

Article

Analyzing Three-Dimensional Laplace Equations Using the Dimension Coupling Method

Fengbin Liu ¹, Mingmei Zuo ², Heng Cheng ² and Ji Ma ^{1,*}

¹ College of Mechanical and Vehicle Engineering, Taiyuan University of Technology, Taiyuan 030024, China; liufengbin@tyut.edu.cn

² School of Applied Science, Taiyuan University of Science and Technology, Taiyuan 030024, China; mingmei9766@163.com (M.Z.); chengheng@shu.edu.cn (H.C.)

* Correspondence: maji@tyut.edu.cn; Tel.: +86-191-0342-1640

Abstract: Due to the low computational efficiency of the Improved Element-Free Galerkin (IEFG) method, efficiently solving three-dimensional (3D) Laplace problems using meshless methods has been a longstanding research direction. In this study, we propose the Dimension Coupling Method (DCM) as a promising alternative approach to address this challenge. Based on the Dimensional Splitting Method (DSM), the DCM divides the 3D problem domain into a coupling of multiple two-dimensional (2D) problems which are handled via the IEFG method. We use the Finite Element Method (FEM) in the third direction to combine the 2D discretized equations, which has advantages over the Finite Difference Method (FDM) used in traditional methods. Our numerical verification demonstrates the DCM's convergence and enhancement of computational speed without losing computational accuracy compared to the IEFG method. Therefore, this proposed method significantly reduces computational time and costs when solving 3D Laplace equations with natural or mixed boundary conditions in a dimensional splitting direction, and expands the applicability of the dimension splitting EFG method.

Keywords: dimension splitting method; improved element-free Galerkin method; dimension coupling method; finite element method; Laplace equation

MSC: 65N22



Citation: Liu, F.; Zuo, M.; Cheng, H.; Ma, J. Analyzing Three-Dimensional Laplace Equations Using the Dimension Coupling Method. *Mathematics* **2023**, *11*, 3717. <https://doi.org/10.3390/math11173717>

Academic Editor: Rade Vignjevic

Received: 19 July 2023

Revised: 16 August 2023

Accepted: 21 August 2023

Published: 29 August 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In science and engineering fields, the meshless method has become an important tool in numerical methods for solving partial differential equations. Compared with the traditional Finite Element Method (FEM), the meshless method [1] is only based on node distribution, eliminating mesh constraints and allowing for higher-accuracy solutions to large deformation problems by establishing appropriate shape functions without requiring grid reconstruction.

The Element-Free Galerkin Method (EFGM) [2–5] is a noteworthy meshless method proposed by Belytschko et al. that utilizes the Moving Least-Squares (MLS) approximation to construct shape functions. The MLS approximate functions have been improved to accelerate computational speed, including the Improved Moving Least-Squares (IMLS) approximation [6], interpolating MLS approximation [7,8], and complex variable MLS approximation [9,10]. Using these methods to construct shape functions resulted in the presentation of the IEFG method [11–14], interpolating EFG method [15–20], and complex variable EFG method [10,21,22], respectively.

Due to the complexity of establishing trial functions, the meshless method has been known to be time-consuming when solving complex 3D mechanics problems. Therefore, finding ways to analyze complex 3D problems more efficiently has become a significant research direction in the field of numerical methods. Recently, researchers have made strides

in this area, leading to the development of numerous dimensional split meshless methods, including the dimensional split complex variable EFG method [23–26], dimensional split EFG method [27–30], dimensional split reproducing kernel particle method [31–34], and interpolating dimensional split EFG method [35,36]. These hybrid meshless methods have demonstrated the capability to efficiently solve a wide range of 3D partial differential equations with smaller relative errors.

However, when natural or mixed boundary conditions occur in a dimensional splitting direction, traditional hybrid meshless methods require the 3D problems to be split into more layers to achieve higher accuracy. In [28], when a natural boundary condition existed in a split direction, Meng et al. had to choose 50 planes to obtain greater computational accuracy. In [33], the wave propagation problem with mixed boundary conditions was investigated. The splitting direction of the second numerical example had natural boundary conditions, so Peng and Cheng had to select 100 planes to achieve a smaller relative error of 0.62% with a computation time of 15.2 s. In contrast, if the splitting direction is selected as an essential boundary condition, only 10 layers are required, resulting in an error of 0.24% and a computation time of 1.1 s. Hence, it is worth noting that time consumption remains a concern when using the FDM in the dimensional split direction to address natural boundary conditions.

To address the above-mentioned issue of time consumption and explore a novel approach to enhance the convenience and efficiency of applying natural or mixed boundary conditions, we propose a modification that replaces the FDM with the FEM in the dimensional splitting direction. The FEM [37,38] is widely recognized for its advantages over the FDM in numerical simulations. Therefore, compared with the Dimensional Splitting EFG Method presented by Meng et al. [28], our proposed method can reduce CPU time when solving partial differential equations with mixed or natural boundary conditions in the splitting direction.

Due to the low computational efficiency of the EFG for solving 3D problems, researchers have been exploring alternative efficient meshless methods. The Dimension Coupling Method (DCM) [39] as a potential approach for solving the 3D Laplace equation is presented in this study. By introducing the Dimensional Split Method (DSM), the problem domain of the 3D Laplace equation is divided into a coupling of multiple 2D problems, and the IEFG method is employed to handle 2D forms. Two-dimensional discrete equations are combined by the FEM in the third direction and the final algebraic equation of the original 3D Laplace’s equation is derived.

The error formula of the DCM is given in numerical examples, and we discuss the influence of meshes of the FEM in the third direction, weight function, other parameters, and node distribution in 2D domains on precision. Furthermore, the convergence of the proposed method in this study is numerically verified. Through three numerical examples, it is demonstrated that the DCM for 3D Laplace equations can significantly improve computational efficiency without reducing computational accuracy.

2. IMLS Approximation

For a point x , the approximation of the corresponding function $u(x)$ can be written as

$$u^h(x) = \tilde{\Phi}u = \sum_{I=1}^n \tilde{\Phi}_I(x)u_I, \quad (x \in \Omega), \tag{1}$$

$$u^T = (u_1, u_2, \dots, u_n), \tag{2}$$

$\tilde{\Phi}$ is the shape function and its form is

$$\tilde{\Phi} = (\tilde{\Phi}_1(x), \tilde{\Phi}_2(x), \dots, \tilde{\Phi}_n(x)) = p^T(x)A^*(x)B(x), \tag{3}$$

$\mathbf{p}^T(\mathbf{x})$ is the vector of the basis function,

$$\mathbf{A}^*(\mathbf{x}) = \begin{bmatrix} \frac{1}{(p_1, p_1)} & 0 & \cdots & 0 \\ 0 & \frac{1}{(p_2, p_2)} & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{1}{(p_n, p_n)} \end{bmatrix}, \tag{4}$$

$$\mathbf{B}(\mathbf{x}) = \mathbf{P}^T \mathbf{W}, \tag{5}$$

$$\mathbf{P} = \begin{bmatrix} p_1(\mathbf{x}_1) & p_2(\mathbf{x}_1) & \cdots & p_m(\mathbf{x}_1) \\ p_1(\mathbf{x}_2) & p_2(\mathbf{x}_2) & \cdots & p_m(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ p_1(\mathbf{x}_n) & p_2(\mathbf{x}_n) & \cdots & p_m(\mathbf{x}_n) \end{bmatrix}, \tag{6}$$

$$\mathbf{W} = \begin{bmatrix} w(\mathbf{x} - \mathbf{x}_1) & 0 & \cdots & 0 \\ 0 & w(\mathbf{x} - \mathbf{x}_2) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & w(\mathbf{x} - \mathbf{x}_n) \end{bmatrix}, \tag{7}$$

and $w(\mathbf{x} - \mathbf{x}_l)$ is the weighting function.

Equations (1)–(7) are the IMLS approximation [6].

