coupled by using the finite difference method on these 2D subdomains. The essential boundary conditions can be enforced directly to improve the calculation accuracy, and the calculation efficiency is greatly improved because of the dimension splitting method, they are the advantages of the IDSEFG method. The effects of the number of nodes, the direction of dimension splitting, and the parameters of the influence domain on the calculation accuracy are studied through four numerical examples. The numerical solutions of the IDSEFG method are compared with the numerical solutions of the IIEFG method and the analytical solutions. It is verified that the numerical solutions of the IDSEFG method are highly consistent with the analytical solution, and the calculation efficiency of the IDSEFG method is significantly better than that of the IIEFG method.

2 The improved IMLS method

The improved IMLS method [25] is proposed by Ren and Cheng. It is based on Lancaster's IMLS method [6].

Defining inner product of any $f(\mathbf{x})$ and $g(\mathbf{x})$ as

$$(f, g)_x = \sum_{I=1}^n w(\mathbf{x} - \mathbf{x}_I) f(\mathbf{x}_I) g(\mathbf{x}_I), \quad \forall f(\mathbf{x}), g(\mathbf{x}) \in C^0(\bar{\Omega}), \quad (1)$$

where $C^0(\bar{\Omega})$ is a set of continuous functions in problem domain $\bar{\Omega}$ and $\bar{\Omega} = \Omega \cup \partial\Omega$, there are $I = 1, 2, \dots, n$ nodes in domain of point \mathbf{x} . Therefore,

$$f_x = [(f, f)_x]^{\frac{1}{2}}. \quad (2)$$

On the space $\text{span}(d_1, d_2, \dots, d_k)$, $d_1(\mathbf{x}) = 1, d_2(\mathbf{x}), \dots, d_k(\mathbf{x})$ are used to express the basis functions, where a function with the subscript \mathbf{x} is related to the selected point \mathbf{x} . Normalizing $d_1(\mathbf{x})$ we have.

$$\beta_x^{(1)}(\mathbf{x}) = \frac{d_1}{\|d_1\|_x} = \frac{1}{\left[\sum_{l=1}^n w(\mathbf{x} - \mathbf{x}_l)\right]^{\frac{1}{2}}}. \tag{3}$$

As we know $i = 2, 3, \dots, k$, the following functions can be generated and they are orthogonal to $\beta_x^{(1)}$,

$$\begin{aligned} b_x^{(i)}(\mathbf{x}) &= d_i(\mathbf{x}) - (d_i, \beta_x^{(1)})_x \beta_x^{(1)} \\ &= d_i(\mathbf{x}) - \frac{\sum_{l=1}^n d_i(\mathbf{x}_l)w(\mathbf{x} - \mathbf{x}_l)}{\sum_{l=1}^n w(\mathbf{x} - \mathbf{x}_l)} = d_i(\mathbf{x}) - \sum_{l=1}^n d_i(\mathbf{x}_l)v(\mathbf{x} - \mathbf{x}_l), \end{aligned} \tag{4}$$

where

$$v(\mathbf{x} - \mathbf{x}_l) = \frac{w(\mathbf{x} - \mathbf{x}_l)}{\sum_{l=1}^n w(\mathbf{x} - \mathbf{x}_l)}, \quad (l = 1, 2, \dots, n). \tag{5}$$

When we make $\beta_x^{(1)}(\mathbf{x}), b_x^{(2)}(\mathbf{x}), \dots, b_x^{(k)}(\mathbf{x})$ as the new basis functions, the approximation function of the MLS approximation is

$$u^h(\mathbf{x}) = \mathbf{v}^T(\mathbf{x})\mathbf{u} + \mathbf{b}^T(\mathbf{x})\mathbf{a}(\mathbf{x}) \tag{6}$$

where

$$\mathbf{u}^T = (u_1, u_2, \dots, u_n), \tag{7}$$

$$\mathbf{v}^T = (v(\mathbf{x} - \mathbf{x}_1), v(\mathbf{x} - \mathbf{x}_2), \dots, v(\mathbf{x} - \mathbf{x}_n)), \tag{8}$$

$$\mathbf{b}^T(\mathbf{x}) = (b_x^{(2)}(\mathbf{x}), b_x^{(3)}(\mathbf{x}), \dots, b_x^{(k)}(\mathbf{x})), \tag{9}$$

$$\mathbf{a}(\mathbf{x}) = (a_1(\mathbf{x}), a_2(\mathbf{x}), \dots, a_{k-1}(\mathbf{x}))^T. \tag{10}$$

Then using the MLS approximation, we have

$$\mathbf{a}(\mathbf{x}) = \mathbf{A}_x^{-1}(\mathbf{x})\mathbf{B}_x(\mathbf{x})\mathbf{u}, \tag{11}$$

where

$$\mathbf{A}_x(\mathbf{x}) = \mathbf{P}_x^T \mathbf{W}(\mathbf{x}) \mathbf{P}_x, \tag{12}$$

$$\mathbf{B}_x(\mathbf{x}) = \mathbf{P}_x^T \mathbf{W}(\mathbf{x}), \tag{13}$$

$$\mathbf{P}_x = \begin{bmatrix} b_x^{(2)}(\mathbf{x}_1) & b_x^{(3)}(\mathbf{x}_1) & \dots & b_x^{(k)}(\mathbf{x}_1) \\ b_x^{(2)}(\mathbf{x}_2) & b_x^{(3)}(\mathbf{x}_2) & \dots & b_x^{(k)}(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ b_x^{(2)}(\mathbf{x}_n) & b_x^{(3)}(\mathbf{x}_n) & \dots & b_x^{(k)}(\mathbf{x}_n) \end{bmatrix}, \tag{14}$$

$$\mathbf{W}(\mathbf{x}) = \begin{bmatrix} w(\mathbf{x} - \mathbf{x}_1) & 0 & \dots & 0 \\ 0 & w(\mathbf{x} - \mathbf{x}_2) & \vdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \vdots & w(\mathbf{x} - \mathbf{x}_n) \end{bmatrix}. \tag{15}$$

It can be obtained by substituting Eq. (10) into Eq. (6)

$$u^h(\mathbf{x}) = \mathbf{v}^T(\mathbf{x})\mathbf{u} + \mathbf{b}^T(\mathbf{x})\mathbf{A}_x^{-1}(\mathbf{x})\mathbf{B}_x(\mathbf{x})\mathbf{u}. \tag{16}$$

Equation (16) can also be written as

$$u^h(\mathbf{x}) = \Phi(\mathbf{x})\mathbf{u}, \tag{17}$$

where

$$\Phi(\mathbf{x}) = (\Phi_1(\mathbf{x}), \Phi_2(\mathbf{x}), \dots, \Phi_n(\mathbf{x})) = \mathbf{v}^T(\mathbf{x}) + \mathbf{b}^T(\mathbf{x})\mathbf{A}_x^{-1}(\mathbf{x})\mathbf{B}_x(\mathbf{x}). \tag{18}$$

For the improved IMLS method, the shape function which shown as Eq. (18) is simpler than MLS approximation, it will improve the calculation efficiency, and

$$\begin{aligned} (\Phi_l(\mathbf{x}))_j &= (v(\mathbf{x} - \mathbf{x}_l) + \mathbf{b}^T(\mathbf{x})\mathbf{A}_x^{-1}(\mathbf{x})(\mathbf{B}_x(\mathbf{x}))_l)_j \\ &= v(\mathbf{x} - \mathbf{x}_l)_j + \mathbf{b}_j^T \mathbf{A}_x^{-1}(\mathbf{B}_x)_l + \mathbf{b}^T (\mathbf{A}_x^{-1})_j (\mathbf{B}_x)_l + \mathbf{b}^T \mathbf{A}_x^{-1} (\mathbf{B}_x)_{lj}, \end{aligned} \tag{19}$$

where

$$(\mathbf{B}_x)_{lj} = w(\mathbf{x} - \mathbf{x}_l)_j b_x(\mathbf{x}_l), \tag{20}$$

$$(\mathbf{A}_x^{-1})_j = -\mathbf{A}_x^{-1}(\mathbf{A}_x)_j \mathbf{A}_x^{-1}, \tag{21}$$

$$\begin{aligned} (\mathbf{A}_x)_j &= \sum_{l=1}^n w(\mathbf{x} - \mathbf{x}_l)_j \mathbf{b}_x(\mathbf{x}_l) \mathbf{b}_x^T(\mathbf{x}_l) + \sum_{l=1}^n w(\mathbf{x} - \mathbf{x}_l) \mathbf{b}_x(\mathbf{x}_l)_j \mathbf{b}_x^T(\mathbf{x}_l) \\ &+ \sum_{l=1}^n w(\mathbf{x} - \mathbf{x}_l) \mathbf{b}_x(\mathbf{x}_l) \mathbf{b}_x^T(\mathbf{x}_l)_j. \end{aligned} \tag{22}$$

Then we have

$$\mathbf{a}(\mathbf{x}) = \mathbf{A}_x^{-1}(\mathbf{x})\mathbf{B}_x(\mathbf{x})(\mathbf{u} - (\mathbf{v}^T(\mathbf{x})\mathbf{u})d_1), \tag{23}$$

where $d_1(\mathbf{x}_l) = 1$.

The weight function $w(\mathbf{x} - \mathbf{x}_l)$ in the improved IMLS method can be selected as

$$w = w(\mathbf{x} - \mathbf{x}_l) = \begin{cases} \frac{\rho^\sigma}{|\mathbf{x} - \mathbf{x}_l|^\sigma} (1 - \frac{\rho}{|\mathbf{x} - \mathbf{x}_l|})^2, & |\mathbf{x} - \mathbf{x}_l| \leq \rho \\ 0, & \text{otherwise} \end{cases}, \tag{24}$$

where ρ is the radius of the influence domain in the function and σ is an even positive integer, we take $\sigma = 4$ in all examples in this paper.

