

The neutron transport equation in exact differential form

Dong Liu^{1,2,4*}, Yang Liu², Haoning Dang³, Kai Wang^{4*}, Bin Zhang¹, Fei Wang^{3*},
Zhouyu Liu⁵, and Yong Jiang¹

¹Nuclear Power Institute of China, State Key Laboratory of Advanced Nuclear Energy Technology, Chengdu 610213, China

²College of Computer Science (School of Software and School of Artificial Intelligence), Sichuan University, Chengdu 610064, China

³School of Mathematics and Statistics, Xi'an Jiaotong University, Xi'an 710049, China

⁴Zhejiang Institute of Modern Physics, School of Physics, Zhejiang University, Hangzhou 310058, China

⁵School of Energy and Power Engineering, Xi'an Jiaotong University, Xi'an 710049, China

Received October 27, 2024; accepted March 11, 2025; published online April 15, 2025

Derived from the Boltzmann equation, the neutron transport equation describes the motions and interactions of neutrons with nuclei in nuclear devices such as nuclear reactors. The collision or fission effect are described as integral terms which arrive in an integro-differential neutron transport equation (IDNT). Only for mono-material or simple geometries conditions, elegant approximation can simplify the transport equation to provide analytic solutions. To solve this integro-differential equation becomes a practical engineering challenge. Recent development of deep-learning techniques provides a new approach to solve them but for some complicated conditions, it is also time consuming. To optimize solving the integro-differential equation particularly under the deep-learning method, we propose to convert the integral terms in the integro-differential neutron transport equation into their corresponding antiderivatives, providing a set of fixed solution constraint conditions for these antiderivatives, thus yielding an exact differential neutron transport equation (EDNT). The paper elucidates the physical meaning of the antiderivatives and analyzes the continuity and computational complexity of the new transport equation form. To illustrate the significant advantage of EDNT, numerical validations have been conducted using various numerical methods on typical benchmark problems. The numerical experiments demonstrate that the EDNT is compatible with various numerical methods, including the finite difference method (FDM), finite volume method (FVM), and PINN. Compared to the IDNT, the EDNT offers significant efficiency advantages, with reductions in computational time ranging from several times to several orders of magnitude. This EDNT approach may also be applicable for other integro-differential transport theories such as radiative energy transport and has potential application in astrophysics or other fields.

neutron transport equation, exact differential form, integro-differential equation, deep learning, antiderivative transformation

PACS number(s): 02.60.Cb, 02.60.Nm, 28.20.Gd, 28.20.-v

Citation: D. Liu, Y. Liu, H. Dang, K. Wang, B. Zhang, F. Wang, Z. Liu, and Y. Jiang, The neutron transport equation in exact differential form, Sci. China-Phys. Mech. Astron. **68**, 270511 (2025), <https://doi.org/10.1007/s11433-024-2642-3>

1 Introduction

The neutron transport equation is the fundamental equation in nuclear reactor core physics, and solving it efficiently and

accurately is essential for reactor engineering design, development, and operational support. The generalized neutron transport equation, which is in integro-differential form, captures the anisotropy of the scattering source and is widely

*Corresponding authors (Dong Liu, email: liudong@uestc.edu.cn; Kai Wang, email: wangkai1@zju.edu.cn; Fei Wang, email: feiwang.xjtu@xjtu.edu.cn)

applicable. The equation is expressed as:

$$\begin{aligned} & \frac{1}{v} \frac{\partial \Psi(\mathbf{r}, \hat{\Omega}, E, t)}{\partial t} + \hat{\Omega} \cdot \nabla \Psi(\mathbf{r}, \hat{\Omega}, E, t) + \Sigma_t(\mathbf{r}, E) \Psi(\mathbf{r}, \hat{\Omega}, E, t) \\ &= \iint \Sigma_s(\mathbf{r}, E') f(\mathbf{r}, \hat{\Omega}' \rightarrow \hat{\Omega}, E' \rightarrow E) \\ & \quad \Psi(\mathbf{r}, \hat{\Omega}', E', t) d\hat{\Omega}' dE' \\ &+ \frac{\chi(E)}{4\pi} \iint v \Sigma_f(\mathbf{r}, E') \Psi(\mathbf{r}, \hat{\Omega}', E', t) d\hat{\Omega}' dE' + Q_e, \end{aligned} \quad (1)$$

where $\Psi(\mathbf{r}, \hat{\Omega}, E, t)$ is the neutron angular flux at time t , position \mathbf{r} , angle $\hat{\Omega}$, and energy E . Note that \mathbf{r} represents the position vector and $\hat{\Omega}$ represents the angular vector, which includes the direction cosine μ and the azimuthal angle φ . Σ_t is the total cross section; v is the neutron velocity; Q_e is the external source; $f(\mathbf{r}, \hat{\Omega}', E' \rightarrow \hat{\Omega}, E)$ is the scattering function; Σ_s is the scattering cross section; Σ_f is the fission cross section; v is the number of neutrons produced per reaction; and $\chi(E)$ is the fission neutron energy spectrum. The scattering source can be anisotropic, while the fission source is normally isotropic.

In steady-state conditions without external sources, the multigroup transport equation [1] can be expressed as eq. (2):

$$\begin{aligned} & \hat{\Omega} \cdot \nabla \Psi_g(\mathbf{r}, \hat{\Omega}) + \Sigma_{t,g}(\mathbf{r}, \hat{\Omega}) \Psi_g(\mathbf{r}, \hat{\Omega}) \\ &= \sum_{g'=1}^G \int \Sigma_{s,g' \rightarrow g}(\mathbf{r}, \hat{\Omega}' \rightarrow \hat{\Omega}) \Psi_{g'}(\mathbf{r}, \hat{\Omega}') d\hat{\Omega}' \\ &+ \frac{\chi_g}{4\pi k_{\text{eff}}} \sum_{g'=1}^G (v \Sigma_f(\mathbf{r}))_{g'} \int \Psi_{g'}(\mathbf{r}, \hat{\Omega}') d\hat{\Omega}', \end{aligned} \quad (2)$$

where k_{eff} denotes the effective multiplication factor, while g represents the energy group index, with $g = 1, 2, \dots, G$. The term $\Sigma_{s,g' \rightarrow g}(\mathbf{r}, \hat{\Omega}' \rightarrow \hat{\Omega})$ refers to the scattering cross-section from energy group g' to g , and from direction $\hat{\Omega}'$ to $\hat{\Omega}$.

Solving the neutron transport equation in its integral form is an extremely complex task. To address this, various methods have been developed. In practical, the neutron transport equation can be expressed in various mathematically equivalent forms, including the integro-differential form, integral form, adjoint form, and even-parity form, among others [1-3]. Meanwhile, The neutron transport equation are usually solved by the traditionally deterministic methods, such as the discrete ordinates method (s_N), the method of characteristics (MOC) [2-5] and the spherical harmonics method (P_n), as well as stochastic methods represented by the Monte Carlo approach [1]. The form of the transport equation is closely tied to the numerical methods employed to solve it. A specific form of the transport equation often corresponds to a matching numerical discretization method. The specific form of the transport equation is also affected by considering

factors such as the geometry, the number of energy groups discretized, the characteristics of the scattering sources, and the presence of external sources.

In the past decade, artificial intelligence, specifically deep learning methods, has become popular for solving partial differential equations due to its advantages, such as well-structured computation, strong continuity properties, and excellent data assimilation capabilities [6-9]. Progress has also been made in using deep learning methods to solve the multi-group complex geometries neutron diffusion equation, a simplified form of the neutron transport equation [9-13]. Exploratory research has also begun on using deep learning methods to solve neutron transport equations with structured geometries and few energy groups [14, 15]. Ref. [14], inspired by the traditional s_N method [2, 3], discretizes the angular variable of the integrand and approximates the integral using Gaussian quadrature [2, 3], resulting in some systematic errors. Ref. [15] proposes a deep learning-based variable-order method that solves simplified transport equations with isotropic scattering and fission sources in few-group structured geometries. However, this method can not deal with anisotropic scattering sources which commonly encountered in practical reactor. Further research for complex geometries and multi-group transport equations is urgently needed.

However, most existing forms of the neutron transport equation contain multiple integral terms for scattering and fission sources over energy and angular variables, which brings significant challenges for applying deep learning methods to solve these equations. The integral terms are traditionally approximated as finite sums as followings:

$$\int_a^b \Psi(x) = \sum_{j=1}^n w_j \Psi(x_j). \quad (3)$$

Current studies show that the traditional forms of the neutron transport equation do not align well with deep learning numerical methods. Firstly, eq. (3) introduces systematic errors due to the finite-sum approximation of the integral terms. Secondly, for deep learning methods, for each machine learning sample point set by eqs. (1) and (2), a finite sum should be computed for all the other sample points, in addition to performing the differentiation operation itself. When the machine learning sample space is large, this characteristic can significantly impact computational efficiency, resulting in very long training time. This represents a technical challenge for deep learning numerical methods used to solve the neutron transport equations.

Therefore, this paper proposes an exact differential form of the neutron transport equation. The technical approach is to convert the integral terms in the integro-differential neutron

transport equation into their corresponding antiderivatives, thereby converting the neutron transport equation into an exact differential form. The physical meaning of the antiderivatives is provided, and the continuity and computational complexity of the new transport equation form are analyzed. Finally, numerical validations are performed for typical geometries using finite difference, finite volume, and deep learning methods to solve the EDNT equation. The results are compared with those obtained by solving the original form of the equation, and the performance of the exact differential form is evaluated.

Given that the integro-differential form (1) and (2) of the transport equation is concise and comprehensive, including isotropic fission sources and anisotropic scattering sources, this paper uses it as the basis for deriving the exact differential neutron transport equation. The transformation method can also be extended to other forms of transport equations.

2 Derivation of the exact differential form of the neutron transport equation

The fundamental principle in constructing the exact differential form of the neutron transport equation is to do antiderivative transformations on all integral terms in the neutron transport equation. This involves converting the integrands into their corresponding antiderivatives. If the transport equation has a multiple integral form, the multiple integral terms will be transformed to the corresponding antiderivatives by means of higher dimensional or cumulative transformations. Consequently, the integro-differential form of neutron transport equation can be transformed into an exact differential form.

2.1 Derivation of the continuous energy neutron transport equation

According to the principles of calculus [16], a continuous integrand has a specific antiderivative with respect to the angular variable $\hat{\Omega}'$ and energy E' . Referring to the two-dimensional Newton-Leibniz formula [16], we perform an antiderivative transformation on the scattering source terms in the neutron transport equation (1), as detailed in Appendix A1.1. The resulting expression is

$$\begin{aligned} Q_s &= \int_{E_0}^{E_1} \int_{\hat{\Omega}_0}^{\hat{\Omega}_1} \Sigma_s(\mathbf{r}, E') f(\mathbf{r}, \hat{\Omega}', E' \rightarrow \hat{\Omega}, E) \\ &\quad \Psi(\mathbf{r}, \hat{\Omega}', E', t) d\hat{\Omega}' dE' \\ &= F_{c,s}(\mathbf{r}, \hat{\Omega}, E, \hat{\Omega}_1, E_1, t). \end{aligned} \quad (4)$$

Similarly, for the fission source, as detailed in Appendix A1.2, we have

$$\begin{aligned} Q_f &= \frac{\chi(E)}{4\pi} \int_{E_0}^{E_1} \int_{\hat{\Omega}_0}^{\hat{\Omega}_1} v \Sigma_f(\mathbf{r}, E') \Psi(\mathbf{r}, \hat{\Omega}', E', t) d\hat{\Omega}' dE' \\ &= \frac{\chi(E)}{4\pi} F_{c,f}(\mathbf{r}, \hat{\Omega}_1, E_1, t), \end{aligned} \quad (5)$$

where $\hat{\Omega}_1, \hat{\Omega}_0$ are the upper and lower limits of the angular integral, and E_1, E_0 are the upper and lower limits of the energy integral. The specific meanings of $F_{c,s}$ and $F_{c,f}$ can be found in Appendix A1.

Substituting eqs. (4) and (5) into eq. (1), we obtain the general exact differential form of the neutron transport equations:

$$\left\{ \begin{array}{l} \frac{1}{v} \frac{\partial \Psi(\mathbf{r}, \hat{\Omega}, E, t)}{\partial t} + \hat{\Omega} \cdot \nabla \Psi(\mathbf{r}, \hat{\Omega}, E, t) \\ + \Sigma_t(\mathbf{r}, E) \Psi(\mathbf{r}, \hat{\Omega}, E, t) \\ = F_{c,s}(\mathbf{r}, \hat{\Omega}, E, \hat{\Omega}_1, E_1, t) + \frac{\chi(E)}{4\pi} \\ F_{c,f}(\mathbf{r}, \hat{\Omega}_1, E_1, t) + Q_e, \\ \frac{\partial^2 F_{c,s}(\mathbf{r}, \hat{\Omega}, E, \hat{\Omega}', E', t)}{\partial \hat{\Omega}' \partial E'} = \Sigma_s(\mathbf{r}, E') \\ f(\mathbf{r}, \hat{\Omega}', E' \rightarrow \hat{\Omega}, E) \Psi(\mathbf{r}, \hat{\Omega}', E', t), \\ \frac{\partial^2 F_{c,f}(\mathbf{r}, \hat{\Omega}', E', t)}{\partial \hat{\Omega}' \partial E'} = v \Sigma_f(\mathbf{r}, E') \Psi(\mathbf{r}, \hat{\Omega}', E', t), \\ (\hat{\Omega}' = \hat{\Omega}_0) \cup (E' = E_0), F_{c,s}(\mathbf{r}, \hat{\Omega}, E, \hat{\Omega}', E', t) = 0, \\ F_{c,f}(\mathbf{r}, \hat{\Omega}', E', t) = 0. \end{array} \right. \quad (6)$$

Thus, eq. (6) represents the general exact differential form of the neutron transport equation. It is a system of equations consisting of three equations, three unknown functions, and two boundary conditions. It should be noted that if the angular is two-dimensional variable, it is generally represented by the direction cosines μ and the azimuthal angle φ . In this case, the scattering/fission source terms in eq. (1) involve triple integrals, which can be treated using higher-dimensional Newton-Leibniz formulas or cumulative transformations.

2.2 Derivation of the multigroup neutron transport equation

2.2.1 Derivation of the multigroup anisotropic scattering equation

To simplify calculations, the anisotropic scattering source in eq. (2) is typically expanded using Legendre polynomials. The scattering source in the multigroup formulation [1] can

be expressed as:

$$\begin{aligned} Q_{s,g} &= \sum_{g'=1}^G \int \Sigma_{s,g' \rightarrow g}(\mathbf{r}, \hat{\Omega}' \rightarrow \hat{\Omega}) \Psi_{g'}(\mathbf{r}, \hat{\Omega}') d\hat{\Omega}' \\ &= \sum_{g'=1}^G \sum_{l=0}^{\infty} \Sigma_{s,l,g' \rightarrow g}(\mathbf{r}) \sum_{m=-l}^l Y_l^m(\hat{\Omega}) \Psi_{g'}^{l,m}, \end{aligned} \quad (7)$$

where $\Psi_{g'}^{l,m}$ represents the neutron flux moment, and $Y_l^m(\hat{\Omega})$ is the spherical harmonic function.