3. Dimension Coupling Method for 3D Laplace Equations

When solving 3D Laplace equations using the IIEFG method, the numerical solution obtained has low computational speed due to the complexity of shape functions compared to 2D problems.

In this paper, we propose using the DCM to solve this issue. The 3D problem can be transformed into multiple 2D problems as shown in Figure 1, which are then discretized using the IIEFG method. In the splitting direction, 2D discretized equations are coupled via the FEM.

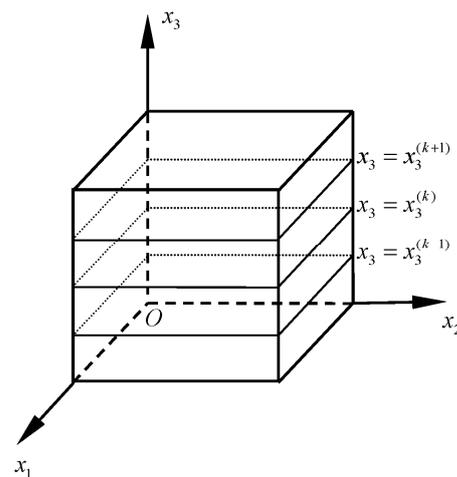


Figure 1. Layers in the direction x_3 of the domain Ω .

The formula of the DCM for 3D Laplace equations is derived in this section. Considering the governing equation [23]

$$\Delta u = 0, (x_1, x_2, x_3) \in \Omega; \tag{8}$$

with boundary conditions

$$u = \bar{u}(x_1, x_2, x_3), (x_1, x_2, x_3) \in \Gamma_u, \tag{9}$$

$$q = \frac{\partial u}{\partial x_1} n_1 + \frac{\partial u}{\partial x_2} n_2 + \frac{\partial u}{\partial x_3} n_3 = \bar{q}(x_1, x_2, x_3), (x_1, x_2, x_3) \in \Gamma_q; \tag{10}$$

\bar{q} and \bar{u} are given values, and n_i is the unit outward normal to the boundary Γ in direction x_i , $\Gamma = \Gamma_u \cup \Gamma_q$, $\Gamma_u \cap \Gamma_q = \emptyset$.

The governing equation of 2D form in the k -th layer based on the DCM is

$$\frac{\partial^2 u^{(k)}}{\partial x_1^2} + \frac{\partial^2 u^{(k)}}{\partial x_2^2} = -\frac{\partial^2 u^{(k)}}{\partial x_3^2}, (x_1, x_2) \in \Omega^{(k)}, x_3 = x_3^{(k)}, \tag{11}$$

where $\Omega^{(k)}$ is the 2D domain of the k -th layer of Ω , and

$$\Omega = \bigcup_{k=1}^L \left\{ \Omega^{(k)} \times [x_3^{(k)}, x_3^{(k+1)}] \right\} \cup \Omega^{(L+1)}, \tag{12}$$

$$u^{(k)} = u(x_1, x_2, x_3^{(k)}), \tag{13}$$

with boundary conditions

$$u^{(k)} = \bar{u}^{(k)}(x_1, x_2) = \bar{u}(x_1, x_2, x_3^{(k)}), (x_1, x_2) \in \Gamma_u^{(k)}, \tag{14}$$

$$q^{(k)} = \bar{q}^{(k)}(x_1, x_2) = \bar{q}(x_1, x_2, x_3^{(k)}), (x_1, x_2) \in \Gamma_q^{(k)}, \tag{15}$$

where $\Gamma_u^{(k)}$ and $\Gamma_q^{(k)}$ are essential and natural boundaries. $\Gamma^{(k)} = \Gamma_u^{(k)} \cup \Gamma_q^{(k)}$, and $\Gamma_u^{(k)} \cap \Gamma_q^{(k)} = \emptyset$.

Equations (11), (14) and (15) are then analyzed using the IIEFG method. The discretization of the second-order partial derivative in the splitting direction is performed using the FEM, then we obtain the discretized equations.

$$\Pi = \int_{\Omega^{(k)}} \left[u \left(\frac{\partial^2 u}{\partial x_3^2} \right) \right] d\Omega^{(k)} - \int_{\Omega^{(k)}} \frac{1}{2} \left[\left(\frac{\partial u}{\partial x_1} \right)^2 + \left(\frac{\partial u}{\partial x_2} \right)^2 \right] d\Omega^{(k)} - \int_{\Gamma_q^{(k)}} u \bar{q} d\Gamma^{(k)}. \tag{16}$$

Equation (16) is the equivalent functional, and the penalty method is selected for exerting boundary conditions, hence we can obtain that the modified functional of each 2D form is

$$\Pi^* = \Pi + \frac{\alpha}{2} \int_{\Gamma_u^{(k)}} (u - \bar{u})(u - \bar{u}) d\Gamma^{(k)}. \tag{17}$$

Let

$$\delta \Pi^* = 0, \tag{18}$$

hence we have

$$\begin{aligned} & \int_{\Omega^{(k)}} \delta u \cdot \frac{\partial^2 u}{\partial x_3^2} d\Omega^{(k)} - \int_{\Omega^{(k)}} \delta(Lu)^T \cdot (Lu) d\Omega^{(k)} - \int_{\Gamma_q^{(k)}} \delta u \cdot \bar{q} d\Gamma^{(k)} \\ & + \alpha \int_{\Gamma_u^{(k)}} \delta u \cdot u d\Gamma^{(k)} - \alpha \int_{\Gamma_u^{(k)}} \delta u \cdot \bar{u} d\Gamma^{(k)} = 0, \end{aligned} \tag{19}$$

where

$$L(\cdot) = \begin{bmatrix} \frac{\partial}{\partial x_1} \\ \frac{\partial}{\partial x_2} \end{bmatrix} (\cdot). \tag{20}$$

In the 2D domain $\Omega^{(k)}$, we select M nodes x_I ; thus, we can obtain the following form:

$$u(x_I^{(k)}) = u(x_I^{(k)}, x_3^{(k)}) = u_I. \tag{21}$$

From Section 2, the expression of the approximate function of Equation (21) is

$$u(x, x_3^{(k)}) = \tilde{\Phi}u = \sum_{I=1}^n \tilde{\Phi}_I(x)u_I, \tag{22}$$

with

$$u = (u_1, u_2, \dots, u_n)^T. \tag{23}$$

Thus,

$$\frac{\partial^2 u(x, x_3^{(k)})}{\partial x_3^2} = \frac{\partial^2}{\partial x_3^2} \sum_{I=1}^n \tilde{\Phi}_I u(x_I, x_3^{(k)}) = \sum_{I=1}^n \tilde{\Phi}_I \frac{\partial^2 u_I}{\partial x_3^2} = \tilde{\Phi}(x)u'', \tag{24}$$

$$Lu(x, x_3^{(k)}) = \sum_{I=1}^n \left[\frac{\partial}{\partial x_1} \frac{\partial}{\partial x_2} \right] \tilde{\Phi}_I u_I = \sum_{I=1}^n B_I u_I = Bu, \tag{25}$$

where

$$u'' = \left(\frac{\partial^2 u(x_1, x_3^{(k)})}{\partial x_3^2}, \frac{\partial^2 u(x_2, x_3^{(k)})}{\partial x_3^2}, \dots, \frac{\partial^2 u(x_n, x_3^{(k)})}{\partial x_3^2} \right)^T, \tag{26}$$

$$B = (B_1, B_2, \dots, B_n), \tag{27}$$

$$B_I = \begin{bmatrix} \tilde{\Phi}_{I,1}(x) \\ \tilde{\Phi}_{I,2}(x) \end{bmatrix}. \tag{28}$$

