



The dimension split element-free Galerkin method for three-dimensional potential problems

Z. J. Meng^{1,2} · H. Cheng³ · L. D. Ma⁴ · Y. M. Cheng¹

Received: 2 August 2017 / Revised: 30 October 2017 / Accepted: 21 November 2017 / Published online: 2 February 2018

© The Chinese Society of Theoretical and Applied Mechanics; Institute of Mechanics, Chinese Academy of Sciences and Springer-Verlag GmbH Germany, part of Springer Nature 2018

Abstract

This paper presents the dimension split element-free Galerkin (DSEFG) method for three-dimensional potential problems, and the corresponding formulae are obtained. The main idea of the DSEFG method is that a three-dimensional potential problem can be transformed into a series of two-dimensional problems. For these two-dimensional problems, the improved moving least-squares (IMLS) approximation is applied to construct the shape function, which uses an orthogonal function system with a weight function as the basis functions. The Galerkin weak form is applied to obtain a discretized system equation, and the penalty method is employed to impose the essential boundary condition. The finite difference method is selected in the splitting direction. For the purposes of demonstration, some selected numerical examples are solved using the DSEFG method. The convergence study and error analysis of the DSEFG method are presented. The numerical examples show that the DSEFG method has greater computational precision and computational efficiency than the IIEFG method.

Keywords Dimension split method · Improved moving least-squares (IMLS) approximation · Improved element-free Galerkin (IEFG) method · Finite difference method (FDM) · Dimension split element-free Galerkin (DSEFG) method · Potential problem

1 Introduction

The meshless method is an effective tool to solve boundary value problems of partial differential equations in addition to the finite difference method (FDM), finite element method (FEM), and boundary element method (BEM). The main difference between these conventional numerical methods and meshless method is the different approaches to obtain the shape functions. However, once the shape function has been formed, they use the same procedure to obtain the equation

system to get the numerical solution of a problem. FDM, FEM, and BEM rely on meshes, then remeshing must be used to solve some complicated problems, such as extremely large deformations and crack growth problems. The meshless method only needs nodes or particles, and it can solve many complicated physical and engineering problems that cannot be solved well with conventional computational methods [1,2].

The potential problem is an important physical and engineering problem, whose governing equations are commonly Laplace's equations or Poisson's equations. The popular numerical methods solving potential problems include FEM, FDM, BEM, and meshless methods [3–7]. The meshless method is actually a more effective numerical method to solve partial differential equations that govern various physical and engineering phenomena than traditional methods.

In recent years, several meshless methods, such as boundary node method [8–10], element-free Galerkin method [11–18], boundary element-free method [19,20], and meshless local Petro-Galerkin method coupled with FEM [21], were developed to solve potential problems. The improved element-free Galerkin (IEFG) method is an effective mesh-

✉ Y. M. Cheng
ymcheng@shu.edu.cn

¹ Shanghai Institute of Applied Mathematics and Mechanics, Shanghai Key Laboratory of Mechanics in Energy Engineering, Shanghai University, Shanghai 200072, China

² School of Applied Science, Taiyuan University of Science and Technology, Taiyuan 030024, China

³ Department of Civil Engineering, Shanghai University, Shanghai 200444, China

⁴ School of Materials Science and Engineering, Taiyuan University of Science and Technology, Taiyuan 030024, China

less method to solve three-dimensional potential problems, because in the IIEFG method, the improved moving least-squares (IMLS) approximation, in which the orthogonal function system is selected as the basis function, is used to obtain the shape function [12,22]. The algebraic equations system in the IMLS approximation is well-conditioned, and it can be solved without deriving the inverse matrix. There are fewer coefficients in the IMLS approximation than the ones in the moving least-squares (MLS) approximation, which is the basis of the EFG method, then the computational efficiency of the IIEFG method is greater than the EFG method. Using the IIEFG method, the three-dimensional potential problem was effectively solved [12]. Furthermore, the IIEFG method can also be successfully applied to other science and engineering problems. In their work, Zhang et al. [23] employed the IIEFG method for two-dimensional elasticity problems, 2D fracture problems [24], three-dimensional wave equation [25], and two-dimensional elastodynamics problems [26].

The dimension split method (DSM) was proposed by Li and Huang [27]. The essential features of the DSM are that the three-dimensional domain is partitioned by several two-dimensional surfaces into several sub-domains. They used optimal control to deal with the stream-function equations of compressible turbomachinery flows and their finite element approximation [27]. They also applied the DSM for the 3D compressible Navier–Stokes equations in turbomachine [28], the three dimensional rotating Navier–Stokes equations [29], the incompressible Navier–Stokes equations in three dimensions [30], and the linearly elastic shell [31]. Additionally, Hansen and Ostermann [32] presented the DSM for evolution and quasilinear parabolic equations [33]. The dimension split method is actually an efficient and convenient numerical method to solve various problems [34–37]. The work of Cheng et al. [38,39] introduced the DSM into the improved complex variable element-free Galerkin method for solving three-dimensional potential and wave propagation problems. It is obvious that the DSM is different from the traditional domain decomposition method because only two-dimensional problems are solved in each sub-domain without solving a three-dimensional problem, then the computational efficiency is improved greatly.

Three-dimensional potential problems are effectively solved with the dimension split element-free Galerkin method. For these two-dimensional problems, the IMLS approximation is applied to construct the shape function. The Galerkin weak form is applied to obtain a discretized system equation, and the penalty method is employed to impose the essential boundary condition. The finite difference method is selected in the splitting direction. Finally, several numerical examples are solved using the dimension split element-free Galerkin (DSEFG) method. These numerical examples test and verify DSEFG theoretical result. The convergence study and error analysis of

the DSEFG method are presented. The numerical examples show that the DSEFG method has greater computational precision and computational efficiency than the IIEFG method.

2 The basic equations of three-dimensional potential problems with dimension split scheme

Consider the following three-dimensional potential problem

$$\frac{\partial^2 u}{\partial x_1^2} + \frac{\partial^2 u}{\partial x_2^2} + \frac{\partial^2 u}{\partial x_3^2} = b(\mathbf{x}), \quad \mathbf{x} = (x_1, x_2, x_3) \in \Omega, \quad (1)$$

$$u(\mathbf{x}) = \bar{u}(\mathbf{x}), \quad \mathbf{x} \in \Gamma_u, \quad (2)$$

$$q(\mathbf{x}) = \frac{\partial u(\mathbf{x})}{\partial \mathbf{n}} = \bar{q}(\mathbf{x}), \quad \mathbf{x} \in \Gamma_q, \quad (3)$$

where $u(\mathbf{x})$ is an unknown function, $b(\mathbf{x})$ is a known function. In the general case, there exist mixed boundary conditions on the boundary Γ of the domain Ω . Γ_u is one part of the boundary Γ with known potential function $\bar{u}(\mathbf{x})$, and Γ_q is the remaining boundary with known normal derivative $\bar{q}(\mathbf{x})$. Notice that $\Gamma = \Gamma_u \cup \Gamma_q$, $\Gamma_u \cap \Gamma_q = \emptyset$, and \mathbf{n} is the unit outward normal to the boundary Γ .

When using the DSEFG method to solve potential problems, we can choose the splitting direction according to the control equations and boundary conditions, which results in that the process of calculating and programming are easy to achieve.

In this paper, we assume that the problem domain Ω is split into L layers along the direction x_3 , and the distance between adjacent layers is Δx_3 . Then we have $L + 1$ two-dimensional sub-domains $\Omega^{(k)}$, $k = 0, 1, \dots, L$, and

$$\Omega = \bigcup_{k=0}^{L-1} \Omega^{(k)} \times [x_3^{(k)}, x_3^{(k+1)}] \cup \Omega^{(L)}, \quad (4)$$

where

$$a = x_3^{(0)} < x_3^{(1)} \dots < x_3^{(L)} = c, \quad x_3 \in [a, c], \quad (5)$$

$$\Delta x_3 = x_3^{(k+1)} - x_3^{(k)} = (c - a)/L. \quad (6)$$

For a fixed $x_3^{(k)}$, u , and $\frac{\partial^2 u}{\partial x_3^2}$ can be considered the function of x_1 and x_2 . Three-dimensional potential problem is translated into a series of two-dimensional boundary value problems, i.e.

$$\frac{\partial^2 u^{(k)}}{\partial x_1^2} + \frac{\partial^2 u^{(k)}}{\partial x_2^2} = b^{(k)} - \frac{\partial^2 u^{(k)}}{\partial x_3^2}, \quad (x_1, x_2) \in \Omega^{(k)}, \quad x_3 = x_3^{(k)}, \quad (7)$$

The corresponding boundary conditions are

$$u(x_1, x_2) = \bar{u}(x_1, x_2) = \bar{u}(x_1, x_2, x_3^{(k)}), \quad (x_1, x_2) \in \Gamma_u^{(k)}, \tag{8}$$

$$q(x_1, x_2) = \bar{q}(x_1, x_2) = \bar{q}(x_1, x_2, x_3^{(k)}), \quad (x_1, x_2) \in \Gamma_q^{(k)}, \tag{9}$$

where

$$u^{(k)} = u(x_1, x_2, x_3^{(k)}), \tag{10}$$

$$b^{(k)} = b(x_1, x_2, x_3^{(k)}), \tag{11}$$

$u(x_1, x_2)$ is the field potential in the sub-domain $\Omega^{(k)}$, $\bar{u}(x_1, x_2, x_3^{(k)})$ is the given potential on essential boundary $\Gamma_u^{(k)}$, $\bar{q}(x_1, x_2, x_3^{(k)})$ is the given gradient on natural boundary $\Gamma_q^{(k)}$, $\Gamma^{(k)} = \Gamma_u^{(k)} \cup \Gamma_q^{(k)}$, and $\Gamma_u^{(k)} \cap \Gamma_q^{(k)} = \emptyset$.