For the multigroup neutron transport equation (2), we apply the antiderivative transformation to the neutron flux moment in the scattering term and the fission source term to derive:

$$\left\{ \begin{array}{l} \hat{\Omega} \cdot \nabla \Psi_g(\mathbf{r}, \hat{\Omega}) + \Sigma_{t,g}(\mathbf{r}, \hat{\Omega}) \Psi_g(\mathbf{r}, \hat{\Omega}) \\ = \sum_{g'=1}^G \sum_{l=0}^{\infty} \Sigma_{s,l,g' \rightarrow g}(\mathbf{r}) \sum_{m=-l}^l Y_l^m(\hat{\Omega}) F_{c,s,g'}^{l,m}(\mathbf{r}, \hat{\Omega}_1) \\ + \frac{\chi_g}{4\pi k_{\text{eff}}} \sum_{g'=1}^G (\nu \Sigma_f(\mathbf{r}))_{g'} F_{c,f,g'}(\mathbf{r}, \hat{\Omega}_1), \\ \frac{dF_{c,s,g'}^{l,m}(\mathbf{r}, \hat{\Omega}')}{d\hat{\Omega}'} = Y_l^m(\hat{\Omega}') \Psi_{g'}(\mathbf{r}, \hat{\Omega}'), \\ \frac{dF_{c,f,g'}(\mathbf{r}, \hat{\Omega}')}{d\hat{\Omega}'} = \Psi_{g'}(\mathbf{r}, \hat{\Omega}'), \\ \hat{\Omega}' = \hat{\Omega}_0, F_{c,s,g'}(\mathbf{r}, \hat{\Omega}_0, \hat{\Omega}') = 0, F_{c,f,g'}(\mathbf{r}, \hat{\Omega}') = 0. \end{array} \right. \quad (8)$$

The specific meanings of $F_{c,s,g'}^{l,m}$ and $F_{c,f,g'}$, and further details of derivation process can be found in Appendix A2.1.

2.2.2 Derivation of the multigroup isotropic scattering neutron transport equation

In practical engineering, for cases where neutron anisotropy is not significant (e.g., neutron flux calculation in the core of a large pressure water reactor), the scattering source is often assumed to be isotropic to simplify the calculation of the neutron transport equation. Alternatively, transport correction methods are applied to modify the cross sections, allowing the isotropic scattering source form of the equation to approximate the precision of the anisotropic neutron transport equation. Therefore, it is necessary to derive the multigroup isotropic scattering neutron transport equation. With isotropic scattering cross sections, the scattering source is independent of the angular variable and can be written as $\Sigma_{s,g' \rightarrow g}(\mathbf{r})$, reducing eq. (2) to:

$$\begin{aligned} \hat{\Omega} \cdot \nabla \Psi_g(\mathbf{r}, \hat{\Omega}) + \Sigma_{t,g}(\mathbf{r}, \hat{\Omega}) \Psi_g(\mathbf{r}, \hat{\Omega}) \\ = \sum_{g'=1}^G \left(\Sigma_{s,g' \rightarrow g}(\mathbf{r}) + \frac{\chi_g}{4\pi k_{\text{eff}}} (\nu \Sigma_f(\mathbf{r}))_{g'} \right) \int \Psi_{g'}(\mathbf{r}, \hat{\Omega}') d\hat{\Omega}'. \end{aligned} \quad (9)$$

By transforming the integral term $\int \Psi_{g'}(\mathbf{r}, \hat{\Omega}') d\hat{\Omega}'$, as detailed in Appendix A2.2, we can obtain

$$\left\{ \begin{array}{l} \hat{\Omega} \cdot \nabla F_{c,g}(\mathbf{r}, \hat{\Omega})' + \Sigma_{t,g}(\mathbf{r}, \hat{\Omega}) F_{c,g}(\mathbf{r}, \hat{\Omega})' \\ = \sum_{g'=1}^G \left(\Sigma_{s,g' \rightarrow g}(\mathbf{r}) + \frac{\chi_g}{4\pi k_{\text{eff}}} (\nu \Sigma_f(\mathbf{r}))_{g'} F_{c,g'}(\mathbf{r}, \hat{\Omega}_1) \right), \\ \hat{\Omega}' = \hat{\Omega}_0, F_{c,g'}(\mathbf{r}, \hat{\Omega}_0) = 0, g' = 1, 2, 3, \dots. \end{array} \right. \quad (10)$$

The specific meanings of $F_{c,g}$ can be found in Appendix A2.2. Comparing with eqs. (6) and (8), the form of eq. (10) is significantly simplified.

3 Analysis and evaluation of the exact differential form of the neutron transport equation

3.1 Physical meaning of the antiderivatives

The antiderivatives in the exact differential form equations (6), (8), and (10), each have practical physical significance, which is somewhat different according to the corresponding integrand of the antiderivative. In eq. (6), $F_{c,s}(\mathbf{r}, \hat{\Omega}, E, \hat{\Omega}', E', t)$ represents the total contribution of scattered neutrons arriving at angle $\hat{\Omega}$ and energy E at location \mathbf{r} and time t , where the neutrons have been scattered from angles $\hat{\Omega}_0$ to $\hat{\Omega}'$ and energy E_0 to E' . Similarly, $F_{c,s}(\mathbf{r}, \hat{\Omega}, E, \hat{\Omega}_1, E_1, t)$ represents the total scattering neutrons arriving at angle $\hat{\Omega}$ and energy E . $F_{c,f}(\mathbf{r}, \hat{\Omega}', E', t)$ represents the total number of fission neutrons produced at location \mathbf{r} at time t from angles $\hat{\Omega}_0$ to $\hat{\Omega}'$ and energy E_0 to E' , and $F_{c,f}(\mathbf{r}, \hat{\Omega}_1, E_1, t)$ gives the total number of fission neutrons produced at location \mathbf{r} at time t . Moreover, for $\hat{\Omega}_0 < \hat{\Omega}_m < \hat{\Omega}_n < \hat{\Omega}_1$ and $E_0 < E_m < E_n < E_1$, the quantity $F_{c,s}(\mathbf{r}, \hat{\Omega}, E, \hat{\Omega}_n, E_n, t) - F_{c,s}(\mathbf{r}, \hat{\Omega}, E, \hat{\Omega}_m, E_m, t)$ represents the total scattering neutrons within the energy range E_m-E_n and angular range $\hat{\Omega}_m-\hat{\Omega}_n$ that arrive at angle $\hat{\Omega}$ and energy E at location \mathbf{r} . Similarly, $F_{c,f}(\mathbf{r}, \hat{\Omega}_n, E_n, t) - F_{c,f}(\mathbf{r}, \hat{\Omega}_m, E_m, t)$ represents the total number of fission neutrons produced within the energy range E_m-E_n and angular range $\hat{\Omega}_m-\hat{\Omega}_n$.

For the multigroup equation (8), $F_{c,s,g'}^{l,m}(\mathbf{r}, \hat{\Omega})$ is the integral of the angular flux moment of group g' at location \mathbf{r} , over angles from $\hat{\Omega}_0$ to $\hat{\Omega}$. Similarly, $F_{c,s,g'}^{l,m}(\mathbf{r}, \hat{\Omega}_1)$ represents the total angular flux moment for that group. $F_{c,f,g'}(\mathbf{r}, \hat{\Omega})$ represents the integral of the angular flux for group g' , and $F_{c,f,g'}(\mathbf{r}, \hat{\Omega}_1)$ represents the scalar flux for that group at \mathbf{r} .

Likewise, in eq. (10), $F_{c,g'}(\mathbf{r}, \hat{\Omega})$ is the integral of the angular flux for group g' over angles from $\hat{\Omega}_0$ to $\hat{\Omega}$, and $F_{c,g'}(\mathbf{r}, \hat{\Omega}_1)$ represents the scalar flux at \mathbf{r} for group g' .

3.2 Continuity analysis of the exact differential form of the neutron transport equation

The antiderivative transformation performed in sect. 2 is based on the assumption that the integrand is continuous. However, in practical applications, the cross sections of core materials are not always continuous, which poses challenges when solving the exact differential form of the neutron transport equation.

The illustration of discontinuous cross sections in reactor core materials is shown in Figure 1. The cross sections of the fuel region, cladding, and water region in a typical core differ significantly. At the boundaries between these regions, the cross sections are discontinuous. Therefore, for the antiderivatives involving the discontinuous cross sections in eq. (6), the following equations are no longer valid:

$$\frac{\partial^2 F_{c,s}(\mathbf{r}, \hat{\Omega}, E, \hat{\Omega}', E', t)}{\partial \hat{\Omega}' \partial E'} = \Sigma_s(\mathbf{r}, E') f(\mathbf{r}, \hat{\Omega}', E' \rightarrow \hat{\Omega}, E) \Psi(\mathbf{r}, \hat{\Omega}', E', t), \quad (11)$$

and

$$\frac{\partial^2 F_{c,f}(\mathbf{r}, \hat{\Omega}', E', t)}{\partial \hat{\Omega}' \partial E'} = v \Sigma_f(\mathbf{r}, E') \Psi(\mathbf{r}, \hat{\Omega}', E', t). \quad (12)$$

However, for multigroup equations (8) and (10), which do not involve cross sections in the integrand, these equations still hold. For eq. (6), despite cross-section discontinuities, the neutron angular flux remains continuous at the interfaces [1-3]. This continuity of angular flux in the specified neutron direction is widely applied in current practical problems. An exact differential form of the neutron transport equation can still be constructed based on this continuity. The domain can be divided into N_{sd} sub-domains based on cross section differences, and angular flux density continuity boundary conditions can be used at the interfaces. The j -th sub-domain should satisfy the following system of equations:

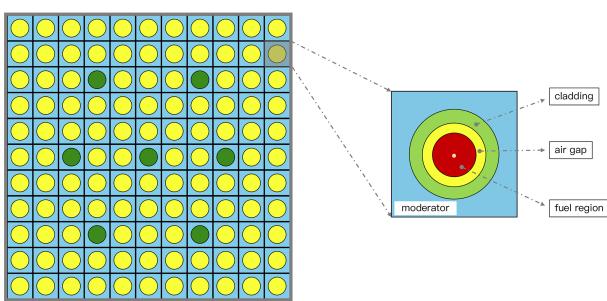


Figure 1 (Color online) Illustration of discontinuous cross sections in reactor core materials.

$$\begin{aligned} & \frac{1}{v} \frac{\partial \Psi_j(\mathbf{r}, \hat{\Omega}, E, t)}{\partial t} + \hat{\Omega} \cdot \nabla \Psi_j(\mathbf{r}_j, \hat{\Omega}, E, t) \\ & + \Sigma_{j,t}(\mathbf{r}_j, E) \Psi_j(\mathbf{r}_j, \hat{\Omega}, E, t) \\ & = F_{j,c,s}(\mathbf{r}_j, \hat{\Omega}, E, \hat{\Omega}_1, E_1, t) \\ & + \frac{x(E)}{4\pi} F_{j,c,f}(\mathbf{r}_j, \hat{\Omega}_1, E_1, t) + Q_{j,e}, \\ & \frac{\partial^2 F_{j,c,s}(\mathbf{r}_j, \hat{\Omega}, E, \hat{\Omega}', E', t)}{\partial \hat{\Omega}' \partial E'} \\ & = \Sigma_{j,s}(\mathbf{r}_j, E') f_j(\mathbf{r}_j, \hat{\Omega}', E' \rightarrow \hat{\Omega}, E) \Psi_j(\mathbf{r}_j, \hat{\Omega}', E', t), \quad (13) \\ & \frac{\partial^2 F_{j,c,f}(\mathbf{r}_j, \hat{\Omega}', E', t)}{\partial \hat{\Omega}' \partial E'} = v_j \Sigma_{j,f}(\mathbf{r}_j, E') \\ & \Psi_j(\mathbf{r}_j, \hat{\Omega}', E', t), \\ & (\hat{\Omega}' = \hat{\Omega}_0) \cup (E' = E_0), \\ & F_{j,c,s}(\mathbf{r}_j, \hat{\Omega}, E, \hat{\Omega}', E', t) = 0, \\ & F_{j,c,f}(\mathbf{r}_j, \hat{\Omega}', E', t) = 0, \\ & \Psi_j(\mathbf{r}_{j \rightarrow j+1}, \hat{\Omega}, E, t) = \Psi_{j+1}(\mathbf{r}_{j+1 \rightarrow j}, \hat{\Omega}, E, t). \end{aligned}$$

Among them, $\mathbf{r}_{j \rightarrow j+1}$ represents the position \mathbf{r} located in sub-domain j , and the distance to the adjacent interface of sub-domain $j+1$ is minimized.

In response to this situation, different neural networks can represent different subdomains, and by using the condition of continuous angular flux between subdomains at the interfaces, a set of differential equations covering the entire region can be formed, thereby solving these equations with discontinuities cross-sections. Relevant implementation strategies can be referred to in ref. [17]. The output of the neural network in the j -th sub-domain is given by

$$\Psi_j(\mathbf{r}_j, \hat{\Omega}, E, t) = \mathcal{N}_j(\mathbf{r}_j, \hat{\Omega}, E, t), \quad j = 1, 2, \dots, N_{sd}. \quad (14)$$

The final solution will be obtained as:

$$\Psi(\mathbf{r}_j, \hat{\Omega}, E, t) = \bigcup_{j=1}^{N_{sd}} \Psi_j(\mathbf{r}_j, \hat{\Omega}, E, t). \quad (15)$$

At the interfaces, the angular flux obey the following conditions as described in eq. (13):

$$\Psi_j(\mathbf{r}_{j \rightarrow j+1}, \hat{\Omega}, E, t) = \Psi_{j+1}(\mathbf{r}_{j+1 \rightarrow j}, \hat{\Omega}, E, t). \quad (16)$$

3.3 Computational complexity analysis

Computational complexity is an important metric for evaluating the efficiency of numerical methods [18]. The factors affecting the complexity of solving neutron transport equations primarily include the computation of differential terms, integral terms, variable discretization, and the iterative solving process. The most commonly used original integro-differential form (2) and the corresponding exact differential form (8) of the multi-group steady-state neutron transport equation are discussed below as examples respectively, and the other equation forms can be analysed similarly.

3.3.1 Complexity analysis of the original integro-differential form of the transport equation

(1) Analysis of differential terms calculation

In eq. (2), the specific form of differential term $\hat{\Omega} \cdot \nabla \Psi_g$ depends on the dimension of \mathbf{r} . Each iteration of every differential form, discrete grid point and machine learning sample point in the equation needs one numerical differentiation calculation. By using Euler's method, the computational complexity of each differentiation is $O(1)$. For instance, in the Cartesian coordinate system, $\hat{\Omega} \cdot \nabla \Psi_g = \mu \frac{\partial \Psi_g}{\partial x} + \eta \frac{\partial \Psi_g}{\partial y} + \xi \frac{\partial \Psi_g}{\partial z}$, For each grid/sample point, three differentiation calculations are required per iteration. Therefore, the overall complexity of the differential term still remains $O(1)$.

(2) Analysis of integral terms calculation

For the scattering source term in the multigroup neutron transport equation (2), the integral term is the neutron flux $\Psi_g^{l,m} = \int_{\hat{\Omega}_0}^{\hat{\Omega}_1} Y_l^m(\hat{\Omega}') \Psi_g(\mathbf{r}, \hat{\Omega}') d\hat{\Omega}'$. By using the trapezoidal rule, each iteration of every grid point or sample point requires calculating the integral of $Y_l^m(\hat{\Omega}') \Psi_g(\mathbf{r}, \hat{\Omega}')$ over the interval $[\hat{\Omega}_0, \hat{\Omega}_1]$. If the angular interval is divided into $N_{\hat{\Omega}}$ subintervals, the computational complexity of each integral calculation is $O(N_{\hat{\Omega}})$. Based on eq. (7), a total of $G \times L \times (2L + 1)$ integral calculations are required, where L is the maximum value in eq. (7). Typically, both G and L are constants, so the overall computational complexity remains $O(N_{\hat{\Omega}})$. If $\hat{\Omega}$ represents two-dimensional angle variable, according to the principle of cumulative transformations, the number of numerical integrations required for the integrand becomes $N_{\mu} \times N_{\varphi}$, and the complexity is $O(N_{\mu} \times N_{\varphi})$, where N_{μ} and N_{φ} represent the divisions of the cosine of the direction and the azimuthal angle respectively. For the continuous energy steady-state neutron transport equation, the computational complexity is $O(N_{\mu} \times N_{\varphi} \times N_E)$, where N_E is the number of energy discretization points.

(3) Analysis of variable discretization

Regardless of the numerical method, variables must be discretized. In finite difference or finite volume methods, the grid is discretized, whereas in deep learning methods, machine learning sample points are discretized. The computational complexity is directly related to the discretization dimension. Let the independent variable \mathbf{r} of the unknown function $\Psi_g(\mathbf{r}, \hat{\Omega})$ in eq. (2) be divided into M_r subintervals/sample points and $\hat{\Omega}$ is divided into $M_{\hat{\Omega}}$ subintervals/sample points, the total number of discretization points is $M_r \times M_{\hat{\Omega}}$, and the total number of calculations is $M_r \times M_{\hat{\Omega}}$. Naturally, as the dimensions of \mathbf{r} and $\hat{\Omega}$ increase, the number of computations will increase accordingly. For the continuous energy steady-state neutron transport equation, the total number of discretization points is $M_r \times M_{\hat{\Omega}} \times M_E$, where M_E is the number of energy discretization points.