At any points $\{\mathbf{x}_l\}_{l=1}^n$, $u^h(\mathbf{x})$ of Eq. (17) can interpolate, so we obtain

$$u^h(x_I) = u(x_I), \tag{25}$$

then we have

$$\Phi_I(x_J) = \begin{cases} 1, & I = J, \\ 0, & I \neq J, \end{cases} \tag{26}$$

In the improved IMLS method, the essential boundary conditions can be enforced directly when establishing the discretized equations because this method satisfies the property of the Kronecker delta function, which will definitely improve the calculation accuracy.

3 The IDSEFG method for 3D potential problems

The governing equation of 3D potential problems is

$$\frac{\partial^2 u}{\partial x_1^2} + \frac{\partial^2 u}{\partial x_2^2} + \frac{\partial^2 u}{\partial x_3^2} = b(x), \quad (x = (x_1, x_2, x_3) \in \Omega), \tag{27}$$

with essential boundary condition

$$u(x) = \bar{u}(x), \quad (x \in \Gamma_u), \tag{28}$$

and natural boundary condition

$$q(x) = \frac{\partial u(x)}{\partial \mathbf{n}(x)} = \bar{q}(x), \quad (x \in \Gamma_q), \tag{29}$$

where Ω is the problem domain, $u(x)$ is an unknown variable, $b(x)$ is a known function, $\bar{u}(x)$ is the known potential function on Γ_u , $\bar{q}(x)$ is the known normal derivative on Γ_q . Notice that $\Gamma = \Gamma_u \cup \Gamma_q$ and $\Gamma_u \cap \Gamma_q = \emptyset$, Γ is the boundary of Ω , and $\mathbf{n}(x)$ is the outward unit normal vector of the boundary Γ .

When using the IDSEFG method to solve 3D potential problems, the splitting direction should be selected reasonably according to the characteristics of the control equation and the boundary conditions, this will bring better computing efficiency and make programming easier.

When applying the dimension splitting method, Ω can be split into L layers along one direction, we choose x_3 here. Δx_3 is the distance between layer to layer, we get $L + 1$ 2D subdomains $\Omega^{(k)} (k = 0, 1, 2, \dots, L)$,

$$\Omega = \bigcup_{k=0}^{L-1} \Omega^{(k)} \times [x_3^{(k)}, x_3^{(k+1)}) \cup \Omega^{(L)}, \tag{30}$$

where

$$a = x_3^{(0)} < x_3^{(1)} \dots < x_3^{(L)} = c, \quad x_3 \in [a, c], \tag{31}$$

$$\Delta x_3 = x_3^{(k+1)} - x_3^{(k)} = (c - a)/L, \tag{32}$$

Because Δx_3 is a fixed value, u and $\frac{\partial^2 u}{\partial x_3^2}$ can be considered as functions of x_1 and x_2 , therefore, the 3D problem of Eq. (27) above can be written as the following series of related 2D boundary value problems,

$$\frac{\partial^2 u^{(k)}}{\partial x_1^2} + \frac{\partial^2 u^{(k)}}{\partial x_2^2} = b^{(k)} - \frac{\partial^2 u^{(k)}}{\partial x_3^2}, \quad ((x_1, x_2) \in \Omega^{(k)}, x_3 = x_3^{(k)}), \tag{33}$$

with the boundary conditions

$$u^{(k)}(x_1, x_2) = \bar{u}^{(k)}(x_1, x_2) = \bar{u}(x_1, x_2, x_3^{(k)}), \quad ((x_1, x_2) \in \Gamma_u^{(k)}), \tag{34}$$

$$q^{(k)}(x_1, x_2) = \bar{q}^{(k)}(x_1, x_2) = \bar{q}(x_1, x_2, x_3^{(k)}), \quad ((x_1, x_2) \in \Gamma_q^{(k)}), \tag{35}$$

where

$$u^{(k)} = u(x_1, x_2, x_3^{(k)}), \tag{36}$$

$$b^{(k)} = b(x_1, x_2, x_3^{(k)}), \tag{37}$$

$u^{(k)}(x_1, x_2)$ is the potential on subdomain $\Omega^{(k)}$, $\bar{u}(x_1, x_2, x_3^{(k)})$ is the known potential function on $\Gamma_u^{(k)}$, $\bar{q}(x_1, x_2, x_3^{(k)})$ is the known gradient function on $\Gamma_q^{(k)}$, $\Gamma^{(k)} = \Gamma_u^{(k)} \cup \Gamma_q^{(k)}$ and $\Gamma_u^{(k)} \cap \Gamma_q^{(k)} = \emptyset$.

The key step of the IDSEFG method is to solve the 2D boundary value problem (33)–(35) with the interpolating EFG method, then the solution of the 3D potential problem (Eqs. (27)–(29)) can be obtained by using the finite difference method in direction x_3 .

The equivalent function of Eqs. (33)–(35) is

$$\begin{aligned} \Pi = & \int_{\Omega^{(k)}} \left[u^{(k)} \left(\frac{\partial^2 u^{(k)}}{\partial x_3^2} - b^{(k)} \right) \right] d\Omega^{(k)} \\ & - \int_{\Omega^{(k)}} \frac{1}{2} \left[\left(\frac{\partial u^{(k)}}{\partial x_1} \right)^2 + \left(\frac{\partial u^{(k)}}{\partial x_2} \right)^2 \right] d\Omega^{(k)} \\ & - \int_{\Gamma_q^{(k)}} u^{(k)} \bar{q}^{(k)} d\Gamma^{(k)}. \end{aligned} \tag{38}$$

Because the improved IMLS method satisfies the property of the Kronecker delta function, thus, the essential boundary conditions can be enforced directly.

Let

$$\delta \Pi = 0, \tag{39}$$

we obtain the equivalent integral weak form

$$\int_{\Omega^{(k)}} \delta u^{(k)} \cdot \frac{\partial^2 u^{(k)}}{\partial x_3^2} d\Omega^{(k)} - \int_{\Omega^{(k)}} \delta(Lu^{(k)})^T \cdot (Lu^{(k)}) d\Omega^{(k)} - \int_{\Omega^{(k)}} \delta u^{(k)} \cdot b^{(k)} d\Omega^{(k)} - \int_{\Gamma_q^{(k)}} \delta u^{(k)} \cdot \bar{q}^{(k)} d\Gamma^{(k)} = 0. \tag{40}$$

where

$$L(\cdot) = \begin{bmatrix} \frac{\partial}{\partial x_1} \\ \frac{\partial}{\partial x_2} \end{bmatrix} (\cdot). \tag{41}$$

We distribute M nodes $\mathbf{x}_I^{(k)}$ in the 2D subdomain $\Omega^{(k)}$ obtained after dimension splitting, $u^{(k)}(\mathbf{x})$ at $\mathbf{x}_I^{(k)}$ can be represented as

$$u^{(k)}(\mathbf{x}_I^{(k)}) = u(\mathbf{x}_I^{(k)}, x_3^{(k)}). \tag{42}$$

The trial function $u^h(\mathbf{x}^{(k)}, x_3^{(k)})$ at $\mathbf{x}^{(k)} = (x_1, x_2)$ is related to the function at the node $\mathbf{x}_I^{(k)}$ where the influence domain covers it. Constructing the shape function with the improved IMLS method, the trial function is

$$u^h(\mathbf{x}^{(k)}, x_3^{(k)}) = \sum_{I=1}^n \Phi_I(\mathbf{x}^{(k)}) u_I^{(k)} = \Phi(\mathbf{x}^{(k)}) \mathbf{u}, \tag{43}$$

where $\Phi(\mathbf{x}^{(k)})$ is the shape function and $\Phi_{I,j}(\mathbf{x}^{(k)})$ is its derivative, they can be expressed as

$$\Phi(\mathbf{x}^{(k)}) = (\Phi_1(\mathbf{x}^{(k)}), \Phi_2(\mathbf{x}^{(k)}), \dots, \Phi_n(\mathbf{x}^{(k)})) = \mathbf{v}^T(\mathbf{x}^{(k)}) + \mathbf{b}^T(\mathbf{x}^{(k)}) \mathbf{A}_x^{-1}(\mathbf{x}^{(k)}) \mathbf{B}_x(\mathbf{x}^{(k)}), \tag{44}$$

$$\Phi_{I,j}(\mathbf{x}^{(k)}) = v(\mathbf{x}^{(k)} - \mathbf{x}_I^{(k)})_j + \mathbf{b}_j^T \mathbf{A}_x^{-1}(\mathbf{B}_x)_I + \mathbf{b}^T (\mathbf{A}_x^{-1})_j (\mathbf{B}_x)_I + \mathbf{b}^T \mathbf{A}_x^{-1}(\mathbf{B}_x)_{Ij}. \tag{45}$$

