

A meshless local Petrov–Galerkin method for solving the neutron diffusion equation

Shima Tayefi¹ · Ali Pazirandeh¹ · Mohsen Kheradmand Saadi¹

Received: 5 May 2018 / Revised: 8 June 2018 / Accepted: 11 July 2018 / Published online: 16 October 2018
© Shanghai Institute of Applied Physics, Chinese Academy of Sciences, Chinese Nuclear Society, Science Press China and Springer Nature Singapore Pte Ltd. 2018

Abstract The goal of this study is to solve the neutron diffusion equation by using a meshless method and evaluate its performance compared to traditional methods. This paper proposes a novel method based on coupling the meshless local Petrov–Galerkin approach and the moving least squares approximation. This computational procedure consists of two main steps. The first involved applying the moving least squares approximation to construct the shape function based on the problem domain. Then, the obtained shape function was used in the meshless local Petrov–Galerkin method to solve the neutron diffusion equation. Because the meshless method is based on eliminating the mesh-based topologies, the problem domain was represented by a set of arbitrarily distributed nodes. There is no need to use meshes or elements for field variable interpolation. The process of node generation is simply and fully automated, which can save time. As this method is a local weak form, it does not require any background integration cells and all integrations are performed locally over small quadrature domains. To evaluate the proposed method, several problems were considered. The results were compared with those obtained from the analytical solution and a Galerkin finite element method. In addition, the proposed method was used to solve neutronic calculations in the

small modular reactor. The results were compared with those of the citation code and reference values. The accuracy and precision of the proposed method were acceptable. Additionally, adding the number of nodes and selecting an appropriate weight function improved the performance of the meshless local Petrov–Galerkin method. Therefore, the proposed method represents an accurate and alternative method for calculating core neutronic parameters.

Keywords Neutron diffusion equation · Meshless local Petrov–Galerkin (MLPG) · Moving least squares approximation (MLSA) · Meshless methods

1 Introduction

The numerical tools of partial differential equations (PDE) play a key role in the accuracy and acceleration of the solution process in engineering problems. In the last few years, significant progress has been achieved in this field due to the development of computer technology. Despite this progress, numerical methods are often generalized to solve differential equations in a heuristic manner [1]. Numerical methods consist of two types based on the discretization method. The first employs a mesh (mesh-based method), and the second uses a local approximation by polynomials (meshless method).

Mesh-based numerical methods are divided into three classical groups for solving PDEs: finite difference methods (FDM), finite volume methods (FVM), and finite element methods (FEM) [2]. These methods have been used to solve engineering problems for more than 60 years and cover a wide scope of problems such as material science,

✉ Ali Pazirandeh
Paziran@ut.ac.ir

Shima Tayefi
Shimatayefy@yahoo.com

Mohsen Kheradmand Saadi
Mohsen.kheradmand@gmail.com

¹ Department of Nuclear Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran

aerospace engineering, computational physics, safety, reliability analysis, solid mechanic, fluid mechanical interactions, and molecular interaction. Mesh-based methods are powerful numerical tools for solving problems and evaluating, predicting, and simulating the behavior of systems. However, mesh-based methods have some limitations in that they cannot provide fully acceptable results. Moreover, despite the great progress in mesh generation, creating an accurate mesh remains difficult. In addition, mesh-based methods are typically time-consuming [1].

Therefore, there is an interest in developing alternative methods that remove or reduce the need for meshing. A review of these methods shows that meshless methods have achieved considerable success in recent decades and have attracted substantial interest from researchers because of their applications for solving computational problems [3–9]. The most outstanding of these methods are introduced below.

The advent of the meshless concept dates back to the model of astrophysical phenomena without boundaries (smooth particle hydrodynamics (SPH)), for example, exploding stars and dust clouds. This idea was proposed as a theory for non-spherical stars by Gingold and Monaghan [10]. The SPH scheme has been successfully applied to many problems such as heat conduction, explosion phenomena, and free surface [11, 12]. Then, Nayroles et al. [13] proposed the diffuse element method that has been employed by many researchers. Belytschko et al. [3] introduced an extended version of Nayroles's method, which included a series of improvements over the diffuse element method formulation (the element-less Galerkin methods) [14]. Liu et al. [15, 16] introduced a new method for correcting the lack of consistency in the SPH scheme, known as the reproducing kernel particle method.

Oñate et al. [17] studied the finite point manner to model elasticity and plate bending and fluid flow problems using the collocation point technique, least squares approximation, and weighted least squares approximation. Since then, research has tended to use the radial basis function approximation technique [18–21]. Several studies on radial basis functions have been applied to solve physical problems such as transport phenomena [22], heat conduction [23], neutron diffusion [24, 25], and analysis of Kirchhoff plates [26]. De and Bathe [27] proposed the finite spheres approach for constructing the approximation function. Moradipour and Yousefi [28] used a meshless kernel-based method to solve the Black–Scholes variational inequality of American options. In addition to the analysis of two-dimensional elasticity problems, Ebrahimnejad et al. [29] presented three schemes of the 2D meshless finite volume method. One of the main methods of meshless schemes is the meshless local Petrov–Galerkin (MLPG) method proposed by [30]. The

MLPG method has frequently been used to successfully solve many engineering problems [31–35].

This method is based on the idea of the local weak form, which eliminates the requirement of the background cell. In addition, this method provides numerical integration in a meshless sense to simplify the integrand of the weak form [4, 6, 30, 36–38]. In this global weak form method, a rational basis is provided to construct meshless methods with a larger degree of flexibility [36, 39–44]. The MLPG method is an effective method for solving many problems, which is based on a local weak form and the moving least squares approximation (MLSA) to approximate the shape functions [45, 46]. As this method does not need a “finite element mesh,” it is a truly meshless method [2]. All integrals in the formulation can be easily evaluated over regularly shaped domains and their boundaries [30]. The shape functions must be effectively constructed to achieve a desirable order of continuity.

Accordingly, the capability and flexibility of meshless methods provide the motivation for using this approach to solve the neutron diffusion equation in this study.

2 Methodology

In this research, the MLPG method was used to solve the neutron diffusion equation in two-dimensional geometry. First, the solution was approximated by using a local approximation in the problem domain. This approximation scheme was used to formulate the residual of the governing equations on the problem and find the shape function. The obtained shape function was employed in the MLPG algorithm to solve the problem. Then, the problem was modeled by the analytical solutions and the Galerkin finite element method (GFEM) to evaluate the proposed method. In the following subsections, all steps required to derive the shape function and the weak form of the problem equation are explained. A schematic representation of the local support domains used to create the shape function is demonstrated in Fig. 1. The domain representation of the assumed problem of the mesh-based and meshless approaches is shown in Fig. 2.

2.1 Determining the shape function by MLSA

Selection of an approximation function has an important role for reaching a sufficient number of equations to determine the unknowns [47, 48]. Some of the most important functions consist of Shepard functions [40, 49], least squares [14, 50], local radial point interpolation [51], and local point interpolation [52]. One of the best schemes for interpolating data with an appropriate accuracy is the moving least squares method, which has more compatibility with MLPG method [53]. The moving least squares

Fig. 1 A schematic representation of the local support domains in the meshless approach

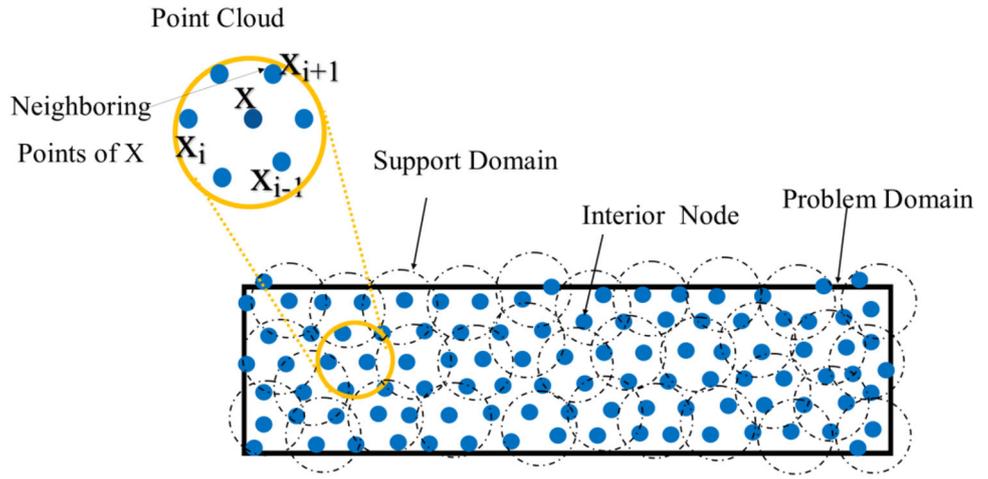
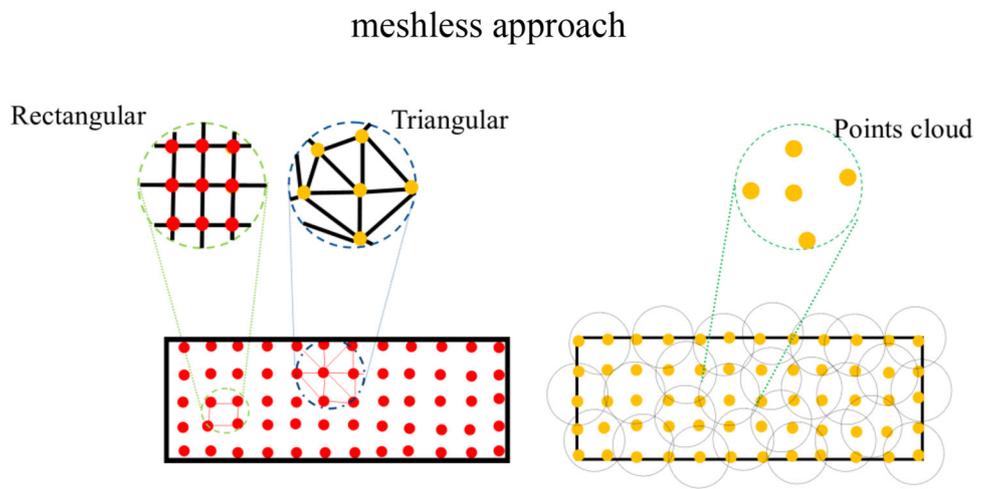


Fig. 2 Comparison of a problem domain in a meshless approach and a mesh-based method



approximation was created by mathematic researchers for data fitting and surface construction [54]. The steps for the determination of the shape function are as follows:

- (1) First, to demonstrate the trial function, the meshless approach should use local interpolation. Therefore, a function of the field unknown variables, $U(\mathbf{X})$, is defined in a subdomain, Ω_s , which indicates the support domain of the MLS approximation for $U(\mathbf{X})$. It should be mentioned that the support domain is located in the problem domain Ω_p . $\mathbf{X} = (x, y)$, which indicates a two-dimensional problem. To find an approximation distribution of function U in the support domain, the local moving squares approximation of function $U(\mathbf{X})$ for all \mathbf{X} in the support domain should be defined as follows [42]:

$$u^h(\mathbf{X}) = \sum_{j=1}^m p_j(\mathbf{X})a_j(\mathbf{X}) = \mathbf{P}^T(\mathbf{X})\mathbf{a}(\mathbf{X}). \tag{1}$$

$\mathbf{P}^T(\mathbf{X})$ is a complete monomial basis of order m . m is

the number of monomial terms, and here the linear basis $m = 3$.

$$\mathbf{P}^T(\mathbf{X}) = \mathbf{P}^T(x, y) = \{1, x, y\} \tag{2}$$

$\mathbf{a}(\mathbf{X})$ is a coefficient vector as follows:

$$\mathbf{a}^T(\mathbf{X}) = \mathbf{a}^T(x, y) = \{a_0(\mathbf{X}), a_1(\mathbf{X}), \dots, a_m(\mathbf{X})\}. \tag{3}$$

The approximated values of the field function are:

$$\mathbf{u}^h(\mathbf{X}, \mathbf{X}_I) = \mathbf{P}(\mathbf{X}_I)\mathbf{a}(\mathbf{X}) \quad I = 1, \dots, n. \tag{4}$$

- (2) In the second step, the weighted discrete L_2 norm is defined as follows:

$$J(\mathbf{X}) = \sum_{i=1}^n W(\mathbf{X} - \mathbf{X}_i) [U^h(\mathbf{X}, \mathbf{X}_i) - U_i(\mathbf{X}_i)]^2 = [\mathbf{P}^T(\mathbf{X}_i)\mathbf{a}(\mathbf{X}) - U_i]^2. \tag{5}$$

U_i is the nodal parameter of the field variable at node i . n indicates the number of neighborhood nodes of node i . The weight function $W(\mathbf{X} - \mathbf{X}_i)$ is the associated weight

function of node i . It is noted that the weight function for all internal and external nodes in the support domain is nonzero and zero, respectively.