Then we can solve Eqs. (7)–(9) in the sub-domain $\Omega^{(k)}$ by using the IIEFG method, and in the direction x_3 finite difference method is used to obtain the solution of the original problem, i.e. Eqs. (1)–(3). This is the idea of the DSEFG method for three-dimensional potential problems.

The equivalent functional of Eqs. (7)–(9) is

$$\begin{aligned} \Pi &= \int_{\Omega^{(k)}} \left[u \left(\frac{\partial^2 u}{\partial x_3^2} - b \right) \right] d\Omega^{(k)} \\ &\quad - \int_{\Omega^{(k)}} \frac{1}{2} \left[\left(\frac{\partial u}{\partial x_1} \right)^2 + \left(\frac{\partial u}{\partial x_2} \right)^2 \right] d\Omega^{(k)} \\ &\quad - \int_{\Gamma_q^{(k)}} u \bar{q} d\Gamma^{(k)}. \end{aligned} \tag{12}$$

The shape function of the IMLS approximation does not satisfy the property of the Kronecker delta function; therefore, the essential boundary cannot be directly imposed to Eq. (12). In this paper, the penalty method is used to impose the essential boundary conditions. The modified functional can then be expressed as

$$\Pi^* = \Pi + \frac{\alpha}{2} \int_{\Gamma_u^{(k)}} (u - \bar{u})(u - \bar{u}) d\Gamma^{(k)}, \tag{13}$$

where α is the penalty factor.

Let

$$\delta \Pi^* = 0, \tag{14}$$

the equivalent integral weak form can be obtained as

$$\begin{aligned} &\int_{\Omega^{(k)}} \delta u \cdot \frac{\partial^2 u}{\partial x_3^2} d\Omega^{(k)} - \int_{\Omega^{(k)}} \delta (\mathbf{L}u)^T \cdot (\mathbf{L}u) d\Omega^{(k)} \\ &\quad - \int_{\Omega^{(k)}} \delta u \cdot b d\Omega^{(k)} - \int_{\Gamma_q^{(k)}} \delta u \cdot \bar{q} d\Gamma^{(k)} \\ &\quad + \alpha \int_{\Gamma_u^{(k)}} \delta u \cdot u d\Gamma^{(k)} - \alpha \int_{\Gamma_u^{(k)}} \delta u \cdot \bar{u} d\Gamma^{(k)} = 0, \end{aligned} \tag{15}$$

where

$$\mathbf{L}(\cdot) = \begin{bmatrix} \frac{\partial}{\partial x_1} \\ \frac{\partial}{\partial x_2} \end{bmatrix} (\cdot). \tag{16}$$

3 The dimension split element-free Galerkin method for three-dimensional potential problems

3.1 Improved moving least-squares approximation

M nodes $\mathbf{x}_I^{(k)}$, $I = 1, 2, \dots, M$ are distributed in a two-dimensional sub-domain $\Omega^{(k)}$. The $u(\mathbf{x})$ at node $\mathbf{x}_I^{(k)}$ is represented as

$$u_I = u(\mathbf{x}_I^{(k)}, x_3^{(k)}). \tag{17}$$

$u(\mathbf{x}^{(k)}, x_3^{(k)})$ at any $\mathbf{x}^{(k)} = (x_1^{(k)}, x_2^{(k)})$ is approximated by nodes $\mathbf{x}_I^{(k)}$, $I = 1, 2, \dots, n$ whose influence domains cover the point $\mathbf{x}^{(k)}$.

The IMLS approximation is used to construct shape functions. The approximation of $u(\mathbf{x})$ at point $\mathbf{x}^{(k)}$ in the two-dimensional sub-domain $\Omega^{(k)}$ is denoted as $u^h(\mathbf{x}^{(k)}, x_3^{(k)})$, and the trial function is

$$\begin{aligned} u^h(\mathbf{x}^{(k)}, x_3^{(k)}) &= \sum_{i=1}^m p_i(\mathbf{x}^{(k)}) a_i(\mathbf{x}^{(k)}) \\ &= \mathbf{p}^T(\mathbf{x}^{(k)}) \mathbf{a}(\mathbf{x}^{(k)}), \quad \mathbf{x}^{(k)} \in \Omega^{(k)}, \end{aligned} \tag{18}$$

where $p_i(\mathbf{x}^{(k)})$ is the basis function, $\mathbf{p}^T(\mathbf{x}^{(k)})$ is a vector of basis functions that consist mostly of monomials of the lowest order to ensure minimum completeness, m is the number of terms of the monomials, and $\mathbf{a}(\mathbf{x}^{(k)})$ is a vector of coefficients. For the two-dimensional sub-domain $\Omega^{(k)}$, in this paper the basis function is chosen as the linear basis, i.e.

$$\mathbf{p}^T = (1, x_1, x_2), \quad m = 3. \tag{19}$$

Define a functional

$$J = \sum_{l=1}^n w(\mathbf{x}^{(k)} - \mathbf{x}_l^{(k)}) \left[\sum_{i=1}^m p_i(\mathbf{x}^{(k)}) a_i(\mathbf{x}^{(k)}) - u_l \right]^2, \tag{20}$$

where $\mathbf{x}_l^{(k)}, l = 1, 2, \dots, n$, are nodes with domains of influence that cover the point $\mathbf{x}^{(k)}$, $w(\mathbf{x}^{(k)} - \mathbf{x}_l^{(k)})$ is a weight function with a domain of influence, and $u_l = u(\mathbf{x}_l^{(k)}, x_3^{(k)})$.

From

$$\frac{\partial J}{\partial \mathbf{a}} = 0, \tag{21}$$

we obtain

$$\mathbf{a}(\mathbf{x}^{(k)}) = \mathbf{A}^{-1}(\mathbf{x}^{(k)}) \mathbf{B}(\mathbf{x}^{(k)}) \mathbf{u}, \tag{22}$$

where

$$\mathbf{A}(\mathbf{x}^{(k)}) = \mathbf{P}^T \mathbf{W} \mathbf{P}, \tag{23}$$

$$\mathbf{B}(\mathbf{x}^{(k)}) = \mathbf{P}^T \mathbf{W}, \tag{24}$$

$$\mathbf{u}^T = (u_1, u_2, \dots, u_n), \tag{25}$$

$$\mathbf{P} = \begin{bmatrix} p_1(\mathbf{x}_1^{(k)}) & p_2(\mathbf{x}_1^{(k)}) & \dots & p_m(\mathbf{x}_1^{(k)}) \\ p_1(\mathbf{x}_2^{(k)}) & p_2(\mathbf{x}_2^{(k)}) & \dots & p_m(\mathbf{x}_2^{(k)}) \\ \vdots & \vdots & \ddots & \vdots \\ p_1(\mathbf{x}_n^{(k)}) & p_2(\mathbf{x}_n^{(k)}) & \dots & p_m(\mathbf{x}_n^{(k)}) \end{bmatrix}, \tag{26}$$

$$\mathbf{W} = \begin{bmatrix} w(\mathbf{x}^{(k)} - \mathbf{x}_1^{(k)}) & 0 & \dots & 0 \\ 0 & w(\mathbf{x}^{(k)} - \mathbf{x}_2^{(k)}) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & w(\mathbf{x}^{(k)} - \mathbf{x}_n^{(k)}) \end{bmatrix}. \tag{27}$$

In the IMLS approximation, a weighted orthogonal polynomial set (p_1, p_2, \dots, p_m) is selected as the basis functions, which results in

$$\mathbf{a}(\mathbf{x}^{(k)}) = \mathbf{A}^*(\mathbf{x}^{(k)}) \mathbf{B}(\mathbf{x}^{(k)}) \mathbf{u}, \tag{28}$$

where

$$\mathbf{A}^*(\mathbf{x}^{(k)}) = \begin{bmatrix} \frac{1}{(p_1, p_1)} & 0 & \dots & 0 \\ 0 & \frac{1}{(p_2, p_2)} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \frac{1}{(p_m, p_m)} \end{bmatrix}. \tag{29}$$

Substituting Eq. (28) into Eq. (18), we can obtain

$$\begin{aligned} u^h(\mathbf{x}^{(k)}, x_3^{(k)}) &= \sum_{l=1}^n \Phi_l^*(\mathbf{x}^{(k)}) u_l(\mathbf{x}^{(k)}) \\ &= \Phi^*(\mathbf{x}^{(k)}) \mathbf{u}, \end{aligned} \tag{30}$$

where $\Phi^*(\mathbf{x}^{(k)})$ is a shape function

$$\begin{aligned} \Phi^*(\mathbf{x}^{(k)}) &= (\Phi_1^*(\mathbf{x}^{(k)}), \Phi_2^*(\mathbf{x}^{(k)}), \dots, \Phi_n^*(\mathbf{x}^{(k)})) \\ &= \mathbf{P}^T(\mathbf{x}^{(k)}) \mathbf{A}^*(\mathbf{x}^{(k)}) \mathbf{B}(\mathbf{x}^{(k)}), \end{aligned} \tag{31}$$

$$\Phi_l^*(\mathbf{x}^{(k)}) = \sum_{j=1}^m p_j(\mathbf{x}^{(k)}) \left[\mathbf{A}^*(\mathbf{x}^{(k)}) \mathbf{B}(\mathbf{x}^{(k)}) \right]_{jl}. \tag{32}$$