The following equations can be obtained from Eqs. (41) and (43),

$$\frac{\partial^2 u(\mathbf{x}^{(k)}, x_3^{(k)})}{\partial x_3^2} = \frac{\partial^2}{\partial x_3^2} \sum_{I=1}^n \Phi_I(\mathbf{x}^{(k)}) u(\mathbf{x}_I^{(k)}, x_3^{(k)}) = \sum_{I=1}^n \Phi_I(\mathbf{x}^{(k)}) \frac{\partial^2 u(\mathbf{x}_I^{(k)}, x_3^{(k)})}{\partial x_3^2} = \Phi(\mathbf{x}^{(k)}) \mathbf{u}'', \tag{46}$$

$$Lu(\mathbf{x}^{(k)}, x_3^{(k)}) = \sum_{I=1}^n \begin{bmatrix} \frac{\partial}{\partial x_1} \\ \frac{\partial}{\partial x_2} \end{bmatrix} \Phi_I(\mathbf{x}^{(k)}) u(\mathbf{x}_I^{(k)}, x_3^{(k)}) = \sum_{I=1}^n \mathbf{B}_I(\mathbf{x}^{(k)}) u(\mathbf{x}_I^{(k)}, x_3^{(k)}) = \mathbf{B}(\mathbf{x}^{(k)}) \mathbf{u}, \tag{47}$$

where

$$\mathbf{u}'' = \left(\frac{\partial^2 u(\mathbf{x}_1^{(k)}, x_3^{(k)})}{\partial x_3^2}, \frac{\partial^2 u(\mathbf{x}_2^{(k)}, x_3^{(k)})}{\partial x_3^2}, \dots, \frac{\partial^2 u(\mathbf{x}_n^{(k)}, x_3^{(k)})}{\partial x_3^2} \right)^T, \tag{48}$$

$$\int_{\Omega^{(k)}} \delta[\mathbf{B}(\mathbf{x}^{(k)}) \mathbf{u}]^T \cdot [\mathbf{B}(\mathbf{x}^{(k)}) \mathbf{u}] d\Omega^{(k)} = \delta \mathbf{u}^T \cdot \left[\int_{\Omega^{(k)}} \mathbf{B}^T(\mathbf{x}^{(k)}) \mathbf{B}(\mathbf{x}^{(k)}) d\Omega^{(k)} \right] \cdot \mathbf{u} = \delta \mathbf{u}^T \cdot \mathbf{K} \cdot \mathbf{u}, \tag{54}$$

$$\mathbf{B}(\mathbf{x}^{(k)}) = (\mathbf{B}_1(\mathbf{x}^{(k)}), \mathbf{B}_2(\mathbf{x}^{(k)}), \dots, \mathbf{B}_n(\mathbf{x}^{(k)})), \tag{49}$$

$$\mathbf{B}_I(\mathbf{x}^{(k)}) = \begin{bmatrix} \Phi_{I,1}(\mathbf{x}^{(k)}) \\ \Phi_{I,2}(\mathbf{x}^{(k)}) \end{bmatrix}. \tag{50}$$

Substituting Eqs. (43), (46) and (47) into Eq. (40) yields

$$\int_{\Omega^{(k)}} \delta[\Phi(\mathbf{x}^{(k)}) \mathbf{u}] \cdot [\Phi(\mathbf{x}^{(k)}) \mathbf{u}''] d\Omega^{(k)} - \int_{\Omega^{(k)}} \delta[\mathbf{B}(\mathbf{x}^{(k)}) \mathbf{u}]^T \cdot [\mathbf{B}(\mathbf{x}^{(k)}) \mathbf{u}] d\Omega^{(k)} - \int_{\Omega^{(k)}} \delta[\Phi(\mathbf{x}^{(k)}) \mathbf{u}] \cdot b^{(k)} d\Omega^{(k)} - \int_{\Gamma_q^{(k)}} \delta[\Phi(\mathbf{x}^{(k)}) \mathbf{u}] \cdot \bar{q}^{(k)} d\Gamma^{(k)} = 0. \tag{51}$$

In order to obtain the discretized equations, every term of Eq. (51) need to be discussed.

The first integration term in Eq. (51) is

$$\int_{\Omega^{(k)}} \delta[\Phi(\mathbf{x}^{(k)}) \mathbf{u}] \cdot [\Phi(\mathbf{x}^{(k)}) \mathbf{u}''] d\Omega^{(k)} = \delta \mathbf{u}^T \cdot \left[\int_{\Omega^{(k)}} \Phi^T(\mathbf{x}^{(k)}) \Phi(\mathbf{x}^{(k)}) d\Omega^{(k)} \right] \cdot \mathbf{u}'' = \delta \mathbf{u}^T \cdot \mathbf{C} \cdot \mathbf{u}'', \tag{52}$$

where

$$\mathbf{C} = \int_{\Omega^{(k)}} \Phi^T(\mathbf{x}^{(k)}) \Phi(\mathbf{x}^{(k)}) d\Omega^{(k)}. \tag{53}$$

The second integration term in Eq. (51) is

$$K = \int_{\Omega^{(k)}} B^T(x^{(k)})B(x^{(k)})d\Omega^{(k)}. \tag{55}$$

The third integration term in Eq. (51) is

$$\int_{\Omega^{(k)}} \delta[\Phi(x^{(k)})u] \cdot b^{(k)} d\Omega^{(k)} = \delta u^T \cdot \int_{\Omega^{(k)}} \Phi^T(x^{(k)})b^{(k)} d\Omega^{(k)} = \delta u^T \cdot F_1, \tag{56}$$

where

$$F_1 = \int_{\Omega^{(k)}} \Phi^T(x^{(k)})b^{(k)} d\Omega^{(k)}. \tag{57}$$

The fourth integration term in Eq. (51) is

$$\int_{\Gamma_q^{(k)}} \delta[\Phi(x^{(k)})u] \cdot \bar{q}^{(k)} d\Gamma^{(k)} = \delta u^T \cdot \int_{\Gamma_q^{(k)}} \Phi^T(x^{(k)})\bar{q}^{(k)} d\Gamma^{(k)} = \delta u^T \cdot F_2, \tag{58}$$

where

$$F_2 = \int_{\Gamma_q^{(k)}} \Phi^T(x^{(k)})\bar{q}^{(k)} d\Gamma^{(k)}. \tag{59}$$

Substituting Eqs. (52), (54), (56) and (58) into Eq. (51) yields

$$\delta u^T \cdot (Cu'' - Ku - F_1 - F_2) = 0. \tag{60}$$

Because δu^T is arbitrary, Eq. (60) can be written as the following differential equation

$$Cu'' - Ku = \hat{F}, \tag{61}$$

where

$$\hat{F} = F_1 + F_2. \tag{62}$$

$L - 1$ nodes are evenly distributed in Ω along the splitting direction x_3 for solving Eq. (61) (that is, Ω is split into L layers along this direction, a total of $L + 1$ planes), except for the first and last plane, we can obtain the numerical solutions of all nodes on the middle planes $x_3 = x_3^{(k)}$, ($k = 1, 2, \dots, L - 1$).

We make $u(x_3^{(k)})$, ($k = 1, 2, \dots, L - 1$) respectively represent the approximate values of potential on every middle plane $x_3 = x_3^{(k)}$, ($k = 1, 2, \dots, L - 1$), let

$$u(x_3^{(1)}) = u^{(1)}, \tag{63}$$

$$u(x_3^{(2)}) = u^{(2)}, \tag{64}$$

⋮

$$u(x_3^{(L-1)}) = u^{(L-1)}, \tag{65}$$

the first and last plane

$$u(x_3^{(0)}) = u^{(0)} = u(a), \tag{66}$$

$$u(x_3^{(L)}) = u^{(L)} = u(c). \tag{67}$$

The following formulas can be obtained by using the finite difference method

$$u^{(k)} \approx \frac{u^{(k-1)} - 2u^{(k)} + u^{(k+1)}}{(\Delta x_3)^2}, \quad (k = 1, 2, \dots, L - 1). \tag{68}$$

Equation (61) can be written as

$$C \cdot \frac{u^{(0)} - 2u^{(1)} + u^{(2)}}{(\Delta x_3)^2} + Ku^{(1)} = \hat{F}^{(1)}, \quad (x_3 = x_3^{(1)}), \tag{69}$$

$$C \cdot \frac{u^{(1)} - 2u^{(2)} + u^{(3)}}{(\Delta x_3)^2} + Ku^{(2)} = \hat{F}^{(2)}, \quad (x_3 = x_3^{(2)}), \tag{70}$$

$$C \cdot \frac{u^{(2)} - 2u^{(3)} + u^{(4)}}{(\Delta x_3)^2} + Ku^{(3)} = \hat{F}^{(3)}, \quad (x_3 = x_3^{(3)}), \tag{71}$$

⋮

$$C \cdot \frac{u^{(L-2)} - 2u^{(L-1)} + u^{(L)}}{(\Delta x_3)^2} + Ku^{(L-1)} = \hat{F}^{(L-1)}, \quad (x_3 = x_3^{(L-1)}). \tag{72}$$

Equations (69)–(72) can be transformed into matrix as

$$\frac{1}{(\Delta x_3)^2} \begin{bmatrix} H & C & & & & & \\ C & H & C & & & & \\ & C & H & C & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & C & H & C & \\ & & & & C & H & \\ & & & & & C & H \end{bmatrix} \begin{bmatrix} u^{(1)} \\ u^{(2)} \\ u^{(3)} \\ \vdots \\ u^{(L-2)} \\ u^{(L-1)} \end{bmatrix} = \begin{bmatrix} \hat{F}^{(1)} - \frac{Cu^{(0)}}{(\Delta x_3)^2} \\ \hat{F}^{(2)} \\ \hat{F}^{(3)} \\ \vdots \\ \hat{F}^{(L-2)} \\ \hat{F}^{(L-1)} - \frac{Cu^{(L)}}{(\Delta x_3)^2} \end{bmatrix}, \tag{73}$$

where

$$H = -2C - (\Delta x_3)^2 K. \tag{74}$$

Let

$$E = \frac{1}{(\Delta x_3)^2} \begin{bmatrix} H & C & & & & & \\ C & H & C & & & & \\ & C & H & C & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & C & H & C & \\ & & & & C & H & \\ & & & & & C & H \end{bmatrix}, \tag{75}$$

$$U = (u^{(1)T}, u^{(2)T}, u^{(3)T}, \dots, u^{(L-2)T}, u^{(L-1)T})^T, \tag{76}$$

$$G = \left(\left(\hat{F}^{(1)} - \frac{Cu^{(0)}}{(\Delta x_3)^2} \right)^T, \hat{F}^{(2)T}, \hat{F}^{(3)T}, \dots, \hat{F}^{(L-2)T}, \left(\hat{F}^{(L-1)} - \frac{Cu^{(L)}}{(\Delta x_3)^2} \right)^T \right)^T, \tag{77}$$

Equation (73) can be written as

$$EU = G. \tag{78}$$

The solutions of Eq. (77) are the potential values of nodes on the middle planes $x_3 = x_3^{(k)}, (k = 1, 2, \dots, L - 1)$, then we can obtain the approximate value of the potential of any node with the linear interpolation method, let $\Delta x_3 = x_3^{(k+1)} - x_3^{(k)}, u(x_3)$ can be expressed as

$$u(x_3) = \frac{x_3 - x_3^{(k)}}{\Delta x_3} u(x_3^{(k)}) + \frac{x_3^{(k+1)} - x_3}{\Delta x_3} u(x_3^{(k+1)}). \tag{79}$$

It is the IDSEFG method for 3D potential problems.