- (3) In the third step, for determining the coefficients matrix, $\mathbf{a}(\mathbf{X})$, J should be minimized with respect to $\mathbf{a}(\mathbf{X})$ [42]:
- $$\frac{\partial J}{\partial \mathbf{a}} = 0. \tag{6}$$

This equation leads to a linear relationship (7) between $\mathbf{a}(\mathbf{X})$ and \mathbf{U} :

$$\mathbf{A}(\mathbf{X})\mathbf{a}(\mathbf{X}) = \mathbf{B}(\mathbf{X})\mathbf{U}_s. \tag{7}$$

Solving Eq. (7) for $\mathbf{a}(\mathbf{X})$ gives [54]:

$$\mathbf{a}(\mathbf{X}) = \mathbf{A}(\mathbf{X})^{-1}\mathbf{B}(\mathbf{X})\mathbf{U}_s, \tag{8}$$

where \mathbf{U}_s is a vector for choosing the field function nodal parameters for interior nodes of the support domain as follows:

$$\mathbf{U}_s = \{\mathbf{U}_1, \mathbf{U}_2, \dots, \mathbf{U}_n\}^T. \tag{9}$$

The matrices $\mathbf{A}(\mathbf{X})$ and $\mathbf{B}(\mathbf{X})$ are given as [54, 55]:

$$\mathbf{A}(\mathbf{X}) = \mathbf{P}^T\mathbf{W}\mathbf{P} = \sum_{i=1}^n W_i(\mathbf{X} - \mathbf{X}_i)\mathbf{P}(\mathbf{X}_i)\mathbf{P}^T(\mathbf{X}_i) \tag{10}$$

$$\mathbf{B}(\mathbf{X}) = \mathbf{P}^T\mathbf{W} = [W_1(\mathbf{X})\mathbf{P}(\mathbf{X}_1), \dots, W_n(\mathbf{X})\mathbf{P}(\mathbf{X}_n)], \tag{11}$$

where $W_i(\mathbf{X}) = \mathbf{W}(\mathbf{X} - \mathbf{X}_i)$.

Hence, for this two-dimensional problem, $\mathbf{A}(\mathbf{X})$ is a symmetric matrix according to the linear basis ($m = 3$) that can be explicitly written as:

$$\begin{aligned} \mathbf{A}_{3 \times 3}(x) &= \sum_{i=1}^n W_i(x)\mathbf{P}(x_i)\mathbf{P}^T(x_i) \\ &= \mathbf{W}(x - x_1) \begin{bmatrix} 1 & x_1 & y_1 \\ x_1 & x_1^2 & x_1y_1 \\ y_1 & x_1y_1 & y_1^2 \end{bmatrix} \\ &\quad + \mathbf{W}(x - x_2) \begin{bmatrix} 1 & x_2 & y_2 \\ x_2 & x_2^2 & x_2y_2 \\ y_2 & x_2y_2 & y_2^2 \end{bmatrix} \\ &\quad + \mathbf{W}(x - x_n) \begin{bmatrix} 1 & x_n & y_n \\ x_n & x_n^2 & x_ny_n \\ y_n & x_ny_n & y_n^2 \end{bmatrix} \\ &= \begin{bmatrix} \sum_{i=1}^n W_i & \sum_{i=1}^n x_iW_i & \sum_{i=1}^n y_iW_i \\ \sum_{i=1}^n x_iW_i & \sum_{i=1}^n x_i^2W_i & \sum_{i=1}^n x_iy_iW_i \\ \sum_{i=1}^n y_iW_i & \sum_{i=1}^n x_iy_iW_i & \sum_{i=1}^n y_i^2W_i \end{bmatrix}_{3 \times 3} \end{aligned} \tag{12}$$

- (4) The fourth step is the derivation of the shape function. If the above equations are substituted into Eq. (4), the form of an interpolation function that is MLSA of $\mathbf{u}(\mathbf{X})$ at \mathbf{X} is written as:

$$\mathbf{u}^h(\mathbf{X}) = \sum_{j=1}^m \phi_j(\mathbf{X})U_j(\mathbf{X}) = \Phi^T(\mathbf{X})\mathbf{U}_s \tag{13}$$

$\Phi^T(\mathbf{X})$ can be rewritten as below [55]:

$$\begin{aligned} \Phi^T(\mathbf{X}) &= \{\phi_1(\mathbf{X})\phi_2(\mathbf{X}) \dots \phi_n(\mathbf{X})\}_{1 \times n} \\ &= \mathbf{p}^T(\mathbf{X})\mathbf{A}^{-1}(\mathbf{X})\mathbf{B}(\mathbf{X}), \end{aligned} \tag{14}$$

where $\Phi(\mathbf{X})$ indicates the MLS shape function vector for n nodes in the support domain of point \mathbf{X} . Therefore, the shape function for the i th node is introduced in Eq. (15) [55]:

$$\phi_i(\mathbf{X}) = \sum_{j=1}^m p_j(\mathbf{X})(\mathbf{A}^{-1}(\mathbf{X})\mathbf{B}(\mathbf{X}))_{ji}. \tag{15}$$

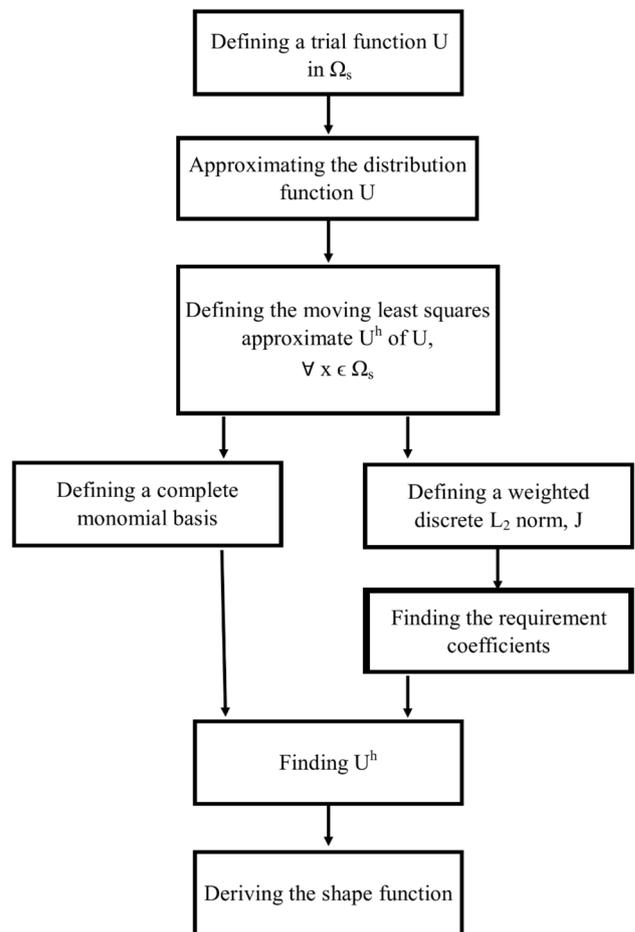


Fig. 3 Flowchart of the applied algorithm for deriving the shape function (MLSA algorithm)

A flowchart of the algorithm for the above steps is shown in Fig. 3.

2.2 Solving the neutron diffusion equation by MLPG

To solve the neutron diffusion equation, the nodal parameters u_j and its derivatives should be defined. u_j is a function of time, while the shape function ϕ_j is a function of \mathbf{X} only [56]. The spatial and temporal derivatives of the above shape function with respect to \mathbf{X} or \mathbf{Y} are extracted from the above equations. Equation (13) can then be rewritten as:

$$\mathbf{u}^h(\mathbf{X}, t) = \sum_{j=1}^n u_j(t) \phi_j(\mathbf{X}), \tag{16}$$

2.2.1 The temporal derivation of \mathbf{u}^h is

$$\mathbf{u}_t^h = \sum_{j=1}^n \dot{u}_j(t) \phi_j(\mathbf{X}). \tag{17}$$

2.2.2 The spatial derivation of \mathbf{u}^h is

$$\mathbf{u}_x^h = \sum_{j=1}^n u_j(t) \phi_j'(\mathbf{X}). \tag{18}$$

2.2.3 In addition, the second-order spatial domain is

$$\mathbf{u}_{xx}^h = \sum_{j=1}^n u_j(t) \phi_j''(\mathbf{X}). \tag{19}$$

The derivatives are denoted by subscripts. In addition, a local quadrature domain, Ω_{Q_i} , is defined for each interior node, x_i . R indicates the residual function. The integral of the weighted residual is zero [57].

$$\int_{\Omega_{Q_i}} RW_i d\Omega_{Q_i} = 0. \tag{20}$$

As mentioned earlier, W_i is a weight function that is nonzero only on Ω_{Q_i} and similar to the weight function used in the MLSA process.

2.3 Determining the residual of the neutron diffusion equation

In this step, two-dimensional neutron diffusion equations are first introduced. For arbitrary volume, v , the

balance neutron equation is considered. This well-known equation is shown below.

$$\begin{aligned} &[\text{Time rate of change of the number of neutron in a volume}] \\ &= [\text{Production rate in the volume}] \\ &- [\text{Absorptions in the volume}] \\ &- [\text{Net leakage from the surface of volume}]. \end{aligned}$$

The first term is mathematically expressed in Eq. (21):

$$\frac{d}{dt} \int_v \frac{1}{v} \varphi(r, t) dv. \tag{21}$$

The production rate is considered as a neutron source in Eq. (22):

$$\int_v S(r, t) dv. \tag{22}$$

The neutron absorption term is presented in Eq. (23):

$$\int_v \Sigma_a \varphi(r, t) dv. \tag{23}$$

The leakage term is shown in Eq. (24):

$$\int_v \nabla \cdot J dv = \frac{d}{dt} \int_v \nabla \cdot [D \nabla \varphi(r, t)] dv. \tag{24}$$

The mathematical formulation of the neutron diffusion theory is obtained as follows:

$$\frac{\partial \varphi}{\partial t} - D \nabla^2 \varphi(r, t) + \Sigma_a \varphi(r, t) = S(r, t). \tag{25}$$

Equation (25) can be rewritten as Eq. (26):

$$u_t - Du_{xx} + \Sigma_a u - S = 0, \tag{26}$$

where u is the solution, D is the diffusion coefficient, s is a neutron source, and Σ_a is the absorption cross section. The residual for this equation is defined according to an approximation solution as follows:

$$\text{Residual} = R = u_t^h - Du_{xx}^h + \Sigma_a u^h - s = 0. \tag{27}$$

Equation (27) is substituted into Eq. (20) to give a system of n linear equations for n nodes:

$$\int_{\Omega_{Q_i}} [u_t^h - Du_{xx}^h + \Sigma_a u^h - s] w_i d\Omega_{Q_i} = 0. \tag{28}$$

Then, each of the neutron diffusion equation terms should be substituted by Eq. (16) into Eq. (19):

- (a) The first term is obtained by substituting Eq. (17) into Eq. (28):

$$\int_{\Omega_{q_i}} [u_i^h] w_i d\Omega_{q_i} = \int \sum_{j=1}^n \dot{u}_j \varphi_j(x) w_i(x) dx. \tag{29}$$