(4) Analysis of the iterative process

For traditional methods like finite difference and finite volume, after the equation is discretized into a matrix form, iterative methods are employed for solving. Common iterative methods include Jacobi and Gauss-Seidel iterations [19], each with its own computational complexity. For equations with integral terms like eq. (2), the discretized matrix tends to be dense, meaning its has higher computational complexity [20].

In deep learning methods, the process of machine learning involves adjusting neural network weights to minimize the weighted loss function, which is composed of the equation and boundary conditions, to approximate the numerical solution of eq. (2). The convergence process in deep learning has a certain "black box" characteristics, making it difficult to analyze quantitatively and mainly dependent on experimental testing.

3.3.2 Complexity analysis of the exact differential form of the transport equation

(1) Analysis of the differential terms calculation

In the exact differential form of eq. (8), as discussed previously, the computational complexity for a single numerical differentiation is $O(1)$. However, compared with eq. (2), the calculation of $\frac{dF_{c,s,g'}^{l,m}(\mathbf{r}, \hat{\Omega}')}{d\hat{\Omega}'}$ and $\frac{dF_{c,f,g'}(\mathbf{r}, \hat{\Omega}')}{d\hat{\Omega}'}$ are added in the exact differential form, both of which have a computational complexity of $O(1)$. Since these differentiation calculations are independent, the total complexity remains $O(1)$.

(2) Analysis of variable discretization

Similar to the previous section, for $\Psi_g(\mathbf{r}, \hat{\Omega})$, $F_{c,s,g'}^{l,m}(\mathbf{r}, \hat{\Omega})$, and $F_{c,f,g'}(\mathbf{r}, \hat{\Omega})$ of eq. (8), after discretizing the independent variables $\mathbf{r}, \hat{\Omega}$ into M_r subintervals/sample points and $M_{\hat{\Omega}}$ subintervals/sample points, the total number of computations is $M_r \times M_{\hat{\Omega}}$. Again, as the dimensions of \mathbf{r} and $\hat{\Omega}$ increase, the number of computations increases proportionally.

(3) Analysis of the iterative process

For traditional methods like finite difference and finite volume, after the exact differential form (8) is discretized, the resulting matrix is sparse, meaning the iterative solving process has a significantly lower complexity compared to the matrix discretized from the integro-differential equation [20].

In deep learning methods, the exact differential form (8) can use multiple neural network outputs to represent $\Psi_g(\mathbf{r}, \hat{\Omega})$, $F_{c,s,g'}^{l,m}(\mathbf{r}, \hat{\Omega})$, and $F_{c,f,g'}(\mathbf{r}, \hat{\Omega})$. Machine learning can then be performed using a composite weighted loss function. The computational cost is similar to that of the single neural network output representing $\Psi_g(\mathbf{r}, \hat{\Omega})$ for eq. (2).

The analysis of the continuous energy steady-state neutron transport equation is consistent with sect. 3.3.1.

3.3.3 Comparison and analysis of computational complexity

From the above analysis, considering only the computation of differential and integral terms as well as variable discretization, the overall complexity of the original integro-differential form of the transport equation (2) is $O_{IDNT}(M_r \times M_{\hat{\Omega}} \times N_{\hat{\Omega}})$, while the complexity of the exact differential form (8) is $O_{EDNT}(M_r \times M_{\hat{\Omega}})$. Assuming the number of discretization points for the variable $\hat{\Omega}$, denoted by $M_{\hat{\Omega}}$, is equal to the number of divisions in the integration domain, denoted by $N_{\hat{\Omega}}$ (i.e., $M_{\hat{\Omega}} = N_{\hat{\Omega}}$), the complexity $O_{EDNT}(M_r \times M_{\hat{\Omega}})$ is significantly smaller than $O_{IDNT}(M_r \times M_{\hat{\Omega}}^2)$. Table 1 provides a detailed comparison of the computational complexities of three numerical methods for both discrete and continuous energy equation. Additionally, considering the iterative process, the iteration count of EDNT equation required for solving the sparse matrix is significantly lower than that of IDNT equation with dense matrix, thus reducing the iterative calculation time. It should be noted that the PINN method for solving neutron transport equations, whether IDNT or EDNT, involves a deep learning process with neural networks, which is widely recognized in academia as having a certain degree of non-interpretability. Therefore, it is currently difficult to provide a quantified measure of the computational complexity of PINN. This efficiency advantage is particularly evident for high-dimensional problems and deep learning methods that require large number of training iterations.

4 Numerical verification

4.1 Critical single-group slab geometry transport problem

Problem description The single-group slab geometry transport equation:

$$\begin{aligned} & \mu \frac{\partial \Psi(x, \mu)}{\partial x} + \Sigma_t(x) \Psi(x, \mu) \\ &= \frac{1}{2} \left(\Sigma_s(x) + \frac{\nu \Sigma_f(x)}{k_{\text{eff}}} \right) \int_{-1}^1 \Psi(x, \mu') d\mu', \end{aligned} \quad (17)$$

where x is the slab thickness, and μ is the direction cosine in Cartesian coordinates. The boundary conditions at both ends of the slab are vacuum boundary conditions:

$$\begin{aligned} \Psi(b, \mu) &= 0, \quad -1 \leq \mu \leq 0, \\ \Psi(-b, \mu) &= 0, \quad 0 \leq \mu \leq 1. \end{aligned} \quad (18)$$

According to the theory in ref. [3], when the material parameters are set to $\Sigma_t = 0.050 \text{ cm}^{-1}$, $\Sigma_s = 0.030 \text{ cm}^{-1}$, $\nu \Sigma_f = 0.0225 \text{ cm}^{-1}$, $b = 66.00527544 \text{ cm}$, the system reaches a critical state with the effective multiplication factor $k_{\text{eff}} = 1$. Table 2 provides the theoretical value of the scalar flux at different positions.

From a mathematical perspective, eq. (17) at the critical state represents an eigenvalue problem, where for each eigenvalue, there are infinitely many corresponding eigenvectors [1, 3]. To ensure numerical stability during the solution process, the following eigenvalue constraint conditions are introduced:

$$\Psi(0, -1) = 0.2, \quad \Psi(0, 1) = 0.2. \quad (19)$$

We define the antiderivative $F_c(x, \mu) = \int_{-1}^{\mu} \Psi(x, \mu') d\mu'$, $\frac{\partial F_c}{\partial \mu}(x, \mu) = \Psi(x, \mu)$, and rewrite the exact differential form of eq. (17) as:

$$\begin{aligned} & \mu \frac{\partial^2 F_c(x, \mu)}{\partial x \partial \mu} + \Sigma_t(x) \frac{\partial F_c(x, \mu)}{\partial \mu} \\ &= \frac{1}{2} \left(\Sigma_s(x) + \frac{\nu \Sigma_f(x)}{k_{\text{eff}}} \right) F_c(x, 1). \end{aligned} \quad (20)$$

The corresponding boundary conditions and eigenvalue constraints are

$$\begin{aligned} & \frac{\partial F_c}{\partial \mu}(b, \mu) = 0, \quad -1 \leq \mu \leq 0, \\ & \frac{\partial F_c}{\partial \mu}(-b, \mu) = 0, \quad 0 \leq \mu \leq 1, \\ & \frac{\partial F_c}{\partial \mu}(0, -1) = 0.2, \quad \frac{\partial F_c}{\partial \mu}(0, 1) = 0.2. \end{aligned} \quad (21)$$

In addition, the fixed solution constraint condition is

$$F_c(x, -1) = 0, \quad -b \leq x \leq b. \quad (22)$$

Table 1 Computational complexity of solving the neutron transport equation^{a)}

Category		Calculus terms complexity			Variable discretization complexity			Overall complexity		
Method		FDM	FVM	PINN	FDM	FVM	PINN	FDM	FVM	PINN
Discrete Energy	IDNT	$O(N_{\hat{\Omega}})$	$O(N_{\hat{\Omega}})$	$O(N_{\hat{\Omega}})$	$O(M_r M_{\hat{\Omega}})$	$O(M_r M_{\hat{\Omega}})$	$O(M_r M_{\hat{\Omega}}^2)$	$O(M_r M_{\hat{\Omega}}^2)$	$O(M_r M_{\hat{\Omega}}^2)$	-
	EDNT	$O(1)$	$O(1)$	$O(1)$	$O(M_r M_{\hat{\Omega}})$	$O(M_r M_{\hat{\Omega}})$	$O(M_r M_{\hat{\Omega}})$	$O(M_r M_{\hat{\Omega}})$	$O(M_r M_{\hat{\Omega}})$	-
Continuous Energy	IDNT	$O(N_{\hat{\Omega}} N_E)$	$O(N_{\hat{\Omega}} N_E)$	$O(N_{\hat{\Omega}} N_E)$	$O(M_r M_{\hat{\Omega}} M_E)$	$O(M_r M_{\hat{\Omega}} M_E)$	$O(M_r M_{\hat{\Omega}} M_E)$	$O(M_r M_{\hat{\Omega}}^2 M_E^2)$	$O(M_r M_{\hat{\Omega}}^2 M_E^2)$	-
	EDNT	$O(1)$	$O(1)$	$O(1)$	$O(M_r M_{\hat{\Omega}} M_E)$	$O(M_r M_{\hat{\Omega}} M_E)$	$O(M_r M_{\hat{\Omega}} M_E)$	$O(M_r M_{\hat{\Omega}} M_E)$	$O(M_r M_{\hat{\Omega}} M_E)$	-

a) Assume $M_E = N_E$ and $M_{\hat{\Omega}} = N_{\hat{\Omega}}$ in the overall complexity.

Table 2 Theoretical scalar flux values for slab geometry

Parameter	Values at different x/b				
	0 (Center)	0.25	0.50	0.75	1.00 (Boundary)
Normalized theoretical value	1	0.94714400	0.79372641	0.55329025	0.21419206

4.1.1 Finite difference method for solving the critical single-group slab geometry problem

In the finite difference method (FDM), a uniform grid is applied, with M and N grid cells selected in the x and μ directions respectively, where h_x and h_μ represent the interval lengths. To discretize the differential terms that appear in the original integro-differential form (17) and the exact differential form (20), the following central difference scheme is used:

$$\begin{aligned}\frac{\partial \Psi}{\partial x}(x_i, \mu_j) &= \frac{\Psi(x_{i+1}, \mu_j) - \Psi(x_{i-1}, \mu_j)}{2h_x}, \\ \frac{\partial F_c}{\partial x}(x_i, \mu_j) &= \frac{F_c(x_{i+1}, \mu_j) - F_c(x_{i-1}, \mu_j)}{2h_x}, \\ \frac{\partial F_c}{\partial \mu}(x_i, \mu_j) &= \frac{F_c(x_i, \mu_{j+1}) - F_c(x_i, \mu_{j-1})}{2h_\mu}.\end{aligned}\quad (23)$$

For the integral terms in the original integro-differential form, the trapezoidal rule is employed for discretization [16]:

$$\int_{-1}^1 \Psi(x, \mu') d\mu' \approx \frac{1}{2} \sum_{j=1}^{N-1} (\Psi(x, \mu_j) + \Psi(x, \mu_{j+1})) h_\mu. \quad (24)$$

The number of grid cells $M \times N$ is selected as 100×100 , 300×300 , and 500×500 respectively, and the corresponding scalar flux and relative error with the theoretical value are shown in Figure 2(a) and (b). The distribution of the exact differential form's antiderivative and the angular flux distribution obtained using the FDM with $M = N = 100$ are presented in Figure 2(c) and (d), respectively.

4.1.2 Finite volume method for solving the critical single-group slab geometry problem

The finite volume method (FVM) is similar to the FDM, which requires discretizing the domain. Uniform grids are applied, dividing the domain into grid cells, and control volumes are formed around each grid node. The crucial step of the FVM is to integrate the governing equations over each control volume. For the integro-differential form (17), integrating over a control volume V yields:

$$\begin{aligned}\int_{\Delta V} \mu \frac{\partial \Psi(x, \mu)}{\partial x} dx d\mu + \int_{\Delta V} \Sigma_t \Psi(x, \mu) dx d\mu \\ - \frac{1}{2} \left(\Sigma_s(x) + \frac{\nu \Sigma_f(x)}{k_{\text{eff}}} \right) \int_{\Delta V} \int_{-1}^1 \Psi(x, \mu') d\mu' dx d\mu = 0.\end{aligned}\quad (25)$$

By approximating the integral terms using the trapezoidal rule and applying the Gauss divergence theorem [21], eq. (25) can be discretized as:

$$\begin{aligned}\mu_{i,j} \Psi_e - \mu_{i,j} \Psi_w + h_x \Sigma_t \Psi_{i,j} \\ - \frac{1}{4} \left(\Sigma_s(x) + \frac{\nu \Sigma_f(x)}{k_{\text{eff}}} \right) h_x h_\mu \sum_{k=1}^{N-1} (\Psi_{i,k} + \Psi_{i,k+1}) = 0.\end{aligned}\quad (26)$$

Similarly, the exact differential form (20) can be discretized in the same manner:

$$\begin{aligned}h_\mu \mu_{i,j} \frac{\partial F_c}{\partial \mu} \Big|_e - h_\mu \mu_{i,j} \frac{\partial F_c}{\partial \mu} \Big|_w + h_x \Sigma_t (F_c|_n - F_c|_s) \\ - \frac{1}{2} \left(\Sigma_s(x) + \frac{\nu \Sigma_f(x)}{k_{\text{eff}}} \right) h_x h_\mu F_{c,i,N-1} = 0.\end{aligned}\quad (27)$$

The variables Ψ and F_c at the control volume faces, as well as their derivatives, are approximated as follows:

$$\begin{aligned}\Psi_e &\approx \frac{\Psi_{i,j} + \Psi_{i+1,j}}{2}, \quad \Psi_w \approx \frac{\Psi_{i-1,j} + \Psi_{i,j}}{2}, \\ F_c|_n &= \frac{F_{c,i,j} + F_{c,i,j+1}}{2}, \quad F_c|_s = \frac{F_{c,i,j} + F_{c,i,j-1}}{2}, \\ \frac{\partial F_c}{\partial \mu} \Big|_e &= \frac{F_{c,i,j+1} + F_{c,i+1,j+1} - F_{c,i,j-1} - F_{c,i+1,j-1}}{4h_\mu}, \\ \frac{\partial F_c}{\partial \mu} \Big|_w &= \frac{F_{c,i,j+1} + F_{c,i-1,j+1} - F_{c,i,j-1} - F_{c,i-1,j-1}}{4h_\mu}.\end{aligned}\quad (28)$$

For control volumes near the boundaries, the coefficients are adjusted accordingly to account for boundary conditions [21].

The grid sizes $M \times N$ selected for numerical verification are the same as those in sect. 4.1.1. Figure 3(a) and (b) show the results of scalar flux and the relative error compared to the theoretical value, computed using the finite volume method. When the grid size is $M = N = 100$, the distribution of the exact differential form's antiderivative and the angular flux distribution are shown in Figure 3(c) and (d).