4 Algorithm implementation process

The IDSEFG method proposed in this paper is used to solve 3D potential problems, it is necessary to select an appropriate dimension splitting direction. In the solution process, the interpolating EFG method requires nodes to be placed evenly in these related subdomains, and the influence domain of these nodes must contain the entire 2D subdomain, which is a necessary condition. In this paper, Gauss integration is used to calculate the numerical integration, and the layout of the integration grid is regular, which provides convenience for establishing Gauss integration points. The integration grid is rectangular.

The algorithm implementation process of the IDSEFG method for 3D potential problems is:

1. According to the different boundary conditions and the governing equation, an appropriate dimension splitting direction need to be selected, then enter the corresponding parameters;
2. Determining each variable and coordinate system of the corresponding 2D potential problem, M nodes $x_I^{(k)}, I = (1, 2, \dots, M)$ are distributed on $\Omega^{(k)}$ and its boundary $\Gamma^{(k)} = \Gamma_u^{(k)} \cup \Gamma_q^{(k)}$, then number these nodes according to certain rules, node coordinates can be generated through the program, so as to quickly select the required nodes later;
3. For numerical integration, a background integration grid needs to be set up;
4. Establishing the integral element and corresponding corner information on the Γ_q boundary;
5. Establishing Gauss integration points, using the background integration grid and boundary information in

- Steps 3 and 4 to calculate the coordinates, integration weights, and Jacobian of the integration points;
6. Establishing the Matrix C , Matrix K and the first term F_1 of Array F ;

- A. Looping through all background integration grids;
 - a) In every integration grid, looping all Gauss points;
 - b) If it is verified that the Gauss point is inside the 2D subdomains $\Omega^{(k)}$, then continue to Steps (c)–(f), if not, proceed directly to Step (f) to end the integration point loop;
 - c) Finding these node numbers located in domain of the Gauss point $x_Q^{(k)}$;
 - d) Calculating shape function $\Phi(x_Q^{(k)})$ and its derivative $\Phi_{I,j}(x_Q^{(k)})$;
 - e) Calculating the contribution values of Gauss points to the Matrix C , K and the first term F_1 of Array F with Eqs. (53), (55), (57);
 - f) Finishing the Gauss integration points loop.

- B. Finishing the background grids loop.

7. The numerical integration calculation on the boundary of $\Gamma_q^{(k)}$: the process is similar to Step 6, and finally obtain the second term F_2 of the load Array \hat{F} by Eq. (59);
8. Adding the F_1 obtained in Step 6 and the F_2 obtained in Step 7 to get \hat{F} ;
9. Substituting displacement boundary conditions directly into the Matrix K and Array \hat{F} ;
10. Using Eqs. (74), (75), (77) and Matrix C , Matrix K , Array \hat{F} to obtain the Matrix E and Matrix G ;
11. Substituting the Matrix E and Matrix G into the system of Eq. (78), the numerical solution U of the potential function at each node in the original 3D problem domain Ω can be obtained;
12. According to the obtained numerical solution U and Eq. (79), the numerical solution of the potential function at any point in Ω can be obtained by fitting;
13. Outputting the potential values.

5 Numerical examples

It is necessary to verify the effectiveness and efficiency of the IDSEFG method for 3D potential problems, the influence domain of the weight function is rectangular, 4×4 Gaussian

integration points are chosen in each integration cell. Four numerical examples are calculated using this method. By selecting different node distributions, the number of dimension splitting layers and the influence domain parameter d_{\max} , its convergence is discussed and the calculation results are compared with the results of the IIEFG method and the analytical solution.

The following relative error is defined to compare the accuracy of the results of the IIEFG method and the IDSEFG method.

$$u - u_{L^2(\Omega)}^{\text{hrel}} = \frac{u - u^h_{L^2(\Omega)}}{u_{L^2(\Omega)}}, \tag{80}$$

where

$$u - u^h_{L^2(\Omega)} = \left[\int_{\Omega} (u - u^h)^2 d\Omega \right]^{1/2}, \tag{81}$$

is the L^2 -norm of the error.

5.1 A Laplace’s equation in a 3D cube

The governing equation is

$$\nabla^2 u = \frac{\partial^2 u(x)}{\partial x_1^2} + \frac{\partial^2 u(x)}{\partial x_2^2} + \frac{\partial^2 u(x)}{\partial x_3^2} = 0, \quad (x \in \Omega), \tag{82}$$

with boundary conditions

$$u = \sin(\pi x_2) \sin(\pi x_3), \quad (x_1 = 0), \tag{83}$$

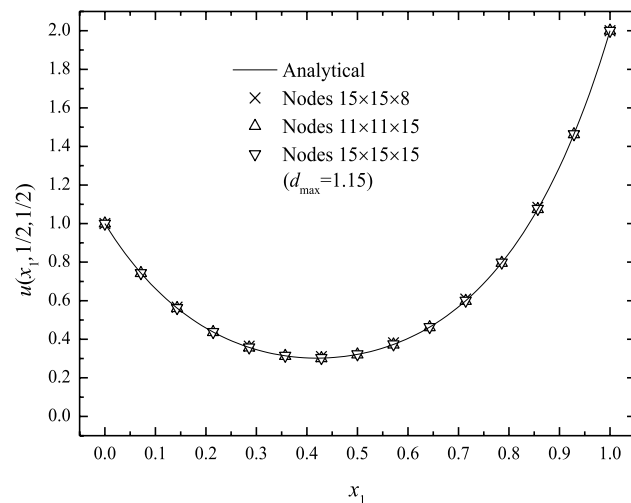


Fig. 1 The numerical results of IDSEFG method for different number of nodes

$$u = 2 \sin(\pi x_2) \sin(\pi x_3), \quad (x_1 = 1), \tag{84}$$

$$u = 0, \quad (x_2 = 0, x_2 = 1, x_3 = 0, x_3 = 1). \tag{85}$$

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

The analytical solution is

$$u = \frac{\sin(\pi x_2) \sin(\pi x_3)}{\sinh(\pi \sqrt{2})} \left[2 \sinh(\pi \sqrt{2} x_1) + \sinh(\pi \sqrt{2} (1 - x_1)) \right]. \tag{86}$$

According to the boundary conditions and the form of governing equation, direction x_1 is chosen as splitting direction. Splitting the problem domain Ω into L subdomains uniformly along x_1 , then M nodes should be distributed in 2D subdomain on the plane $O^{(k)} x_2 x_3$.

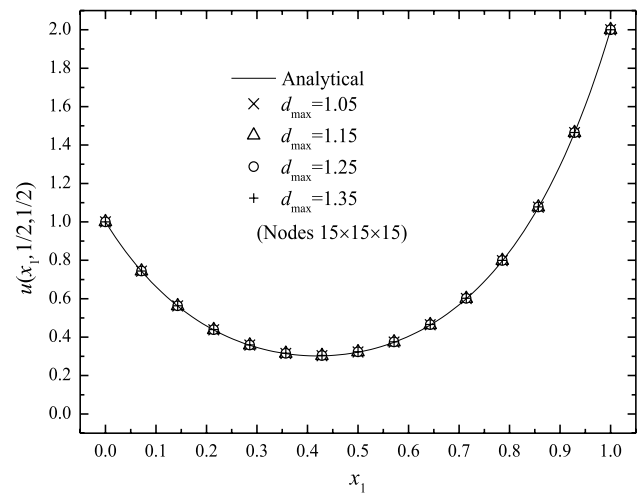


Fig. 2 The numerical results of IDSEFG method for different d_{\max}

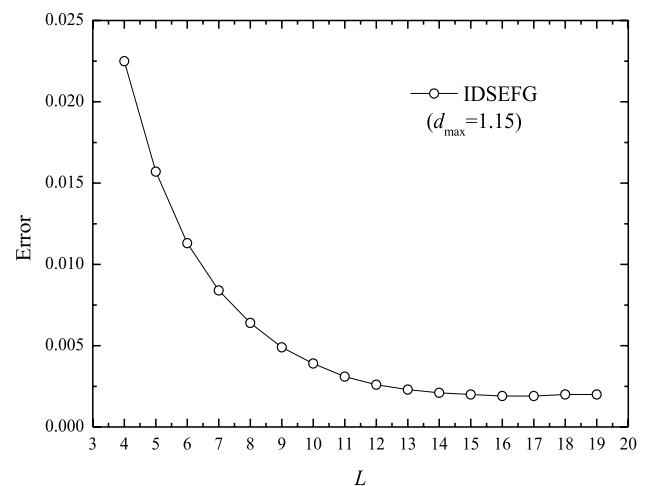


Fig. 3 The relative error for different number of splitting layers

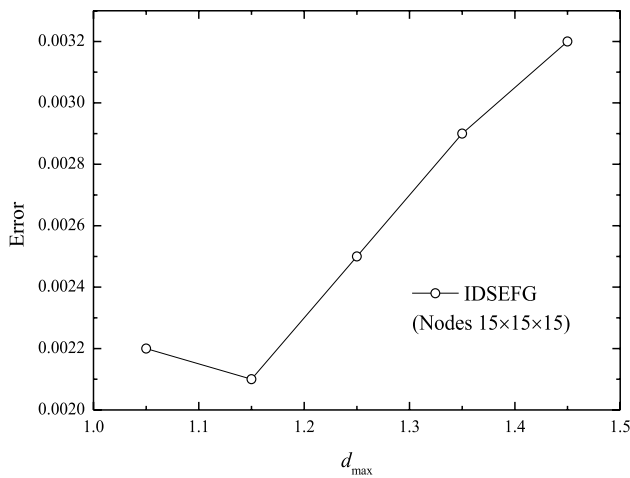


Fig. 4 The relative error for different d_{max}

Table 1 The comparison of relative error and CPU time of the IDSEFG method and IIEFG method with different node distributions