(b) The second term is obtained by substituting Eq. (19) into Eq. (28):

$$\begin{aligned} -D \int_{\Omega_{q_i}} [u_{xx}^h] w_i d\Omega_{q_i} &= \int_{\Omega_{q_i}} Du_x^h w_{i,x}(x) d\Omega_{q_i} - Du_x^h w_i \Big|_{X_{qi1}}^{X_{qi n}} \\ &= D \int \sum_{j=1}^n u_j(t) \dot{\varphi}_j(x) \dot{w}_i(x) dx - Du_j(t) \varphi_j(x) w_i(x). \end{aligned} \tag{30}$$

(c) The third term is obtained by substituting Eq. (16) into Eq. (28):

$$\int_{\Omega_{q_i}} \Sigma_a u^h w_i d\Omega_{q_i} = \int \sum_{j=1}^n \Sigma_a u_j^h(t) \varphi_j(X) w_i(x) dx. \tag{31}$$

(d) The fourth term is obtained by substituting the fission effective factors into Eq. (22):

$$\int_{\Omega_{q_i}} -S w_i d\Omega_{q_i} = - \int \frac{\lambda}{\lambda} v \Sigma_f w_i(x) dx. \tag{32}$$

Therefore, Eq. (28) can be rewritten as:

$$\begin{aligned} &\int \sum_{j=1}^n \dot{u}_j \varphi_j(x) w_i(x) dx + D \int \sum_{j=1}^n u_j(t) \dot{\varphi}_j(x) \dot{w}_i(x) dx \\ &- D \sum_{j=1}^n u_j(t) \varphi_j(x) w_i(x) \Big|_{X_{qi1}}^{X_{qi n}} \\ &+ \int \sum_{j=1}^n \Sigma_a u_j^h(t) \varphi_j(X) w_i(x) dx - \int \frac{\lambda}{\lambda} v \Sigma_f w_i(x) dx = 0. \end{aligned} \tag{33}$$

For clarity, the sums are moved outside the integrals as shown in Eq. (34):

$$\begin{aligned} &\sum_{j=1}^n \int \dot{u}_j(t) \varphi_j(x) w_i(x) dx + D \sum_{j=1}^n \int u_j(t) \dot{\varphi}_j(x) \dot{w}_i(x) dx \\ &- D \sum_{j=1}^n u_j(t) \varphi_j(x) w_i(x) \Big|_{X_{qi1}}^{X_{qi n}} \\ &+ \sum_{j=1}^n \int \Sigma_a u_j(t) \varphi_j(x) w_i(x) dx - \int \frac{\lambda}{\lambda} v \Sigma_f w_i(x) dx = 0. \end{aligned} \tag{34}$$

This equation can be rearranged into its final form as shown in Eq. (35):

$$\begin{aligned} &\sum_{j=1}^n \dot{u}_j(t) \int \varphi_j(x) w_i(x) dx + D \sum_{j=1}^n u_j(t) \int \dot{\varphi}_j(x) \dot{w}_i(x) dx \\ &- D \sum_{j=1}^n u_j(t) \varphi_j(x) w_i(x) \Big|_{X_{qi1}}^{X_{qi n}} \\ &+ \sum_{j=1}^n u_j(t) \int \Sigma_a \varphi_j(x) w_i(x) dx - \frac{\lambda}{\lambda} v \Sigma_f \int w_i(x) dx = 0. \end{aligned} \tag{35}$$

It should be noted that the integrals obtained in discretized equations are regular integrals. To calculate the integrals, the numerical integration method based on the 20-point Gauss–Legendre quadrature was applied [33, 45, 58].

2.4 Solution of the time-dependent neutron diffusion equation

The first stage in the time-dependent solution is the discretization of space and time variables. The space and time variables are essentially discretized to gain a series of (nonlinear) algebraic systems. Therefore, it is necessary to decouple the space and time variables to solve this problem.

In this study, three different approaches to time discretization were applied, the forward Euler, Crank–Nicolson, and backward Euler scheme. It is notable that the results of the second scheme were more acceptable than other methods because the method is unconditionally stable. In addition, the time required for this process was shorter than other methods. In the method, time is replaced by the time derivative at the half-step $\dot{u}_j(t + \Delta t/2)$ with the central difference approximation [59]:

$$\dot{u}_j(t + \Delta t/2) = \frac{u_j(t + \Delta t) - u_j(t)}{\Delta t}. \tag{36}$$

In this discretization method, $u_j(t + \Delta t/2)$ is approximated as the average of $u_j(t)$ and $u_j(t + \Delta t)$. It is noted that the second-order accuracy in time and space is provided by the Crank–Nicolson method. In addition, this method is unconditionally stable.

By collecting all the nodal parameters into a vector $\mathbf{u}_j^{(t)} = (u_1^{(t)} \dots u_n^{(t)})$, the matrix form of Eq. (35) is derived as below:

$$M \sum_{j=1}^n \dot{u}_j(t) + N \sum_{j=1}^n u_j(t) - S = 0 \tag{37}$$

$$M \sum_{j=1}^n \dot{u}_j \left(t + \frac{\Delta t}{2} \right) + N \sum_{j=1}^n u_j \left(t + \frac{\Delta t}{2} \right) - S = 0 \tag{38}$$

$$M \left(\frac{u(t + \Delta t) - u(t)}{\Delta t} \right) + N \left(\frac{u(t + \Delta t) + u(t)}{2} \right) - S = 0 \tag{39}$$

$$2Mu(t + \Delta t) - 2Mu(t) + N\Delta tu(t) + N\Delta tu(t + \Delta t) - 2t\Delta t = 0 \tag{40}$$

$$\mathbf{u}^{(t+\Delta t)}(2M + N\Delta t) = 2t\Delta t + \mathbf{u}^{(t)}(2M - N\Delta t), \tag{41}$$

where M and N are coefficients of \dot{u}_j and u_j , respectively.

The proposed final relationship, which was used in the computational algorithm as the main formula at each time step, is:

$$\mathbf{u}^{(t+\Delta t)}\mathbf{L} = \mathbf{s}^* + \mathbf{u}^{(t)}\mathbf{K}. \tag{42}$$

All approaches were calculated by a code written entirely in FORTRAN 95. To enforce the boundary conditions in the problem, the following provisions should be considered. Each element of matrix \mathbf{L} , L_{ij} , is defined based on the three conditions. If x_j is in the support domain of boundary node x_i , L_{ij} is $\phi_j(x_i)$. If x_j is in the support domain of any point in the quadrature domain of the internal node x_i , L_{ij} is $2M_{ij} + N_{ij}\Delta t$. Otherwise, the value of L_{ij} is zero.

In addition, each element of matrix \mathbf{K} , K_{ij} , is defined based on the three conditions. If x_j is in the support domain of boundary node x_i , L_{ij} is zero. If x_j is in the support domain of any point in the quadrature domain of the internal node x_i , L_{ij} is $2M_{ij} - N_{ij}\Delta t$. Otherwise, the value of L_{ij} is zero and S^* is based on the problem conditions.

As mentioned previously, the shape function is an important requirement for applying meshless methods. A qualified approach should satisfy several conditions to provide the shape functions of meshless methods [48]. These conditions consist of having an arbitrary nodal distribution, being numerically stable, satisfying a certain order of consistency, and being compactly supported [48]. The accuracy of interpolation for the arbitrary points depends on the number of nodes in the support domain [53]. The number of nodes in each interpolation domain should be used to determine the appropriate shape function and local interpolation schemes to match the sensitivity of any meshless interpolation methods to a variable. Therefore, an appropriate support domain should be selected to apply an accurate approximation [4]. It should be noted that, in order to evaluate the performance of the proposed method, several weight functions such as the cubic spline, the quadratic spline, the compact support radial basis, and Gaussian weight functions were studied. These weight functions are presented below:

1. Cubic spline weight function

$$w_i(x) = \begin{cases} \frac{2}{3} - 4r_i^2 + 4r_i^3, & \text{for } 0 \leq r_i \leq \frac{1}{2} \\ \frac{4}{3} - 4r_i + 4r_i^2 - \frac{4}{3}r_i^3, & \text{for } \frac{1}{2} \leq r_i \leq 1 = 0, \\ \text{for } r_i > 1. \end{cases} \tag{43}$$

2. Quadratic spline weight function

$$w_i(x) = 1 - 6r_i^2 + 8r_i^3 - 3r_i^4 \quad \text{for } 0 \leq r_i \leq 1 = 0, \tag{44}$$

for $r_i > 1$.

3. Compact support radial basis function (CSRBF)

$$w_i(x) = (1 - r_i)^6 + [6 + 36r_i + 82r_i^2 + 72r_i^3 + 30r_i^4 + 5r_i^5], \tag{45}$$

for $0 \leq r_i \leq 1 = 0$, for $r_i > 1$.

4. Gaussian weight function

$$w_i(x) = \begin{cases} \frac{\exp\left[-\left(\frac{d_i}{c_i}\right)^2\right] - \exp\left[-\left(\frac{r_i}{c_i}\right)^2\right]}{1 - \exp\left[-\left(\frac{r_i}{c_i}\right)^2\right]} & \text{for } 0 \leq d_i \leq r_i \\ 0 & \text{for } d_i \geq r_i \end{cases} \tag{46}$$

where r_i is the size of the support for the weight function, w_i , and $d_i = |x - x_i|$ is the distance between node x_i and x . The distance between nodes is equal to half the mean free path of the neutron. The value of the mean free path for a thermal neutron in a reactor core is 0.7. In addition, the radius of the support domain for the weight function is approximately three mean free paths of the neutron. Moreover, c_i is a constant controlling the shape of the weight function, w_i , which is an effective parameter for the stability and accuracy of the method that is obtained through experiments. In this problem, c_i in the range of 0.3–0.8 leads to more stable and accurate results. Then, for each interior node, x_i , a local quadrature domain, Ω_{Q_i} , is defined [60].

The main steps of the proposed algorithm include:

1. Defining the initial condition of the problem;
2. Defining the boundary condition of the problem;
3. Setting up the cloud points;
4. Finding the affect domain of each node;
5. Looping over the time steps;
6. Applying the MLSA algorithm:
 - 6.1 Selecting the neighboring nodes;
 - 6.2 Deriving the shape functions and its derivative for the nodes;

- 6.3 Evaluating the nodal matrices and vectors;
- 6.4 Setting the nodal portion to the global matrices and vectors;
- 6.5 End loop (Fig. 3);
7. Solving the government equations at each node;
8. Applying the moving least squares shape function for interior nodes of a local domain for recalculation;
9. Determining the coefficients at each node to obtain higher accuracy;
10. Determining and evaluating the obtained results;
11. Evaluating the effects of increasing the amount of nodes in the problem domain;
12. Recording the history of the state variables and their derivatives;
13. Back to 5;
14. End.

The flowchart of the MLPG algorithm is demonstrated in Fig. 4.

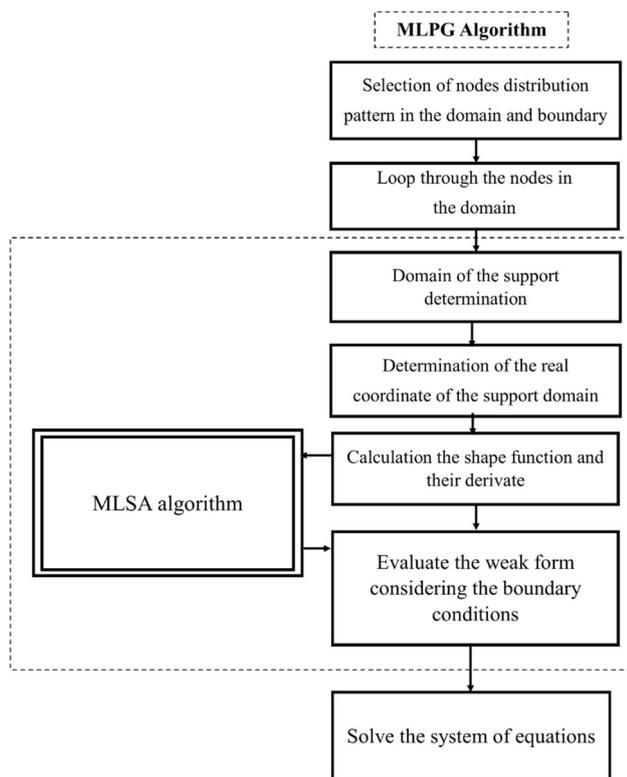


Fig. 4 Flowchart of the MLPG algorithm

3 Results and discussion

Several example problems were studied to validate the proposed method. The results of the MLPG method were then compared with the results of the analytical solution and the Galerkin finite element method.