For the monomial basis function

$$\mathbf{q} = (q_i) = (1, x_1, x_2, x_1^2, x_1 x_2, x_2^2, \dots), \tag{33}$$

the weighted orthogonal basis function set can be written as

$$p_i = q_i - \sum_{k=1}^{i-1} \frac{(q_i, p_k)}{(p_k, p_k)} p_k, \quad i = 1, 2, 3, \dots \tag{34}$$

In this paper, we select cubic spline function

$$w(d) = \begin{cases} \frac{2}{3} - 4d^2 + 4d^3, & d \leq \frac{1}{2}, \\ \frac{4}{3} - 4d + 4d^2 - \frac{4}{3}d^3, & \frac{1}{2} < d \leq 1, \\ 0, & d > 1, \end{cases} \tag{35}$$

as the weight function. Here

$$d = \frac{d_l}{\hat{d}} = \frac{|\mathbf{x} - \mathbf{x}_l|}{\hat{d}}, \tag{36}$$

\hat{d} is the size of the domain of influence of the node \mathbf{x}_l , $\hat{d} = d_{\max} \cdot c_l$, d_{\max} is a scaling parameter, and c_l is the distance between \mathbf{x}_l and the nearest node from it.

3.2 The dimension split element-free Galerkin method for three-dimensional potential problems

To obtain the solution of the original problem, i.e. Eqs. (1)–(3), the Galerkin weak form of the three-dimensional potential problem is used to get the final discretized equations.

From Eq. (30), we have

$$\begin{aligned} \frac{\partial^2 u(\mathbf{x}^{(k)}, x_3^{(k)})}{\partial x_3^2} &= \frac{\partial}{\partial x_3^2} \sum_{l=1}^n \Phi_l^*(\mathbf{x}^{(k)}) u_l(\mathbf{x}^{(k)}, x_3^{(k)}) \\ &= \sum_{l=1}^n \Phi_l^*(\mathbf{x}^{(k)}) \frac{\partial^2 u_l}{\partial x_3^2} = \Phi^*(\mathbf{x}^{(k)}) \mathbf{u}'', \end{aligned} \tag{37}$$

$$\begin{aligned} Lu(\mathbf{x}^{(k)}, x_3^{(k)}) &= \sum_{l=1}^n \left[\frac{\partial}{\partial x_1} \frac{\partial}{\partial x_2} \right] \Phi_l^*(\mathbf{x}^{(k)}) u_l \\ &= \sum_{l=1}^n \mathbf{B}_l(\mathbf{x}^{(k)}) u_l = \mathbf{B}(\mathbf{x}^{(k)}) \mathbf{u}, \end{aligned} \tag{38}$$

where

$$\mathbf{u}'' = \left(\frac{\partial^2 u_1}{\partial x_3^2}, \frac{\partial^2 u_2}{\partial x_3^2}, \dots, \frac{\partial^2 u_n}{\partial x_3^2} \right)^T, \tag{39}$$

$$\mathbf{B}(\mathbf{x}^{(k)}) = (\mathbf{B}_1(\mathbf{x}^{(k)}), \mathbf{B}_2(\mathbf{x}^{(k)}), \dots, \mathbf{B}_n(\mathbf{x}^{(k)})), \tag{40}$$

$$\mathbf{B}_l(\mathbf{x}^{(k)}) = \begin{bmatrix} \Phi_{l,1}^*(\mathbf{x}^{(k)}) \\ \Phi_{l,2}^*(\mathbf{x}^{(k)}) \end{bmatrix}. \tag{41}$$

Substituting Eqs. (30), (37), and (38) into Eq. (15) yields

$$\begin{aligned} &\int_{\Omega^{(k)}} \delta [\Phi^*(\mathbf{x}^{(k)}) \mathbf{u}] \cdot [\Phi^*(\mathbf{x}^{(k)}) \mathbf{u}''] d\Omega^{(k)} \\ &- \int_{\Omega^{(k)}} \delta [\mathbf{B}(\mathbf{x}^{(k)}) \mathbf{u}]^T \cdot [\mathbf{B}(\mathbf{x}^{(k)}) \mathbf{u}] d\Omega^{(k)} \\ &- \int_{\Omega^{(k)}} \delta [\Phi^*(\mathbf{x}^{(k)}) \mathbf{u}] \cdot b d\Omega^{(k)} \\ &- \int_{\Gamma_q^{(k)}} \delta [\Phi^*(\mathbf{x}^{(k)}) \mathbf{u}] \cdot \bar{q} d\Gamma^{(k)} \\ &+ \int_{\Gamma_u^{(k)}} \delta [\Phi^*(\mathbf{x}^{(k)}) \mathbf{u}] \cdot \alpha \cdot [\Phi^*(\mathbf{x}^{(k)}) \mathbf{u}] d\Gamma^{(k)} \\ &- \int_{\Gamma_u^{(k)}} \delta [\Phi^*(\mathbf{x}^{(k)}) \mathbf{u}] \cdot \alpha \bar{u} d\Gamma^{(k)} = 0. \end{aligned} \tag{42}$$

In order to obtain the discrete equations, we discuss each integration term of Eq. (42) as follows.

The first integration term of Eq. (42) is

$$\begin{aligned} &\int_{\Omega^{(k)}} \delta [\Phi^*(\mathbf{x}^{(k)}) \mathbf{u}] \cdot [\Phi^*(\mathbf{x}^{(k)}) \mathbf{u}''] d\Omega^{(k)} \\ &= \delta \mathbf{u}^T \cdot \left[\int_{\Omega^{(k)}} \Phi^{*T}(\mathbf{x}^{(k)}) \Phi^*(\mathbf{x}^{(k)}) d\Omega^{(k)} \right] \cdot \mathbf{u}'' \\ &= \delta \mathbf{u}^T \cdot \mathbf{C} \cdot \mathbf{u}'', \end{aligned} \tag{43}$$

where

$$\mathbf{C} = \int_{\Omega^{(k)}} \Phi^{*T}(\mathbf{x}^{(k)}) \Phi^*(\mathbf{x}^{(k)}) d\Omega^{(k)}. \tag{44}$$

The second integration term of Eq. (42) is

$$\begin{aligned} &\int_{\Omega^{(k)}} \delta [\mathbf{B}(\mathbf{x}^{(k)}) \mathbf{u}]^T \cdot [\mathbf{B}(\mathbf{x}^{(k)}) \mathbf{u}] d\Omega^{(k)} \\ &= \delta \mathbf{u}^T \cdot \left[\int_{\Omega^{(k)}} \mathbf{B}^T(\mathbf{x}^{(k)}) \mathbf{B}(\mathbf{x}^{(k)}) d\Omega^{(k)} \right] \cdot \mathbf{u} \\ &= \delta \mathbf{u}^T \cdot \mathbf{K} \cdot \mathbf{u}, \end{aligned} \tag{45}$$

where

$$\mathbf{K} = \int_{\Omega^{(k)}} \mathbf{B}^T(\mathbf{x}^{(k)}) \mathbf{B}(\mathbf{x}^{(k)}) d\Omega^{(k)}. \tag{46}$$

The third integration term of Eq. (42) is

$$\begin{aligned} &\int_{\Omega^{(k)}} \delta [\Phi^*(\mathbf{x}^{(k)}) \mathbf{u}] \cdot b d\Omega^{(k)} \\ &= \delta \mathbf{u}^T \cdot \int_{\Omega^{(k)}} \Phi^{*T}(\mathbf{x}^{(k)}) b d\Omega^{(k)} \\ &= \delta \mathbf{u}^T \cdot \mathbf{F}_1, \end{aligned} \tag{47}$$

where

$$\mathbf{F}_1 = \int_{\Omega^{(k)}} \Phi^{*T}(\mathbf{x}^{(k)}) b d\Omega^{(k)}. \tag{48}$$

The fourth integration term of Eq. (42) is

$$\begin{aligned} &\int_{\Gamma_q^{(k)}} \delta [\Phi^*(\mathbf{x}^{(k)}) \mathbf{u}] \cdot \bar{q} d\Gamma^{(k)} \\ &= \delta \mathbf{u}^T \cdot \int_{\Gamma_q^{(k)}} \Phi^{*T}(\mathbf{x}^{(k)}) \bar{q} d\Gamma^{(k)} = \delta \mathbf{u}^T \cdot \mathbf{F}_2, \end{aligned} \tag{49}$$

where

$$\mathbf{F}_2 = \int_{\Gamma_q^{(k)}} \Phi^{*T}(\mathbf{x}^{(k)}) \bar{q} d\Gamma^{(k)}. \tag{50}$$