Nodes	Error		CPU time (s)	
	IDSEFG (%)	IEFG (%)	IDSEFG	IEFG
9×9×9	1.01	1.85	1.54	30.14
10×10×10	0.73	1.3	2.18	33.89
11×11×11	0.54	1.24	2.92	35.64
12×12×12	0.41	1.23	4.02	35.91
13×13×13	0.32	0.67	5.18	37.7
14×14×14	0.26	1.32	6.22	40.8
15×15×15	0.21	1.35	7.88	46.8
16×16×16	0.17	1.88	10.29	53.11
17×17×17	0.14	1.25	12.98	59.16
18×18×18	0.12	0.86	15.64	67.41
19×19×19	0.1	0.25	19.1	81.23

The following results are given to verify the effectiveness and convergence of the IDSEFG method.

Figure 1 is the result of the IDSEFG method with different node distributions for this numerical example. It can be seen that in the case of different node distributions but d_{max} is taken to be 1.15, the numerical solutions obtained by this method are very close to the analytical solution.

d_{max} is a parameter of the size of the influence domain of a node, and it is directly related to the computational accuracy of the numerical results. The value of d_{max} should ensure the problem domain covered by the influence domains of the weight functions of all nodes, and the inverse matrix in the improved IMLS method can be obtained. If the value of d_{max} is too small, the singularity of the equation systems maybe exists. Therefore, we need to find an appropriate value of d_{max} for different examples according to the numerical results.

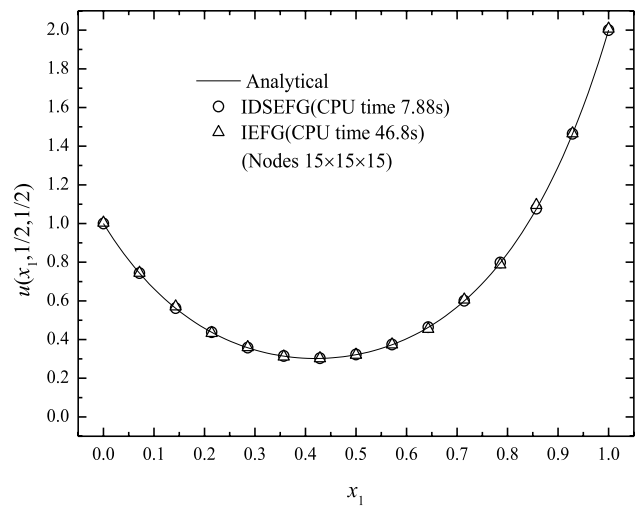


Fig. 5 The numerical results of IDSEFG method and IEFG method at $(x_1, 1/2, 1/2)$

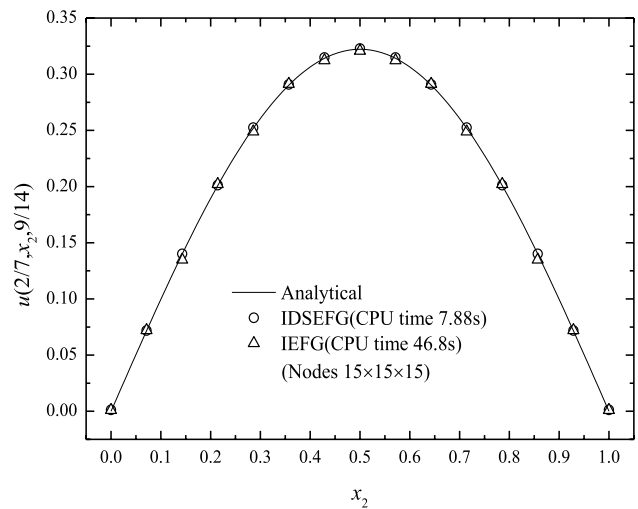


Fig. 6 The numerical results of IDSEFG method and IEFG method at $(2/7, x_2, 9/14)$

Figure 2 shows the result that the IDSEFG method with different d_{max} . When the node distribution is $15 \times 15 \times 15$ but different d_{max} , it can be seen that the relative error of the numerical solutions is very small compared with the analytical solution.

Figure 3 shows the results obtained by the IDSEFG method in this example when a different number of splitting layer is selected. $(L + 1) \times 15 \times 15$ is chosen as the node distribution on Ω . We can see that the relative error gradually decreases and tends to be stable as the number of splitting layer increases, indicating that the method is convergent.

Figure 4 shows the relative error of the IDSEFG method with different d_{max} . We found that when $d_{max} = 1.15$ and

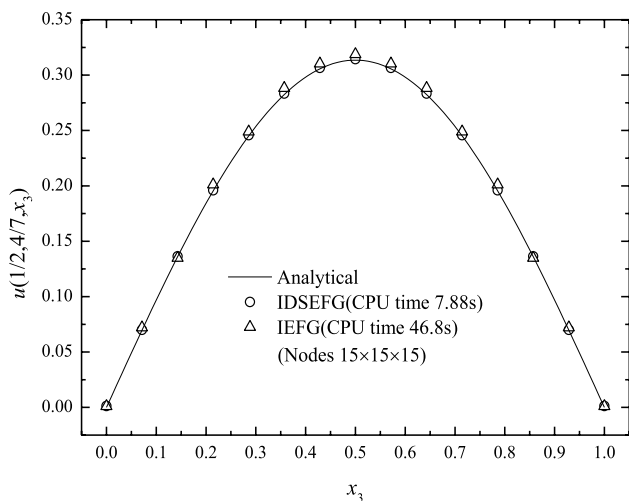


Fig. 7 The numerical results of IDSEFG method and IIEFG method at $(1/2, 4/7, x_3)$

$15 \times 15 \times 15$ node distribution is used, the relative error reaches the minimum.

Table 1 shows that when different node distributions are chosen, the CPU time and calculation accuracy of the IDSEFG method and the IIEFG method. It can be concluded that the IDSEFG method is significantly better than the IIEFG method in terms of calculation accuracy and efficiency when the same node distribution is chosen.

Further analysis of the results, the IIEFG method has a tendency to improve the calculation accuracy when the number of nodes continues to increase, but the results are not stable and the CPU time keeps extending. In contrast, the IDSEFG method will improve the calculation accuracy when the number of nodes gradually increases. We can select an appropriate number of nodes according to the accuracy requirements to save calculation time. In this example, $15 \times 15 \times 15$ node distribution is used, then we got a relatively high calculation accuracy while taking relatively little time to calculate.

If the smaller relative error 0.25% obtained by the IIEFG method is compared with the relative error 0.26% obtained by the IDSEFG method, we found that the calculation efficiency of the IDSEFG method is about 13 times that of the IIEFG method when the calculation accuracy is similar.

Figures 5, 6, 7 are the comparison results of the IDSEFG, IIEFG method and analytical solution at $(x_1, 1/2, 1/2)$, $(2/7, x_2, 9/14)$, $(1/2, 4/7, x_3)$ respectively. The node distribution of the two methods is $15 \times 15 \times 15$, the d_{\max} of the IDSEFG method is 1.15, and the d_{\max} of the IIEFG method is 1.3, penalty factor $\alpha = 1.0 \times 10^3$. It can be concluded that the accuracy of the results of the IDSEFG method is good enough, and there is still small error on individual nodes of the IIEFG method.

To sum up, the IDSEFG method is not only significantly higher in accuracy than the IIEFG method, but its efficiency has increased several times or even more than ten times.

5.2 A Poisson’s equation in a cuboid

The governing equation is

$$\nabla^2 u(x) = f(x), \quad (x \in \Omega), \tag{87}$$

$$f(x) = -\left(\frac{1}{a^2} + \frac{1}{b^2} + \frac{1}{c^2}\right)\pi^2 \sin \frac{\pi x_1}{a} \sin \frac{\pi x_2}{b} \sin \frac{\pi x_3}{c}, \tag{88}$$

with the boundary condition

$$u(x) = 0, \quad (x \in \Gamma). \tag{89}$$

In the example, the values of a, b and c are 1, 2, 3, $\Omega = [0, 1] \times [0, 2] \times [0, 3]$ is the problem domain. The analytical solution is

$$u(x) = \sin \pi x_1 \sin \frac{\pi x_2}{2} \sin \frac{\pi x_3}{3}. \tag{90}$$

Because the solution domain is a cuboid with different lengths in three directions, it is necessary to consider that different splitting directions may affect the calculation accuracy and efficiency.

First, according to the length of the problem domain in the three directions of x_1, x_2 , and x_3 are 1, 2, and 3, respectively, using $9 \times 17 \times 25$ node distribution.

5.2.1 x_1 as the splitting direction

In 2D subdomain on the plane $O^{(k)}x_2x_3$, the node distribution is 17×25 , and 9 nodes are distributed in the splitting direction x_1 , $d_{\max} = 1.91$, running the program to get a relative error of 0.96% and a CPU time of 13.81 s.