Substituting Equations (22), (24) and (25) into Equation (19),

$$\begin{aligned} & \int_{\Omega^{(k)}} \delta[\tilde{\Phi}u]^T \cdot [\tilde{\Phi}u''] d\Omega^{(k)} - \int_{\Gamma_q^{(k)}} \delta[\tilde{\Phi}u]^T \cdot \bar{q} d\Gamma^{(k)} - \int_{\Omega^{(k)}} \delta[Bu]^T [Bu] d\Omega^{(k)} \\ & + \alpha \int_{\Gamma_u^{(k)}} \delta[\tilde{\Phi}u]^T \cdot [\tilde{\Phi}u] d\Gamma^{(k)} - \alpha \int_{\Gamma_u^{(k)}} \delta[\tilde{\Phi}u]^T \cdot \bar{u} d\Gamma^{(k)} = 0. \end{aligned} \tag{29}$$

Next, we write Equation (29) in its matrix form

$$\int_{\Omega^{(k)}} \delta[\tilde{\Phi}u]^T \cdot [\tilde{\Phi}u''] d\Omega^{(k)} = \delta u^T \cdot \left[\int_{\Omega^{(k)}} \tilde{\Phi}^T \tilde{\Phi} d\Omega^{(k)} \right] \cdot u'' = \delta u^T \cdot C \cdot u'', \tag{30}$$

$$\int_{\Omega^{(k)}} \delta[Bu]^T \cdot [Bu] d\Omega^{(k)} = \delta u^T \cdot \left[\int_{\Omega^{(k)}} B^T B d\Omega^{(k)} \right] \cdot u = \delta u^T \cdot K \cdot u, \tag{31}$$

$$\int_{\Gamma_q^{(k)}} \delta[\tilde{\Phi}u]^T \cdot \bar{q} d\Gamma^{(k)} = \delta u^T \cdot \int_{\Gamma_q^{(k)}} \tilde{\Phi}^T \bar{q} d\Gamma^{(k)} = \delta u^T \cdot f_q, \tag{32}$$

$$\alpha \int_{\Gamma_u^{(k)}} \delta[\tilde{\Phi}u]^T \cdot [\tilde{\Phi}u] d\Gamma^{(k)} = \delta u^T \cdot \left[\alpha \int_{\Gamma_u^{(k)}} \tilde{\Phi}^T \tilde{\Phi} d\Gamma^{(k)} \right] \cdot u = \delta u^T \cdot K_\alpha \cdot u, \tag{33}$$

$$\alpha \int_{\Gamma_u^{(k)}} \delta[\tilde{\Phi}u]^T \cdot \bar{u} d\Gamma^{(k)} = \delta u^T \cdot \alpha \int_{\Gamma_u^{(k)}} \tilde{\Phi}^T \bar{u} d\Gamma^{(k)} = \delta u^T \cdot f_\alpha, \tag{34}$$

where

$$K = \int_{\Omega^{(k)}} B^T B d\Omega^{(k)}, \tag{35}$$

$$C = \int_{\Omega^{(k)}} \tilde{\Phi}^T \tilde{\Phi} d\Omega^{(k)}, \tag{36}$$

$$K_\alpha = \alpha \int_{\Gamma_u^{(k)}} \tilde{\Phi}^T \tilde{\Phi} d\Gamma^{(k)}, \tag{37}$$

$$f_q = \int_{\Gamma_q^{(k)}} \tilde{\Phi}^T \bar{q} d\Gamma^{(k)}, \tag{38}$$

$$f_\alpha = \alpha \int_{\Gamma_u^{(k)}} \tilde{\Phi}^T \bar{u} d\Gamma^{(k)}. \tag{39}$$

Substituting Equations (30)–(34) into Equation (29),

$$\delta \mathbf{u}^T \cdot (\mathbf{C} \mathbf{u}'' + \mathbf{K}_\alpha \mathbf{u} - \mathbf{K} \mathbf{u} - \mathbf{f}_\alpha - \mathbf{f}_q) = 0. \tag{40}$$

Let

$$\mathbf{F} = \mathbf{f}_q + \mathbf{f}_\alpha, \tag{41}$$

$$\hat{\mathbf{K}} = \mathbf{K}_\alpha - \mathbf{K}. \tag{42}$$

Hence, Equation (40) can be transformed as

$$\mathbf{C} \mathbf{u}'' + \hat{\mathbf{K}} \mathbf{u} = \mathbf{F}. \tag{43}$$

Let

$$\mathbf{u}(x_3^{(2)}) = \mathbf{u}^{(2)}, \tag{44}$$

$$\mathbf{u}(x_3^{(3)}) = \mathbf{u}^{(3)}, \tag{45}$$

⋮

$$\mathbf{u}(x_3^{(L)}) = \mathbf{u}^{(L)}, \tag{46}$$

suppose $x_3 \in [a, c]$, thus we have

$$\mathbf{C} \mathbf{u}'(c)v(x_3^{(L+1)}) - \mathbf{C} \mathbf{u}'(a)v(x_3^{(1)}) - \mathbf{C} \int_a^c \mathbf{u}' v' dx_3 + \hat{\mathbf{K}} \int_a^c \mathbf{u} v dx_3 = \int_a^c \mathbf{F} v dx_3, \tag{47}$$

the test function $v = \varphi_i$ ($i = 1, 2, \dots, N + 1$) is selected with the shape function based on piecewise linear interpolation functions, thus $v(x_3^{(1)}) = 0$ and $v(x_3^{(L+1)}) = 0$ when $x_3 = x_3^{(2)}, x_3^{(3)}, \dots, x_3^{(L-1)}, x_3^{(L)}$. Hence, Equation (47) is changed to the following form

$$\hat{\mathbf{K}} \sum_{k=1}^{L+1} \mathbf{u}^{(k)} \int_a^c \varphi_k \varphi_i dx_3 - \mathbf{C} \sum_{k=1}^{L+1} \mathbf{u}^{(k)} \int_a^c \varphi'_k \varphi'_i dx_3 = \int_a^c \mathbf{F}^{(k)} \varphi_i dx_3, \quad i = 2, 3, \dots, N. \tag{48}$$

Let

$$\mathbf{H}_{ik} = \left(\sum_{k=1}^{L+1} \int_a^c \varphi_k \varphi_i dx_3 \right) \hat{\mathbf{K}} - \left(\sum_{k=1}^{L+1} \int_a^c \varphi'_k \varphi'_i dx_3 \right) \mathbf{C}, \quad i = 2, 3, \dots, N, \tag{49}$$

$$\mathbf{W}_i = \int_a^c \mathbf{F}^{(k)} \varphi_i dx_3, \quad i = 1, 2, \dots, N + 1. \tag{50}$$

Therefore, Equation (48) is transformed as

$$\mathbf{H}_{ik} \mathbf{u}^{(k)} = \mathbf{W}_i, \quad i = 2, 3, \dots, N. \tag{51}$$

(2) Scale parameter.

We choose 19×19 regular nodes and 18×18 integral cells. The mesh number of the FEM in the x_3 split direction is 18 and we set $\alpha = 2.1 \times 10^6$. Figure 2 shows the relationship between the scale parameter d_{\max} and the error. From Figure 2, we can conclude that when $d_{\max} = 1.4\sim 1.5$, the computational accuracy of the numerical solution is higher.

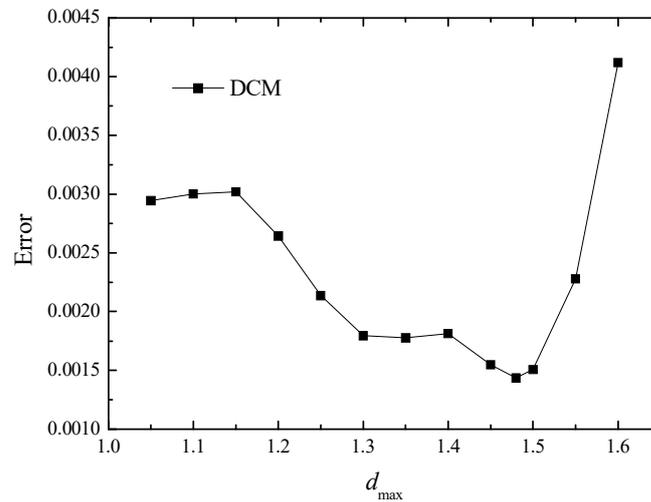


Figure 2. The correlation between d_{\max} and the errors.