The fifth integration term of Eq. (42) is

$$\begin{aligned} &\int_{\Gamma_u^{(k)}} \delta [\Phi^*(\mathbf{x}^{(k)}) \mathbf{u}] \cdot \alpha \cdot [\Phi^*(\mathbf{x}^{(k)}) \mathbf{u}] d\Gamma^{(k)} \\ &= \delta \mathbf{u}^T \cdot \left[\int_{\Gamma_u^{(k)}} \Phi^{*T}(\mathbf{x}^{(k)}) \alpha \Phi^*(\mathbf{x}^{(k)}) d\Gamma^{(k)} \right] \cdot \mathbf{u} \\ &= \delta \mathbf{u}^T \cdot \mathbf{K}_\alpha \cdot \mathbf{u}, \end{aligned} \tag{51}$$

where

$$\mathbf{K}_\alpha = \alpha \int_{\Gamma_u^{(k)}} \Phi^{*T}(\mathbf{x}^{(k)}) \Phi^*(\mathbf{x}^{(k)}) d\Gamma^{(k)}. \tag{52}$$

The sixth integration term of Eq. (42) is

$$\begin{aligned} &\int_{\Gamma_u^{(k)}} \delta \left[\Phi^*(x^{(k)}) \mathbf{u} \right] \cdot \alpha \bar{u} d\Gamma^{(k)} = \delta \mathbf{u}^T \\ &\quad \cdot \int_{\Gamma_u^{(k)}} \Phi^{*T}(x^{(k)}) \alpha \bar{u} d\Gamma^{(k)} \\ &= \delta \mathbf{u}^T \cdot \mathbf{F}_\alpha, \end{aligned} \tag{53}$$

where

$$\mathbf{F}_\alpha = \alpha \int_{\Gamma_u^{(k)}} \Phi^{*T}(x^{(k)}) \bar{u} d\Gamma^{(k)}. \tag{54}$$

Substituting Eqs. (43), (45), (47), (49), (51), and (53) into Eq. (42), we obtain

$$\delta \mathbf{u}^T \cdot (\mathbf{C} \mathbf{u}'' - \mathbf{K} \mathbf{u} + \mathbf{K}_\alpha \mathbf{u} - \mathbf{F}_1 - \mathbf{F}_2 - \mathbf{F}_\alpha) = \mathbf{0}. \tag{55}$$

As $\delta \mathbf{u}^T$ is arbitrary, we have

$$\mathbf{C} \mathbf{u}'' + \bar{\mathbf{K}} \mathbf{u} = \mathbf{F}, \tag{56}$$

where

$$\bar{\mathbf{K}} = \mathbf{K}_\alpha - \mathbf{K}, \tag{57}$$

$$\mathbf{F} = \mathbf{F}_1 + \mathbf{F}_2 + \mathbf{F}_\alpha. \tag{58}$$

To solve Eq. (56), $L - 1$ points are uniformly inserted to the domain $[a, c]$, and $[a, c]$ in direction x_3 is equally divided into L parts. That is, the domain Ω is divided into $L - 1$ planes in direction x_3 , then we consider the potential of the plane when $x_3 = x_3^{(1)}, x_3^{(2)}, \dots, x_3^{(L-1)}$.

Suppose $\mathbf{u}(x_3^{(1)}), \mathbf{u}(x_3^{(2)}), \dots, \mathbf{u}(x_3^{(L-1)})$ represent the approximation of the potential of the plane when $x_3 = x_3^{(1)}, x_3^{(2)}, \dots, x_3^{(L-1)}$. Let

$$\mathbf{u}(x_3^{(1)}) = \mathbf{u}^{(1)}, \tag{59}$$

$$\mathbf{u}(x_3^{(2)}) = \mathbf{u}^{(2)}, \tag{60}$$

\vdots

$$\mathbf{u}(x_3^{(L-1)}) = \mathbf{u}^{(L-1)}, \tag{61}$$

and

$$\mathbf{u}(x_3^{(0)}) = \mathbf{u}^{(0)} = \mathbf{u}(a), \tag{62}$$

$$\mathbf{u}(x_3^{(L)}) = \mathbf{u}^{(L)} = \mathbf{u}(c), \tag{63}$$

by using the finite difference method, we have

$$\mathbf{u}''^{(k)} \approx \frac{\mathbf{u}^{(k-1)} - 2\mathbf{u}^{(k)} + \mathbf{u}^{(k+1)}}{(\Delta x_3)^2}, \quad k = 1, 2, \dots, L - 1. \tag{64}$$

Then, Eq. (56) can be written as

$$\mathbf{C} \cdot \frac{\mathbf{u}^{(0)} - 2\mathbf{u}^{(1)} + \mathbf{u}^{(2)}}{(\Delta x_3)^2} + \bar{\mathbf{K}} \mathbf{u}^{(1)} = \mathbf{F}^{(1)}, \tag{65}$$

$$\mathbf{C} \cdot \frac{\mathbf{u}^{(1)} - 2\mathbf{u}^{(2)} + \mathbf{u}^{(3)}}{(\Delta x_3)^2} + \bar{\mathbf{K}} \mathbf{u}^{(2)} = \mathbf{F}^{(2)}, \tag{66}$$

$$\mathbf{C} \cdot \frac{\mathbf{u}^{(2)} - 2\mathbf{u}^{(3)} + \mathbf{u}^{(4)}}{(\Delta x_3)^2} + \bar{\mathbf{K}} \mathbf{u}^{(3)} = \mathbf{F}^{(3)}, \tag{67}$$

$$\begin{aligned} &\vdots \\ \mathbf{C} \cdot \frac{\mathbf{u}^{(L-2)} - 2\mathbf{u}^{(L-1)} + \mathbf{u}^{(L)}}{(\Delta x_3)^2} + \bar{\mathbf{K}} \mathbf{u}^{(L-1)} &= \mathbf{F}^{(L-1)}. \end{aligned} \tag{68}$$

The corresponding matrix form is

$$\begin{aligned} &\frac{1}{(\Delta x_3)^2} \begin{bmatrix} \mathbf{H} & \mathbf{C} & & & & & \\ \mathbf{C} & \mathbf{H} & \mathbf{C} & & & & \\ & \mathbf{C} & \mathbf{H} & \mathbf{C} & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & \mathbf{C} & \mathbf{H} & \mathbf{C} & \\ & & & & \mathbf{C} & \mathbf{H} & \end{bmatrix} \begin{bmatrix} \mathbf{u}^{(1)} \\ \mathbf{u}^{(2)} \\ \mathbf{u}^{(3)} \\ \vdots \\ \mathbf{u}^{(L-2)} \\ \mathbf{u}^{(L-1)} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{F}^{(1)} - \frac{\mathbf{C} \mathbf{u}^{(0)}}{(\Delta x_3)^2} \\ \mathbf{F}^{(2)} \\ \mathbf{F}^{(3)} \\ \vdots \\ \mathbf{F}^{(L-2)} \\ \mathbf{F}^{(L-1)} - \frac{\mathbf{C} \mathbf{u}^{(L)}}{(\Delta x_3)^2} \end{bmatrix}, \end{aligned} \tag{69}$$

where

$$\mathbf{H} = -2\mathbf{C} + (\Delta x_3)^2 \bar{\mathbf{K}}. \tag{70}$$

Let

$$\mathbf{E} = \frac{1}{(\Delta x_3)^2} \begin{bmatrix} \mathbf{H} & \mathbf{C} & & & & & \\ \mathbf{C} & \mathbf{H} & \mathbf{C} & & & & \\ & \mathbf{C} & \mathbf{H} & \mathbf{C} & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & \mathbf{C} & \mathbf{H} & \mathbf{C} & \\ & & & & \mathbf{C} & \mathbf{H} & \end{bmatrix}, \tag{71}$$

$$\mathbf{U} = \left[\mathbf{u}^{(1)T}, \mathbf{u}^{(2)T}, \mathbf{u}^{(3)T}, \dots, \mathbf{u}^{(L-2)T}, \mathbf{u}^{(L-1)T} \right]^T, \tag{72}$$

$$\begin{aligned} \mathbf{R} = &\left[\left(\mathbf{F}^{(1)} - \frac{\mathbf{C} \mathbf{u}^{(0)}}{(\Delta x_3)^2} \right)^T, \mathbf{F}^{(2)T}, \mathbf{F}^{(3)T}, \dots, \mathbf{F}^{(L-2)T}, \right. \\ &\left. \left(\mathbf{F}^{(L-1)} - \frac{\mathbf{C} \mathbf{u}^{(L)}}{(\Delta x_3)^2} \right)^T \right]^T. \end{aligned} \tag{73}$$

Equation (69) can be simplified as

$$EU = R. \tag{74}$$

By solving Eq. (74), we can obtain the potential of nodes on each layer $\Omega^{(k)}$, $k = 1, 2, \dots, L - 1$. Then the potential of any nodes in the domain $[a, c]$ can be obtained using linear interpolation method. Let $\Delta x_3 = x_3^{(k+1)} - x_3^{(k)}$, $u(x_3)$ can be expressed as

$$u(x_3) = \frac{x_3 - x_3^{(k)}}{\Delta x_3} u(x_3^{(k)}) + \frac{x_3^{(k+1)} - x_3}{\Delta x_3} u(x_3^{(k+1)}). \tag{75}$$

4 Example problems

Four example problems are selected to demonstrate the effectiveness and advantages of the DSEFG method presented in this paper. The convergence of the DSEFG method is discussed by analyzing the final potential function values under different nodes distribution and different scaling factors. The numerical results of these examples are compared with analytical solutions and the ones of the IIEFG method.

In this section, Gaussian quadrature scheme with 4×4 points is used for numerical integrations on each cell of the background mesh. The cubic spline function is used as the weight function, and the linear basis function is selected.