3.1 Convergence degree test

In order to calculate the convergence degree of the MLPG method, a slab is considered. The boundary conditions are a perfect reflector on the left and a bare surface on the right of the slab. The model geometry is presented in Fig. 5. The logarithm of the error versus the number of nodes is presented in Fig. 6. It is clear that increasing the number of nodes decreases the error. Figure 7 illustrates the logarithm of the error versus the logarithm of the inverse of the number of nodes. It is well known that the slope of this diagram depicts the convergence rate. For

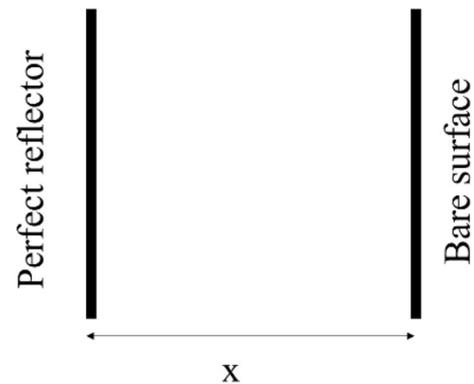


Fig. 5 Geometry of the convergence degree test

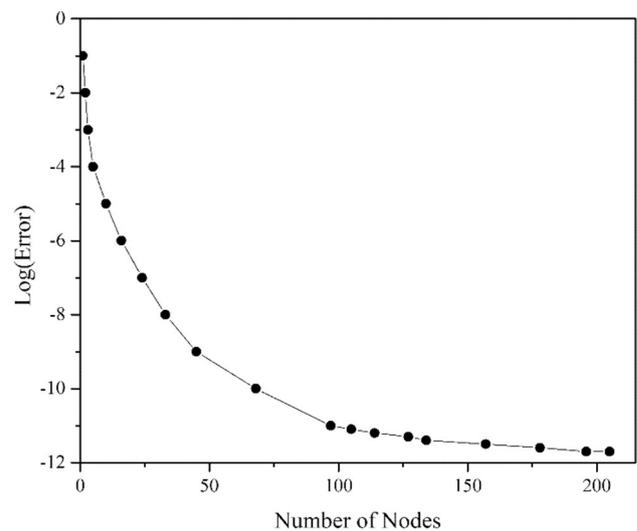


Fig. 6 Error logarithm versus the number of nodes

Fig. 7 **a** Logarithm of the error versus the logarithm of the inverse of the number of nodes and **b** the convergence rate versus the number of nodes

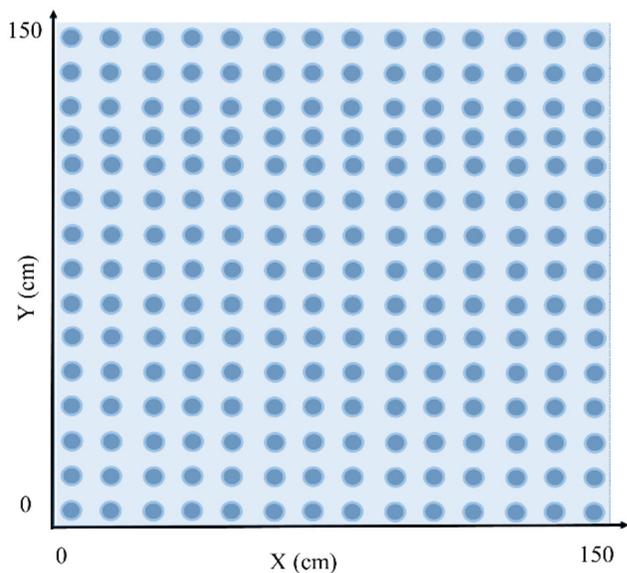
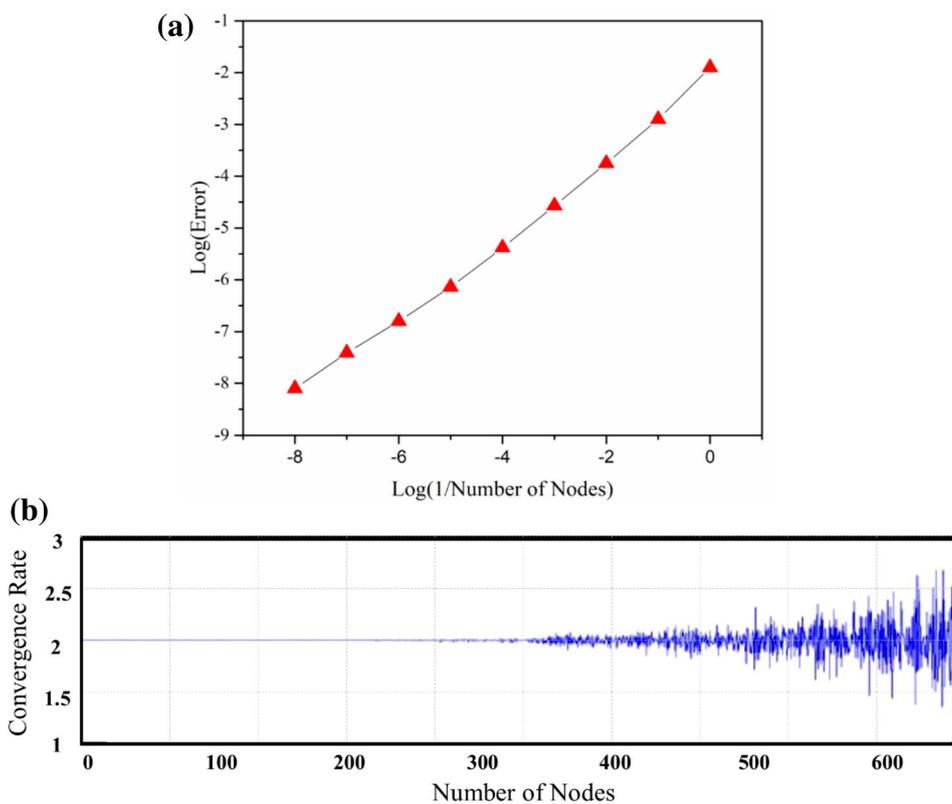


Fig. 8 Geometry of the single region

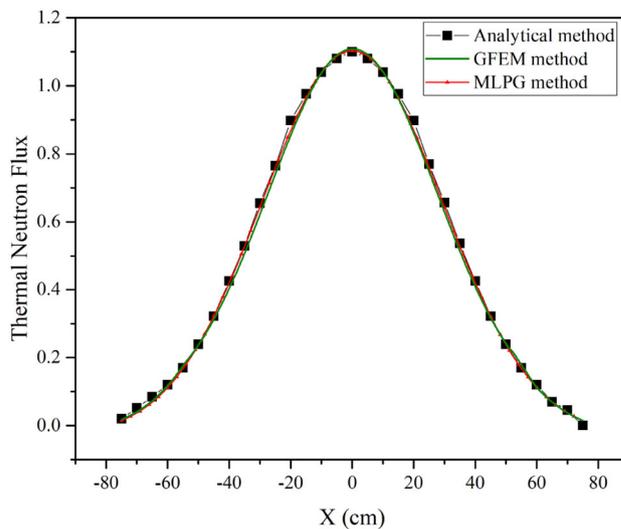


Fig. 9 Thermal neutron flux obtained from the MLPG and GFEM and the analytical solution (Case I)

Table 1 Applied neutronic parameters for Case I

Applied models	Energy group	Absorption cross section	Nu-fission cross section	Diffusion coefficient
Slab I	1	0.03051	0.000256	1.440130
	2	0.10582	0.154726	0.015723

further clarification, the convergence rate versus the number of nodes is presented in Fig. 7b, which demonstrates that the behavior of the proposed method is stable for less than 500 nodes.

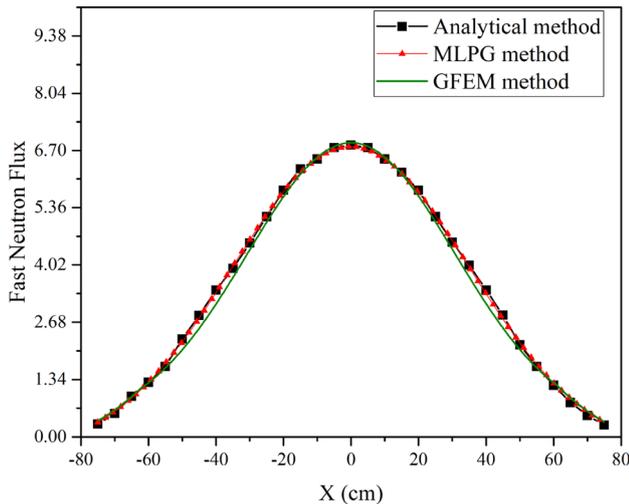


Fig. 10 Fast neutron flux obtained from the MLPG and GFEM and the analytical solution (Case I)

3.2 Constant source problem

In this test problem, a distributed source problem was considered. This problem consists of two models, a single region (Fig. 8) and two regions (Fig. 11) including a source in region 1 and a reflector in region 2 with dimensions of 150 cm × 150 cm. The boundary conditions include a perfect reflector on the left side and a bare surface on the right side. In this section, three types of problem were considered:

Case I The geometry of the single region is presented in Fig. 8. The parameters required for this calculation are given in Table 1. A comparison of the fast and thermal neutron flux of the MLPG method with that of the GFEM and the analytical solution is presented in Figs. 9 and 10, respectively.

Case II The geometry of the two-region problem is presented in Fig. 11, and the required parameters are given in Table 2. A comparison of the fast and the thermal neutron flux of the MLPG method with that of the GFEM and the analytical solution is presented in Figs. 12 and 13, respectively. As shown, the results of the proposed method exhibit good agreement with those of the analytical solution.

Fig. 11 Geometry of the two-region problem

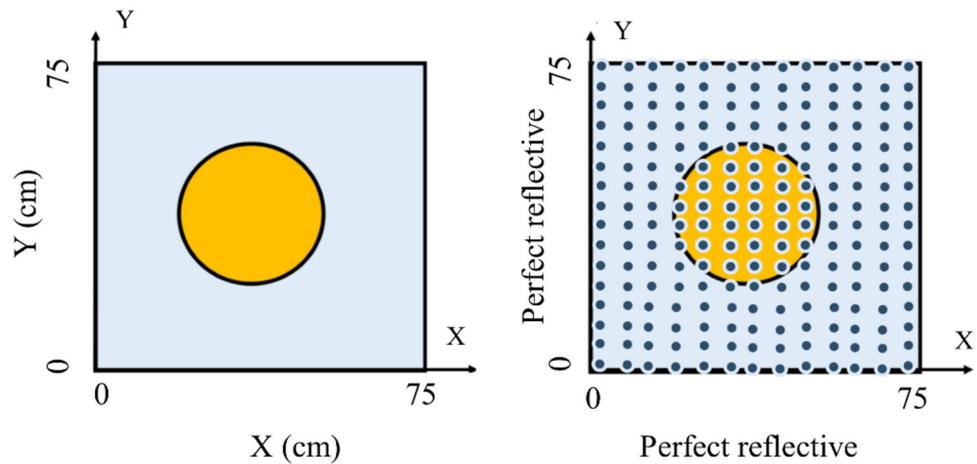


Table 2 Applied neutronic parameters for Case II

Applied models	Energy group	Absorption cross section	Nu-Fission cross section	Diffusion coefficient	Region
Slab II	1	0.03051	0.000256	1.440130	Fuel
	2	0.10582	0.154726	0.015723	
	1	0.035128	0.000000	1.852524	Reflector
	2	0.032540	0.000000	0.294517	