(3) Penalty factor.

We select 19×19 regular nodes and 18×18 integral cells. The mesh number of the FEM in the x_3 split direction is 18 with $d_{\max} = 1.48$. The computational accuracy of the numerical solution is higher when $\alpha = 2.1 \times 10^6$ in Figure 3.

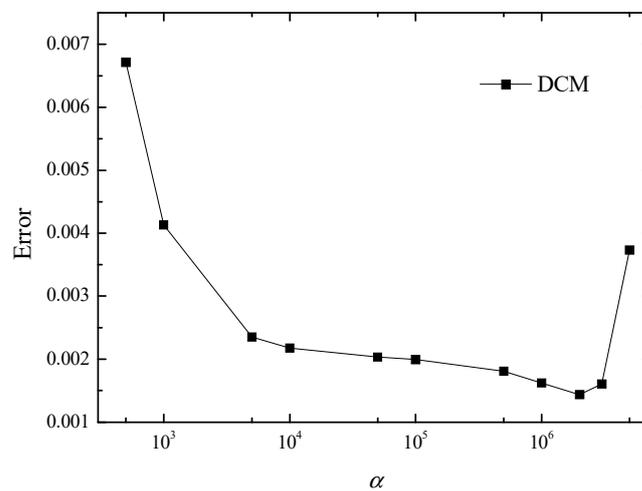


Figure 3. The correlation between α and the errors.

Different node distributions in the k -th plane and meshes in the x_3 direction are analyzed in the following, respectively.

(4) Node distribution.

We select $d_{\max} = 1.48$, $\alpha = 2.1 \times 10^6$, and 18 meshes in the x_3 split direction. The relationship between the relative error and the number of nodes is shown in Figure 4. We can see that as the number of nodes increases, the relative error tends to decrease.

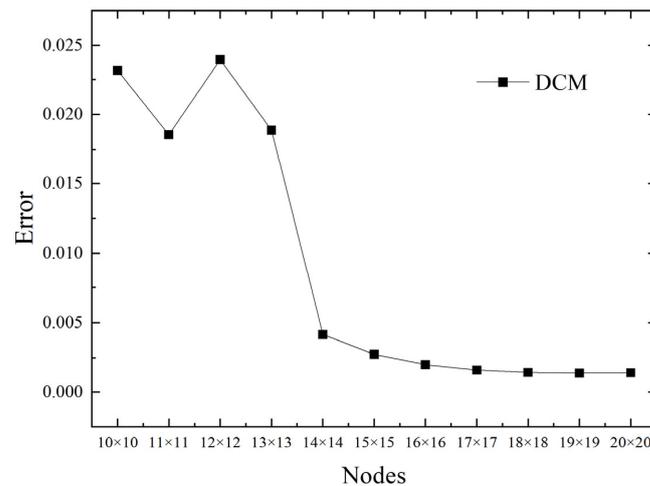


Figure 4. The correlation between nodes and the errors.

(5) Mesh number.

We use 19×19 regular nodes and 18×18 integral cells in each 2D domain, setting $d_{\max} = 1.48$ and $\alpha = 2.1 \times 10^6$. The relationship between the relative error and the mesh number is shown in Figure 5. We can see that as the number of meshes increases in the x_3 splitting direction, the computational accuracy steadily improves.

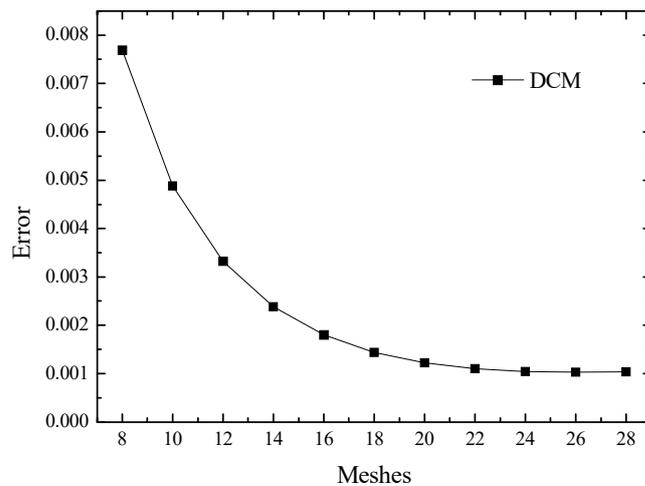


Figure 5. The correlation between meshes and the errors.

As discussed above, the DCM for the 3D Laplace equation is convergent.

Using the proposed DCM in this paper to solve Example 1, we select 19×19 regular nodes and 18×18 integral cells. In the x_3 splitting direction, we employ the FEM with a mesh number of 18, $d_{\max} = 1.48$, and $\alpha = 2.1 \times 10^6$. As a result, the relative error is 0.1435% and the computational time is 1.5 s.

In contrast, when applying the IIEFG method to solve Example 1, we choose $19 \times 19 \times 19$ regular nodes, $18 \times 18 \times 18$ integral cells, and the cubic spline weight function. By selecting $d_{\max} = 1.32$ and $\alpha = 1.5 \times 10^3$, the resulting relative error is 0.2134% with a computational time of 63.9 s.

The comparison of the computational accuracy and time of the DCM and the IIEFG method is shown in Table 1.

Table 1. The comparison of computational accuracy and time of the DCM and the IIEFG.

Method	Regular Nodes	Relative Error	CPU Time (s)
DCM	$19 \times 19 \times 19$	0.1435%	1.5
IIEFG	$19 \times 19 \times 19$	0.2134%	63.9

A comparison is performed between the numerical solutions obtained using the DCM and the IIEFG method, along with the exact solutions depicted in Figures 6–8. It can be observed that the results of these two methods accord well with the analytical solutions.

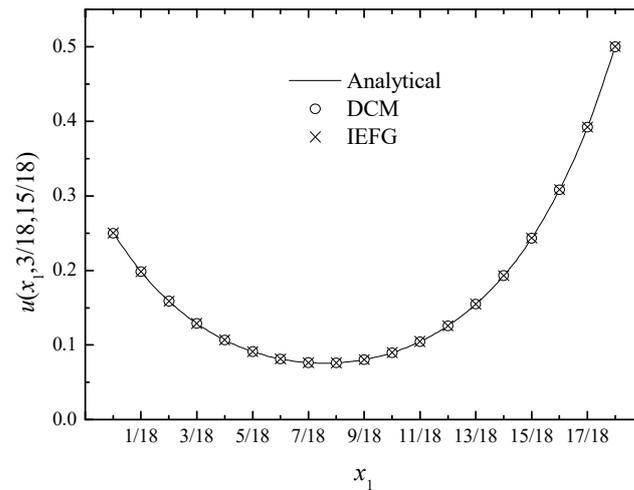


Figure 6. The numerical results of the DCM, IIEFG, and analytical solutions along x_1 .

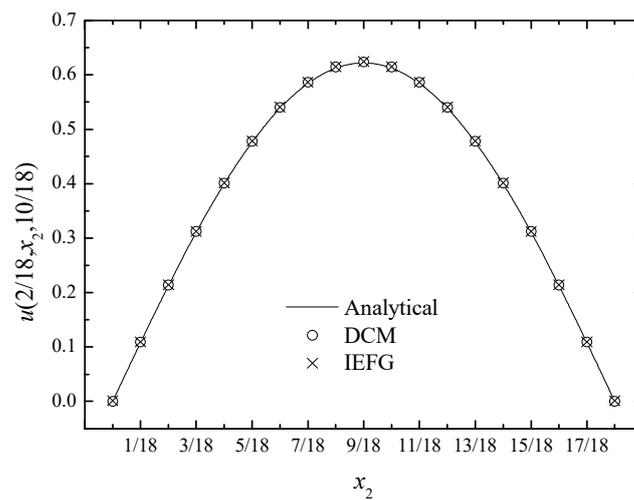


Figure 7. The numerical results of the DCM, IIEFG, and analytical solutions along x_2 .