4.1.3 Deep learning solution for the critical single-group slab geometry problem

In the deep learning approach, specifically using PINN [8], we constructed two fully connected neural networks, $N_\Psi(x, \mu)$ and $N_{F_c}(x, \mu)$. Each with 8 hidden layers and 16 neurons per layer, were constructed to solve for the numerical solutions $\Psi(x, \mu)$ and $F_c(x, \mu)$ of the IDNT equation and EDNT equation respectively. The integral terms in the IDNT

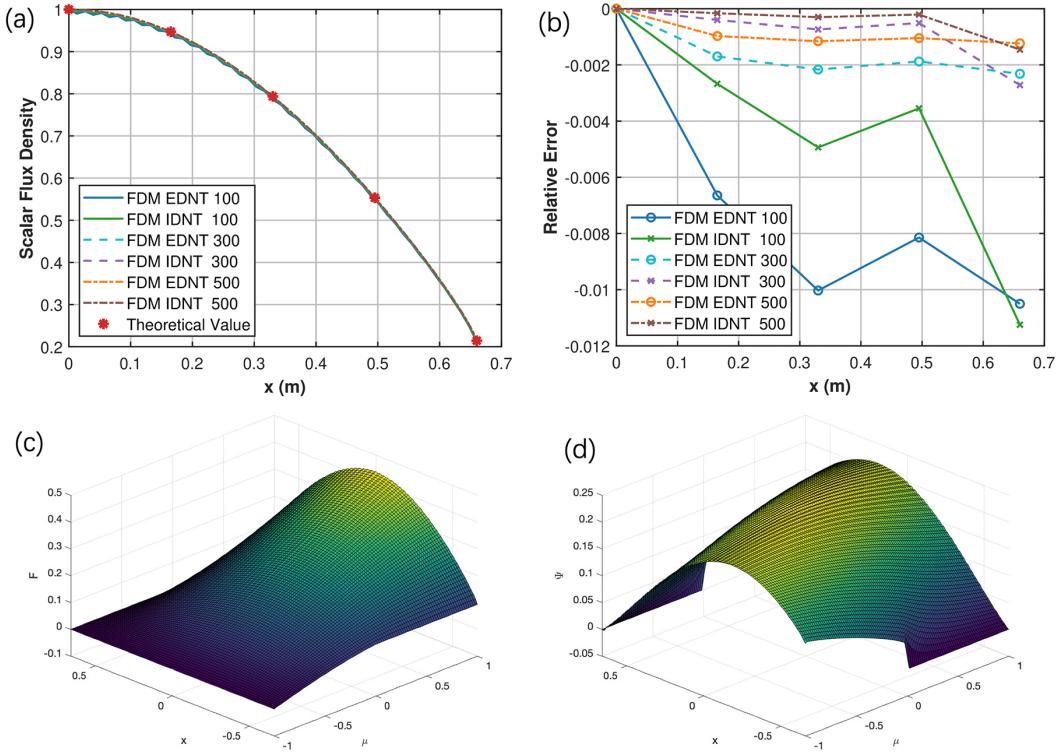


Figure 2 (Color online) Numerical results for the critical single-group slab geometry transport problem using the FDM. (a) Scalar flux density, (b) relative error, (c) antiderivative distribution of the EDNT, (d) angular flux distribution of the EDNT.

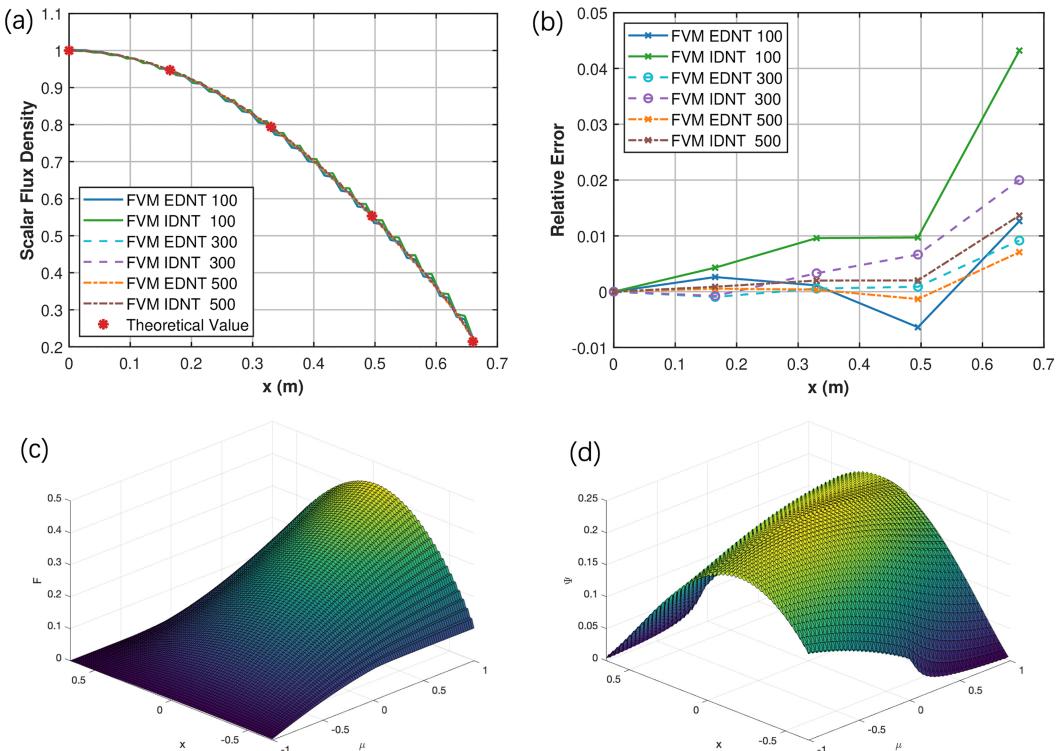


Figure 3 (Color online) Numerical results for the critical single-group slab geometry transport problem using the FVM. (a) Scalar flux density, (b) relative error, (c) antiderivative distribution of the EDNT, (d) angular flux distribution of the EDNT.

equation are approximated using the trapezoidal rule as shown in eq. (24). The network parameters were initialized with Gaussian random distributions, and the activation function was set as tanh. Training was conducted using full-batch training with the ADAM optimizer, leveraging automatic differentiation [22]. The learning rate was initialized at 1×10^{-3} , decayed gradually to 1×10^{-6} , and stopped after 1×10^5 iterations. A detailed analysis of the loss function \mathcal{L} can be found in the ref. [15]. Additionally, a schematic of the model architecture (Figure A11) is provided in Appendix A3 to facilitate replication of the experiments.

The network was trained using 1500, 3000, and 6000 sam-

pled points respectively. The results of scalar flux and relative errors compared to the theoretical value are shown in Figure 4(a) and (b). When the number of sampling points is 6000, the antiderivative distribution and the angular flux distribution are shown in Figure 4(c) and (d).

The numerical validation results indicate that when the same numerical method is applied to both the IDNT and the EDNT equations, there's not much difference in accuracy of the numerical solutions. Additionally, Table 3 summarizes the CPU times, maximum relative errors (MaxRE), and mean squared errors (MSE) for solving the critical single-group slab geometry transport problem using the FDM, FVM, and

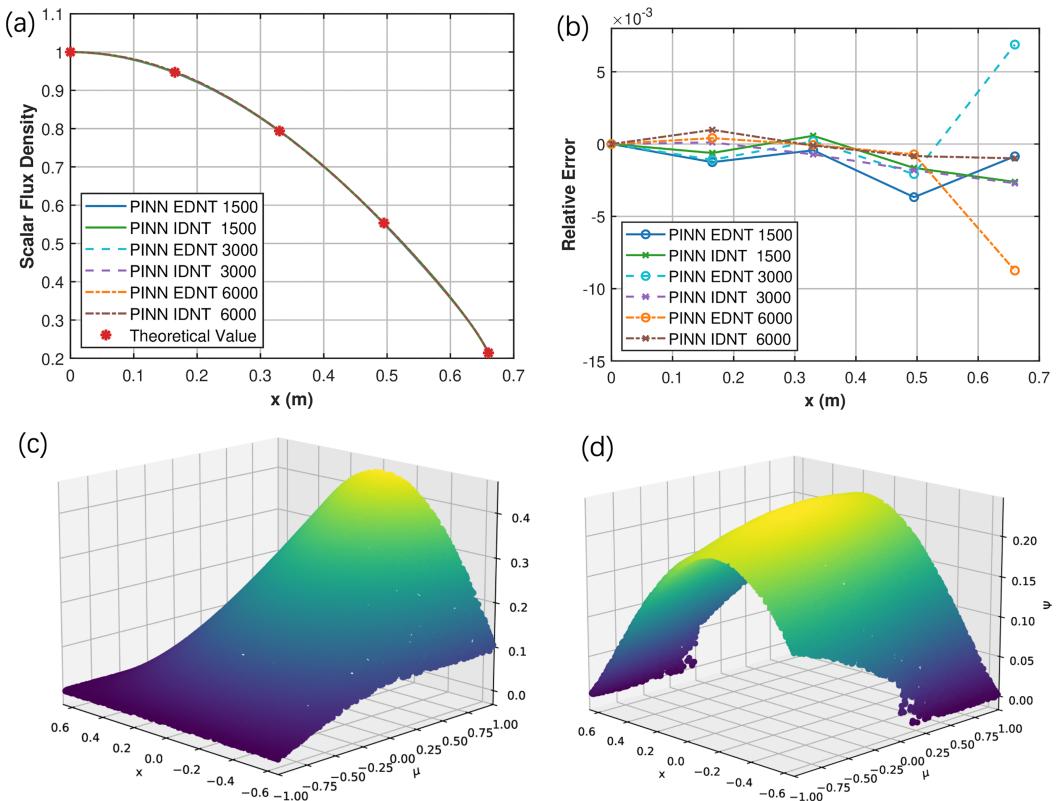


Figure 4 (Color online) Numerical results for the critical single-group slab geometry transport problem using the PINN. (a) Scalar flux density, (b) relative error, (c) antiderivative distribution of the EDNT, (d) angular flux distribution of the EDNT.

Table 3 CPU times and relative error results for three numerical methods applied to the critical single-group slab geometry transport problem

Method	FDM			FVM			PINN		
Grid size/Sampling points	100 × 100	300 × 300	500 × 500	100 × 100	300 × 300	500 × 500	1500	3000	6000
IDNT MaxRE	1.83×10^{-1}	7.24×10^{-2}	4.67×10^{-2}	4.32×10^{-2}	2.00×10^{-2}	7.07×10^{-3}	-2.62×10^{-3}	-2.71×10^{-3}	-9.89×10^{-4}
EDNT MaxRE	1.77×10^{-1}	7.12×10^{-2}	4.61×10^{-2}	1.26×10^{-2}	9.17×10^{-3}	1.36×10^{-2}	-3.67×10^{-3}	6.88×10^{-3}	-8.75×10^{-3}
IDNT MSE	4.17×10^{-4}	6.02×10^{-5}	2.45×10^{-5}	3.78×10^{-5}	7.85×10^{-6}	2.60×10^{-6}	1.14×10^{-6}	9.15×10^{-7}	7.64×10^{-7}
EDNT MSE	3.80×10^{-4}	5.67×10^{-5}	2.32×10^{-5}	5.34×10^{-6}	1.02×10^{-6}	6.37×10^{-7}	3.40×10^{-7}	3.35×10^{-7}	2.25×10^{-7}
IDNT time (s)	0.16	4.80	36.60	0.43	8.50	50.84	2410.33	5786.88	13350.21
EDNT time (s)	0.04	0.45	1.60	0.07	0.65	1.67	1176.75	1205.62	2235.76
Time ratio: $\frac{\text{IDNT}}{\text{EDNT}}$	4.00	9.07	22.88	6.14	13.08	30.44	2.05	4.80	5.97

PINN. As the grid density or the number of sampling points increases, the EDNT equation gradually demonstrates a significantly advantage in computational efficiency over the IDNT equation. This finding is consistent with the analysis of computational complexity in sect. 3.3, where $O_{\text{EDNT}}(M_r \times M_\Omega) \ll O_{\text{IDNT}}(M_r \times M_\Omega^2)$.

4.2 Single-group cylindrical geometry transport problem with fixed source

Problem Description Assuming the neutron source is isotropic and neglecting higher-order scattering terms, the single-group cylindrical geometry transport equation with fixed source is as follows:

$$\sqrt{1 - \mu^2} \left(\cos \phi \frac{\partial \Psi}{\partial r} - \frac{\sin \phi}{r} \frac{\partial \Psi}{\partial \phi} \right) + \Sigma_t \Psi = Q_s + Q_f, \quad (29)$$

where $Q_s = \frac{1}{4\pi} \Sigma_s(r) \int_{-1}^1 \int_0^{2\pi} \Psi(r, \phi', \mu') d\phi' d\mu'$, and $Q_f = 0.2 \cos\left(\frac{\pi r}{2R}\right)$. The total cross section is $\Sigma_t = 0.050 \times 10^2 \text{ m}^{-1}$, and the scattering cross section is $\Sigma_s = 0.030 \times 10^2 \text{ m}^{-1}$. The radius of the cylinder is $R = 1.08225766 \text{ m}$. The boundary conditions are vacuum conditions:

$$\Psi(R, \phi, \mu) = 0, \quad \frac{\pi}{2} \leq \phi \leq \frac{3\pi}{2}. \quad (30)$$

Define $F_c(r, \phi, \mu) = \int_{-1}^1 \int_0^{2\pi} \Psi(r, \phi', \mu') d\phi' d\mu'$, and $\frac{\partial^2 [F_c(r, \phi', \mu')]}{\partial \mu' \partial \phi'} = \Psi(r, \phi', \mu')$. The corresponding exact differential form of eq. (29) can be written as:

$$\begin{aligned} & \sqrt{1 - \mu^2} \left(\cos \phi \frac{\partial^3 F_c}{\partial \mu \partial \phi \partial r} - \frac{\sin \phi}{r} \frac{\partial^3 F_c}{\partial \mu \partial^2 \phi} \right) + \Sigma_t \frac{\partial^2 F_c}{\partial \mu \partial \phi} \\ &= \frac{1}{4\pi} \Sigma_s(r) F_c(r, 2\pi, 1) + 0.2 \cos\left(\frac{\pi r}{2R}\right). \end{aligned} \quad (31)$$

The vacuum boundary conditions for this form are

$$\frac{\partial^2 F_c(R, \phi, \mu)}{\partial \mu \partial \phi} = 0, \quad \frac{\pi}{2} \leq \phi \leq \frac{3\pi}{2}. \quad (32)$$

Additionally, the following fixed solution constraint condition are applied:

$$\begin{aligned} F_c(r, 0, \mu) &= 0, \quad 0 \leq r \leq R, \quad -1 \leq \mu \leq 1, \\ F_c(r, \phi, -1) &= 0, \quad 0 \leq r \leq R, \quad 0 \leq \phi \leq 2\pi. \end{aligned} \quad (33)$$

4.2.1 Finite difference method for the single-group cylindrical geometry transport problem with fixed source

Similar to the slab geometry problem, the cylindrical geometry problem is also discretized using a uniform grid. Grid cells are selected in the r , ϕ , and μ directions, with M , N , and S grid cells respectively, where h_r , h_ϕ and h_μ represent the

interval lengths. Central difference schemes are used to discretize the differential terms in both the IDNT and EDNT equations of the equation. For the integral terms in the IDNT equation, the two-dimensional trapezoidal rule is used for discretisation [16]:

$$\begin{aligned} & \int_{-1}^1 \int_0^{2\pi} N_\Psi(r_i, \phi', \mu') d\phi' d\mu' \\ &= \frac{1}{4} \sum_{j=1}^{S-1} \sum_{k=1}^{N-1} \left(N_\Psi(r_i, \phi_k, \mu_j) + N_\Psi(r_i, \phi_{k+1}, \mu_j) \right. \\ & \quad \left. + N_\Psi(r_i, \phi_k, \mu_{j+1}) + N_\Psi(r_i, \phi_{k+1}, \mu_{j+1}) \right) h_\phi h_\mu. \end{aligned} \quad (34)$$

The number of grid cells $M \times N \times S$ is selected as $24 \times 24 \times 24$, $28 \times 28 \times 28$, and $32 \times 32 \times 32$, respectively. Since this problem lacks an analytical solution or theoretical estimate, we use the flux results obtained from the transport simulation code OpenMC [23] as the reference solution. The flux computed from the IDNT and EDNT equations, along with their relative errors compared to the OpenMC reference solution, are shown in Figure 5(a) and (b). When the grid cell number is $M = N = S = 32$, the antiderivative of the EDNT equation obtained using the FDM, as well as the angular flux distribution, are shown in Figure 5(c)-(f).