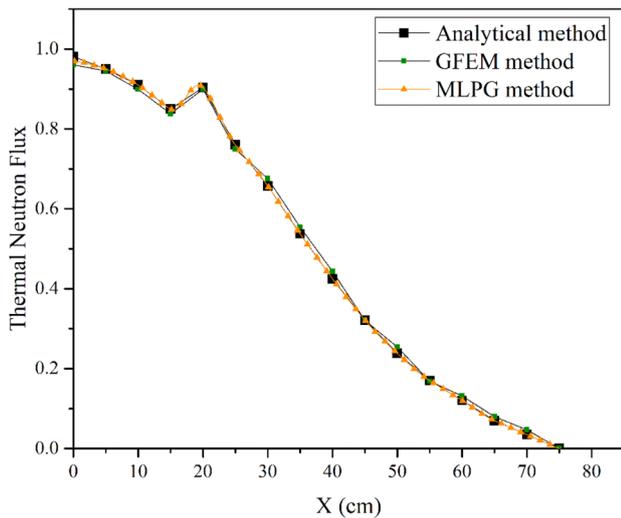


Fig. 12 Thermal neutron flux obtained from the MLPG and GFEM and the analytical solution (Case I)

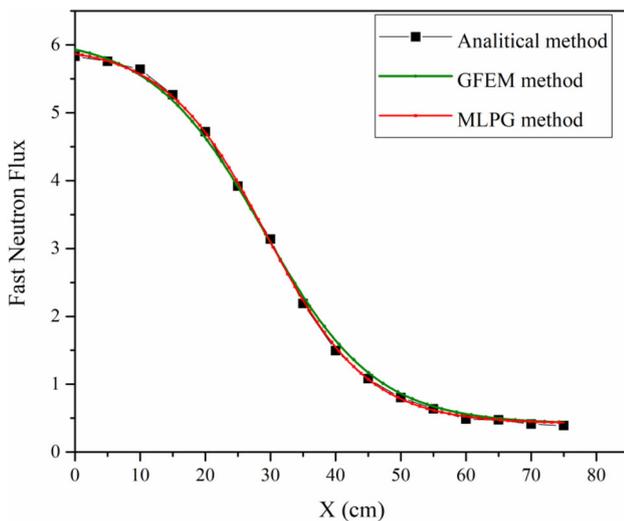


Fig. 13 Fast neutron flux obtained from the MLPG and GFEM and the analytical solution (Case II)

Table 3 Comparison between average errors obtained by applying various weight functions

Error	MLPG applied weight functions			
	Gaussian	CSRBF	Cubic spline	Quadratic spline
Case I	0.0035	0.073	0.094	0.085
Case II	0.0038	0.070	0.091	0.083

The average error of these results obtained by applying various weight functions is presented in Table 3, which indicates that the accuracy of the solution is improved by using Gaussian weight functions. In Table 4, the average

Table 4 Comparison between average errors and CPU time obtained by applying Gaussian and CSRBF weight functions

Problem No.	MLPG applied weight functions			
	Gaussian		CSRBF	
	Error	CPU time (s)	Error	CPU time (s)
Case I	0.0035	2	0.073	2.5
Case II	0.0038	3.5	0.070	6

error and CPU time of this problem are demonstrated for the Gaussian and CSRBF weight functions. Figure 14 presents a comparison of the fast neutron flux in Case I and Case II obtained by inserting 50×50 nodes.

In addition, to evaluate the performance of the number of inserted nodes, MLPG methods with 20×20 nodes and 40×40 nodes were compared. As shown, an accurate value of the thermal flux was achieved by the MLPG with 40×40 nodes for Case I (Fig. 15). Figure 15 indicates that the performance of the MLPG method was improved by increasing the number of nodes. Figure 16 demonstrates the contour of the fast and the thermal neutron flux. The best results for Case II were obtained with 40×40 nodes.

Case III In addition, in order to evaluate the performance of the proposed method for dealing with curve boundaries, a problem with a curved domain was considered. The geometry of this problem is presented in Fig. 17. A comparison of the contours of the thermal neutron flux of this problem is shown in Fig. 18. Two MLPG methods with various numbers of nodes were employed in this problem. The results show good agreement between the proposed method and analytical solutions. In addition, the obtained results illustrate that, to achieve the required accuracy, a large number of nodes can be inserted in the problem domain. Furthermore, increasing the number of nodes in a curved domain by the MLPG method is much simpler than mesh generation for mesh-based methods. However, the integration process for a curved domain is challenging. As mentioned, a parallel algorithm was used in this study to improve the calculation.

Case IV Moreover, in order to evaluate the performance of the proposed method, two types of node distribution, normal and scatter models, were applied to Case I. The average error of these results obtained by applying various weight functions is presented in Table 4. Table 5 indicates that the accuracy of the solution was improved by using Gaussian weight functions and the scatter distribution. The patterns of the node distributions are presented in Fig. 19.

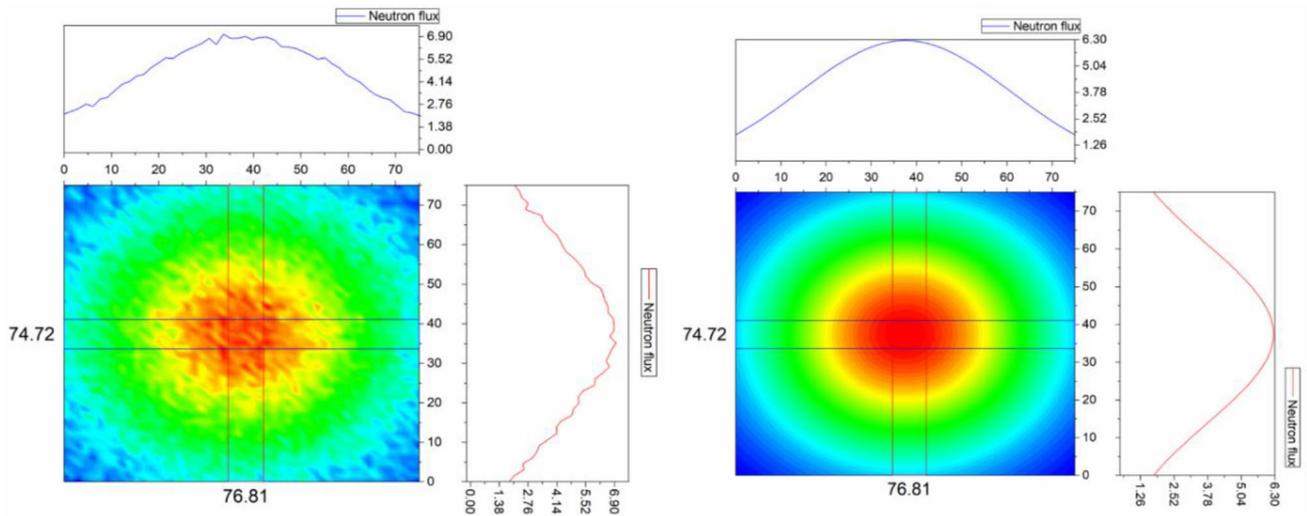


Fig. 14 Comparison of the fast neutron flux in Case I (left) and Case II (right)