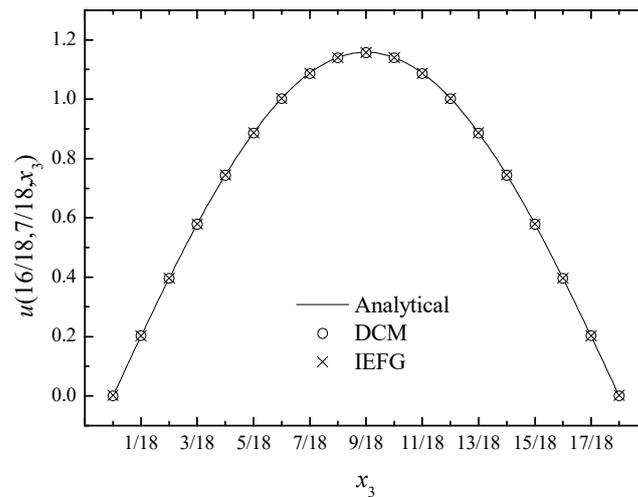


Figure 8. The numerical results of the DCM, IIEFG, and analytical solutions along x_3 .

We can see that the DCM not only exhibits superior calculation accuracy but also significantly enhances the computational efficiency of the IIEFG method in the numerical results of Figures 6–8.

The second example [1] is

$$\nabla^2 u = 0, \tag{74}$$

this problem domain is $\Omega = [0, 1]^3$, and

$$u_{,1}|_{x_1=0} = u_{,1}|_{x_1=1} = u_{,2}|_{x_2=0} = u_{,2}|_{x_2=1} = 0, \tag{75}$$

$$u|_{x_3=0} = -\frac{\cos(\pi x_1) \cos(\pi x_2)}{\sqrt{2}\pi \tanh(\sqrt{2}\pi)}, \tag{76}$$

$$u_{,3}|_{x_2=1} = 0. \tag{77}$$

Equations (75)–(77) are boundary conditions, and

$$u = \cos(\pi x_1) \cos(\pi x_2) \left[\frac{\sinh(\sqrt{2}\pi x_3)}{\sqrt{2}\pi} - \frac{\cosh(\sqrt{2}\pi x_3)}{\sqrt{2}\pi \tanh(\sqrt{2}\pi)} \right] \tag{78}$$

is the analytical solution.

In Example 2, 15×15 regular nodes and 14×14 integral cells are selected in each 2D domain. We employ the FEM in the x_3 split direction with a mesh number of 14 and $d_{\max} = 1.34$. The resulting relative error is 0.3796% and the computational time is 0.6 s.

In contrast, when analyzing Example 2 using the IIEFG method, we choose the cubic spline weight function along with $15 \times 15 \times 15$ regular nodes and $14 \times 14 \times 14$ integral cells. Moreover, we set $d_{\max} = 1.0$ and $\alpha = 5.8 \times 10^2$. The corresponding relative error is 0.3745% and the computational time is 12.9 s.

The comparison of the computational accuracy and time of the DCM and the IIEFG method is shown in Table 2.

Table 2. The comparison of computational accuracy and time of the DCM and the IIEFG.

Method	Regular Nodes	Relative Error	CPU Time (s)
DCM	$15 \times 15 \times 15$	0.3796%	0.6
IIEFG	$15 \times 15 \times 15$	0.3745%	12.9

The numerical solution of the DCM is compared with that of the IIEFG method and the exact solutions in Figures 9–11. It can be observed that the results of these two numerical methods accord well with the exact ones.

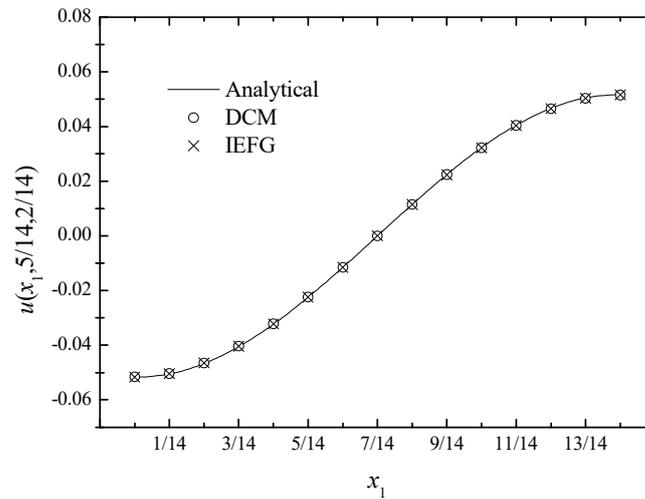


Figure 9. The numerical results of the DCM, IIEFG, and analytical solutions along x_1 .

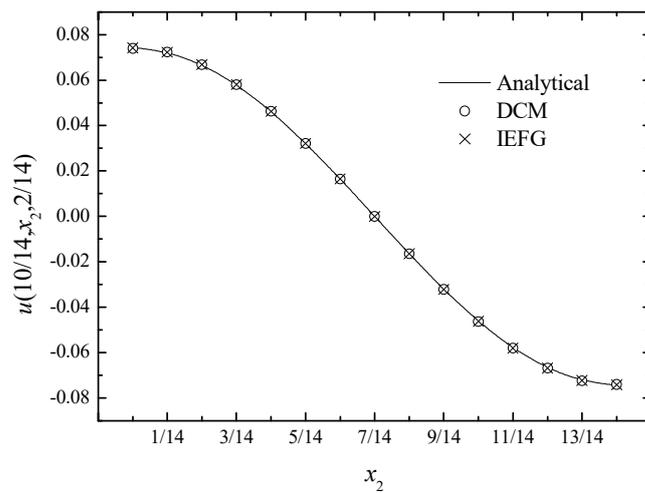


Figure 10. The numerical results of the DCM, IIEFG, and analytical solutions along x_2 .

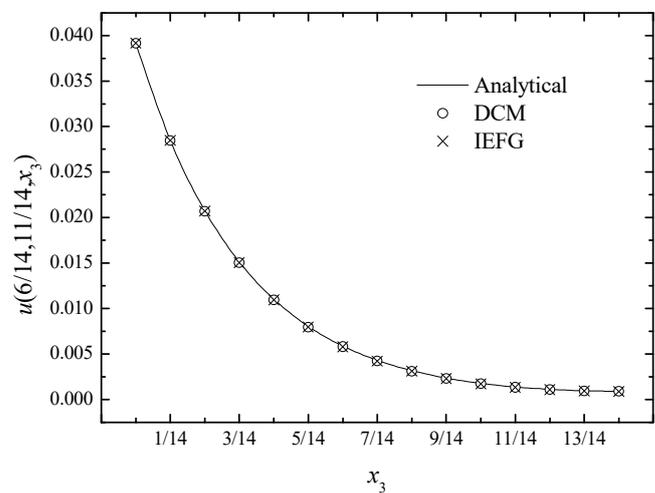


Figure 11. The numerical results of the DCM, IIEFG, and analytical solutions along x_3 .

The results demonstrate that the DCM exhibits higher efficiency compared to the IIEFG method, despite both methods yielding similar errors. Furthermore, the DCM has the advantage of efficiently handling mixed boundary conditions without requiring excessive layers in the dimensional splitting direction.

The third example is

$$\nabla^2 u = 0, \tag{79}$$

this problem domain is $\Omega = [0, 1]^3$, and

$$u = x_2^2 + x_3^2, (x_1 = 0), \tag{80}$$

$$u = -2 + x_2^2 + x_3^2, (x_1 = 1) \tag{81}$$

$$u = -2x_1^2 + x_3^2, (x_2 = 0), \tag{82}$$

$$u = -2x_1^2 + 1 + x_3^2, (x_2 = 1), \tag{83}$$

$$u = -2x_1^2 + x_2^2, (x_3 = 0), \tag{84}$$

$$u = -2x_1^2 + x_2^2 + 1, (x_3 = 1). \tag{85}$$

Equations (80)–(85) are boundary conditions.

$$u = -2x_1^2 + x_2^2 + x_3^2 \tag{86}$$

is the analytical solution.