5.2.2 x_2 as the splitting direction

In 2D subdomain on the plane $O^{(k)}x_1x_3$, the node distribution is 9×25 , and 17 nodes are distributed in the splitting direction x_2 , $d_{\max} = 1.15$, running the program to get a relative error of 0.80% and a CPU time of 10.18 s.

5.2.3 x_3 as the splitting direction

In 2D subdomain on the plane $O^{(k)}x_1x_2$, the node distribution is 9×17 , and 25 nodes are distributed in the splitting direction x_3 , $d_{\max} = 1.15$, running the program to get a relative error of 0.95% and a CPU time of 9.65 s.

It can be seen from the above results that when the node distribution is the same, selecting different splitting

Table 2 The comparison of relative error and CPU time of the IDSEFG method at different splitting layers

Splitting layers (L)	Error (%)	CPU time (s)
6	1.0	2.98
7	0.98	3.42
8	0.97	3.75
9	0.97	4.27
10	0.96	4.71
11	0.96	5.17
12	0.96	5.53
13	0.96	5.88
14	0.96	6.4
15	0.95	6.67
16	0.95	6.95
17	0.95	7.31
18	0.95	7.78
19	0.95	8.21
20	0.95	8.62
21	0.95	9.04
22	0.95	9.43
23	0.95	9.65

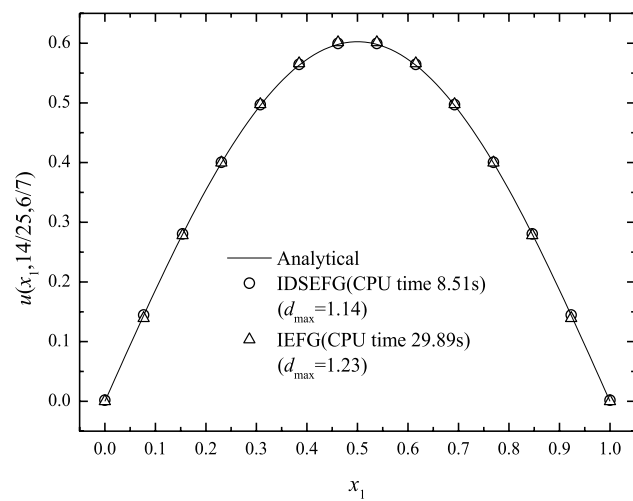


Fig. 8 The numerical results of IDSEFG method and IIEFG method at $(x_1, 14/25, 6/7)$

directions will get different calculation accuracy and efficiency, so it is necessary to select an appropriate splitting direction to obtain better calculation results. In this example, the results when the splitting direction is x_2 or x_3 are significantly better than the results obtained by choosing direction x_1 , but there are both advantages and disadvantages in accuracy and efficiency with choosing x_2 and x_3 , we need to continue the discussion.

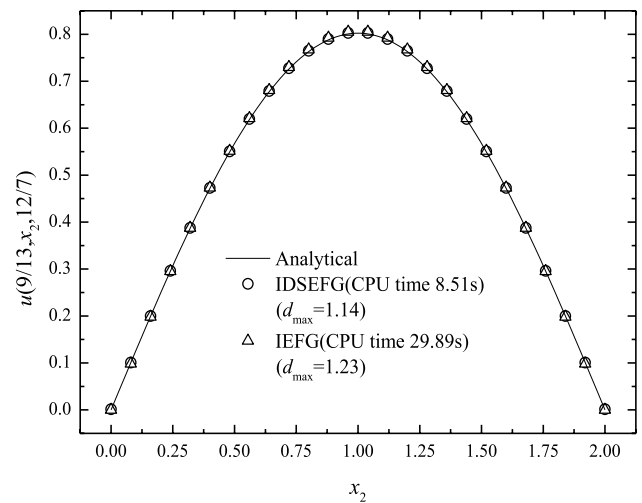


Fig. 9 The numerical results of IDSEFG method and IIEFG method at $(9/13, x_2, 12/7)$

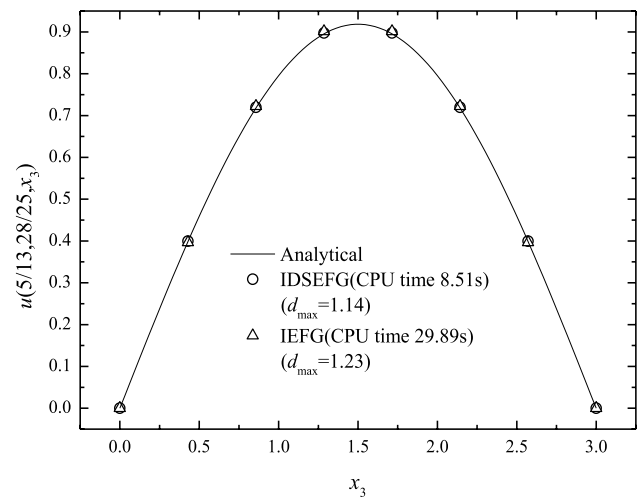


Fig. 10 The numerical results of IDSEFG method and IIEFG method at $(5/13, 28/25, x_3)$

Table 2 shows the results of the IDSEFG method with different splitting layers in this example when x_3 is chosen as the splitting direction, 9×17 node distribution is used in 2D subdomain on the plane $O^{(k)}x_1x_2$. From this table, we can see that with the continuous increase of the number of splitting layers, the calculation accuracy is slightly improved and then the relative error is stabilized at 0.95%, and the calculation time continues to increase. Therefore, the number of splitting layers has a relatively small impact on the calculation accuracy for this example, but it has a greater impact on the CPU time. In general, choosing a direction with long length as the splitting direction can save time and get a better result.

To sum up, we select x_3 as the splitting direction finally, the node distribution in 2D subdomain on the

plane $O^{(k)}x_1x_2$ is 14×26 , and 8 nodes are distributed on the splitting direction, $d_{\max} = 1.14$, then the relative error is 0.28% and CPU time is 8.51 s. The node distribution of the IIEFG method is also $14 \times 26 \times 8$, $d_{\max} = 1.23$, penalty factor $\alpha = 4.5 \times 10^6$, then the relative error we got is 0.68% and the CPU time is 29.89 s.

Figures 8, 9, 10 are the comparison results of IDSEFG, IIEFG method and analytical solution at $(x_1, 14/25, 6/7)$, $(9/13, x_2, 12/7)$, $(5/13, 28/25, x_3)$ respectively. The node distribution of the two methods is $14 \times 26 \times 8$, the d_{\max} of the IDSEFG method is 1.14 and the d_{\max} of the IIEFG method is 1.22. It can be concluded that the results obtained by the two methods are very close to the analytical solution, but the calculation accuracy and calculation efficiency of the IDSEFG method are 2.43 times and 3.51 times that of the IIEFG method, respectively, indicating that the IDSEFG method has significant advantages.

5.3 A Laplace’s equation in a 3D cube

The governing equation is

$$\nabla^2 u(\mathbf{x}) = 0, \quad (\mathbf{x} \in \Omega), \tag{91}$$

with the boundary conditions

$$\frac{\partial u(0, x_2, x_3)}{\partial x_1} = \frac{\partial u(1, x_2, x_3)}{\partial x_1} = 0, \tag{92}$$

$$\frac{\partial u(x_1, 0, x_3)}{\partial x_2} = \frac{\partial u(x_1, 1, x_3)}{\partial x_2} = 0, \tag{93}$$

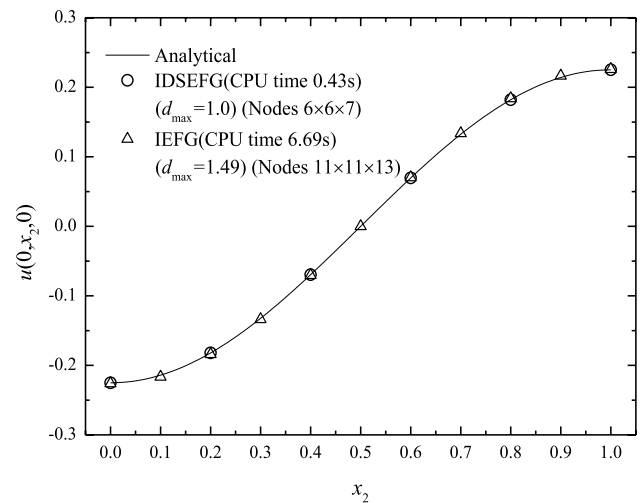


Fig. 12 The numerical results of IDSEFG method and IIEFG method at $(0, x_2, 0)$

$$\frac{\partial u(x_1, x_2, 0)}{\partial x_3} = \cos(\pi x_1) \cos(\pi x_2), \tag{94}$$

$$\frac{\partial u(x_1, x_2, 1)}{\partial x_3} = 0. \tag{95}$$

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

$$u(\mathbf{x}) = \left[\frac{\sinh(\sqrt{2}\pi x_3)}{\sqrt{2}\pi} - \frac{\cosh(\sqrt{2}\pi x_3)}{\sqrt{2}\pi \tanh(\sqrt{2}\pi)} \right] \cos(\pi x_1) \cos(\pi x_2). \tag{96}$$