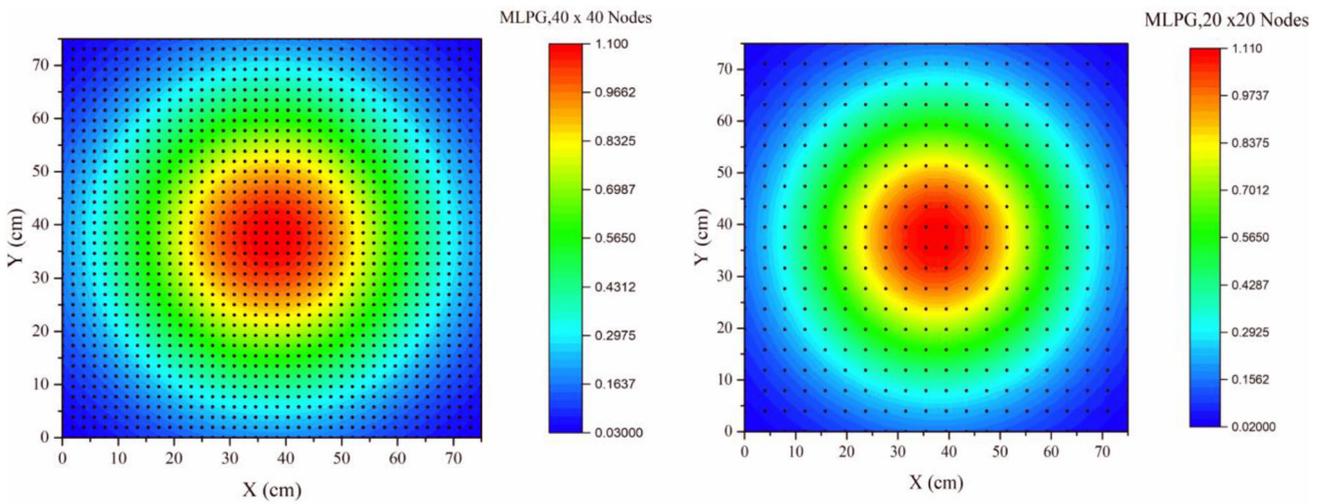


Fig. 15 Comparison of the effect of the number of nodes on the thermal flux

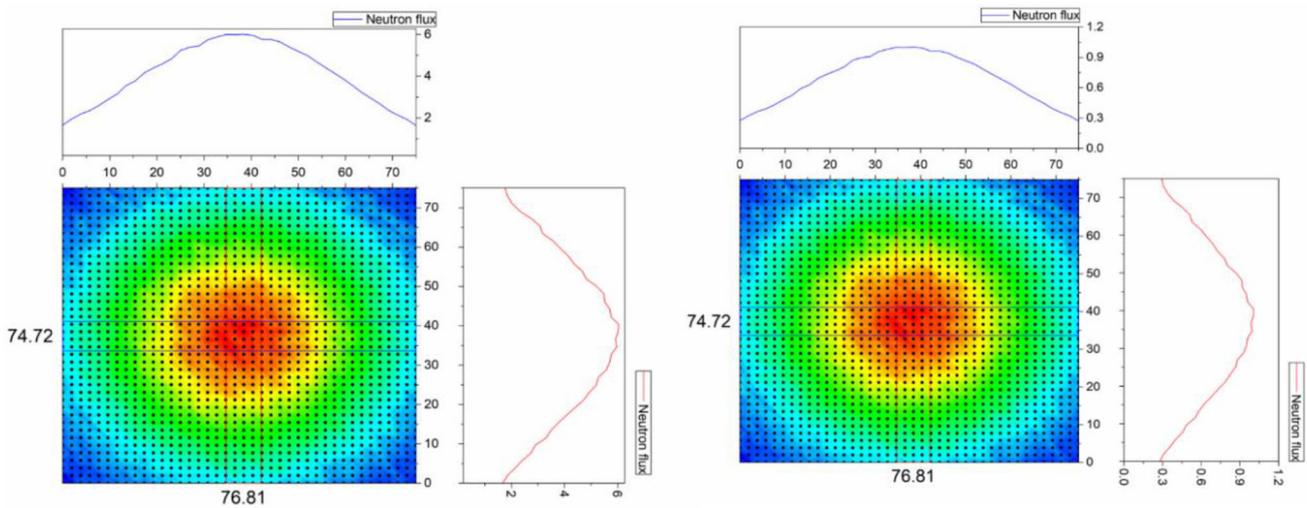


Fig. 16 Contour of the fast (left) and the thermal (right) neutron flux

Fig. 17 Geometry of Case III

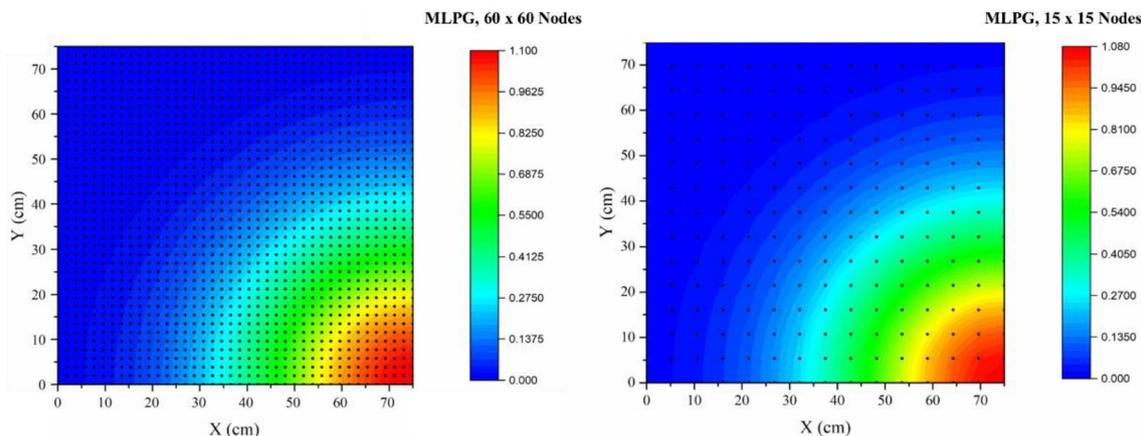
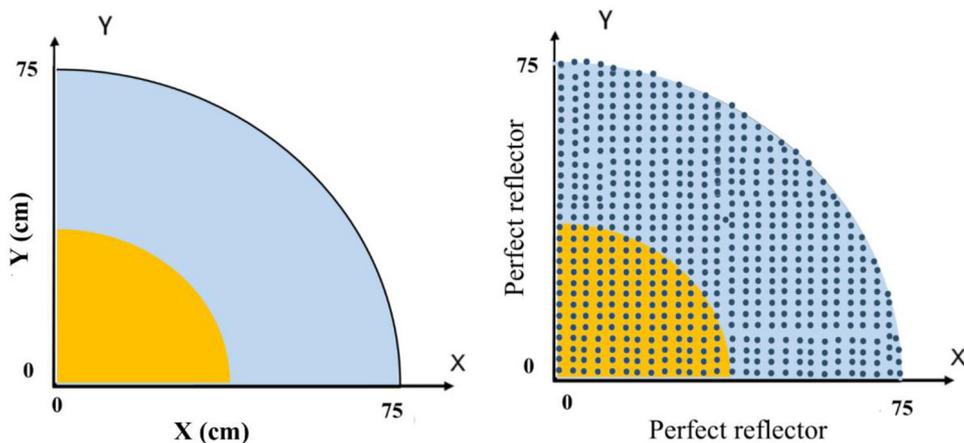


Fig. 18 Thermal neutron flux contour for Case III

Table 5 Comparison between average errors obtained by applying normal and scatter node distributions for Case I

Distribution pattern	MLPG applied weight functions			
	Gaussian	CSRBF	Cubic spline	Quadratic spline
Normal distribution	0.0035	0.073	0.094	0.085
Scatter distribution	0.0030	0.059	0.082	0.065

Fig. 19 a Normal and b scatter node distributions

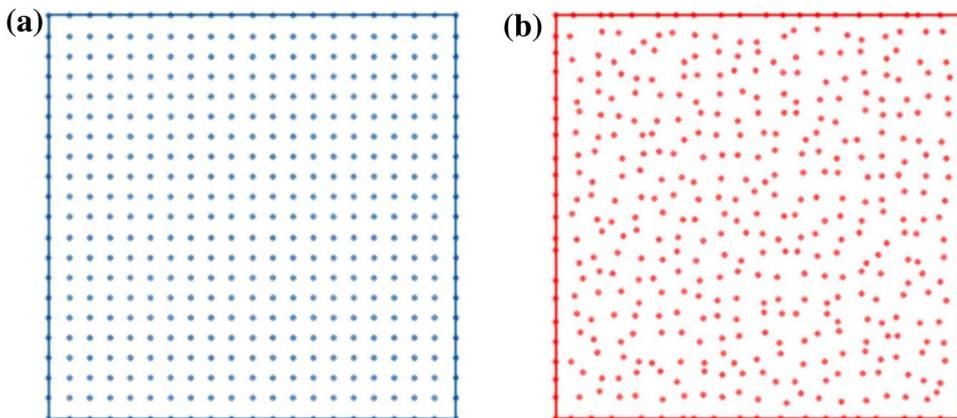


Fig. 20 Geometry of the void test problem

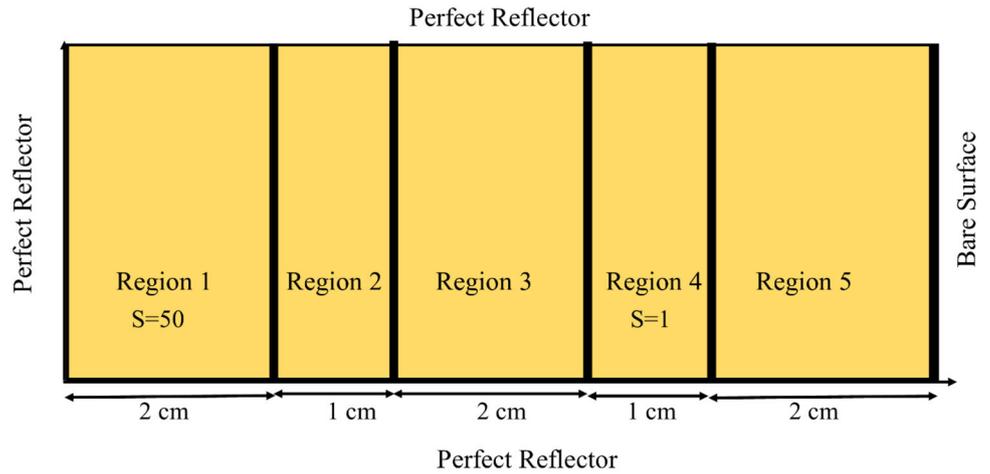
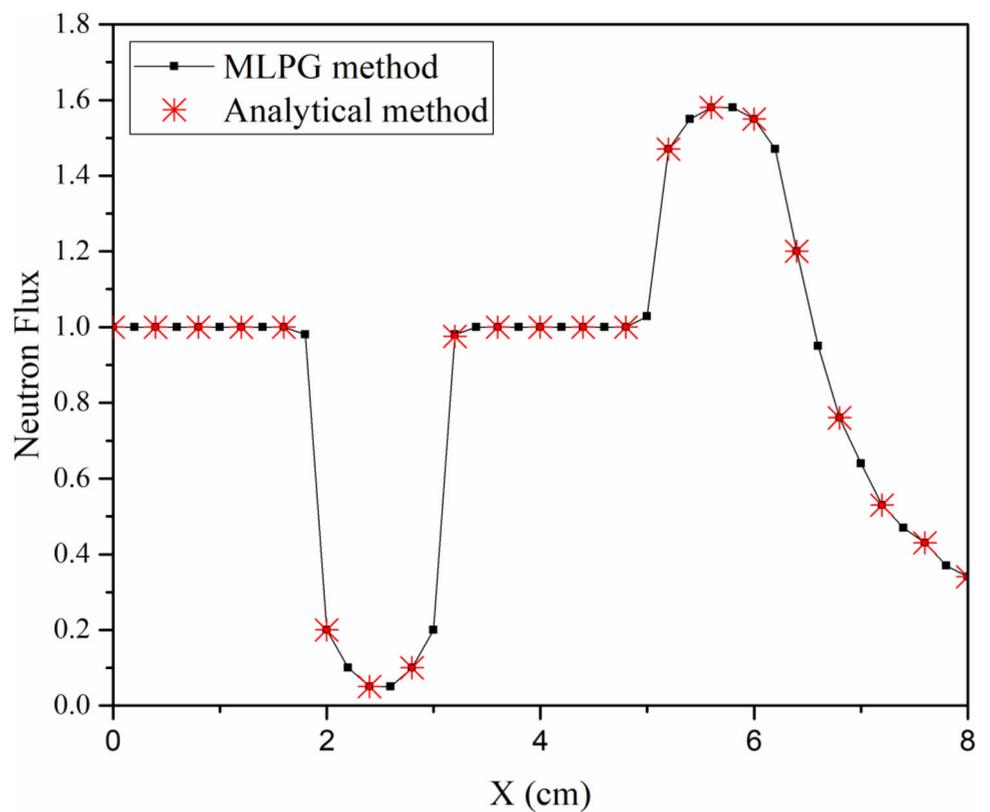


Fig. 21 Comparison of the MLPG method and the analytical solution in the void test problem



3.3 Void test problem

The void test problem was first studied by Abuzaid and Gashut [61]. The problem involves a slab including multiple regions with a perfect reflector on the left side and a bare surface on the right side. Figure 20 presents the geometry and properties of the problem. A comparison between the results of the proposed method and the analytical solution is provided in Fig. 21, which indicates a good agreement.

3.4 Neutronic calculations for a SMR reactor

The proposed method was applied to calculate the neutronic parameters in a small modular reactor (SMR). The core used for the analysis was Westinghouse’s SMR, with an electrical output of 200 MW and an active core height of 2.4 m. The core consists of 89 assemblies contained within a core barrel and reactor vessel. These assemblies include 52 fuel assemblies and 37 control rod assemblies. The SMR core map is presented in Fig. 22. The reactor vessel components were based on an AP1000

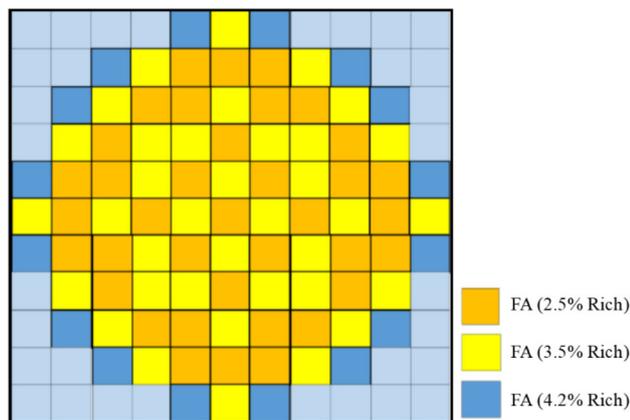


Fig. 22 SMR core map

design but modified to a reduced diameter and height of 3.5 m and 24.7 m, respectively [62].

The fuel assembly was a square lattice in a standard 17×17 layout with 264 fuel rod locations, 24 guide tube locations, and 1 central location for instrumentation, incorporating the standard Westinghouse design specifications. The fuel of the core was uranium oxide (UOX) with less than 5% U-235 enrichment (2.5w/o, 3.5w/o, 4.2%w/o U-235 fuel) and a density of 10.36 g/cm^3 . The fuel rod

consisted of a uniform cylindrical pellet stacked together within a Zircaloy clad tube. Between the fuel stack and the clad, clearance was provided to accommodate fuel swelling due to accumulation of fission products, thereby preventing clad rupture. The gap was filled with helium gas to improve heat conduction from fuel to cladding. The guide tubes in the fuel assembly served as a location for the insertion of the rod cluster control assembly (RCCA). The RCCA is a spider assembly consisting of evenly spaced control rods of either silver indium cadmium or boron carbide rods based on the type of fuel used. Detailed specifications for the fuel rod, cladding, structure, control rod, and burnable poison (i.e., discrete and integral) were taken from CASL (Consortium for Advanced Simulation of LWRs) VERA core physics benchmark specifications [62].