Then, we apply the DCM to solve Example 3 and analyze three situations in which the FEM is applied in different directions.

(1) The FEM applied in the x_1 direction.

The mesh number is 18 with $d_{\max} = 1.51$ and $\alpha = 2.5 \times 10^4$. We select 11×11 regular nodes and 10×10 integral cells in each 2D domain. The resulting relative error is 0.1025% and the computational time is 0.3 s.

(2) The FEM applied in the x_2 or x_3 direction.

The mesh number is 10 with $d_{\max} = 1.32$ and $\alpha = 4.2 \times 10^4$. We select 11×11 regular nodes and 10×10 integral cells in each 2D domain. The resulting relative error is 0.1934% and the computational time is 0.2 s.

Furthermore, we employ the IIEFG method for analysis. In this case, we choose $11 \times 11 \times 11$ regular nodes, $10 \times 10 \times 10$ integral cells, and the cubic spline weight function. The parameters used are $d_{\max} = 1.33$ and $\alpha = 1.4 \times 10^3$. The resulting error is 0.1544% with a calculation time of 7.1 s.

The comparison of the computational accuracy and time of the DCM and the IIEFG method is shown in Table 3.

Table 3. The comparison of computational accuracy and time of the DCM and the IIEFG.

Method	Regular Nodes	Relative Error	CPU Time (s)
DCM (split in x_1 direction)	$11 \times 11 \times 11$	0.1025%	0.3
DCM (split in x_2 or x_3 direction)	$11 \times 11 \times 11$	0.1934%	0.2
IIEFG	$11 \times 11 \times 11$	0.1544%	7.1

When the FEM is used in the x_1 -axis splitting direction, it can be observed that the DCM can obtain a smaller error. Comparing the numerical solution of the DCM with those

of the IIEFG method and the exact ones in Figures 12–14, the results of these two methods agree well with the exact ones.

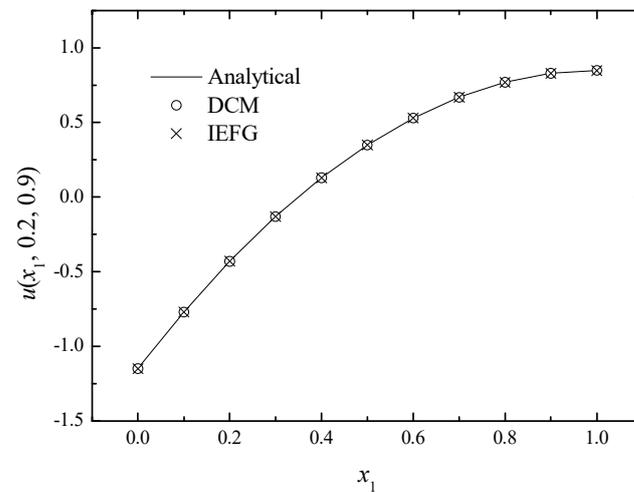


Figure 12. The numerical results of the DCM, IIEFG, and analytical solutions along x_1 .

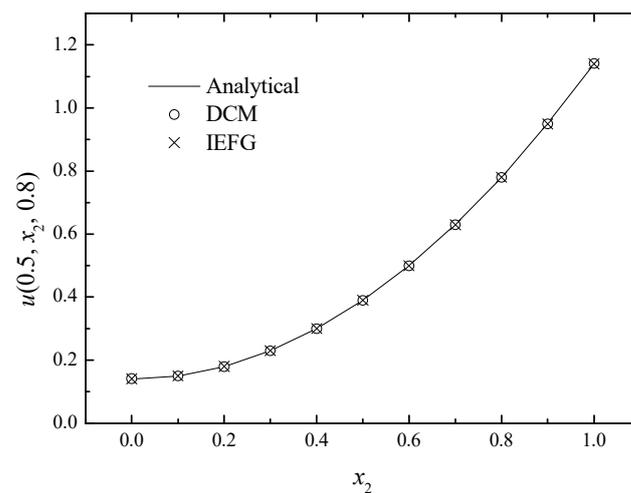


Figure 13. The numerical results of the DCM, IIEFG, and analytical solutions along x_2 .

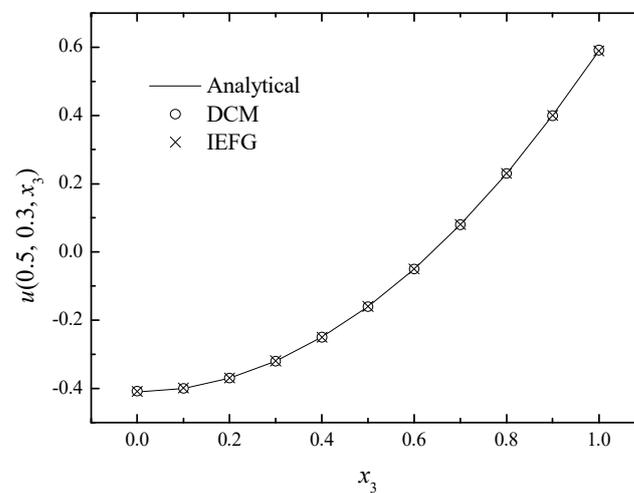


Figure 14. The numerical results of the DCM, IIEFG, and analytical solutions along x_3 .

The fourth example [27,31] is in an irregular-shaped domain selected as

$$\nabla^2 u = 0, (r \in [1,2], \theta \in [0, \pi], x_3 \in [0,1]), \tag{87}$$

The boundary conditions are

$$u(1, \theta, x_3) = \sin \theta + x_3, \tag{88}$$

$$u(2, \theta, x_3) = x_3, \tag{89}$$

$$u(r, 0, x_3) = x_3, \tag{90}$$

$$u(r, \pi, x_3) = x_3, \tag{91}$$

$$u(r, \theta, 0) = \frac{4}{3} \left(\frac{1}{r} - \frac{r}{4} \right) \sin \theta, \tag{92}$$

$$u(r, \theta, 1) = \frac{4}{3} \left(\frac{1}{r} - \frac{r}{4} \right) \sin \theta + 1. \tag{93}$$

The analytical solution of this problem is

$$u(r, \theta, x_3) = \frac{4}{3} \left(\frac{1}{r} - \frac{r}{4} \right) \sin \theta + x_3. \tag{94}$$

For this example, the FEM is applied in the x_3 split direction with a mesh number of 10. A total of 9×31 nodes are distributed in a half-torus domain for the 2D problem. Among these nodes, 9 nodes are positioned along the radial direction r , and 31 nodes are uniformly distributed along the angle axis θ as shown in Figure 15 given in [27]. This means that the integral node distribution is $9 \times 31 \times 11$ with $d_{\max} = 1.0$ and $\alpha = 0.12$. The resulting relative error is 0.1510% and the computational time is 1.3 s.

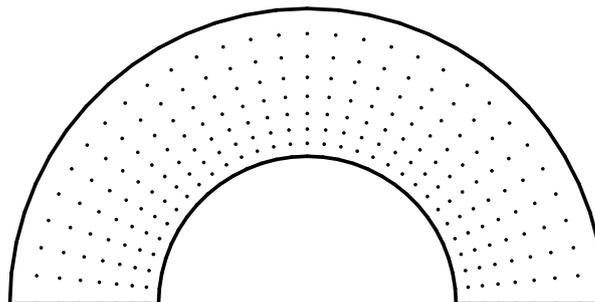


Figure 15. Node distribution in half-torus 2D sub-domain [27].

For this example, in order to verify the effectiveness and correctness of the DCM, the results of the DCM are compared with those of the IIEFG method given in [27] using $9 \times 31 \times 11$ nodes, $8 \times 30 \times 10$ integral cells, and the cubic spline weight function. When $d_{\max} = 1.2$, and $\alpha = 1.0 \times 10^4$ are selected, the relative error and the computational time of the IIEFG method are 0.2377% and 132.1 s, respectively.