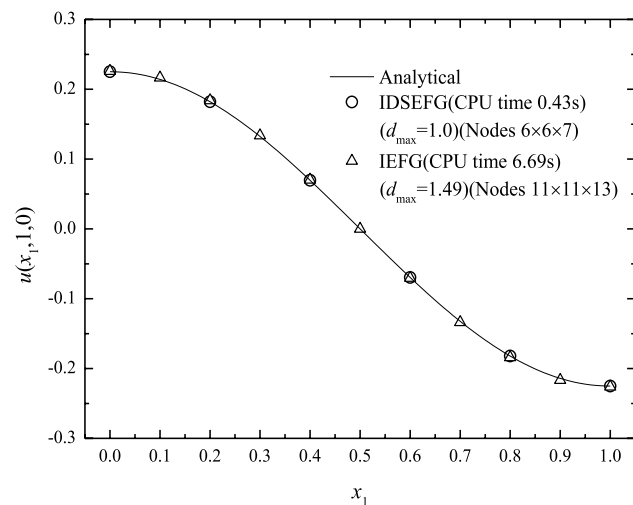


Fig. 11 The numerical results of IDSEFG method and IIEFG method at $(x_1, 1, 0)$

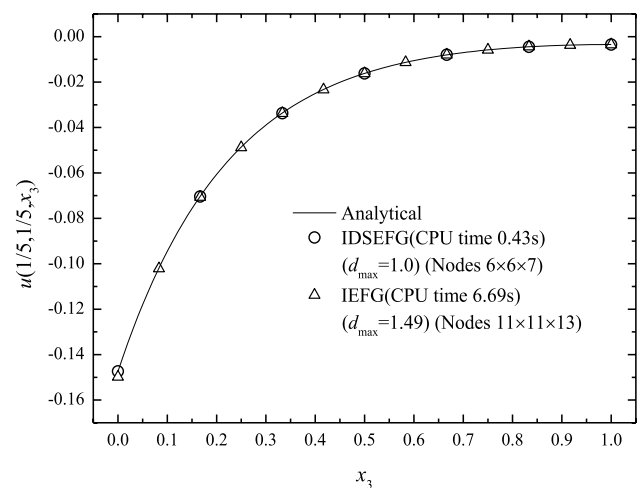


Fig. 13 The numerical results of IDSEFG method and IIEFG method at $(1/5, 1/5, x_3)$

The splitting direction is direction x_3 , $6 \times 6 \times 7$ node distribution is chosen and d_{\max} is 1.0. Running the program to get a relative error 0.074% and a CPU time 0.43 s of the IDSEFG method. As a comparison, the node distribution of the IIEFG method is $11 \times 11 \times 13$, d_{\max} is taken as 1.49, then the relative error is 0.36% and the CPU time is 6.69 s.

Figures 11, 12, 13 are the comparison results of the IDSEFG, IIEFG method and analytical solution at $(x_1, 1, 0)$, $(0, x_2, 0)$, $(1/5, 1/5, x_3)$ respectively. It can be concluded that the numerical results of two methods are close to the analytical solution, but the calculation efficiency of the IDSEFG method is 15.6 times that of the IIEFG method, and even if the IDSEFG method has fewer nodes, the calculation accuracy is still improved by 386.5% that of the IIEFG method.

5.4 A Laplace’s equation in a half circular cylinder

The governing equation is

$$\nabla^2 u(\mathbf{x}) = 0, \quad (r \in [1, 2], \theta \in [0, \pi], x_3 \in [0, 1]), \quad (97)$$

with following boundary conditions

$$u(1, \theta, x_3) = \sin \theta + x_3, \quad (98)$$

$$u(2, \theta, x_3) = x_3, \quad (99)$$

$$u(r, 0, x_3) = x_3, \quad (100)$$

$$u(r, \pi, x_3) = x_3, \quad (101)$$

$$u(r, \theta, 0) = \frac{4}{3} \left(\frac{1}{r} - \frac{r}{4} \right) \sin \theta, \quad (102)$$

$$u(r, \theta, 1) = \frac{4}{3} \left(\frac{1}{r} - \frac{r}{4} \right) \sin \theta + 1. \quad (103)$$

The analytical solution is

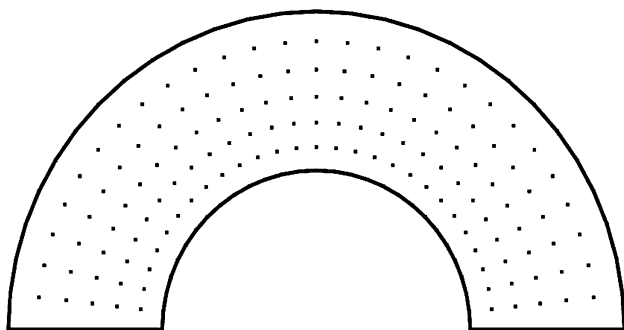


Fig. 14 The node distribution in the 2D subdomain of a half-torus

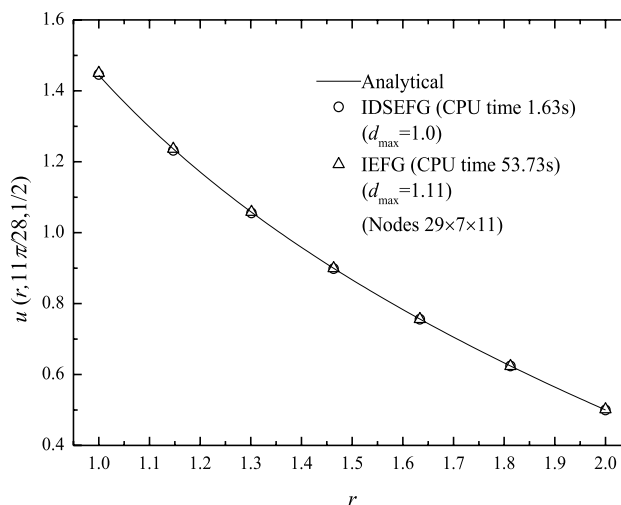


Fig. 15 The numerical results of IDSEFG method and IIEFG method along direction r

$$u(r, \theta, x_3) = \frac{4}{3} \left(\frac{1}{r} - \frac{r}{4} \right) \sin \theta + x_3. \quad (104)$$

The splitting direction is direction x_3 , $29 \times 7 \times 11$ is the node distribution, and $d_{\max} = 1.0$. Splitting 11 subdomains uniformly in the direction x_3 , the node distribution on the 2D half-circle subdomain is 29×7 as shown in Fig. 14.

The relative error and the CPU time obtained by the IDSEFG method are 0.14% and 1.63 s. With the same node distribution and $d_{\max} = 1.11$, penalty factor $\alpha = 1.0 \times 10^4$, the relative error and the CPU time of the IIEFG method are 0.21% and 53.73 s.

Figures 15, 16, 17 are the comparisons of the numerical solutions of the IDSEFG, IIEFG method and the analytical

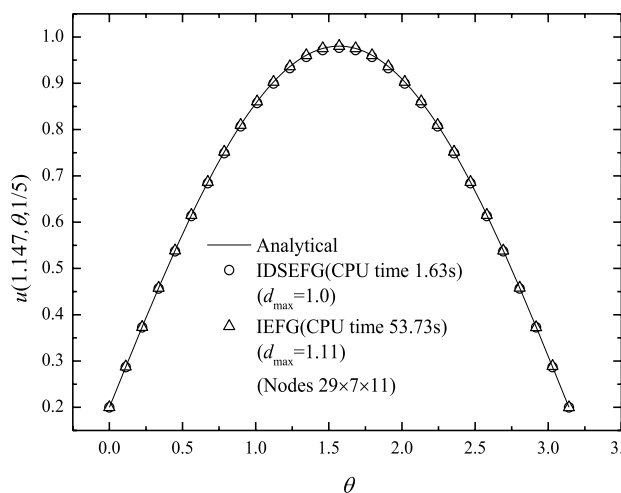


Fig. 16 The numerical results of IDSEFG method and IIEFG method along direction theta

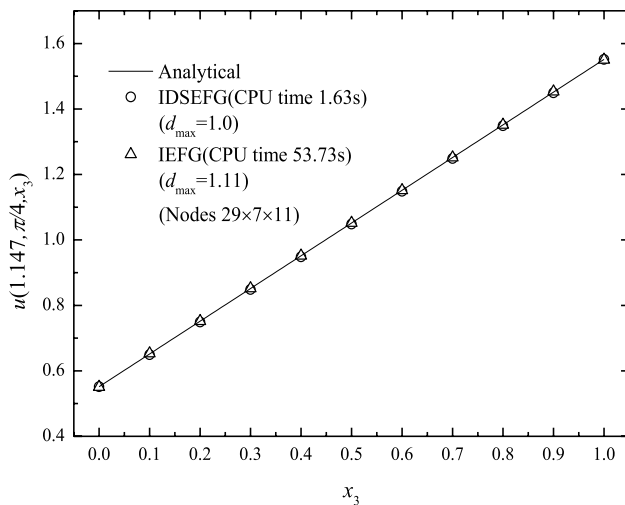


Fig. 17 The numerical results of IDSEFG method and IIEFG method along direction x_3

solutions in three directions. The numerical results of the IIEFG method and the IDSEFG method are very accurate, but we can see that the running time of the IDSEFG method is only 3.03% of the IIEFG method. This result once illustrates the excellent computational efficiency of the IDSEFG method again.

6 Conclusions

In this paper, the IDSEFG method for 3D potential problems is established. The key of the IDSEFG method is to split a 3D solution domain into many related 2D subdomains. The shape function is constructed by the improved IMLS method on the 2D subdomain, and the discretized equations are obtained by the Galerkin weak form based on dimension splitting method. The discrete equations on these 2D subdomains are coupled by the finite difference method. Finally, the numerical solution of the potential of any node can be obtained. The number of undetermined coefficients of the approximation function constructed by the improved IMLS method is less than that of the MLS approximation, which improves the computational efficiency. The essential boundary conditions can be enforced directly because the IDSEFG method satisfies the property of the Kronecker delta function. Comparing the results of IDSEFG method and IIEFG method with analytical solution through four numerical examples, it is proved that both methods are effective and further shows that the IDSEFG method improves the calculation efficiency and calculation

accuracy. At the same time, the last numerical example shows that the IDSEFG method is also applicable to the ring-like problem domain.