4.2.2 Finite volume method for the single-group cylindrical geometry transport problem with fixed source

The FVM for the cylindrical geometry transport problem also employs uniform grids. Grid nodes are selected in the r , ϕ , and μ directions, with M , N , and S nodes, and corresponding control volumes are constructed around each node. As in the slab geometry case, the control equations are integrated over each control volume, and the discrete equations are derived using the Gauss divergence theorem and the trapezoidal integration rule (34), combined with an upwind scheme [21].

The grid sizes $M \times N \times S$ selected for numerical verification are the same as those in sect. 4.2.1. The flux computed from the IDNT and EDNT equations, along with their relative errors compared to the OpenMC reference solution, are shown in Figure 6(a) and (b). For a grid size of $M = N = S = 32$, the numerical results for the EDNT equation obtained using the FVM are shown in Figure 6(c)-(f).

4.2.3 Deep learning method for the single-group cylindrical geometry transport problem with fixed source

In this approach, two fully connected neural networks with 8 hidden layers and 16 neurons per layer, $N_\Psi(r, \phi, \mu)$ and $N_{F_c}(r, \phi, \mu)$, are used to solve the original integro-differential form $\Psi(r, \phi, \mu)$ and the exact differential form $F_c(r, \phi, \mu)$ respectively. Integral terms in the IDNT equation are discretized using the trapezoidal rule (34). The rest of the

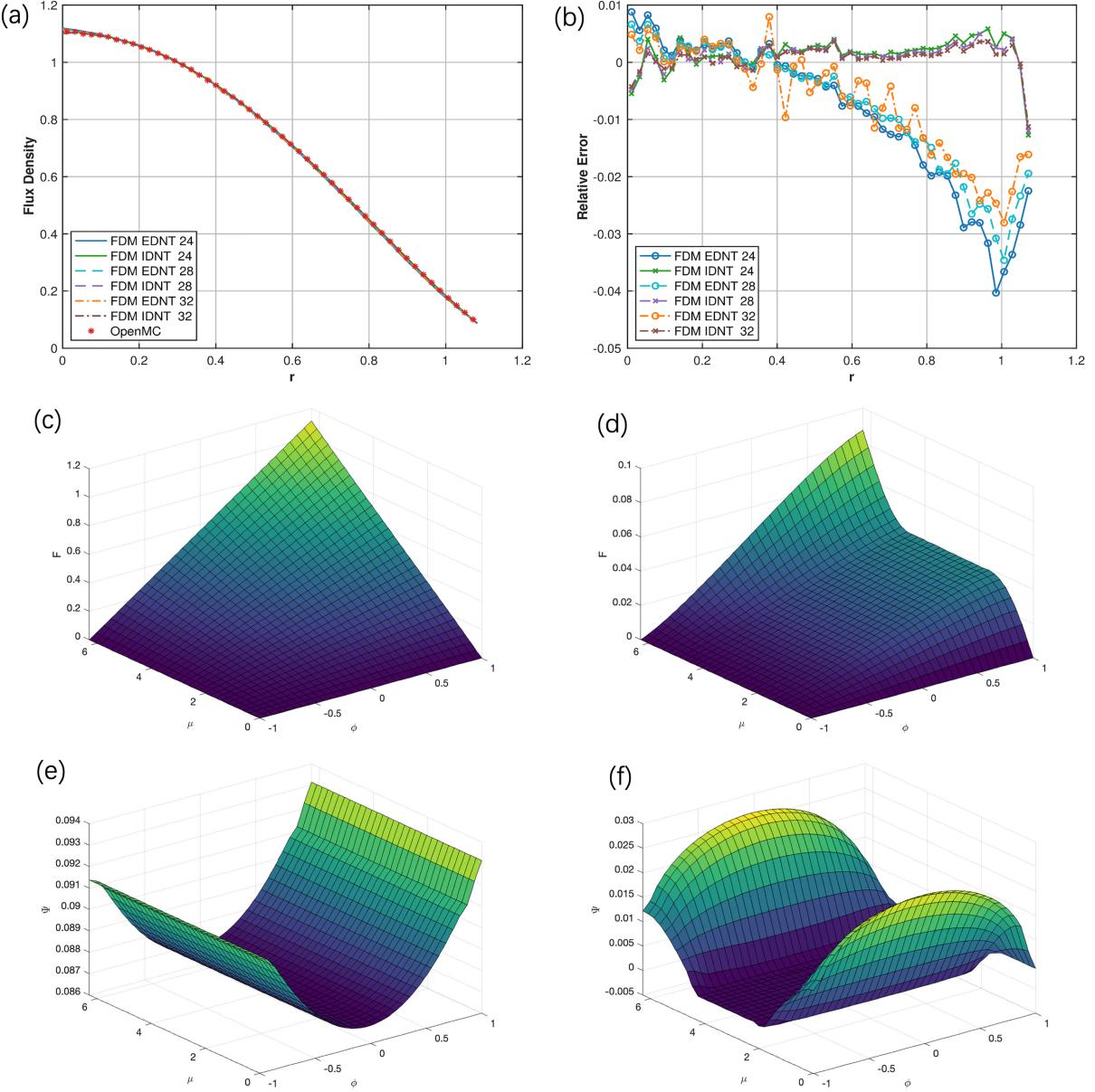


Figure 5 (Color online) Numerical results for the single-group cylindrical geometry transport problem with fixed source using the FDM. (a) Flux density, (b) relative error, (c) antiderivative distribution of the EDNT at $r = 0$, (d) antiderivative distribution of the EDNT at $r = R$, (e) angular flux distribution of the EDNT at $r = 0$, (f) angular flux distribution of the EDNT at $r = R$.

training parameters and strategies are same as that of sect. 4.1.3. The model architecture schematic (Figure A12) is in Appendix A3. The model is trained with 2000, 4000, and 8000 sampling points respectively. The flux obtained from the IDNT and EDNT equations, along with their relative errors compared to the OpenMC reference solution, are shown in Figure 7(a) and (b). For 8000 sampling points, the numerical results for the EDNT equation obtained using the PINN are shown in Figure 7(c)-(f).

The relative errors between the numerical solutions obtained from three different methods and the reference solutions indicate that the EDNT and IDNT equations achieve

comparable accuracy when using the same numerical methods. Additionally, Table 4 summarizes the CPU times and relative error results for solving the single-group cylindrical geometry transport problem with a fixed source. As the grid is refined or the number of sampling points increases, the EDNT equation demonstrates a more significant advantage in computational efficiency compared to the IDNT equation. Compared to the slab geometry case, this advantage becomes even more obvious. It is primarily due to the increasing difference in complexity as the number of angular variables grows. This result further validates the computational complexity analysis presented in sect. 3.3.

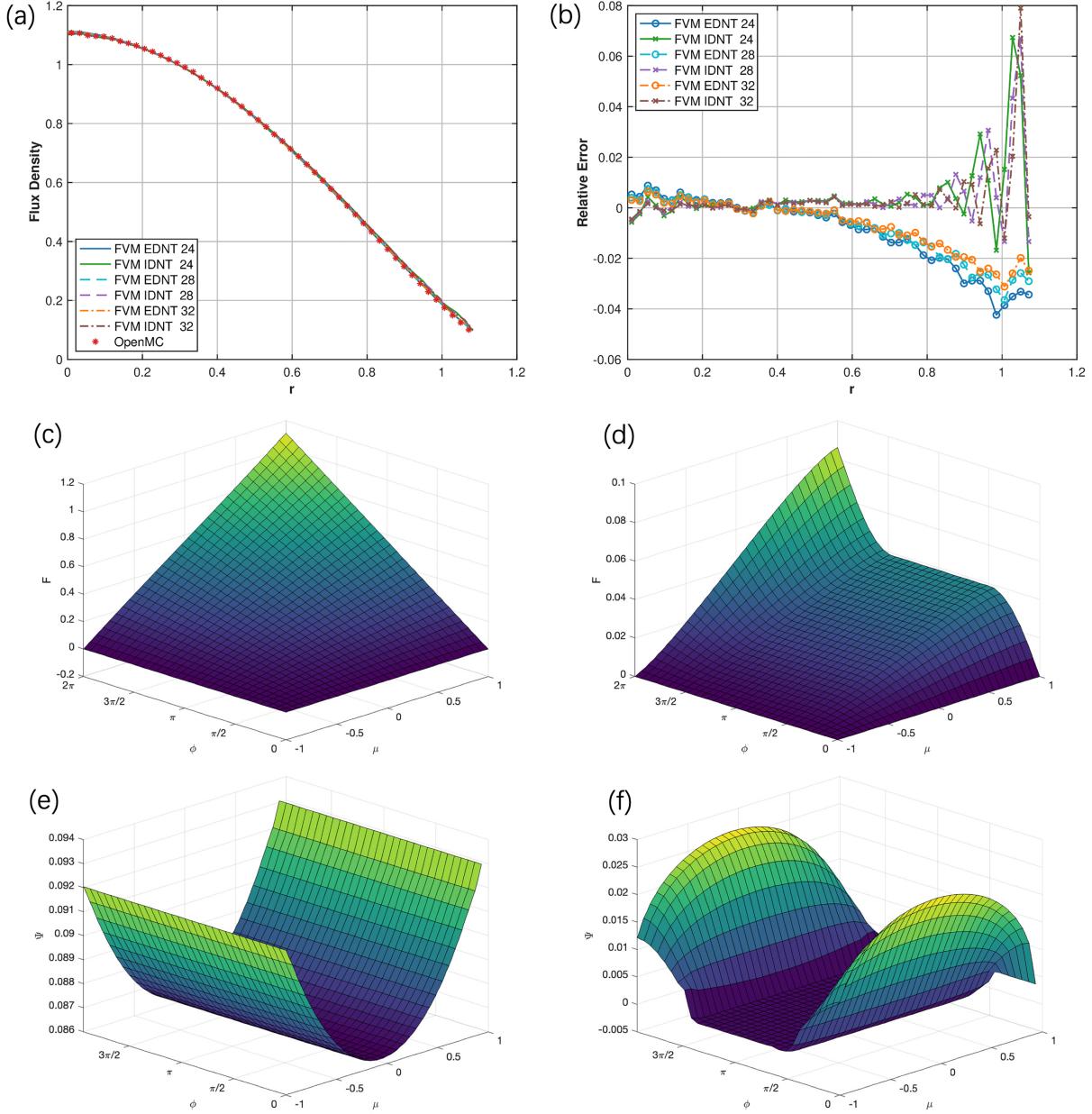


Figure 6 (Color online) Numerical results for the single-group cylindrical geometry transport problem with fixed source using the FVM. (a) Flux density, (b) relative error, (c) antiderivative distribution of the EDNT at $r = 0$, (d) antiderivative distribution of the EDNT at $r = R$, (e) angular flux distribution of the EDNT at $r = 0$, (f) angular flux distribution of the EDNT at $r = R$.

It is worth noting that a significant characteristic of deep learning methods is to generate continuous solutions that are inherent in neural networks. In contrast, traditional methods, such as finite difference and finite volume methods, produce discrete solutions.

4.3 Multi-material cylindrical geometry transport problem with fixed source

Problem Description In this case, we extend the single-group cylindrical geometry transport problem to a multi-

material scenario. The transport equation form of EDNT and IDNT is similar to that of the single-material case in sect. 4.2, but the total cross section $\Sigma_t(r)$ and scattering cross section $\Sigma_s(r)$ are defined in Figure 8. The boundary conditions remain the same as in the single-material case in sect. 4.2.

4.3.1 Finite difference method for the multi-material cylindrical geometry transport problem with fixed source

The problem is discretized using a uniform grid in the r , ϕ , and μ directions. The main difference from sect. 4.2.1 lies

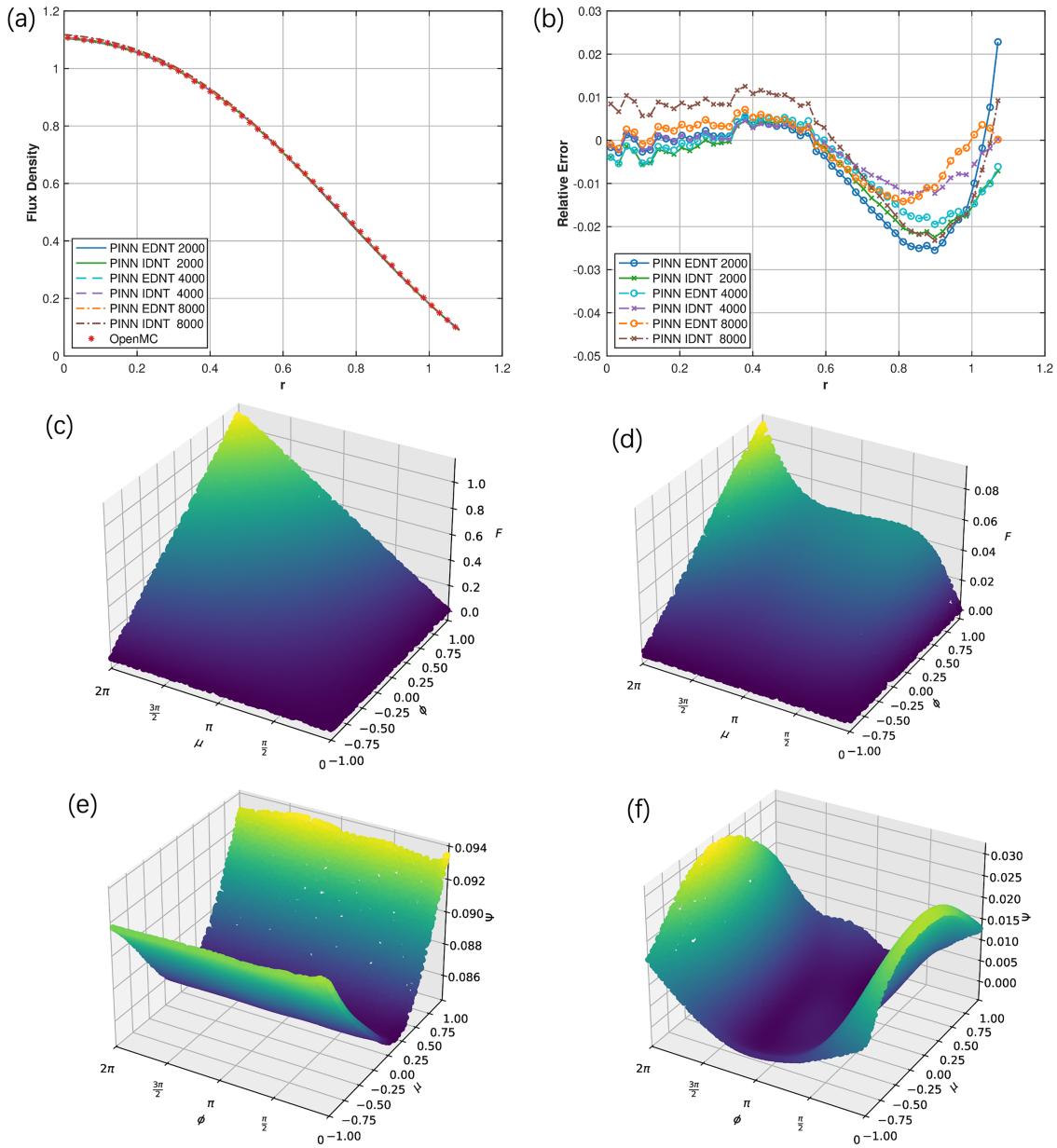


Figure 7 (Color online) Numerical results for the single-group cylindrical geometry transport problem with fixed source using the PINN. (a) Flux density, (b) relative error, (c) antiderivative distribution of the EDNT at $r = 0$, (d) antiderivative distribution of the EDNT at $r = R$, (e) angular flux distribution of the EDNT at $r = 0$, (f) angular flux distribution of the EDNT at $r = R$.