4.1 Laplace’s equation with Dirichlet boundary conditions on a cube

As the first example, we consider Laplace’s equation

$$\nabla^2 u(x) = \frac{\partial^2 u(x)}{\partial x_1^2} + \frac{\partial^2 u(x)}{\partial x_2^2} + \frac{\partial^2 u(x)}{\partial x_3^2} = 0, \quad x \in \Omega, \tag{76}$$

with boundary conditions

$$u = \sin(\pi x_2) \sin(\pi x_3), \quad x_1 = 0, \tag{77}$$

$$u = 2 \sin(\pi x_2) \sin(\pi x_3), \quad x_1 = 1, \tag{78}$$

$$u = 0, \quad x_2 = 0, x_2 = 1, x_3 = 0, x_3 = 1, \tag{79}$$

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

The analytical solution is

$$u = \frac{\sin(\pi x_2) \sin(\pi x_3)}{\sinh(\pi \sqrt{2})} [2 \sinh(\pi \sqrt{2} x_1) + \sinh(\pi \sqrt{2} (1 - x_1))]. \tag{80}$$

Depending on the boundary conditions, the domain Ω is divided into L equal parts along x_1 direction. Uniform node

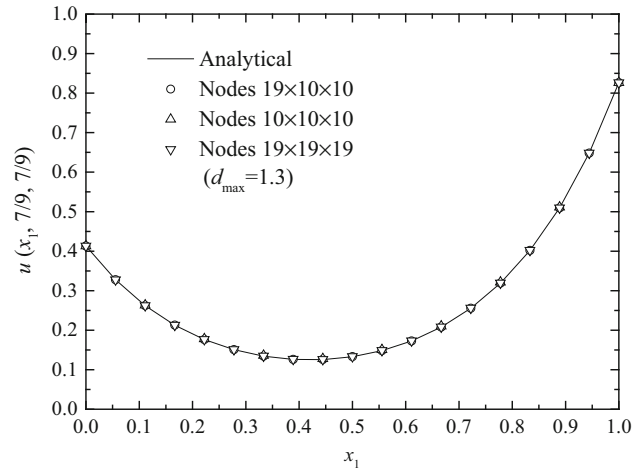


Fig. 1 Results obtained by the DSEFG method with different node distributions

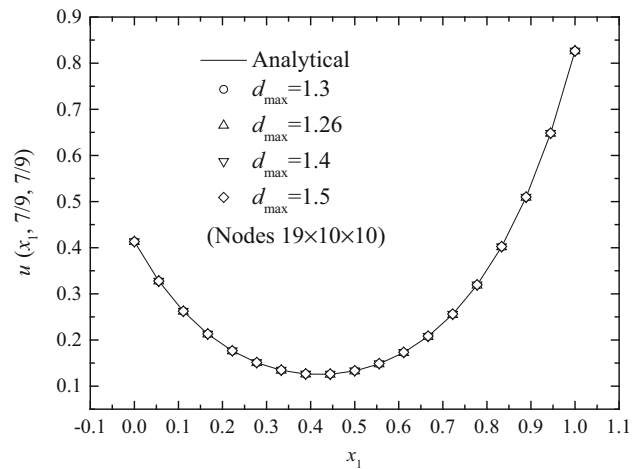


Fig. 2 Results obtained by the DSEFG method with different d_{max}

distribution is used in the plane Ox_2x_3 . That is, the domain Ω can be represented as $\Omega = \bigcup_{k=0}^{L-1} \Omega^{(k)} \times [x_1^{(k)}, x_1^{(k+1)}] \cup \Omega^{(L)}$.

Results obtained with the DSEFG method with different node distributions and different d_{max} along the direction x_1 are shown in Figs. 1 and 2.

The convergence of the DSEFG method with different node distributions and different d_{max} along the direction x_1 are shown in Figs. 3 and 4. In Fig. 3, the domain Ω is divided into L equal parts in direction x_1 , and the node distribution in the plane Ox_2x_3 is 10×10 , that is, the node distribution in the problem domain Ω is $(L + 1) \times 10 \times 10$.

From Figs. 1–4, we can observe that the DSEFG method has greater computational precision under the node distribution $19 \times 10 \times 10$ when $d_{max} = 1.3$.

The relationship between the relative error norm and the CPU time of the DSEFG and IIEFG methods under different

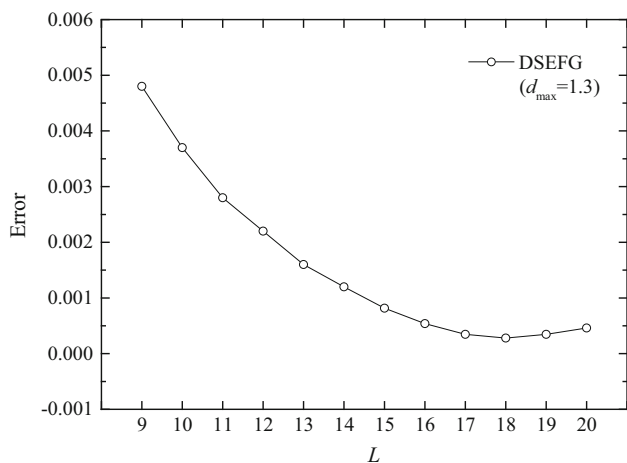


Fig. 3 The convergence of the DSEFG method with different node distributions.

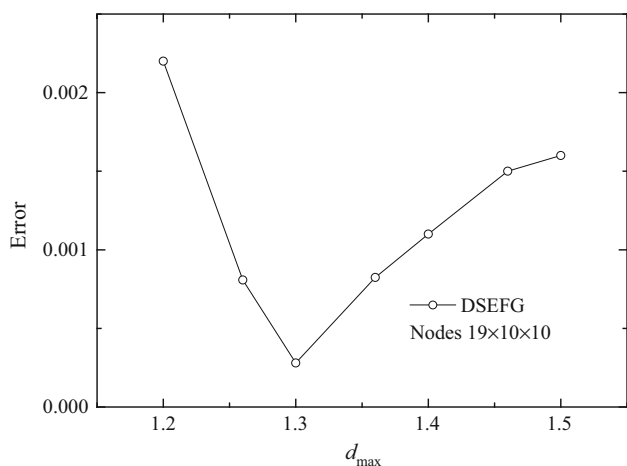


Fig. 4 The convergence of the DSEFG method with different d_{max}

Table 1 The relative error norm and CPU time of the DSEFG and IIEFG methods under different node distributions

Number of nodes	Relative error norm		CPU time (s)	
	DSEFG	IEFG	DSEFG	IEFG
10 × 10 × 10	0.0048	0.013	1.37	53.84
11 × 10 × 10	0.0037	0.0135	1.48	55.83
12 × 10 × 10	0.0028	0.0173	2.064	56.23
13 × 10 × 10	0.0022	0.0126	1.75	56.62
14 × 10 × 10	0.0016	0.0125	1.88	57.65
15 × 10 × 10	0.0012	0.0169	1.95	58.7
16 × 10 × 10	8.17 × 10 ⁻⁴	0.0199	2.1	60.07
17 × 10 × 10	5.40 × 10 ⁻⁴	0.0178	2.2	60.05
18 × 10 × 10	3.46 × 10 ⁻⁴	0.0164	2.4	61.29
19 × 10 × 10	2.80 × 10 ⁻⁴	0.0147	2.5	62.05
20 × 10 × 10	3.46 × 10 ⁻⁴	0.0145	2.579	62.72
21 × 10 × 10	4.60 × 10 ⁻⁴	0.0155	2.7	63.96

Table 2 The relative error norm and CPU time of the DSEFG and IEFG methods under different d_{max}

d _{max}	Relative error norm		CPU time (s)	
	DSEFG	IEFG	DSEFG	IEFG
1.2	0.0022	0.0143	2.5	58.99
1.26	8.08 × 10 ⁻⁴	0.0147	2.49	64.33
1.3	2.80 × 10 ⁻⁴	0.0147	2.5	63.74
1.36	8.24 × 10 ⁻⁴	0.0149	2.65	63.31
1.4	1.10 × 10 ⁻³	0.0152	2.63	64.06
1.46	1.50 × 10 ⁻³	0.016	2.63	68.57
1.5	1.60 × 10 ⁻³	0.0167	3.1	76.67

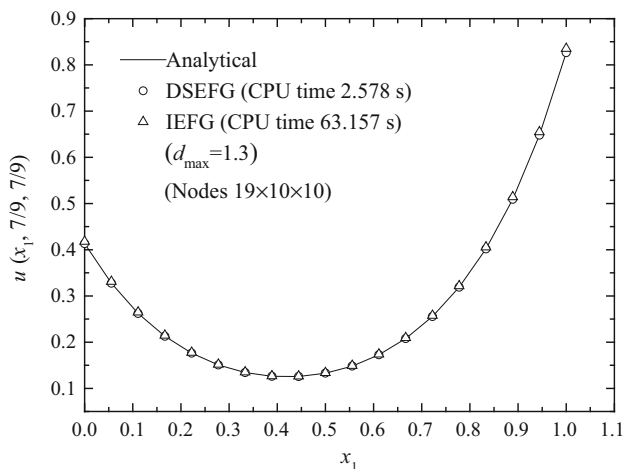


Fig. 5 The results of the DSEFG and IEFG methods along the direction x₁

node distributions and different d_{max} are shown in Tables 1 and 2.

From Tables 1 and 2, we can see that the DSEFG method has higher accuracy and computational efficiency than the IEFG method under the same node distribution and d_{max}.