This study reveals the ability of the MLPG method to solve the neutronic diffusion equation and evaluates the performance of this method when different selectable parameters were incorporated into the solution. In order to automate the procedure of the proposed method for this problem, a computer program was developed. The first stage of computational procedures consists of creating the cross section database by WIMSD4 code in three-group energy. Then, some of the main neutronic parameters of the

Table 6 Comparison of the effective multiplication factor between MLPG methods and citation code

Parameter	Citation	Referenced value	MLPG applied weight functions				Nodes insertion
			Gaussian	CSRBF	Cubic spline	Quadratic spline	
Effective multiplication factor	1.12414	1.12511	1.12332	1.11032	1.11054	1.11902	37×37
			1.12403	1.11940	1.11891	1.11974	91×91
			1.12413	1.12439	1.12475	1.11996	289×289

Table 7 Comparison of the maximum-to-average flux ratio between MLPG methods and citation code

Parameter	Citation	Referenced value	MLPG applied weight functions				Nodes insertion
			Gaussian	CSRBF	Cubic spline	Quadratic spline	
Maximum-to-average flux ratio	2.48520	2.4980	2.48213	2.43191	2.44521	2.39120	37×37
			2.48500	2.45225	2.46378	2.44032	91×91
			2.48490	2.48821	2.49707	2.45318	289×289

Table 8 Comparison of the control rod worth between MLPG methods and citation code

Parameter	Citation	Referenced value	MLPG applied weight functions				Nodes insertion
			Gaussian	CSRBF	Cubic spline	Quadratic spline	
Control rod worth	0.15613	0.1571	0.15243	0.14254	0.15500	0.14382	37×37
			0.15629	0.14367	0.15630	0.14630	91×91
			0.15610	0.14975	0.15679	0.14587	289×289

Table 9 Comparison between average errors of results obtained by applying various weight functions (for 289×289 nodes)

Error	MLPG applied weight functions			
	Gaussian	CSRBF	Cubic spline	Quadratic spline
Effective multiplication factor	0.001	0.0064	0.032	0.045
Maximum-to-average flux ratio	0.002	0.041	0.037	0.179
Control rod worth	0.001	0.046	0.020	0.071

Table 10 Comparison between average errors of results and CPU time obtained by applying Gaussian weight function and various node distributions

Parameter	MLPG Gaussian weight function					
	Nodes normal distribution			Nodes scatter distribution		
	Error	CPU time (s)	Nodes	Error	CPU time (s)	Nodes
Effective multiplication factor	0.012	4.0	37×37	0.010	3.5	37×37
	0.087	5.5	91×91	0.063	4.0	91×91
	0.001	10	289×289	0.001	7.5	289×289

core were calculated by the proposed method. The results of the MLPG method were compared with the results of the citation code and the reference values. In addition, as the

selection of the weight function and the number of nodes in the problem domain play an important role in solution accuracy, the performance of the MLPG method was examined with respect to these parameters. Table 6 demonstrates the effective multiplication factor as a main parameter of the core calculation. Tables 7 and 8 present the maximum-to-average flux ratio and the control rod worth, respectively. The choice of weight function is clearly a significant factor for obtaining an accurate solution. The most important frequently used weight functions are the cubic spline, quadratic spline, CSRBF, and Gaussian. Table 9 compares the average errors of results obtained from the various weight functions and reveals that the Gaussian function produces better accuracy than other functions.

In addition, the performance of the MLPG method with respect to the number of nodes was evaluated for the above parameters. The values obtained for these parameters are promisingly close to the citation code results and reference values for 289×289 nodes. Moreover, a comparison between the average errors and the CPU time by applying Gaussian weight functions and various node distributions to derive an effective multiplication factor is given in Table 10. The errors decrease with increasing node number. In addition, the scatter distribution results are more accurate than those of the normal distribution.

The fast flux and the thermal flux of the reactor core are demonstrated in Fig. 23 for the MLPG method with various node numbers. In addition, to evaluate the performance of the MLPG method with respect to the number of nodes, 37×37 , 91×91 , and 289×289 nodes in the radial direction and 21 layers in the axial direction were employed in the problem domain for any of the fuel assembly types.

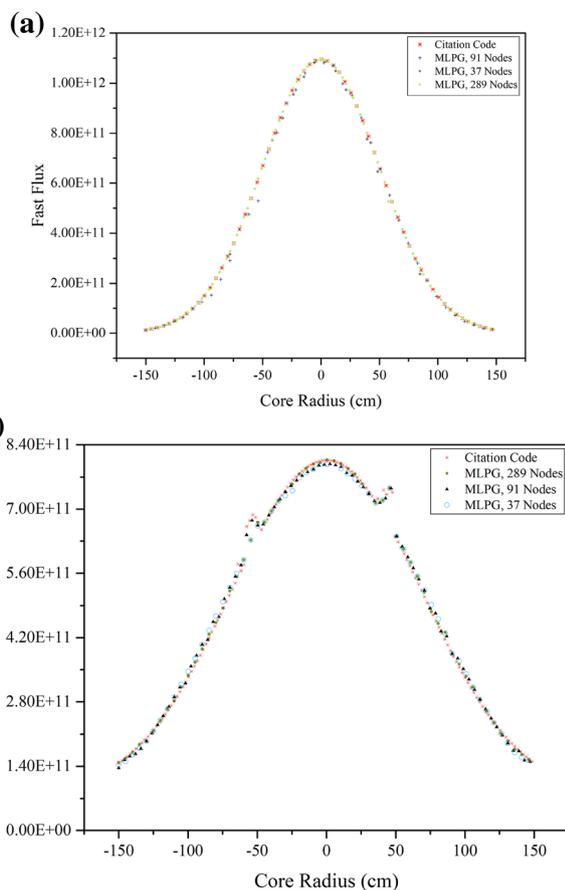


Fig. 23 Comparison of **a** the fast flux and **b** the thermal flux of the applied MLPG method for 37×37 , 91×91 , and 289×289 nodes with the citation code

Fig. 24 Power factor distributions in a 1/8 symmetry of the SMR core

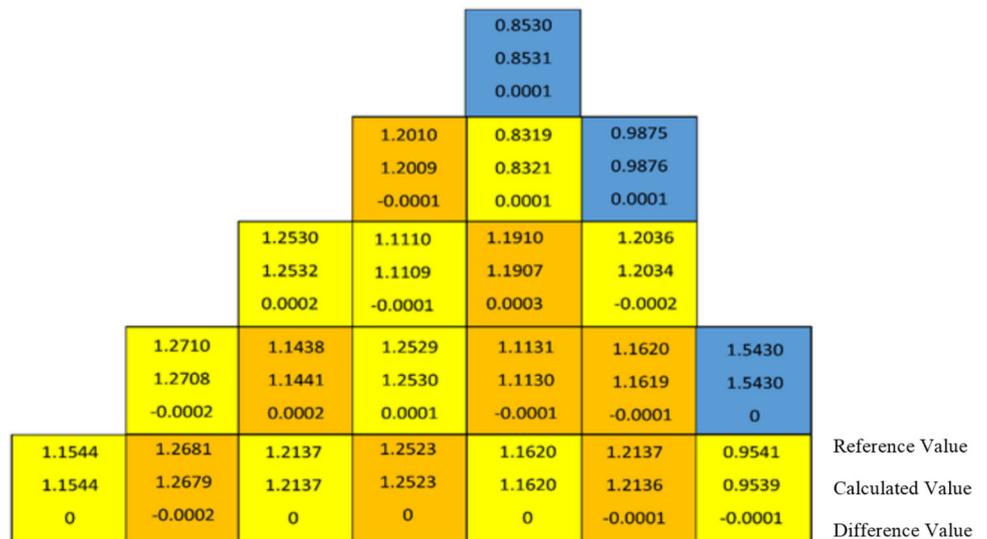


Figure 24 shows the power factor distributions for a 1/8 symmetry of this core. In this figure, the reference values are compared with the values of the proposed method (MLPG 289 × 289 nodes). The reference value exhibits good agreement with the calculated value. The differences between these parameters are also demonstrated in Fig. 24. The comparison of these methods with the citation code and reference values indicates that increasing the number of nodes leads to a more accurate performance of the MLPG method.

4 Conclusion

In this work, the meshless local Petrov–Galerkin method and moving least squares approximation were coupled to solve the neutron diffusion equation. Several problems were examined to demonstrate the degree of accuracy that can be obtained by employing this method as well as the applicability of the proposed method with respect to effective factors. The results of the proposed method were compared with the analytical solution and the GFEM, indicating good agreement. Additionally, the proposed method was used to solve neutronic calculations in the SMR reactor. The results were compared with those obtained from the citation code and reference values.

The accuracy and precision of the proposed method were both acceptable and further improved by increasing the number of nodes and through selection of an appropriate weight function. It should be noted that the runtime varied from 0.5 to 10 s, depending mainly on the weight function type, the number of nodes per cell domain, the number of nodes in the boundary, the pattern of node distribution, the type of boundary conditions applied, and the particular computer system on which the code is run.

As mentioned above, the numerical proposed method removes the mesh generation process by inserting nodes into the problem domain. Therefore, this study eliminated the mesh generation time plus the interactive and CPU time and decreased the time costs.

In summary, using the MLPG method to solve the neutron diffusion equations is promising due to improved computational accuracy and execution time. Accordingly, the outstanding performance of the MLPG method has been confirmed for nuclear engineering calculations, indicating that it can be successfully applied to the desired nuclear computational codes.

References

1. M. Steinhauser, *Computational Multi Scale Modeling of Fluids and Solids* (Springer, Berlin, 2017)
2. G.R. Liu, *Mesh Free Methods: Moving Beyond the Finite Element Method* (Taylor & Francis, Boca Raton, 2009)
3. T. Belytschko, T. Rabczuk, A. Huerta et al., Mesh free methods, in *Encyclopedia of Computational Mechanics*, ed. by E. Stein, R. de Borst, J.R. Hughes (Wiley, Chichester, 2004), pp. 1–48
4. S. Li, W.K. Liu, Mesh free and particle methods and their applications. *Appl. Mech. Rev.* **55**, 1–34 (2002). <https://doi.org/10.1115/1.1431547>
5. Y.T. Gu, Mesh free methods and their comparisons. *Int. J. Comput. Methods* **2**, 477–515 (2005). <https://doi.org/10.1142/S0219876205000673>
6. A. Tayebi, Y. Shekari, M.H. Heydari, A meshless method for solving two-dimensional variable-order time fractional advection–diffusion equation. *J. Comput. Phys.* **340**, 655–669 (2017). <https://doi.org/10.1016/j.jcp.2017.03.061>
7. Q.H. Li, S.S. Chen, G.X. Kou, Transient heat conduction analysis using the MLPG method and modified precise time step integration method. *J. Comput. Phys.* **230**, 2736–2750 (2011). <https://doi.org/10.1016/j.jcp.2011.01.019>