Table 4 CPU time and relative error results for three numerical methods applied to the single-group cylindrical geometry transport problem with fixed source

Numerical method	FDM				FVM			PINN		
	$24 \times 24 \times 24$	$28 \times 28 \times 28$	$32 \times 32 \times 32$	$24 \times 24 \times 24$	$28 \times 28 \times 28$	$32 \times 32 \times 32$	2000	4000	8000	
IDNT MaxRE	1.28×10^{-2}	1.20×10^{-2}	1.13×10^{-2}	6.74×10^{-2}	6.70×10^{-2}	7.90×10^{-2}	2.25×10^{-2}	1.23×10^{-2}	2.32×10^{-2}	
EDNT MaxRE	4.03×10^{-2}	3.46×10^{-2}	2.81×10^{-2}	4.24×10^{-2}	3.65×10^{-2}	3.12×10^{-2}	2.55×10^{-2}	1.95×10^{-2}	1.42×10^{-2}	
IDNT MSE	3.70×10^{-6}	2.35×10^{-6}	1.70×10^{-6}	9.44×10^{-6}	6.84×10^{-6}	5.47×10^{-6}	2.16×10^{-5}	7.54×10^{-6}	5.51×10^{-5}	
EDNT MSE	3.04×10^{-5}	2.13×10^{-5}	1.98×10^{-5}	3.12×10^{-5}	2.24×10^{-5}	1.69×10^{-5}	2.48×10^{-5}	1.67×10^{-5}	1.46×10^{-5}	
IDNT time (s)	17.64	91.44	538.38	9.88	40.84	163.26	12339.09	28991.51	113200.76	
EDNT time (s)	0.33	1.11	3.38	0.48	1.04	2.74	2468.72	4423.05	6525.75	
Time ratio: $\frac{\text{IDNT}}{\text{EDNT}}$	53.02	82.38	159.19	20.66	39.16	59.51	5.00	6.55	17.35	

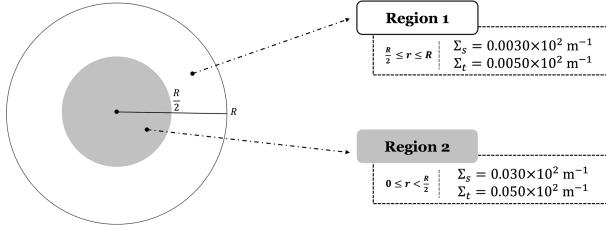


Figure 8 Geometric description of multi-material cylindrical geometry transport problem with fixed source.

in the piecewise definitions of $\Sigma_t(r)$ and $\Sigma_s(r)$. The integral terms are discretized using the two-dimensional trapezoidal rule, as described before eq. (34). The grid sizes are set as $32 \times 32 \times 32$. The flux results and relative errors compared to the OpenMC reference are shown in Figure 9(a) and (b).

4.3.2 Finite volume method for the multi-material cylindrical geometry transport problem with fixed source

The FVM for the problem also employs uniform grids. Grid nodes are selected in the r , ϕ , and μ directions, with M , N , and S nodes, and corresponding control volumes are constructed around each node. The control equations are integrated over each control volume, and the discrete equations are derived using the Gauss divergence theorem and the trapezoidal integration rule (34), combined with an upwind scheme [21] as same as sect. 4.2.2. The grid sizes are set as $32 \times 32 \times 32$. The flux results and relative errors compared to the OpenMC reference are shown in Figure 9(a) and (b).

4.3.3 Deep learning method for the multi-material cylindrical geometry transport problem with fixed source

Two fully connected neural networks, $N_\Psi(r, \phi, \mu)$ and $N_{F_c}(r, \phi, \mu)$, are used to solve the problem. The integral terms are discretized as same as sect. 4.2.3. The model is trained

with 4000 sampling points. The results and relative errors compared to the OpenMC reference are presented in Figure 9(a) and (b).

Additionally, Table 5 summarizes the CPU times and relative error results of the multi-material cylindrical geometry transport problem with a fixed source. The results indicate that EDNT achieves comparable computational accuracy to IDNT, while significantly improving computational efficiency with reductions in computational time ranging from several times to several orders of magnitude.

4.4 Non-critical two-group slab geometry transport problem

Problem Description For the slab geometry two-group single-material region, the two-group slab theory and transport equation IDNT form are based on ref. [1]. The slab thickness $b = 50$ cm, and the material properties are listed in Table 6. We use this example to verify the applicability of EDNT to the PINN method in multi-group scenarios.

The corresponding EDNT equation is as follows:

$$\left\{ \begin{aligned} & \mu \frac{\partial F_{c,1}(x, \mu)}{\partial x \partial \mu} + \Sigma_{t,1}(x) F_{c,1}(x, \mu)' \\ &= \frac{1}{2} (\Sigma_{s,1-1}(x)[F_{c,1}(x, 1)] + \Sigma_{s,2-1}(x)[F_{c,2}(x, 1)]) \\ &+ \frac{1}{2k_{\text{eff}}} ((v\Sigma_f(x))_1[F_{c,1}(x, 1)] + (v\Sigma_f(x))_2[F_{c,2}(x, 1)]) \\ & \mu \frac{\partial F_{c,2}(x, \mu)}{\partial x \partial \mu} + \Sigma_{t,2}(x) F_{c,2}(x, \mu)' \\ &= \frac{1}{2} (\Sigma_{s,1-2}(x)[F_{c,1}(x, 1)] + \Sigma_{s,2-2}(x)[F_{c,2}(x, 1)]) \\ & F_{c,1}(x, \mu)' = \frac{dF_{c,1}(x, \mu)}{d\mu} = \Psi_1(x, \mu), F_{c,1}(x, -1) = 0 \\ & F_{c,2}(x, \mu)' = \frac{dF_{c,2}(x, \mu)}{d\mu} = \Psi_2(x, \mu), F_{c,2}(x, -1) = 0 \\ & x \in [-0.5, 0.5], \mu \in [-1, 1] \end{aligned} \right\}, \quad (35)$$

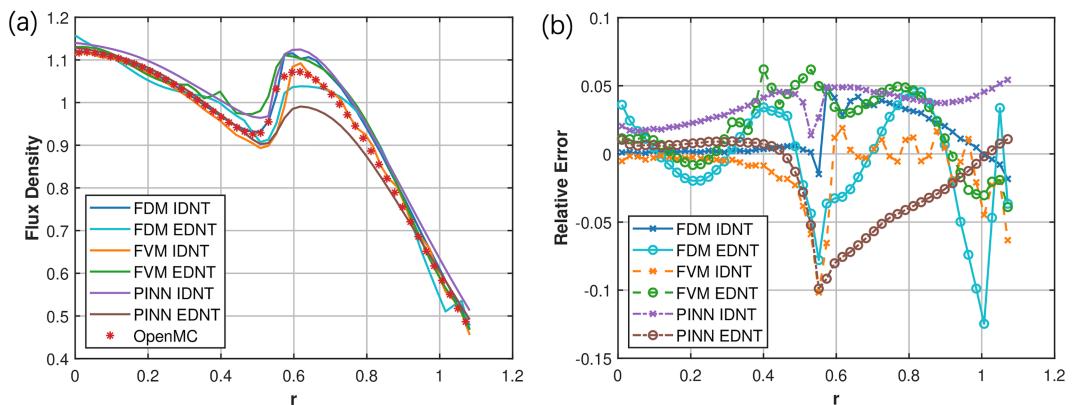


Figure 9 (Color online) Numerical results for the multi-material cylindrical geometry transport problem with fixed source. (a) Flux density, (b) relative error.

Table 5 CPU time and relative error results for three numerical methods applied to the multi-material cylindrical geometry transport problem with fixed source

Numerical method	FDM	FVM	PINN
IDNT MSE	3.99×10^{-4}	3.98×10^{-4}	1.17×10^{-3}
EDNT MSE	9.15×10^{-4}	9.06×10^{-4}	1.38×10^{-3}
IDNT time (s)	91.78	179.76	34706.51
EDNT time (s)	0.86	7.46	4542.73
Time ratio: $\frac{\text{IDNT}}{\text{EDNT}}$	106.72	24.10	7.64

Table 6 Material properties in the calculation region of two-group slab geometry

g	$\Sigma_t (\text{cm}^{-1})$	$v\Sigma_f (\text{cm}^{-1})$	x_g	$\Sigma_{s,1-g} (\text{cm}^{-1})$	$\Sigma_{s,2-g} (\text{cm}^{-1})$
1	0.33285210	0.01266922	1	0.31876071	0
2	0.44925527	0.22110247	0	0.00093413	0.31466259

where Ψ_1 and Ψ_2 represent the angular flux densities of the thermal and fast groups, respectively, while $F_{c,1}$ and $F_{c,2}$ are corresponding antiderivatives.

The PINN deep learning method and parameter choices are as follows: Two neural networks are used to represent the thermal group and fast group, respectively. The network architecture, machine learning rate, and other hyperparameters are the same as in sect. 4.1.3. During the learning process, the weights of the two neural networks are alternately updated to solve for the equation. The eigenvalue constraint for

the fast group is $F'_{c,1}(0, -1) = F'_{c,1}(0, 1) = 1$, while no eigenvalue constraint is applied to the thermal group. The detailed methods for eigenvalue constraints and the construction of the loss function can be found in ref. [15].

Since this problem does not have an analytical solution or theoretical estimate, the normalized scalar flux results of PINN are compared with that of OpenMC. The results are shown in Figure 10 and Table 7. As shown in Figure 10, the numerical results also provide a continuous angular flux distribution for the two-group system. The validation work for this problem demonstrates that the EDNT is suitable for numerical solution techniques based on deep learning, such as PINN.

5 Conclusion

The neutron transport equation has various equivalent forms suited for different numerical discretization methods. The traditional form is not well-suited for deep learning-based numerical methods, which typically require long training time. To address this, the main contribution of this paper is the proposal of a new exact differential form of the neutron transport equation. By converting the integral terms in the integro-differential neutron transport equation into their corresponding antiderivatives, an exact differential form of neutron transport equations is proposed, and the physical

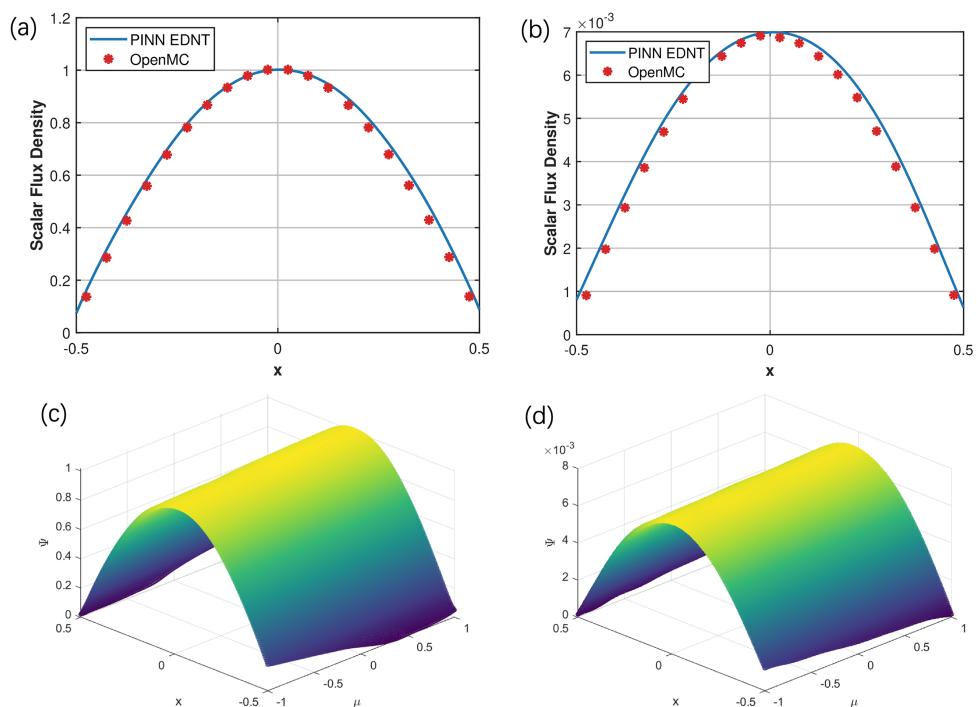


Figure 10 (Color online) Numerical calculation results of the non-critical two-group slab geometry transport problem. (a) Comparison of fast group flux with OpenMC software; (b) comparison of thermal group flux with OpenMC software; (c) angular flux distribution of the EDNT for the fast group; (d) angular flux distribution of the EDNT for the thermal group.

Table 7 Calculation results for non-critical two-group scalar flux of slab geometry

Energy group	k_{eff}		Relative deviation of k_{eff}	EDNT MSE	Maximum absolute error of EDNT
	PINN	OpenMC			
Fast Group	0.9346	0.9464	-3.80×10^{-2}	7.93×10^{-4}	5.22×10^{-2}
Thermal Group				6.43×10^{-8}	3.96×10^{-4}

meanings of the antiderivatives are elucidated. A comparative analysis of the computational complexity between the two forms was conducted. Numerical experiments, performed on typical benchmark problems using different numerical methods, further verified the results.

Both theoretical analysis and numerical experiments demonstrate that the EDNT equation is universally applicable across various numerical methods, offering a significantly advantage in computational efficiency over the IDNT equation. It approves that the EDNT equation is well-suited for deep learning-based numerical method.

Future research is recommended to focus on improving traditional numerical methods based on the EDNT equation to enhance the computational efficiency of current reactor core design software, and we will apply the EDNT format to address the corresponding continuous energy neutron transport problems. In terms of numerical computation methods involving deep learning, subsequent work should concentrate on optimizing the construction of neural network architectures. Additionally, This EDNT approach may also be applicable for other integro-differential transport theories such as radiative energy transport and has potential application in astrophysics or other fields.

This work was supported by the National High Level Talent Special Support Program of China (Grant No. J705981200002001), the Basic Research and Stable Support Research Project of Science and Technology Industry Bureau of China (Grant No. WDZC-2023050305), the Sichuan Province Unveiled and Leading Common Technology Project of China and Natural Science Foundation of Sichuan Province (Grant No. 2023NSFSC0066), and the National Natural Science Foundation of China (Grant No. 12175220).

Conflict of interest The authors declare that they have no conflict of interest.

- 1 R. M. Kuridan, *Neutron Transport: Theory, Modeling, and Computations* (Springer Nature Switzerland, Cham, 2023).
- 2 L. Cao, *Deterministic Numerical Methods for Unstructured-Mesh Neutron Transport Calculation* (Elsevier Science & Technology, San Diego, 2020).
- 3 E. E. Lewis, and W. F. Miller, *Computational Methods of Neutron Transport* (John Wiley and Sons, Inc., New York, 1984).
- 4 A. M. Weinberg, and E. P. Wigner, *The Physical Theory of Neutron Chain Reactors* (University of Chicago Press, Chicago, 1958).
- 5 J. J. Duderstadt, and W. R. Martin, *Transport Theory* (Wiley-Interscience, New York, 1979).
- 6 D. A. Pratama, M. A. Bakar, N. B. Ismail, and M. M. Arab J. Basic Appl. Sci. **29**, 233 (2022).