The comparison of the computational accuracy and time of the present DCM and the IIEFG method given in [27] is shown in Table 4.

Table 4. The comparison of computational accuracy and time of the present DCM and the IIEFG method given in [27].

Method	Nodes	Relative Error	CPU Time (s)
DCM	$9 \times 31 \times 11$	0.1510%	1.3
IEFG [27]	$9 \times 31 \times 11$	0.2377%	132.1

Comparing the numerical solutions of the present DCM with those of the IIEFG method given in [27] and the exact ones in Figures 16–18, the results of the DCM agree well with the exact ones.

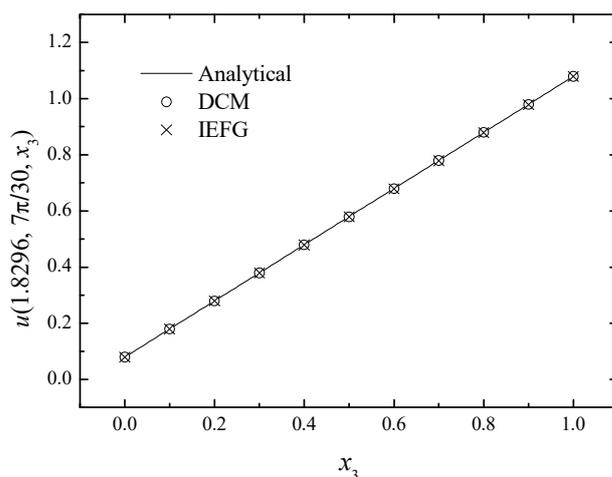


Figure 16. The numerical results of the present DCM, the IIEFG method [27], and analytical solutions along x_3 .

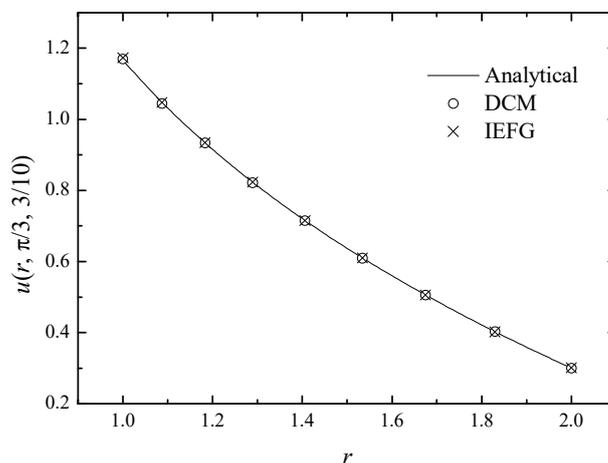


Figure 17. The numerical results of the present DCM, the IIEFG method [27], and analytical solutions along the radial direction r .

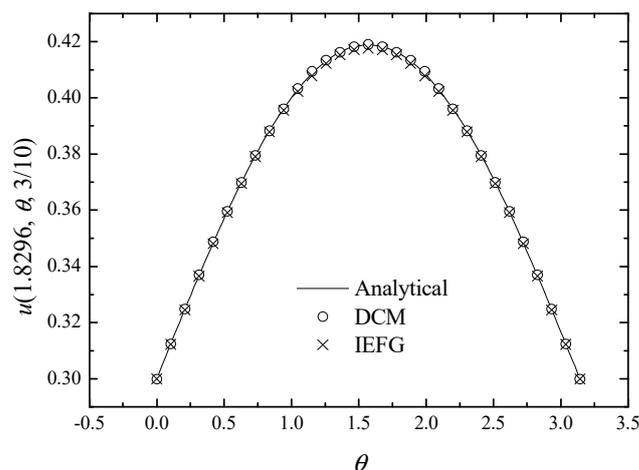


Figure 18. The numerical results of the present DCM, the IIEFG method [27], and analytical solutions along the angle axis θ .

5. Discussions

In this study, a new dimensional split EFG method called the DCM is studied. The proposed method is used to solve 3D Laplace equations.

The influence of meshes of the FEM in the third direction, the weight function, other parameters, and the node distribution in 2D domains on precision is discussed. Additionally, the convergence of the proposed method in this study is numerically verified. Four numerical examples are provided to illustrate that the new method significantly enhances computational efficiency without reducing computational accuracy compared to the IIEFG method.

6. Conclusions

Due to the lower computational efficiency of the EFG and IIEFG methods in solving 3D Laplace equations, we propose a new dimension splitting EFG method named the DCM, which efficiently solves the 3D Laplace equation. Moreover, it conveniently and efficiently handles mixed boundary conditions by using the FEM instead of the FDM in the splitting direction. Consequently, this study provides an efficient and potential numerical method that can reduce computational time and costs without sacrificing accuracy to solve 3D complex engineering problems, including but not limited to crack propagation, high-speed collisions and explosions, metal press forming, and large deformation problems in gradient materials. Because of the large-scale calculation of such problems, research into the DCM combined with parallel algorithms needs to be investigated in the future, which has not been addressed before.

However, the DCM still has some limitations. First, the MATLAB program for the DCM is more complex than that for the IIEFG method. Moreover, challenges may arise when using the DCM presented in this study combined with parallel algorithms. Therefore, this new method still needs further improvement in future research.

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

Funding: This work was supported by the Natural Science Foundation of Shanxi Province (Grant No. 202203021212233 and 20210302124388).

Data Availability Statement: Not applicable.

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

Nomenclature

List of symbols in formulas.

Symbol	Nomenclature	Symbol	Nomenclature
$u(x)$	Approximation function	Γ_q	Natural boundary
$\tilde{\Phi}$	Shape function of IIEFG method	$\Omega^{(k)}$	2D domain of k -th layer
$p(x)$	Basis function of IIEFG method	$\Gamma_u^{(k)}$	Essential boundary in k -th layer
$w(x - x_I)$	Weighting function of IIEFG method	$\Gamma_q^{(k)}$	Natural boundary in k -th layer
Ω	Problem domain	d_{\max}	Scale parameter
Γ	Boundary	α	Penalty factor
Γ_u	Essential boundary	$v = \varphi_i$	Test function of FEM

List of abbreviations and corresponding full names.

Abbreviation	Full name	Abbreviation	Full name
DCM	Dimension Coupling Method	IEFG	Improved Element-Free Galerkin
DSM	Dimensional Split Method	MLS	Moving Least-Squares
FEM	Finite Element Method	IMLS	Improved Moving Least-Squares
FDM	Finite Difference Method	3D	Three-dimensional
EFG	Element-Free Galerkin	2D	Two-dimensional