8. X.H. Wu, W.Q. Tao, S.P. Shen et al., A stabilized MLPG method for steady state incompressible fluid flow simulation. *J. Comput. Phys.* **229**, 8564–8577 (2010). <https://doi.org/10.1016/j.jcp.2010.08.001>
9. R. Salehi, M. Dehghan, A moving least square reproducing polynomial meshless method. *Appl. Numer. Math.* **69**, 34–58 (2013). <https://doi.org/10.1016/j.apnum.2013.03.001>
10. R.A. Gingold, J.J. Monaghan, Smoothed particle hydrodynamics: theory and application to non-spherical stars. *Mon. Not. R. Astron. Soc.* **181**, 375–389 (1977). <https://doi.org/10.1093/mnras/181.3.375>
11. P.W. Cleary, Modeling confined multi-material heat and mass flows using SPH. *Appl. Math. Model.* **22**, 981–993 (1998). [https://doi.org/10.1016/S0307-904X\(98\)10031-8](https://doi.org/10.1016/S0307-904X(98)10031-8)
12. G.R. Liu, K.Y. Dai, K.M. Lim, A point interpolation mesh free method for static and frequency analysis of two-dimensional piezoelectric structures. *Comput. Mech.* **29**, 510–519 (2002). <https://doi.org/10.1007/s00466-002-0360-9>
13. B. Nayroles, G. Touzot, P. Villon, Generalizing the finite element method: diffuse approximation and diffuse elements. *Comput. Mech.* **10**, 307–318 (1992). <https://doi.org/10.1007/BF00364252>
14. W.K. Liu, S. Li, T. Belytschko, Moving least-square reproducing kernel methods. Part I: methodology and convergence. *Comput. Method Appl. Mech.* **143**, 113–154 (1977). [https://doi.org/10.1016/S0045-7825\(96\)01132-2](https://doi.org/10.1016/S0045-7825(96)01132-2)
15. W.K. Liu, J. Adee, S. Jun et al., Reproducing kernel particle methods for elastic and plastic problems, in *Advanced Computational Methods for Material Modeling*, ed. by D.A. Siginer, W.E. VanArsdale, C.M. Altan, A.N. Alexandrou (ASME, New Orleans, 1993), pp. 175–189
16. W.K. Liu, S. Jun, Y.F. Zhang, Reproducing kernel particle methods. *Numer. Method Fluids* **20**, 1081–1106 (1995). <https://doi.org/10.1002/flid.1650200824>
17. E. Oñate, S. Idelsohn, O.C. Zienkiewicz et al., A finite point method in computational mechanics—applications to convective transport and fluid flow. *Int. J. Numer. Methods Eng.* **39**, 3839–3866 (1996). [https://doi.org/10.1002/\(SICI\)1097-0207\(19961130\)39:22<3839::AID-NME27>3.0.CO;2-R](https://doi.org/10.1002/(SICI)1097-0207(19961130)39:22<3839::AID-NME27>3.0.CO;2-R)
18. Y.T. Gu, G.R. Liu, Using radial function basis in a boundary-type meshless method, boundary point method, in *International Conference on Science & Engineering Computation* (Beijing, 2001)
19. M.D. Buhmann, *Radial Basis Functions: Theory and Implementations* (Cambridge University, Cambridge, 2003)
20. H. Mehrabi, B. Voosoghi, On estimating the curvature attributes and strain invariants of deformed surface through radial basis functions. *Comp. Appl. Math.* (2016). <https://doi.org/10.1007/s40314-016-0380-2>
21. Q. Li, J. Soric, T. Jarak et al., A locking-free meshless local Petrov–Galerkin formulation for thick and thin plates. *J. Comput. Phys.* **208**, 116–133 (2005). <https://doi.org/10.1016/j.jcp.2005.02.008>
22. P. Mycek, G. Pinon, G. Germain, Formulation and analysis of a diffusion-velocity particle model for transport-dispersion equations. *Comp. Appl. Math.* **35**, 447–473 (2014). <https://doi.org/10.1007/s40314-014-0200-5>
23. S. Abbasbandy, A. Shirzadi, MLPG method for two-dimensional diffusion equation with Neumann’s and non-classical boundary conditions. *Appl. Numer. Math.* **61**, 170–180 (2011). <https://doi.org/10.1016/j.apnum.2010.09.002>
24. B. Rokrok, H. Minuchehr, A. Zolfaghari, Application of radial point interpolation method to neutron diffusion field. *Trend Appl. Sci. Res.* **7**, 18–31 (2012). <https://doi.org/10.3923/tasr.2012.18.31>
25. T. Tanbay, B. Ozgener, A comparison of the meshless RBF collocation method with finite element and boundary element methods in neutron diffusion calculations. *Eng. Anal. Bound. Elem.* **46**, 30–40 (2014). <https://doi.org/10.1016/j.enganabound.2014.05.005>
26. J.R. Xiao, R.C. Batra, D.F. Gilhooley et al., Analysis of thick plates by using a higher-order shear and normal deformable plate theory and MLPG method with radial basis functions. *Comput. Methods Appl. Mech. Eng.* **196**, 979–987 (2007). <https://doi.org/10.1016/j.cma.2006.08.002>
27. S. De, K.J. Bathe, The method of finite spheres. *Comput. Mech.* **25**, 329–345 (2000). <https://doi.org/10.1007/s004660050481>
28. M. Moradipour, S. Yousefi, Using a meshless kernel-based method to solve the Black-Scholes variational inequality of American options. *Comp. Appl. Math.* (2016). <https://doi.org/10.1007/s40314-016-0351-7>
29. M. Ebrahimnejad, N. Fallah, A.R. Khoei, Three types of meshless finite volume method for the analysis of two-dimensional elasticity problems. *Comp. Appl. Math.* **36**, 971 (2015). <https://doi.org/10.1007/s40314-015-0273-9>
30. S. Atluri, T. Zhu, New concepts in meshless methods. *Comput. Mech. Int. J. Numer. Methods Eng.* **47**, 117–127 (2000). [https://doi.org/10.1002/\(SICI\)1097-0207\(200011030\)47:1/3<537::AID-NME783>3.0.CO;2-E](https://doi.org/10.1002/(SICI)1097-0207(200011030)47:1/3<537::AID-NME783>3.0.CO;2-E)
31. M. Dehghan, M. Abbaszadeh, Numerical investigation based on direct meshless local Petrov–Galerkin (direct MLPG) method for solving generalized Zakharov system in one and two dimensions and generalized Gross-Pitaevskii equation. *Eng. Comput.* **33**, 983–996 (2017). <https://doi.org/10.1007/s00366-017-0510-5>
32. A. Taleei, M. Dehghan, Direct meshless local Petrov–Galerkin method for elliptic interface problems with applications in electrostatic and elastostatic. *Comput. Methods Appl. Mech. Eng.* **278**, 479–498 (2014). <https://doi.org/10.1016/j.cma.2014.05.016>
33. M. Dehghan, R. Salehi, A meshless local Petrov–Galerkin method for the time-dependent Maxwell equations. *J. Comput. Appl. Math.* **268**, 93–110 (2014). <https://doi.org/10.1016/j.cam.2014.02.013>
34. M. Kamranian, M. Dehghan, M. Tatari, An adaptive meshless local Petrov–Galerkin method based on a posteriori error estimation for the boundary layer problems. *Appl. Numer. Math.* **111**, 181–196 (2017). <https://doi.org/10.1016/j.apnum.2016.09.007>
35. M. Dehghan, M. Abbaszadeh, A. Mohebbi, Meshless local Petrov–Galerkin and RBFs collocation methods for solving 2D fractional Klein-Kramers dynamics equation on irregular domains. *Comput. Model. Eng. Sci.* **107**, 481–516 (2015)
36. W.H. Chen, X.M. Guo, Element free Galerkin method for three-dimensional structural analysis. *Comput. Model. Eng. Sci.* **4**, 497–508 (2001). <https://doi.org/10.3970/cmescs.2001.002.497>
37. Q.W. Ma, Meshless local Petrov–Galerkin method for two-dimensional nonlinear water wave problems. *J. Comput. Phys.* **205**, 611–625 (2005). <https://doi.org/10.1016/j.jcp.2004.11.010>
38. J. Amani Rad, K. Parand, Numerical pricing of American options under two stochastic factor models with jumps using a meshless local Petrov–Galerkin method. *Appl. Numer. Math.* **115**, 252–274 (2017). <https://doi.org/10.1016/j.apnum.2017.01.015>
39. H. Lin, S.N. Atluri, The Meshless Local Petrov–Galerkin (MLPG) method for solving incompressible Navier–Stokes equations. *Comput. Model. Eng. Sci.* **2**, 117–142 (2001). <https://doi.org/10.3970/cmescs.2001.002.117>
40. S.Y. Long, S.N. Atluri, A meshless local Petrov–Galerkin (MLPG) method for solving the bending problems of a thin plate. *CMES* **3**, 53–64 (2002). <https://doi.org/10.3970/cmescs.2002.003.053>
41. T. Belytschko, Y.Y. Lu, L. Gu, Element-free Galerkin method. *Int. J. Numer. Methods Eng.* **37**, 229–256 (1994). <https://doi.org/10.1002/nme.1620370205>
42. G.R. Liu, Y.T. Gu, Meshless local Petrov–Galerkin (MLPG) method in combination with finite element and boundary element

- approaches. *Comput. Mech.* **26**, 536–546 (2000). <https://doi.org/10.1007/s004660000203>
43. M. Li, F.F. Dou, T. Korakianitis et al., Boundary node Petrov–Galerkin method in solid structures. *Comp. Appl. Math.* (2016). <https://doi.org/10.1007/s40314-016-0335-7>
 44. I. Debbabi, H. BelhadjSalah, Analysis of thermo-elastic problems using the improved element-free Galerkin method. *Comp. Appl. Math.* (2016). <https://doi.org/10.1007/s40314-016-0401-1>
 45. D. Mirzaei, R. Schaback, Direct meshless local Petrov–Galerkin (DMLPG) method: a generalized MLS approximation. *Appl. Numer. Math.* **68**, 73–82 (2013). <https://doi.org/10.1016/j.apnum.2013.01.002>
 46. M. Dehghan, D. Mirzaei, Meshless local Petrov–Galerkin (MLPG) method for the unsteady magnetohydrodynamic (MHD) flow through pipe with arbitrary wall conductivity. *Appl. Numer. Math.* **59**, 1043–1058 (2009). <https://doi.org/10.1016/j.apnum.2008.05.001>
 47. S. Atluri, S. Shen, The basis of meshless domain discretization: the meshless local Petrov–Galerkin (MLPG) method. *Adv. Comput. Math.* **23**, 73–93 (2005). <https://doi.org/10.1007/s10444-004-1813-9>
 48. G.R. Liu, Y.T. Gu, *An introduction to mesh free methods and their programming*, 1st edn. (Springer, Dordrecht, 2005)
 49. Y.T. Gu, G.R. Liu, A meshless local Petrov–Galerkin (MLPG) method for free and forced vibration analyses for solids. *Comput. Mech.* **27**, 188–198 (2001). <https://doi.org/10.1007/s004660100237>
 50. P. Lancaster, K. Salkauskas, Surfaces generation by moving least squares methods. *Math. Comput.* **37**, 141–158 (1981). <https://doi.org/10.1090/S0025-5718-1981-0616367-1>
 51. V.R. Hosseini, E. Shivanian, W. Chen, Local radial point interpolation (MLRPI) method for solving time fractional diffusion-wave equation with damping. *J. Comput. Phys.* **312**, 307–332 (2016). <https://doi.org/10.1016/j.jcp.2016.02.030>
 52. Y.T. Gu, G.R. Liu, A local point interpolation method for static and dynamic analysis of thin beams. *Comput. Methods Appl. Mech. Eng.* **190**, 5515–5528 (2001). [https://doi.org/10.1016/S0045-7825\(01\)00180-3](https://doi.org/10.1016/S0045-7825(01)00180-3)
 53. G. Liu, *Mesh Free Methods: Moving Beyond the Finite Element Method* (CRC Press, Boca Raton, 2003)
 54. M. Dehghan, D. Mirzaei, Numerical solution to the unsteady two-dimensional Schrödinger equation using meshless local boundary integral equation method. *Int. J. Numer. Methods Eng.* **76**, 501–520 (2008). <https://doi.org/10.1002/nme.2338>
 55. J. Zheng, S. Long, G. Li, Topology optimization of free vibrating continuum structures based on the element free Galerkin method. *Struct. Multidiscip. Optim.* **45**, 119–127 (2012). <https://doi.org/10.1007/s00158-011-0667-2>
 56. M. Sterk, B. Robic, R. Trobec, Mesh free method applied to the diffusion equation. *Parallel Numer.* **5**, 57–66 (2005). <https://doi.org/10.1007/s00158-011-0667-2>
 57. T. Tanbay, B. Ozgener, Numerical solution of the multigroup neutron diffusion equation by the meshless RBF collocation method. *Math. Comput. Appl.* **18**, 399–407 (2013). <https://doi.org/10.3390/mca18030399>
 58. D. Mirzaei, M. Dehghan, Meshless local Petrov–Galerkin (MLPG) approximation to the two dimensional sine-Gordon equation. *J. Comput. Appl. Math.* **233**, 2737–2754 (2010). <https://doi.org/10.1016/j.cam.2009.11.022>
 59. D. Jeong, J. Kim, A Crank–Nicolson scheme for the Landau–Lifshitz equation without damping. *J. Comput. Appl. Math.* **234**, 613–623 (2010). <https://doi.org/10.1016/j.cam.2010.01.002>
 60. O. Abbasi, A. Rostami, G. Karimian, Identification of exonic regions in DNA sequences using cross-correlation and noise suppression by discrete wavelet transform. *BMC Bioinformatics* **12**, 430 (2011). <https://doi.org/10.1186/1471-2105-12-430>
 61. O.A. Abuzaid, A.H.M. Gashut, Discontinuous finite element methods for reactor calculations, in *Third Arab Conf. Peac. Uses At. Energy* (Damascus, 1996), pp. 39–46
 62. IAEA, Status of small and medium sized reactors (2011)