- 7 W. Zha, D. Li, L. Shen, W. Zhang, and X. Liu, Chin. J. Mech. **54**, 543 (2022).
- 8 M. Raissi, P. Perdikaris, and G. E. Karniadakis, J. Comput. Phys. **378**, 686 (2019).
- 9 L. Lu, P. Jin, G. Pang, Z. Zhang, and G. E. Karniadakis, Nat. Mach. Intell. **3**, 218 (2021).
- 10 Y. Xie, Y. Wang, Y. Ma, and Z. Wu, J. Nucl. Eng. **2**, 533 (2021).
- 11 D. Liu, Q. Luo, L. Tang, P. An, and F. Yang, Nucl. Power Eng. **43**, 1 (2022).
- 12 Y. Yang, H. Gong, S. Zhang, Q. Yang, Z. Chen, Q. He, and Q. Li, Ann. Nucl. Energy **183**, 109656 (2023), arXiv: 2208.13483.
- 13 J. Wang, X. Peng, Z. Chen, B. Zhou, Y. Zhou, and N. Zhou, Ann. Nucl. Energy **176**, 109234 (2022).
- 14 Z. Chen, L. Liu, and L. Mu, Discrete Cont. Dyn. Syst.-S **15**, 669 (2022).
- 15 D. Liu, X. Q. Wang, B. Zhang, C. Y. Yu, Z. H. Gong, and Q. L. Chen, At. Energy Sci. Technol. **57**, 946 (2023).
- 16 R. Courant, F. John, R. Courant, and F. John, *Introduction to Calculus and Analysis* (Springer, Berlin, Heidelberg, 1999).
- 17 A. D. Jagtap, E. Kharazmi, and G. E. Karniadakis, Comput. Methods Appl. Mech. Eng. **365**, 113028 (2020).
- 18 S. Arora, and B. Barak, *Computational Complexity: A Modern Approach*, 1st ed (Cambridge University Press, Cambridge, 2009).
- 19 W. F. Ames, *Numerical Methods for Partial Differential Equations*, 3rd ed (Elsevier Science, Burlington, 2014).
- 20 P. Polyanin, and A. V. Manzhirov, *Handbook of Integral Equations: Second Edition*, 2nd ed (Chapman and Hall/CRC, New York, 2008).
- 21 F. Moukalled, L. Mangani, and M. Darwish, *The Finite Volume Method in Computational Fluid Dynamics: An Advanced Introduction with OpenFOAM® and Matlab®* (Springer International Publishing, Cham, 2016).
- 22 A. G. Baydin, B. A. Pearlmutter, A. A. Radul, and J. M. Siskind, J. Mach. Learn. Res. **18**, 1 (2018).
- 23 P. K. Romano, and B. Forget, Ann. Nucl. Energy **51**, 274 (2013).

Appendix

A1 Derivation of the exact differential form of the continuous energy neutron transport equation

A1.1 Antiderivative transformation for scattering sources

The anisotropic scattering source in eq. (1) is given by

$$Q_s = \iint \Sigma_s(\mathbf{r}, E') f(\mathbf{r}, \hat{\Omega}', E' \rightarrow \hat{\Omega}, E) \Psi(\mathbf{r}, \hat{\Omega}', E', t) d\hat{\Omega}' dE'. \quad (\text{a1})$$

In general, the angular flux $\Psi(\mathbf{r}, \hat{\Omega}, E, t)$ is a continuous function with respect to the angular variable $\hat{\Omega}$ and the energy E [1-3]. Suppose that the integrand $\Sigma_s(\mathbf{r}, E') f(\mathbf{r}, \hat{\Omega}', E' \rightarrow \hat{\Omega}, E)$ is continuous, then the integrand $\Sigma_s(\mathbf{r}, E') f(\mathbf{r}, \hat{\Omega}', E' \rightarrow \hat{\Omega}, E) \Psi(\mathbf{r}, \hat{\Omega}', E', t)$ is also

continuous. According to the principles of calculus [16], a continuous integrand has an antiderivative with respect to the angular variable $\hat{\Omega}'$ and energy E' , denoted as $F_s(\mathbf{r}, \hat{\Omega}, E, \hat{\Omega}', E', t)$. Referring to the two-dimensional Newton-Leibniz formula [16], we perform an antiderivative transformation on the scattering source terms in the neutron transport equation (1).

For the scattering source term in eq. (1), we have

$$\begin{aligned} Q_s &= \int_{E_0}^{E_1} \int_{\hat{\Omega}_0}^{\hat{\Omega}_1} \Sigma_s(\mathbf{r}, E') f(\mathbf{r}, \hat{\Omega}', E' \rightarrow \hat{\Omega}, E) \\ &\quad \Psi(\mathbf{r}, \hat{\Omega}', E', t) d\hat{\Omega}' dE' \\ &= F_s(\mathbf{r}, \hat{\Omega}, E, \hat{\Omega}_1, E_1, t) - F_s(\mathbf{r}, \hat{\Omega}, E, \hat{\Omega}_1, E_0, t) \\ &\quad - F_s(\mathbf{r}, \hat{\Omega}, E, \hat{\Omega}_0, E_1, t) + F_s(\mathbf{r}, \hat{\Omega}, E, \hat{\Omega}_0, E_0, t), \end{aligned} \quad (\text{a2})$$

where $\hat{\Omega}_1, \hat{\Omega}_0$ are the upper and lower limits of the angular integral, and E_1, E_0 are the upper and lower limits of the energy integral.

According to the definition of antiderivatives:

$$\begin{aligned} &\frac{\partial F_s(\mathbf{r}, \hat{\Omega}, E, \hat{\Omega}', E', t)}{\partial \hat{\Omega}' \partial E'} \\ &= \Sigma_s(\mathbf{r}, E') f(\mathbf{r}, \hat{\Omega}', E' \rightarrow \hat{\Omega}, E) \Psi(\mathbf{r}, \hat{\Omega}', E', t). \end{aligned} \quad (\text{a3})$$

Based on the principles of calculus, we know that $F_s(\mathbf{r}, \hat{\Omega}, E, \hat{\Omega}', E', t)$ is a cluster of functions. These functions differ from each other by any continuous function with respect to the variable $\hat{\Omega}', E'$, and all partial derivatives of $F_s(\mathbf{r}, \hat{\Omega}, E, \hat{\Omega}', E', t)$ must be equal.

Let $F_{0,s}(\mathbf{r}, \hat{\Omega}, E, \hat{\Omega}', E', t)$ be one specific function, there is $F_s(\mathbf{r}, \hat{\Omega}, E, \hat{\Omega}', E', t) = F_{0,s}(\mathbf{r}, \hat{\Omega}, E, \hat{\Omega}', E', t) + C(\hat{\Omega}', E')$, $\frac{\partial^2 C(\hat{\Omega}', E')}{\partial \hat{\Omega}' \partial E'} = 0$. Set

$$\begin{aligned} C(\hat{\Omega}', E') &= -F_{0,s}(\mathbf{r}, \hat{\Omega}, E, \hat{\Omega}_0, E', t) \\ &\quad -F_{0,s}(\mathbf{r}, \hat{\Omega}, E, \hat{\Omega}', E_0, t) \\ &\quad +F_{0,s}(\mathbf{r}, \hat{\Omega}, E, \hat{\Omega}_0, E_0, t), \end{aligned} \quad (\text{a4})$$

then a specific antiderivative is given by

$$\begin{aligned} F_{cs}(\mathbf{r}, \hat{\Omega}, E, \hat{\Omega}', E', t) &= F_{0,s}(\mathbf{r}, \hat{\Omega}, E, \hat{\Omega}', E', t) - F_{0,s}(\mathbf{r}, \hat{\Omega}, E, \hat{\Omega}_0, E', t) \\ &\quad - F_{0,s}(\mathbf{r}, \hat{\Omega}, E, \hat{\Omega}', E_0, t) + F_{0,s}(\mathbf{r}, \hat{\Omega}, E, \hat{\Omega}_0, E_0, t). \end{aligned} \quad (\text{a5})$$

When the conditions $(\hat{\Omega}' = \hat{\Omega}_0) \cup (E' = E_0)$ are satisfied, we have $F_{cs}(\mathbf{r}, \hat{\Omega}, E, \hat{\Omega}', E', t) = 0$. Therefore

$$\begin{aligned} &\frac{\partial^2 F_{cs}(\mathbf{r}, \hat{\Omega}, E, \hat{\Omega}', E', t)}{\partial \hat{\Omega}' \partial E'} \\ &= \frac{\partial^2 F_{0,s}(\mathbf{r}, \hat{\Omega}, E, \hat{\Omega}', E', t)}{\partial \hat{\Omega}' \partial E'} \\ &= \Sigma_s(\mathbf{r}, E') f(\mathbf{r}, \hat{\Omega}', E' \rightarrow \hat{\Omega}, E) \Psi(\mathbf{r}, \hat{\Omega}', E', t). \end{aligned} \quad (\text{a6})$$

Thus, we find a specific function $F_{cs}(\mathbf{r}, \hat{\Omega}, E, \hat{\Omega}', E', t)$ for the cluster of antiderivatives that satisfies the given conditions. Substituting into eq. (a2) yields:

$$\begin{aligned} Q_s &= \int_{E_0}^{E_1} \int_{\hat{\Omega}_0}^{\hat{\Omega}_1} \Sigma_s(\mathbf{r}, E') f(\mathbf{r}, \hat{\Omega}', E' \rightarrow \hat{\Omega}, E) \\ &\quad \Psi(\mathbf{r}, \hat{\Omega}', E', t) d\hat{\Omega}' dE' \\ &= F_{cs}(\mathbf{r}, \hat{\Omega}, E, \hat{\Omega}_1, E_1, t). \end{aligned} \quad (\text{a7})$$

A1.2 Antiderivative transformation for fission sources

The isotropic fission source in eq. (1) is given by

$$Q_f = \frac{\chi(E)}{4\pi} \iint \nu \Sigma_f(\mathbf{r}, E') \Psi(\mathbf{r}, \hat{\Omega}', E', t) d\hat{\Omega}' dE'. \quad (\text{a8})$$

Similarly, suppose the integrand $\nu \Sigma_f(\mathbf{r}, E')$ is a continuous function, then the integrand $\nu \Sigma_f(\mathbf{r}, E') \Psi(\mathbf{r}, \hat{\Omega}', E', t)$ is also continuous. Therefore, there exists an antiderivative $F_f(\mathbf{r}, \hat{\Omega}', E', t)$ for the fission source term in eq. (1), an antiderivative transformation of fission sources can be given:

$$\begin{aligned} Q_f &= \frac{\chi(E)}{4\pi} \int_{E_0}^{E_1} \int_{\hat{\Omega}_0}^{\hat{\Omega}_1} \nu \Sigma_f(\mathbf{r}, E') \Psi(\mathbf{r}, \hat{\Omega}', E', t) d\hat{\Omega}' dE' \\ &= \frac{\chi(E)}{4\pi} (F_f(\mathbf{r}, \hat{\Omega}_1, E_1, t) - F_f(\mathbf{r}, \hat{\Omega}_1, E_0, t) \\ &\quad - F_f(\mathbf{r}, \hat{\Omega}_0, E_1, t) + F_f(\mathbf{r}, \hat{\Omega}_0, E_0, t)). \end{aligned} \quad (\text{a9})$$

Moreover,

$$\frac{\partial^2 F_f(\mathbf{r}, \hat{\Omega}', E', t)}{\partial \hat{\Omega}' \partial E'} = \nu \Sigma_f(\mathbf{r}, E') \Psi(\mathbf{r}, \hat{\Omega}', E', t). \quad (\text{a10})$$

Similar to the analysis for the scattering source, $F_f(\mathbf{r}, \hat{\Omega}', E', t)$ belongs to a cluster of functions, from which a specific antiderivative $F_{c,f}(\mathbf{r}, \hat{\Omega}', E', t)$ can be found. Under the conditions $(\hat{\Omega}' = \hat{\Omega}_0) \cup (E' = E_0)$, we have $F_{c,f}(\mathbf{r}, \hat{\Omega}', E', t) = 0$, and the following relationship holds:

$$\frac{\partial^2 F_{c,f}(\mathbf{r}, \hat{\Omega}', E', t)}{\partial \hat{\Omega}' \partial E'} = \nu \Sigma_f(\mathbf{r}, E') \Psi(\mathbf{r}, \hat{\Omega}', E', t). \quad (\text{a11})$$

Substituting the constraint conditions into eq. (a9), we obtain:

$$\begin{aligned} Q_f &= \frac{\chi(E)}{4\pi} \int_{E_0}^{E_1} \int_{\hat{\Omega}_0}^{\hat{\Omega}_1} \nu \Sigma_f(\mathbf{r}, E') \Psi(\mathbf{r}, \hat{\Omega}', E', t) d\hat{\Omega}' dE' \\ &= \frac{\chi(E)}{4\pi} F_{c,f}(\mathbf{r}, \hat{\Omega}_1, E_1, t). \end{aligned} \quad (\text{a12})$$

A1.3 Exact differential form of the continuous energy neutron transport equation

Substituting eqs. (a7) and (a12) into eq. (1), we obtain the general exact differential form of the neutron transport equations:

$$\left\{ \begin{array}{l} \frac{1}{v} \frac{\partial \Psi(\mathbf{r}, \hat{\Omega}, E, t)}{\partial t} + \hat{\Omega} \cdot \nabla \Psi(\mathbf{r}, \hat{\Omega}, E, t) \\ + \Sigma_t(\mathbf{r}, E) \Psi(\mathbf{r}, \hat{\Omega}, E, t) \\ = F_{c,s}(\mathbf{r}, \hat{\Omega}, E, \hat{\Omega}_1, E_1, t) + \frac{\chi(E)}{4\pi} F_{c,f}(\mathbf{r}, \hat{\Omega}_1, E_1, t) \\ + Q_e, \\ \frac{\partial^2 F_{c,s}(\mathbf{r}, \hat{\Omega}, E, \hat{\Omega}', E', t)}{\partial \hat{\Omega}' \partial E'} \\ = \Sigma_s(\mathbf{r}, E') f(\mathbf{r}, \hat{\Omega}', E' \rightarrow \hat{\Omega}, E) \Psi(\mathbf{r}, \hat{\Omega}', E', t), \\ \frac{\partial^2 F_{c,f}(\mathbf{r}, \hat{\Omega}', E', t)}{\partial \hat{\Omega}' \partial E'} = v \Sigma_f(\mathbf{r}, E') \Psi(\mathbf{r}, \hat{\Omega}', E', t), \\ (\hat{\Omega}' = \hat{\Omega}_0) \cup (E' = E_0), F_{c,s}(\mathbf{r}, \hat{\Omega}, E, \hat{\Omega}', E', t) = 0, \\ F_{c,f}(\mathbf{r}, \hat{\Omega}', E', t) = 0. \end{array} \right. \quad (a13)$$

A2 Derivation of the multigroup neutron transport equation

A2.1 Derivation of the multigroup anisotropic scattering equation

To simplify the calculation, the anisotropic scattering source in eq. (2) is typically expanded using Legendre polynomials. The anisotropic scattering source for the multigroup form [1] can be expressed as follows:

$$\begin{aligned} Q_{s,g} &= \sum_{g'=1}^G \int \Sigma_{s,g' \rightarrow g}(\mathbf{r}, \hat{\Omega}' \rightarrow \hat{\Omega}) \Psi_{g'}(\mathbf{r}, \hat{\Omega}') d\hat{\Omega}' \\ &= \sum_{g'=1}^G \sum_{l=0}^{\infty} \Sigma_{s,l,g' \rightarrow g}(\mathbf{r}) \sum_{m=-l}^l Y_l^m(\hat{\Omega}) \Psi_{g'}^{l,m}. \end{aligned} \quad (a14)$$

Here, the neutron flux moment is

$$\Psi_{g'}^{l,m} = \int Y_l^m(\hat{\Omega}') \Psi_{g'}(\mathbf{r}, \hat{\Omega}') d\hat{\Omega}', \quad (a15)$$

and the spherical harmonic function is

$$Y_l^m(\hat{\Omega}) = \begin{cases} P_l^m(\cos \theta) \cos m\varphi, & \text{for } m = 0, 1, 2, \dots, l, \\ P_l^m(\cos \theta) \sin |m|\varphi, & \text{for } m = -1, -2, \dots, -l, \end{cases} \quad (a16)$$

where P_l^m is the associated Legendre polynomial. Its form is given as:

$$P_l^m(\mu_0) = \frac{(-1)^m}{2^l l!} (1 - \mu_0^2)^{m/2} \frac{d^{l+m}}{d\mu_0^{l+m}} (1 - \mu_0^2)^l, \quad (a17)$$

where θ and φ are the polar and azimuthal angles of the neutron motion direction, and μ_0 generally denotes the direction cosine between two different neutron motion directions. The detailed expression can be found in ref. [1].