Figures 5–7 show the numerical solutions obtained by both the IEFG method and DSEFG method along the three axes with the node distribution 19 × 10 × 10 and d_{max} = 1.3. We can observe that the result of the DSEFG method and the IEFG method are in agreement with the analytical solution, and the DSEFG method has greater computational precision and efficiency than the IEFG method.

4.2 Poisson’s equation with Dirichlet boundary conditions

As a second example, we solve a three-dimensional Poisson’s equation

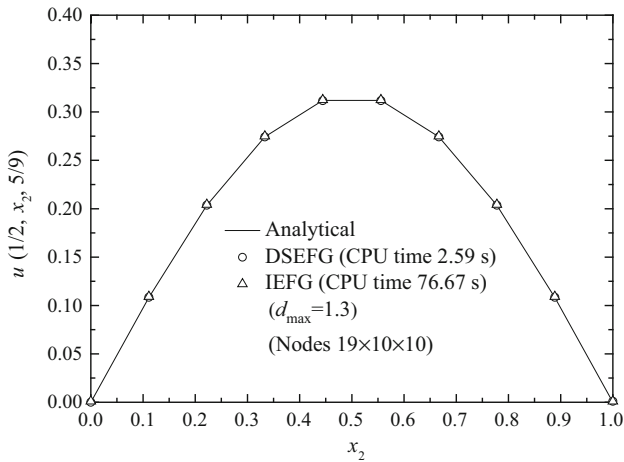


Fig. 6 The results of the DSEFG and IIEFG methods along the direction x_2

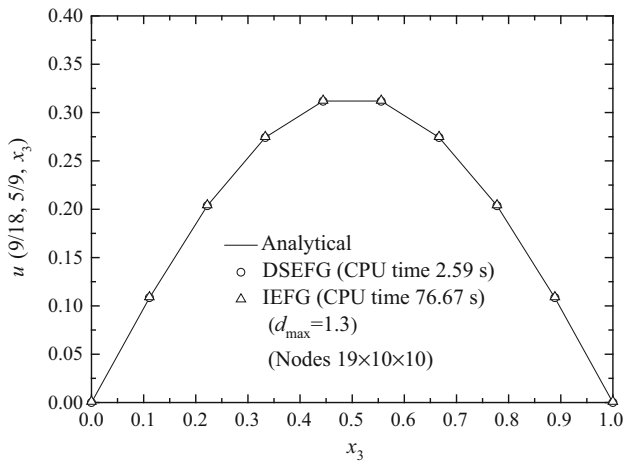


Fig. 7 The results of the DSEFG and IIEFG methods along the direction x_3

$$\nabla^2 u(x) = f(x), \quad x \in \Omega, \tag{81}$$

$$f(x) = -\left(1 + \frac{1}{4} + \frac{1}{9}\right) \pi^2 \sin(\pi x_1) \sin \frac{\pi x_2}{2} \sin \frac{\pi x_3}{3}, \tag{82}$$

with Dirichlet boundary conditions

$$u(x) = 0, \quad x \in \Gamma, \tag{83}$$

where $\Omega = [0, 1] \times [0, 2] \times [0, 3]$.

The analytical solution of this problem is

$$u(x) = \sin(\pi x_1) \sin \frac{\pi x_2}{2} \sin \frac{\pi x_3}{3}. \tag{84}$$

Using the IIEFG method to solve this problem, $9 \times 21 \times 17$ regular nodes are distributed in the problem domain Ω and $d_{\max} = 1.22$. The relative error is 0.0025 and CPU time is 68.3 s.

Then using the DSEFG method to solve this problem, and three cases in which different splitting direction is selected are discussed.

- (1) The splitting direction is x_1 . For the first case, the problem domain Ω is divided into 8 planes equally along the direction x_1 . And on each plane Ox_2x_3 , 21×17 nodes are regularly chosen. It means that the integral node distribution is $9 \times 21 \times 17$. And $d_{\max} = 2.1$, $\alpha = 3.0 \times 10^3$. Then the relative error is 0.0091 and the CPU time is 7.06 s.
- (2) The splitting direction is x_2 . For the second case, the problem domain Ω is divided into 20 planes equally along the direction x_2 . And on each plane Ox_1x_3 , 9×17 nodes are regularly chosen. This means that the integral node distribution is also $9 \times 21 \times 17$ and $d_{\max} = 1.7$, $\alpha = 3.0 \times 10^3$. Then the relative error is 0.0016 and the CPU time is 5.24 s.
- (3) The splitting direction is x_3 . For the third case, the problem domain Ω is divided into 16 planes equally along the direction x_3 . On each plane Ox_1x_2 , 9×21 nodes are regularly chosen. This means that the integral node distribution is also $9 \times 21 \times 17$. And $d_{\max} = 1.13$, $\alpha = 3.0 \times 10^3$. Then the relative error is 0.0013 and the CPU time is 4.55 s.

From the discussion above, we can see that splitting direction will influence the computational accuracy. Thus, we should select the apposite splitting direction according to the control equations and boundary conditions.

In this paper, we choose x_3 as the splitting direction and the better result can be obtained. Figures 8–10 show the numerical solutions obtained with the IIEFG method and the DSEFG method along the three axes under the same node distribution $9 \times 21 \times 17$. From these figures, we can observe that the numerical results obtained with the IIEFG method and the DSEFG method are in agreement with the analytical ones. The CPU time of the DSEFG is less than the one for the IIEFG method.

4.3 Laplace’s equation with Neumann boundary conditions on a cube

As a third example, we study Laplace’s equation

$$\nabla^2 u(x) = 0, \quad x \in \Omega, \tag{85}$$

with Neumann boundary conditions

$$\frac{\partial u(0, x_2, x_3)}{\partial x_1} = \frac{\partial u(1, x_2, x_3)}{\partial x_1} = 0, \tag{86}$$

$$\frac{\partial u(x_1, 0, x_3)}{\partial x_2} = \frac{\partial u(x_1, 1, x_3)}{\partial x_2} = 0, \tag{87}$$

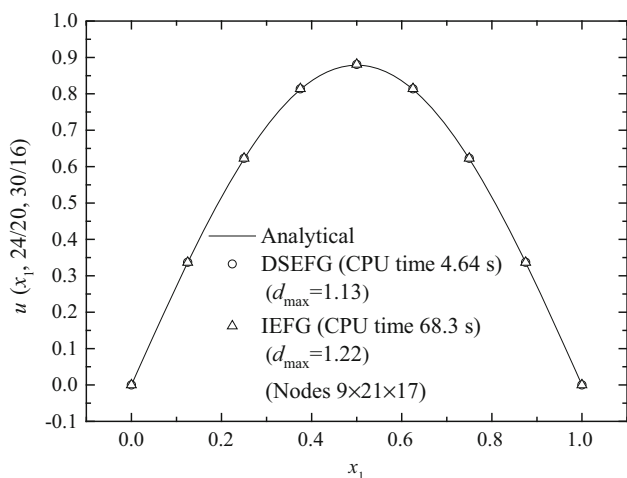


Fig. 8 The results of the DSEFG and IIEFG methods along the direction x_1

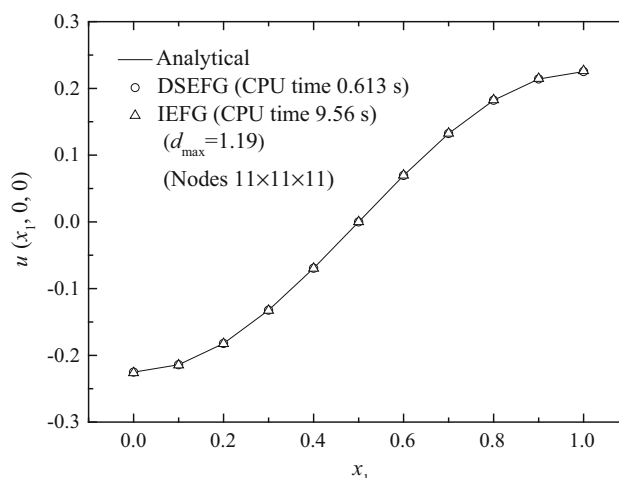


Fig. 11 The results of the DSEFG and IIEFG methods along the direction x_1

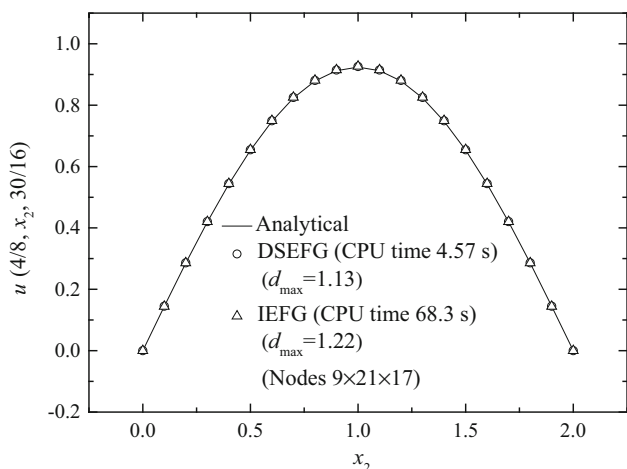


Fig. 9 The results of the DSEFG and IIEFG methods along the direction x_2

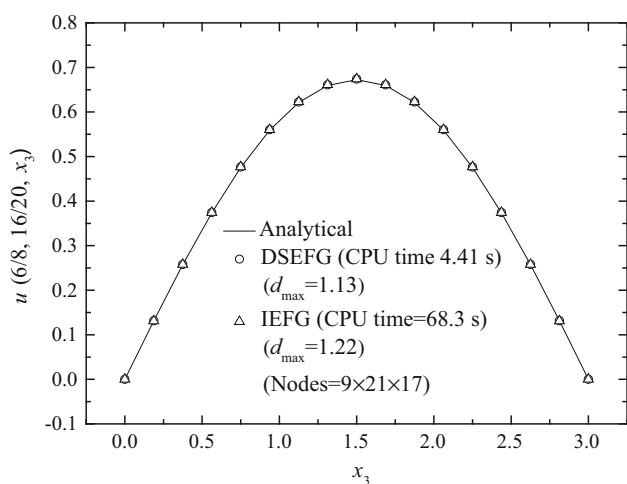


Fig. 10 The results of the DSEFG and IIEFG methods along the direction x_3

$$\frac{\partial u(x_1, x_2, 0)}{\partial x_3} = \cos(\pi x_1) \cos(\pi x_2), \tag{88}$$