References

- Cheng, Y.M. *Meshless Methods*; Science Press: Beijing, China, 2015.
- Belytschko, T.; Lu, Y.Y.; Gu, L. Element-free Galerkin Methods. *Int. J. Numer. Methods Eng.* **1994**, *37*, 229–256.
- Gavete, L.; Falcon, S.; Ruiz, A. An error indicator for the element-free Galerkin method. *Eur. J. Mech. A/Solids* **2001**, *20*, 327–341.
- Gavete, L.; Guesta, J.L.; Ruiz, A. A procedure for approximation of the error in the EFG method. *Int. J. Numer. Methods Eng.* **2002**, *53*, 677–690.
- Gavete, L.; Gavete, M.L. A posteriori error approximation in EFG method. *Int. J. Numer. Methods Eng.* **2003**, *58*, 2239–2263.
- Cheng, Y.M.; Chen, M.J. A boundary element-free method for linear elasticity. *Acta Mech. Sin.* **2003**, *35*, 181–186.
- Ren, H.P.; Cheng, Y.M. The interpolating element-free Galerkin (IEFG) method for two-dimensional elasticity problems. *Int. J. Appl. Mech.* **2011**, *3*, 735–758.
- Sun, F.X.; Wang, J.F.; Cheng, Y.M. An improved interpolating element-free Galerkin method for elastoplasticity via nonsingular weight functions. *Int. J. Appl. Mech.* **2016**, *8*, 1650096.
- Cheng, Y.M.; Peng, M.J.; Li, J.H. The complex variable moving least-square approximation and its application. *Acta Mech. Sin.* **2005**, *37*, 719–723.
- Bai, F.N.; Li, D.M.; Wang, J.F.; Cheng, Y.M. An improved complex variable element-free Galerkin method for two-dimensional elasticity problems. *Chin. Phys. B* **2012**, *21*, 020204.
- Zhang, Z.; Zhao, P.; Liew, K.M. Analyzing three-dimensional potential problems with the improved element-free Galerkin method. *Comput. Mech.* **2009**, *44*, 273–284.
- Zheng, G.D.; Cheng, Y.M. The improved element-free Galerkin method for diffusional drug release problems. *Int. J. Appl. Mech.* **2020**, *12*, 2050096.
- Cheng, H.; Zheng, G.D. Analyzing 3D advection-diffusion problems by using the improved element-free Galerkin method. *Math. Probl. Eng.* **2020**, *2020*, 4317538.
- Cheng, H.; Peng, M.J. The improved element-free Galerkin method for 3D Helmholtz equations. *Mathematics* **2022**, *10*, 14. [[CrossRef](#)]
- Liu, D.; Cheng, Y.M. The interpolating element-free Galerkin method for three-dimensional transient heat conduction problems. *Results Phys.* **2020**, *19*, 103477.
- Wu, Q.; Liu, F.B.; Cheng, Y.M. The interpolating element-free Galerkin method for three-dimensional elastoplasticity problems. *Eng. Anal. Bound. Elem.* **2020**, *115*, 156–167.
- Wu, Q.; Peng, P.P.; Cheng, Y.M. The interpolating element-free Galerkin method for elastic large deformation problems. *Sci. China Technol. Sci.* **2021**, *64*, 364–374.
- Liu, F.B.; Cheng, Y.M. The improved element-free Galerkin method based on the nonsingular weight functions for elastic large deformation problems. *Int. J. Comput. Mater. Sci. Eng.* **2018**, *7*, 1850023.
- Liu, F.B.; Wu, Q.; Cheng, Y.M. A meshless method based on the nonsingular weight functions for elastoplastic large deformation problems. *Int. J. Appl. Mech.* **2019**, *11*, 1950006.

20. Liu, F.B.; Cheng, Y.M. The improved element-free Galerkin method based on the nonsingular weight functions for inhomogeneous swelling of polymer gels. *Int. J. Appl. Mech.* **2018**, *10*, 1850047.
21. Li, D.M.; Liew, K.M.; Cheng, Y.M. An improved complex variable element-free Galerkin method for two-dimensional large deformation elastoplasticity problems. *Comput. Methods Appl. Mech. Eng.* **2014**, *269*, 72–86.
22. Cheng, H.; Peng, M.J.; Cheng, Y.M. Analyzing wave propagation problems with the improved complex variable element-free Galerkin method. *Eng. Anal. Bound. Elem.* **2019**, *100*, 80–87. [[CrossRef](#)]
23. 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.* **2017**, *84*, 52–62. [[CrossRef](#)]
24. 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.* **2017**, *9*, 1750090. [[CrossRef](#)]
25. Cheng, H.; Peng, M.J.; Cheng, Y.M. The dimension splitting and improved complex variable element-free Galerkin method for 3-dimensional transient heat conduction problems. *Int. J. Numer. Methods Eng.* **2018**, *114*, 321–345. [[CrossRef](#)]
26. Cheng, H.; Peng, M.J.; Cheng, Y.M.; Meng, Z.J. The hybrid complex variable element-free Galerkin method for 3D elasticity problems. *Eng. Struct.* **2020**, *219*, 110835. [[CrossRef](#)]
27. Meng, Z.J.; Cheng, H.; Ma, L.D.; Cheng, Y.M. The dimension split element-free Galerkin method for three-dimensional potential problems. *Acta Mech. Sin.* **2018**, *34*, 462–474. [[CrossRef](#)]
28. Meng, Z.J.; Cheng, H.; Ma, L.D.; Cheng, Y.M. The dimension splitting element-free Galerkin method for 3D transient heat conduction problems. *Sci. China Phys. Mech. Astron.* **2019**, *62*, 040711. [[CrossRef](#)]
29. Meng, Z.J.; Cheng, H.; Ma, L.D.; Cheng, Y.M. The hybrid element-free Galerkin method for three-dimensional wave propagation problems. *Int. J. Numer. Methods Eng.* **2019**, *117*, 15–37. [[CrossRef](#)]
30. Ma, L.D.; Meng, Z.J.; Chai, J.F.; Cheng, Y.M. Analyzing 3D advection-diffusion problems by using the dimension splitting element-free Galerkin method. *Eng. Anal. Bound. Elem.* **2020**, *111*, 167–177. [[CrossRef](#)]
31. Peng, P.P.; Wu, Q.; Cheng, Y.M. The dimension splitting reproducing kernel particle method for three-dimensional potential problems. *Int. J. Numer. Methods Eng.* **2020**, *121*, 146–164. [[CrossRef](#)]
32. Peng, P.P.; Cheng, Y.M. Analyzing three-dimensional transient heat conduction problems with the dimension splitting reproducing kernel particle method. *Eng. Anal. Bound. Elem.* **2020**, *121*, 180–191. [[CrossRef](#)]
33. Peng, P.P.; Cheng, Y.M. Analyzing three-dimensional wave propagation with the hybrid reproducing kernel particle method based on the dimension splitting method. *Eng. Comput.* **2022**, *38*, 1131–1147. [[CrossRef](#)]
34. Peng, P.P.; Cheng, Y.M. A hybrid reproducing kernel particle method for three-dimensional advection-diffusion problems. *Int. J. Appl. Mech.* **2021**, *13*, 2150085. [[CrossRef](#)]
35. Wu, Q.; Peng, M.J.; Cheng, Y.M. The interpolating dimension splitting element-free Galerkin method for 3D potential problems. *Eng. Comput.* **2022**, *38*, 2703–2717. [[CrossRef](#)]
36. Wu, Q.; Peng, M.J.; Cheng, Y.M. The dimension splitting interpolating element-free Galerkin method for solving three-dimensional transient heat conduction problems. *Eng. Anal. Bound. Elem.* **2021**, *128*, 326–431. [[CrossRef](#)]
37. Ali, B.; Shafiq, A.; Alanazi, M.M.; Hendi, A.A.; Hussein, A.K.; Shah, N.A. Significance of Nanoparticle Radius and Gravity Modulation on Dynamics of Nanofluid over Stretched Surface via Finite Element Simulation: The Case of Water-Based Copper Nanoparticles. *Mathematics* **2023**, *11*, 1266. [[CrossRef](#)]
38. Zhang, B.; Tang, L.Q.; Zhang, H.W.; Ali, B.; Shah, N.A.; Jeon, Y.S. Finite element study of nanoparticles spacing and radius on dynamics of water fluid subject to microgravity environment. *Results Phys.* **2023**, *47*, 106355. [[CrossRef](#)]
39. Cheng, H.; Xing, Z.B.; Yao, L.L. The dimension coupling method for 3D convection-diffusion-reaction problems with variable coefficients. *Eng. Anal. Bound. Elem.* **2023**, *151*, 164–174. [[CrossRef](#)]
40. Kouatchou, J.; Zhang, J. Optimal injection operator and high order schemes for multigrid solution of 3D poisson equation. *Int. J. Comput. Math.* **2000**, *76*, 173–190. [[CrossRef](#)]
41. Romão, E.C.; Campos, M.D.; Moura, L.F.M. Application of the Galerkin and least-squares finite element methods in the solution of 3D Poisson and Helmholtz equations. *Comput. Math. Appl.* **2011**, *62*, 4288–4299. [[CrossRef](#)]

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