For the multigroup neutron transport equation (2), we apply the antiderivative transformation to the neutron flux moment in the scattering term (a14). Since the angular variables $\hat{\Omega}$ and $\hat{\Omega}'$ are decoupled, the transformation is significantly simplified. Let $F_{s,g'}^{l,m}(\mathbf{r}, \hat{\Omega}')$ be the antiderivative of $Y_l^m(\hat{\Omega}') \Psi_{g'}(\mathbf{r}, \hat{\Omega}')$, as described in sect. A1.1, according to the Newton-Leibniz formula, we have

$$\begin{aligned} \Psi_{g'}^{l,m} &= \int_{\hat{\Omega}_0}^{\hat{\Omega}_1} Y_l^m(\hat{\Omega}') \Psi_{g'}(\mathbf{r}, \hat{\Omega}') d\hat{\Omega}' \\ &= F_{s,g'}^{l,m}(\mathbf{r}, \hat{\Omega}_1) - F_{s,g'}^{l,m}(\mathbf{r}, \hat{\Omega}_0). \end{aligned} \quad (a18)$$

Moreover,

$$\begin{aligned} F_{s,g'}^{l,m}(\mathbf{r}, \hat{\Omega}')' &= \frac{d}{d\hat{\Omega}'} \int_{\hat{\Omega}_0}^{\hat{\Omega}'} Y_l^m(\hat{\Omega}') \Psi_{g'}(\mathbf{r}, \hat{\Omega}') d\hat{\Omega}' \\ &= Y_l^m(\hat{\Omega}') \Psi_{g'}(\mathbf{r}, \hat{\Omega}'). \end{aligned} \quad (a19)$$

Similarly, there exists a specific function $F_{c,s,g'}^{l,m}$ such that $F_{c,s,g'}^{l,m}(\mathbf{r}, \hat{\Omega}_0) = 0$ when $\hat{\Omega}' = \hat{\Omega}_0$. Hence,

$$\Psi_{g'}^{l,m} = \int_{\hat{\Omega}_0}^{\hat{\Omega}_1} Y_l^m(\hat{\Omega}') \Psi_{g'}(\mathbf{r}, \hat{\Omega}') d\hat{\Omega}' = F_{c,s,g'}^{l,m}(\mathbf{r}, \hat{\Omega}_1). \quad (a20)$$

Substituting this into eq. (a14), we get

$$Q_{s,g} = \sum_{g'=1}^G \sum_{l=0}^{\infty} \Sigma_{s,l,g' \rightarrow g}(\mathbf{r}) \sum_{m=-l}^l Y_l^m(\hat{\Omega}) F_{c,s,g'}^{l,m}(\mathbf{r}, \hat{\Omega}_1). \quad (a21)$$

Similarly, for the fission source in the multigroup transport equation, we have

$$\int_{\hat{\Omega}_0}^{\hat{\Omega}_1} \Psi_{g'}(\mathbf{r}, \hat{\Omega}') d\hat{\Omega}' = F_{f,g'}(\mathbf{r}, \hat{\Omega}_1) - F_{f,g'}(\mathbf{r}, \hat{\Omega}_0), \quad (a22)$$

and there exists a specific function $F_{c,f,g'}(\mathbf{r}, \hat{\Omega}')$, under the conditions $\hat{\Omega}' = \hat{\Omega}_0$, $F_{c,f,g'}(\mathbf{r}, \hat{\Omega}') = 0$, and it satisfies the following equation:

$$F_{c,f,g'}^{l,m}(\mathbf{r}, \hat{\Omega}')' = \frac{dF_{c,f,g'}(\mathbf{r}, \hat{\Omega}')}{d\hat{\Omega}'} = \Psi_{g'}(\mathbf{r}, \hat{\Omega}'). \quad (a23)$$

Thus, we obtain

$$\left\{ \begin{array}{l} \hat{\Omega} \cdot \nabla \Psi_g(\mathbf{r}, \hat{\Omega}) + \Sigma_{t,g}(\mathbf{r}, \hat{\Omega}) \Psi_g(\mathbf{r}, \hat{\Omega}) \\ = \sum_{g'=1}^G \sum_{l=0}^{\infty} \sum_{s,l,g' \rightarrow g}(\mathbf{r}) \sum_{m=-l}^l Y_l^m(\hat{\Omega}) F_{c,s,g'}^{l,m}(\mathbf{r}, \hat{\Omega}_1) \\ + \frac{\chi_g}{4\pi k_{\text{eff}}} \sum_{g'=1}^G (\nu \Sigma_f(\mathbf{r}))_{g'} F_{c,f,g'}(\mathbf{r}, \hat{\Omega}_1), \\ \frac{dF_{c,s,g'}^{l,m}(\mathbf{r}, \hat{\Omega}')}{d\hat{\Omega}'} = Y_l^m(\hat{\Omega}') \Psi_{g'}(\mathbf{r}, \hat{\Omega}'), \\ \frac{dF_{c,f,g'}(\mathbf{r}, \hat{\Omega}')}{d\hat{\Omega}'} = \Psi_{g'}(\mathbf{r}, \hat{\Omega}'), \\ \hat{\Omega}' = \hat{\Omega}_0, F_{c,s,g'}(\mathbf{r}, \hat{\Omega}, \hat{\Omega}') = 0, F_{c,f,g'}(\mathbf{r}, \hat{\Omega}') = 0. \end{array} \right. \quad (\text{a24})$$

A2.2 Derivation of the multigroup isotropic scattering neutron transport equation

With isotropic scattering cross sections, the scattering source is independent of the angular variable and can be written as $\Sigma_{s,g' \rightarrow g}(\mathbf{r})$, reducing eq. (2) to

$$\begin{aligned} & \hat{\Omega} \cdot \nabla \Psi_g(\mathbf{r}, \hat{\Omega}) + \Sigma_{t,g}(\mathbf{r}, \hat{\Omega}) \Psi_g(\mathbf{r}, \hat{\Omega}) \\ &= \sum_{g'=1}^G \Sigma_{s,g' \rightarrow g}(\mathbf{r}) \int \Psi_{g'}(\mathbf{r}, \hat{\Omega}') d\hat{\Omega}' \\ &+ \frac{\chi_g}{4\pi k_{\text{eff}}} \sum_{g'=1}^G (\nu \Sigma_f(\mathbf{r}))_{g'} \int \Psi_{g'}(\mathbf{r}, \hat{\Omega}') d\hat{\Omega}' \\ &= \sum_{g'=1}^G \left(\Sigma_{s,g' \rightarrow g}(\mathbf{r}) + \frac{\chi_g}{4\pi k_{\text{eff}}} (\nu \Sigma_f(\mathbf{r}))_{g'} \right) \int \Psi_{g'}(\mathbf{r}, \hat{\Omega}') d\hat{\Omega}'. \end{aligned} \quad (\text{a25})$$

By transforming the integral term $\int \Psi_{g'}(\mathbf{r}, \hat{\Omega}') d\hat{\Omega}'$, we can obtain its specific antiderivative $F_{c,g'}(\mathbf{r}, \hat{\Omega}')$. When $\hat{\Omega}' = \hat{\Omega}_0$, $F_{c,g'}(\mathbf{r}, \hat{\Omega}_0) = 0$, and the corresponding exact differential form of the transport equation is given by

$$\begin{aligned} & \hat{\Omega} \cdot \nabla \Psi_g(\mathbf{r}, \hat{\Omega}) + \Sigma_{t,g}(\mathbf{r}, \hat{\Omega}) \Psi_g(\mathbf{r}, \hat{\Omega}) \\ &= \sum_{g'=1}^G \left(\Sigma_{s,g' \rightarrow g}(\mathbf{r}) + \frac{\chi_g}{4\pi k_{\text{eff}}} (\nu \Sigma_f(\mathbf{r}))_{g'} \right) F_{c,g'}(\mathbf{r}, \hat{\Omega}_1). \end{aligned} \quad (\text{a26})$$

Moreover, $\frac{dF_{c,g'}(\mathbf{r}, \hat{\Omega}')}{d\hat{\Omega}'} = \Psi_{g'}(\mathbf{r}, \hat{\Omega}')$. Replacing $\hat{\Omega}'$ with $\hat{\Omega}$ and g' with g , we have

$$F_{c,g}(\mathbf{r}, \hat{\Omega})' = \frac{dF_{c,g}(\mathbf{r}, \hat{\Omega})}{d\hat{\Omega}} = \Psi_g(\mathbf{r}, \hat{\Omega}). \quad (\text{a27})$$

If the angular is two-dimensional variable, then

$$F_{c,g}(\mathbf{r}, \hat{\Omega})' = \frac{\partial^2 F_{c,g}(\mathbf{r}, \mu, \varphi)}{\partial \mu \partial \varphi} = \Psi_g(\mathbf{r}, \mu, \varphi). \quad (\text{a28})$$

Substituting into eq. (a24), we obtain

$$\left\{ \begin{array}{l} \hat{\Omega} \cdot \nabla F_{c,g}(\mathbf{r}, \hat{\Omega})' + \Sigma_{t,g}(\mathbf{r}, \hat{\Omega}) F_{c,g}(\mathbf{r}, \hat{\Omega})' \\ = dl \sum_{g'=1}^G \left(\Sigma_{s,g' \rightarrow g}(\mathbf{r}) + \frac{\chi_g}{4\pi k_{\text{eff}}} (\nu \Sigma_f(\mathbf{r}))_{g'} \right) F_{c,g'}(\mathbf{r}, \hat{\Omega}_1), \\ \hat{\Omega}' = \hat{\Omega}_0, F_{c,g'}(\mathbf{r}, \hat{\Omega}_0) = 0, g' = 1, 2, 3, \dots \end{array} \right. \quad (\text{a29})$$

A3 PINN architecture for solving single-group slab and cylindrical transport problems

Figures A11 and A12 illustrate the PINN architectures for solving single-group slab and cylindrical geometry transport problems, respectively. Both architectures employ fully connected neural networks with 8 hidden layers, each containing 16 neurons and utilizing the tanh activation function. The network outputs, denoted as N_Ψ and N_F , correspond to the solutions of the IDNT and EDNT, respectively. The whole loss function \mathcal{L} comprises the residual loss \mathcal{L}_r of transport equation at N_r collocation points set, boundary condition loss \mathcal{L}_b at N_b collocation points set, eigenvalue constraint loss \mathcal{L}_e at N_e collocation points set, and fixed solution constraint condition loss \mathcal{L}_f at N_f collocation points set. During training, the loss function \mathcal{L} is minimized using the Adam optimizer via backpropagation.

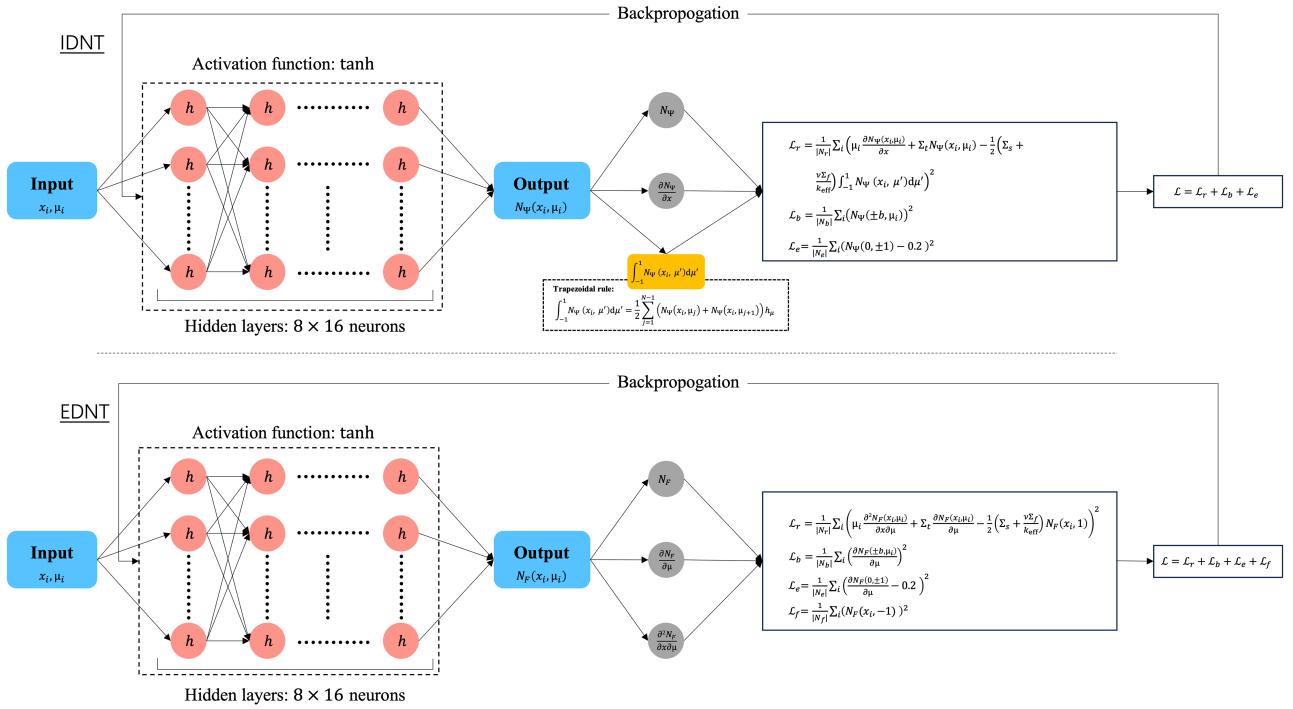


Figure A11 (Color online) PINN architecture for solving the critical single-group slab geometry problem. IDNT (top) and EDNT (bottom).

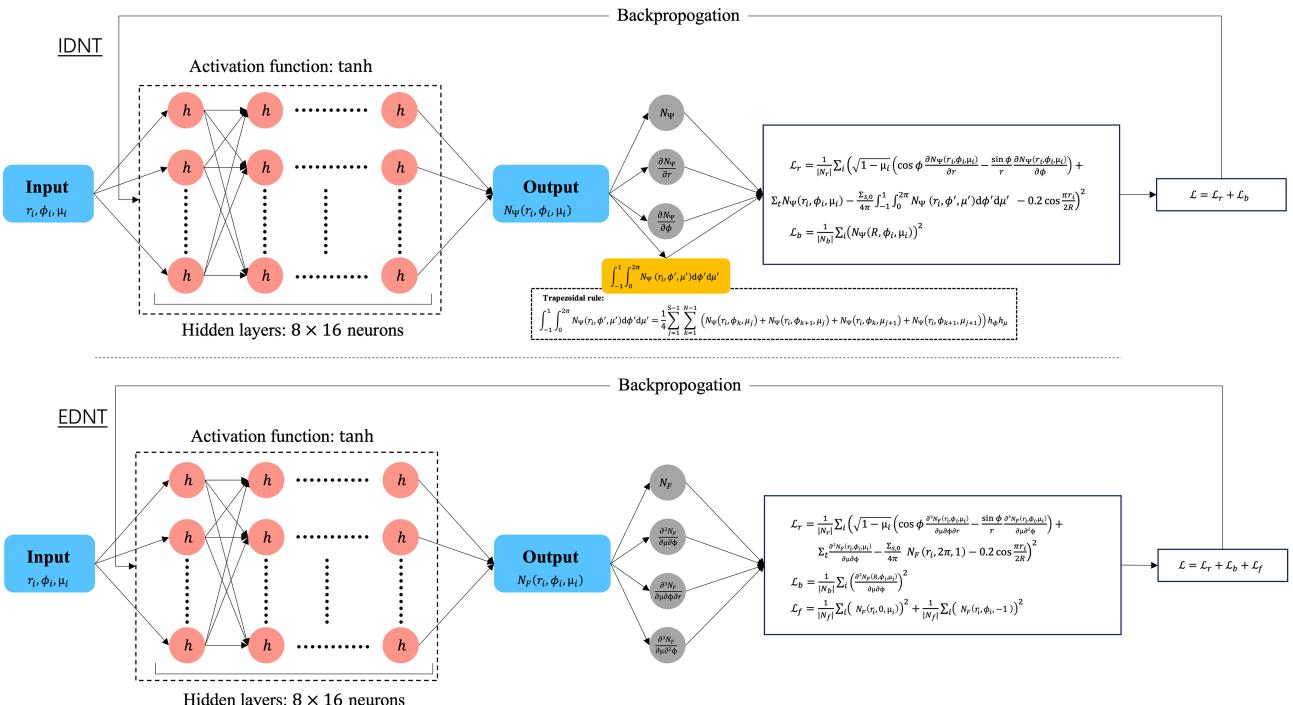


Figure A12 (Color online) PINN architecture for the single-group cylindrical geometry transport problem with fixed source. IDNT (top) and EDNT (bottom).