$$\frac{\partial u(x_1, x_2, 1)}{\partial x_3} = 0, \tag{89}$$

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

The analytical solution of this problem is

$$u(x) = \left[\frac{\sinh(\sqrt{2}\pi x_3)}{\sqrt{2}\pi} - \frac{\cosh(\sqrt{2}\pi x_3)}{\sqrt{2}\pi \tanh(\sqrt{2}\pi)} \right] \times \cos(\pi x_1) \cos(\pi x_2). \tag{90}$$

For this example, we select x_3 as the splitting direction and the uniform node distribution is also adopted. Figures 11–13 show the numerical results obtained with the IIEFG method and the DSEFG method along the three axes under the node distribution $11 \times 11 \times 11$ and $d_{\max} = 1.19$. For the IIEFG method, the error is 0.0072, and the CPU time is 9.56 s. For the DSEFG method, the error is 0.0043, and the CPU time is 0.613 s. Comparing both methods, the computation accuracy and computation speed of the DSEFG method is higher than IIEFG method.

4.4 Laplace’s equation with Dirichlet boundary conditions on a half-torus cylinder

As a fourth example, we study Laplace’s equation

$$\nabla^2 u(x) = 0, \quad r \in [1, 2], \theta \in [0, \pi], x_3 \in [0, 1], \tag{91}$$

with Dirichlet boundary conditions

$$u(1, \theta, x_3) = \sin \theta + x_3, \tag{92}$$

$$u(2, \theta, x_3) = x_3, \tag{93}$$

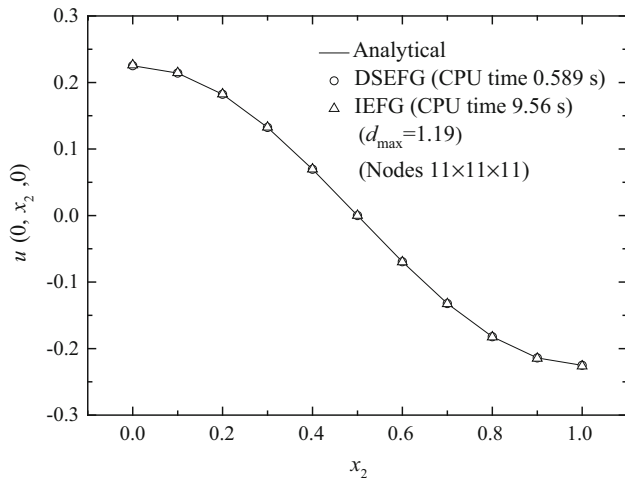


Fig. 12 The results of the DSEFG and IIEFG methods along the direction x_2

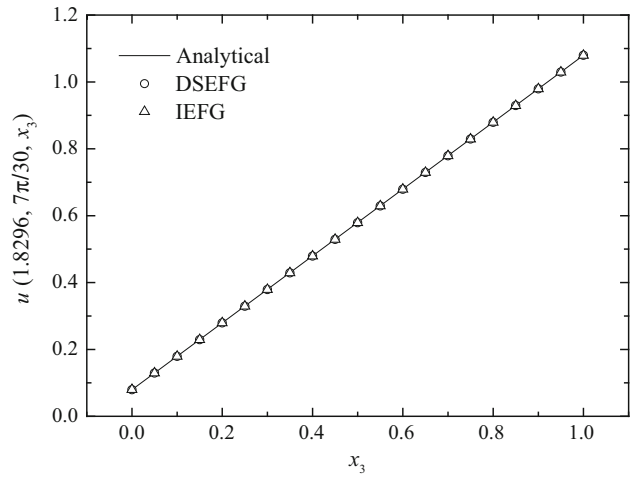


Fig. 15 The results of the DSEFG and IIEFG methods along the direction x_3

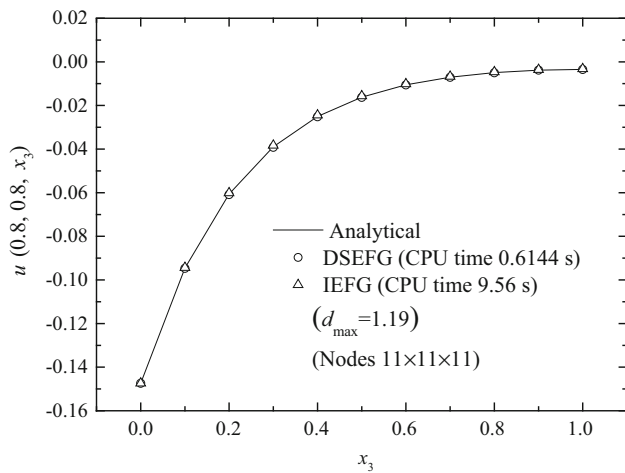


Fig. 13 The results of the DSEFG and IIEFG methods along the direction x_3

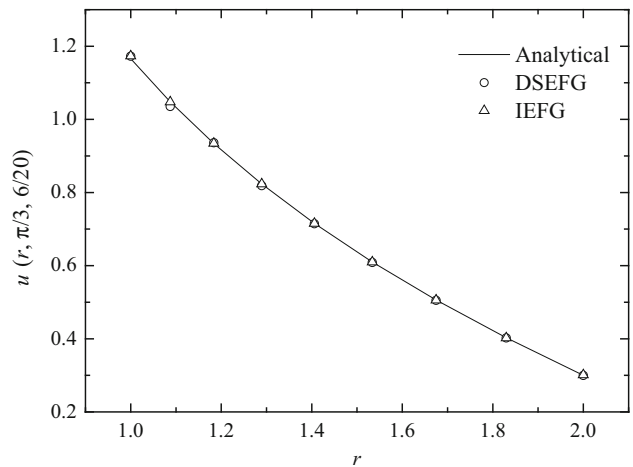


Fig. 16 The results of the DSEFG and IIEFG methods along the radial direction r

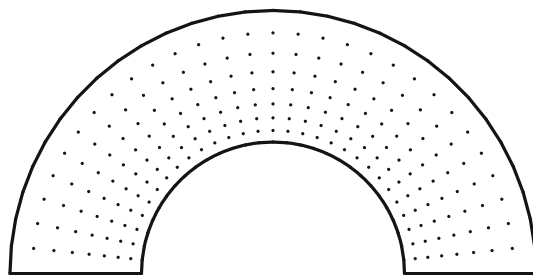


Fig. 14 Node distribution in two-dimensional sub-domain of a half-torus

$$u(r, 0, x_3) = x_3, \tag{94}$$

$$u(r, \pi, x_3) = x_3, \tag{95}$$

$$u(r, \theta, 0) = \frac{4}{3} \left(\frac{1}{r} - \frac{r}{4} \right) \sin \theta, \tag{96}$$

$$u(r, \theta, 1) = \frac{4}{3} \left(\frac{1}{r} - \frac{r}{4} \right) \sin \theta + 1. \tag{97}$$

The analytical solution of this problem is

$$u(r, \theta, x_3) = \frac{4}{3} \left(\frac{1}{r} - \frac{r}{4} \right) \sin \theta + x_3. \tag{98}$$

For this example, the problem domain is divided into 20 equal parts along the x_3 direction. 9×31 nodes are distributed on a half-torus domain of two-dimensional problem. Nine nodes are laid along the radial direction r in a 1.1 proportion and 31 nodes are uniformly laid along the angle axis θ as shown in Fig. 14. It means that the integral node distribution is $9 \times 31 \times 21$, and $d_{\max} = 1.2$, $\alpha = 1.0 \times 10^4$.

Figures 15–17 show the results obtained using the IIEFG method and the DSEFG method. It can be found that the results of two methods are in agreement with the analytical

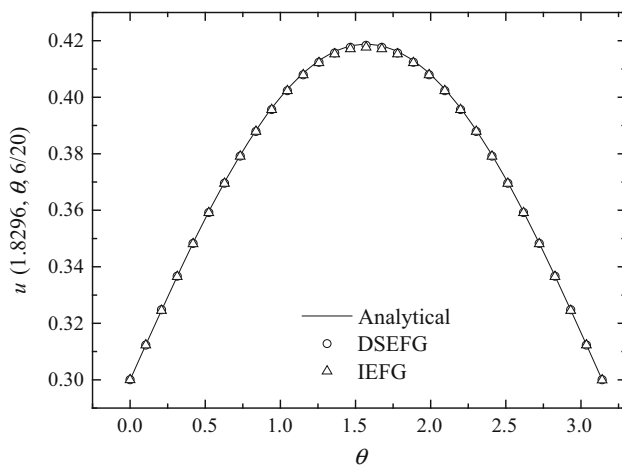


Fig. 17 The results of the DSEFG and IIEFG methods along the angle axis θ

solution. Under similar precision, the CPU time of the IIEFG method is 224.57s, and the one for the DSEFG method is 3.74s. Then the DSEFG method has greater computational efficiency than the IIEFG method.