We have discussed how to choose the splitting direction in Eg. 2 because its problem domain is a cuboid with different lengths in three directions. In the Example, we finally found that if we choose a direction with a long length as the splitting direction can save CPU time and obtain better results. In other examples, we chose the direction with more complicated boundary conditions as the splitting direction. During the test of the examples, we found that if the boundary conditions of the two-dimensional problem after the dimension splitting are complicated, it will not only bring some difficulties to program code but also reduce the efficiency. Choosing the direction with more complicated boundary conditions as the dimension splitting direction is easier to program code, and better results can be obtained.

Funding National Natural Science Foundation of China, Grant/Award Number: 11571223.

References

- Cheng Y (2015) Meshless method, vol 1. Science Press, Beijing
- Cheng Y (2016) Meshless method, vol 2. Science Press, Beijing
- Belytschko T, Krongauz Y, Organ D et al (1996) Meshless methods: an overview and recent developments. *Comput Methods Appl Mech Eng* 139(1–4):3–47
- Chen L, Cheng Y (2010) The complex variable reproducing kernel particle method for elasto-plasticity problems. *Sci China Phys Mech Astron* 53(5):954–965
- Chen L, Cheng Y (2018) The complex variable reproducing kernel particle method for bending problems of thin plates on elastic foundations. *Comput Mech* 62:67–80
- Lancaster P, Salkauskas K (1981) Surfaces generated by moving least squares methods. *Math Comput* 37:141–158
- Cheng J (2020) Data analysis of the factors influencing the industrial land leasing in Shanghai based on mathematical models. *Math Probl Eng* 2020:9346863
- Cheng J (2020) Analyzing the factors influencing the choice of the government on leasing different types of land uses: Evidence from Shanghai of China. *Land Use Policy* 90:104303
- Cheng J (2021) Analysis of commercial land leasing of the district governments of Beijing in China. *Land Use Policy* 100:104881
- Cheng J (2021) Residential land leasing and price under public land ownership. *J Urban Plan Dev* 147(2):05021009
- Cheng Y, Chen M (2003) A boundary element-free method for linear elasticity. *Acta Mech Sin* 35(2):181–186
- Cheng Y, Peng M (2005) Boundary element-free method for elastodynamics. *Sci China Ser G Phys Mech Astron* 48(6):641–657
- Cheng Y, Li J (2005) A meshless method with complex variables for elasticity. *Acta Phys Sin* 54(10):4463–4471
- Cheng Y, Li J (2006) A complex variable meshless method for fracture problems. *Sci China Ser G* 49(1):46–59

15. Bai F, Li D, Wang J, Cheng Y (2012) An improved complex variable element-free Galerkin method for two-dimensional elasticity problems. *Chin Phys B* 21(2):020204
16. Zhang Z, Li D, Cheng Y, Liew K (2012) The improved element-free Galerkin method for three-dimensional wave equation. *Acta Mech Sin* 28(3):808–818
17. Zhang Z, Wang J, Cheng Y, Liew K (2013) The improved element-free Galerkin method for three-dimensional transient heat conduction problems. *Sci China Phys Mech Astron* 56(8):1568–1580
18. Zhang Z, Hao S, Liew K, Cheng Y (2013) The improved element-free Galerkin method for two-dimensional elastodynamics problems. *Eng Anal Bound Elem* 37(12):1576–1584
19. Peng M, Li R, Cheng Y (2014) Analyzing three-dimensional viscoelasticity problems via the improved element-free Galerkin (IEFG) method. *Eng Anal Bound Elem* 40:104–113
20. Yu S, Peng M, Cheng H, Cheng Y (2019) The improved element-free Galerkin method for three-dimensional elastoplasticity problems. *Eng Anal Bound Elem* 104:215–224
21. Zheng G, Cheng Y (2020) The improved element-free Galerkin method for diffusional drug release problems. *Int J Appl Mech* 12(8):2050096
22. Deng Y, Liu C, Peng M, Cheng Y (2015) The interpolating complex variable element-free Galerkin method for temperature field problems. *Int J Appl Mech* 7(2):1550017
23. Cheng Y, Liu C, Bai F, Peng M (2015) Analysis of elastoplasticity problems using an improved complex variable element-free Galerkin method. *Chin Phys B* 24(10):100202
24. Wang B, Ma Y, Cheng Y (2019) The improved complex variable element-free Galerkin method for bending problem of thin plate on elastic foundations. *Int J Appl Mech* 11(10):1950105
25. Ren H, Cheng Y, Zhang W (2010) An interpolating boundary element-free method (IBEFM) for elasticity problems. *Sci China Phys Mech Astron* 53(4):758–766
26. Ren H, Cheng Y (2011) The interpolating element-free Galerkin (IEFG) method for two-dimensional elasticity problems. *Int J Appl Mech* 3(4):735–758
27. Ren H, Cheng Y (2012) The interpolating element-free Galerkin (IEFG) method for two-dimensional potential problems. *Eng Anal Bound Elem* 36(5):873–880
28. Cheng Y, Bai F, Peng M (2014) A novel interpolating element-free Galerkin (IEFG) method for two-dimensional elastoplasticity. *Appl Math Model* 38:5187–5197
29. Cheng Y, Bai F, Liu C, Peng M (2016) Analyzing nonlinear large deformation with an improved element-free Galerkin method via the interpolating moving least-squares method. *Int J Comput Mater Sci Eng* 5(4):1650023
30. Wu Q, Peng P, Cheng Y (2021) The interpolating element-free Galerkin method for elastic large deformation problems. *Sci China Technol Sci* 64:364–374
31. Liu D, Cheng Y (2019) The interpolating element-free Galerkin (IEFG) method for three-dimensional potential problems. *Eng Anal Bound Elem* 108:115–123
32. Liu D, Cheng Y (2020) The interpolating element-free Galerkin method for three-dimensional transient heat conduction problems. *Results Phys* 19:103477
33. Wu Q, Liu F, Cheng Y (2020) The interpolating element-free Galerkin method for three-dimensional elastoplasticity problems. *Eng Anal Bound Elem* 115:156–167
34. Wang J, Sun F, Cheng Y, Huang A (2014) Error estimates for the interpolating moving least-squares method. *Appl Math Comput* 245:321–342
35. Sun F, Wang J, Cheng Y, Huang A (2015) Error estimates for the interpolating moving least-squares method in n-dimensional space. *Appl Numer Math* 98:79–105
36. Wang J, Sun F, Cheng Y (2012) An improved interpolating element-free Galerkin method with a nonsingular weight function for two-dimensional potential problems. *Chin Phys B* 21(9):090204
37. Sun F, Liu C, Cheng Y (2014) An improved interpolating element-free Galerkin method based on nonsingular weight functions. *Math Probl Eng* 2014:323945
38. Sun F, Wang J, Cheng Y (2013) An improved interpolating element-free Galerkin method for elasticity. *Chin Phys B* 22(12):120203
39. Sun F, Wang J, Cheng Y (2016) An improved interpolating element-free Galerkin method for elastoplasticity via nonsingular weight functions. *Int J Appl Mech* 8(8):1650096
40. Wang J, Hao S, Cheng Y (2014) The error estimates of the interpolating element-free Galerkin method for two-point boundary value problems. *Math Probl Eng* 2014:641592
41. Liu F, Cheng Y (2018) The improved element-free Galerkin method based on the nonsingular weight functions for inhomogeneous swelling of polymer gels. *Int J Appl Mech* 10(4):1850047
42. Liu F, Cheng Y (2018) The improved element-free Galerkin method based on the nonsingular weight functions for elastic large deformation problems. *Int J Comput Mater Sci Eng* 7(3):1850023
43. Liu F, Wu Q, Cheng Y (2019) A meshless method based on the nonsingular weight functions for elastoplastic large deformation problems. *Int J Appl Mech* 11(1):1950006
44. Cheng H, Peng M, Cheng Y (2017) A hybrid improved complex variable element-free Galerkin method for three-dimensional potential problems. *Eng Anal Bound Elem* 84:52–62
45. Cheng H, Peng M, Cheng Y (2017) A fast complex variable element-free Galerkin method for three-dimensional wave propagation problems. *Int J Appl Mech* 9(6):1750090
46. Cheng H, Peng M, Cheng Y (2018) The dimension splitting and improved complex variable element-free Galerkin method for 3-dimensional transient heat conduction problems. *Int J Numer Meth Eng* 114(3):321–345
47. Cheng H, Peng M, Cheng Y (2018) A hybrid improved complex variable element-free Galerkin method for three-dimensional advection-diffusion problems. *Eng Anal Bound Elem* 97:39–54
48. Cheng H, Peng M, Cheng Y, Meng Z (2020) The hybrid complex variable element-free Galerkin method for 3D elasticity problems. *Eng Struct* 219:110835
49. Meng Z, Cheng H, Ma L, Cheng Y (2018) The dimension splitting element-free Galerkin method for three-dimensional potential problems. *Acta Mech Sin* 34(3):462–474
50. Meng Z, Cheng H, Ma L, Cheng Y (2019) The dimension splitting element-free Galerkin method for 3D transient heat conduction problems. *Sci China Phys Mech Astron* 62(4):040711
51. Meng Z, Cheng H, Ma L, Cheng Y (2019) The hybrid element-free Galerkin method for three-dimensional wave propagation problems. *Int J Numer Meth Eng* 117(1):15–37
52. Peng P, Wu Q, Cheng Y (2020) The dimension splitting reproducing kernel particle method for three-dimensional potential problems. *Int J Numer Meth Eng* 121:146–164
53. Peng P, Cheng Y (2020) Analyzing three-dimensional transient heat conduction problems with the dimension splitting reproducing kernel particle method. *Eng Anal Bound Elem* 121:180–191
54. Peng P, Cheng Y (2021) Analyzing three-dimensional wave propagation with the hybrid reproducing kernel particle method based on the dimension splitting method. *Eng Comput*. <https://doi.org/10.1007/s00366-020-01256-9>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.