5 Conclusions

This paper presents a new fast meshless method to solve the three-dimensional potential problems. The main idea of the DSEFG method is that a three-dimensional problem can be transformed into a series of two-dimensional problems. We only need to solve a two-dimensional problem in each sub-domain. For two-dimensional problems, the IIEFG method is applied, which uses an orthogonal function system with a weight function as the basis functions. It is efficient to avoid an ill-conditioned system of equations. Then, the finite difference method is selected for the splitting direction. From numerical results obtained by the DSEFG method, together with comparisons with analytical solutions and the ones for the IIEFG method, we can observe that DSEFG method is efficient to solve three-dimensional potential problems and generally has greater computational precision and higher computation speed than the IIEFG method.

Acknowledgements This work was supported by the National Natural Science Foundation of China (Grants 11571223, 51404160) and Shanxi Province Science Foundation for Youths (Grant 2014021025-1).

References

- Belytschko, T., Krongauz, Y., Organ, D., et al.: Meshless methods: an overview and recent developments. *Comput. Methods Appl. Mech. Eng.* **139**, 3–47 (1996)
- Dolbow, J., Belytschko, T.: An introduction to programming the meshless element free Galerkin method. *Arch. Comput. Methods Eng.* **5**, 207–241 (1998)
- Canelas, A., Laurain, A., Novotny, A.A.: A new reconstruction method for the inverse potential problem. *J. Comput. Phys.* **268**, 417–431 (2014)
- Sun, L.L., Chen, W., Zhang, C.Z.: A new formulation of regularized meshless method applied to interior and exterior anisotropic potential problems. *Appl. Math. Model.* **37**, 7452–7464 (2013)
- Khaji, N., Javaran, S.H.: New complex Fourier shape functions for the analysis of two-dimensional potential problems using boundary element method. *Eng. Anal. Bound. Elem.* **37**, 260–272 (2013)
- Gong, Y.P., Dong, C.Y., Qin, X.C.: An isogeometric boundary element method for three dimensional potential problems. *J. Comput. Appl. Math.* **313**, 454–468 (2017)
- Sun, F.L., Zhang, Y.M., Young, D.L., et al.: A new boundary mesh-free method for potential problems. *Adv. Eng. Softw.* **100**, 32–42 (2016)
- Mukherjee, Y.X., Mukherjee, S.: The boundary node method for potential problems. *Int. J. Numer. Methods Eng.* **40**, 797–815 (1997)
- Chati, M.K., Mukherjee, S.: The boundary node method for three-dimensional problems in potential theory. *Int. J. Numer. Methods Eng.* **47**, 1523–1547 (2000)
- Zhang, Y.M., Sun, F.L., Young, D.L., et al.: Average source boundary node method for potential problems. *Eng. Anal. Bound. Elem.* **70**, 114–125 (2016)
- Li, X.L.: Error estimates for the moving least-square approximation and the element-free Galerkin method in n-dimensional spaces. *Appl. Numer. Math.* **99**, 77–97 (2016)
- Lu, Y.Y., Belytschko, T., Gu, L.: A new implementation of the element free Galerkin method. *Comput. Methods Appl. Mech. Eng.* **113**, 397–414 (1994)
- Li, X.L., Zhang, S.G., Wang, Y., et al.: Analysis and application of the element-free Galerkin method for nonlinear Sine-Gordon and generalized Sinh-Gordon equations. *Comput. Math. Appl.* **71**, 1655–1678 (2016)
- Vahid, S.: Topology optimization using bi-directional evolutionary structural optimization based on the element-free Galerkin method. *Eng. Optim.* **48**, 380–396 (2016)
- Dehghan, M., Abbaszadeh, M., Mohebbi, A.: The use of interpolating element-free Galerkin technique for solving 2D generalized Benjamin–Bona–Mahony–Burgers and regularized long-wave equations on non-rectangular domains with error estimate. *J. Comput. Appl. Math.* **286**, 211–231 (2015)
- Joldes, G.R., Wittek, A., Miller, K.: Adaptive numerical integration in element-free Galerkin methods for elliptic boundary value problems. *Eng. Anal. Bound. Elem.* **51**, 52–63 (2015)
- Chen, L., Liu, C., Ma, H.P., et al.: An interpolating local Petrov–Galerkin method for potential problems. *Int. J. Appl. Mech.* **6**, 1450009 (2014)
- Deng, Y.J., Liu, C., Peng, M.J., et al.: The interpolating complex variable element-free Galerkin method for temperature field problems. *Int. J. Appl. Mech.* **7**, 1550017 (2015)
- Peng, M.J., Cheng, Y.M.: A boundary element-free method (BEFM) for two-dimensional potential problems. *Eng. Anal. Bound. Elem.* **33**, 77–82 (2009)
- Lian, H., Kerfriden, P., Bordas, S.: Implementation of regularized isogeometric boundary element methods for gradient-based shape optimization in two-dimensional linear elasticity. *Int. J. Numer. Methods Eng.* **106**, 972–1017 (2016)
- Chen, T., Raju, I.S.: A coupled finite element and meshless local Petrov–Galerkin method for two-dimensional potential problems. *Comput. Methods Appl. Mech. Eng.* **192**, 4533–4550 (2003)
- Cheng, Y.M., Chen, M.J.: A boundary element-free method for linear elasticity. *Acta Mech. Sin.* **19**, 181–186 (2003)
- Zhang, Z., Liew, K.M., Cheng, Y.M.: Coupling of the improved element-free Galerkin and boundary element methods for two-

- dimensional elasticity problems. *Eng. Anal. Bound. Elem.* **32**, 100–107 (2008)
24. Zhang, Z., Liew, K.M., Cheng, Y.M., et al.: Analyzing 2D fracture problems with the improved element-free Galerkin method. *Eng. Anal. Bound. Elem.* **32**, 241–250 (2008)
 25. Zhang, Z., Li, D.M., Cheng, Y.M., et al.: The improved element-free Galerkin method for three-dimensional wave equation. *Acta Mech. Sin.* **28**, 808–818 (2012)
 26. Zhang, Z., Hao, S.Y., Liew, K.M., et al.: The improved element-free Galerkin method for two-dimensional elastodynamics problems. *Eng. Anal. Bound. Elem.* **37**, 1576–1584 (2013)
 27. Li, K.T., Huang, A.X.: Mathematical aspect of the stream-function equations of compressible turbomachinery flows and their finite element approximation using optimal control. *Comput. Methods Appl. Mech. Eng.* **41**, 175–194 (1983)
 28. Li, K.T., Huang, A.X., Zhang, W.L.: A dimension split method for the 3-D compressible Navier–Stokes equations in turbomachine. *Commun. Numer. Methods Eng.* **18**, 1–14 (2002)
 29. Li, K.T., Yu, J.P., Shi, F., et al.: Dimension splitting method for the three dimensional rotating Navier–Stokes equations. *Acta Math. Appl. Sinica* **28**, 417–442 (2012)
 30. Chen, H., Li, K.T., Wang, S.: A dimension split method for the incompressible Navier–Stokes equations in three dimensions. *Int. J. Numer. Methods Fluids* **73**, 409–435(2013)
 31. Li, K.T., Shen, X.Q.: A dimensional splitting method for the linearly elastic shell. *Int. J. Comput. Math.* **84**, 807–824 (2007)
 32. Hansen, E., Ostermann, A.: Dimension splitting for evolution equations. *Numer. Math.* **108**, 557–570 (2008)
 33. Hansen, E., Ostermann, A.: Dimension splitting for quasilinear parabolic equations. *IMA J. Numer. Anal.* **30**, 857–869 (2010)
 34. Hou, R.Y., Wei, H.B.: Dimension splitting algorithm for a three-dimensional elliptic equation. *Int. J. Comput. Math.* **89**, 112–127 (2012)
 35. ter Maten, E.J.W.: Splitting methods for fourth order parabolic partial differential equations. *Computing* **37**, 335–350 (1986)
 36. Bragin, M.D., Rogov, B.V.: On exact dimensional splitting for a multidimensional scalar quasilinear Hyperbolic conservation law. *Dokl. Math.* **94**, 382–386 (2016)
 37. D’souza, R.M., Margolus, N.H., Smith, M.A.: Dimension-splitting for simplifying diffusion in lattice-gas models. *J. Stat. Phys.* **107**, 401–422 (2002)
 38. Cheng, H., Peng, M.J., Cheng, Y.M.: The hybrid improved complex variable element-free Galerkin method for three-dimensional potential problems. *Eng. Anal. Bound. Elem.* **84**, 52–62 (2017)
 39. Cheng, H., Peng, M.J., Cheng, Y.M.: A fast complex variable element-free Galerkin method for three-dimensional wave propagation problems. *Int. J. Appl. Mech.* **9**, 1750090 (2017)