



Application research of a hybrid data- and knowledge-driven artificial intelligence scientific computing model in neutron diffusion calculation for nuclear reactors

Fu-Lin Zeng¹ · Xiao-Long Zhang¹ · Peng-Cheng Zhao¹ · Zi-Jing Liu¹

Received: 26 July 2024 / Revised: 4 November 2024 / Accepted: 15 November 2024 / Published online: 3 January 2026

© The Author(s), under exclusive licence to China Science Publishing & Media Ltd. (Science Press), Shanghai Institute of Applied Physics, the Chinese Academy of Sciences, Chinese Nuclear Society 2025

Abstract

Amidst the growing global emphasis on nuclear safety, the integrity of nuclear reactor systems has garnered attention in the aftermath of consequential events. Moreover, the rapid development of artificial intelligence technology has provided immense opportunities to enhance the safety and economy of nuclear energy. However, data-driven deep learning techniques often lack interpretability, which hinders their applicability in the nuclear energy sector. To address this problem, this study proposes a hybrid data-driven and knowledge-driven artificial intelligence model based on physics-informed neural networks to accurately compute the neutron flux distribution inside a nuclear reactor core. Innovative techniques, such as regional decomposition, intelligent k_{eff} (effective multiplication factor) search, and k_{eff} inversion, have been introduced for the calculation. Furthermore, hyperparameters of the model are automatically optimized using a whale optimization algorithm. A series of computational examples are used to validate the proposed model, demonstrating its applicability, generality, and high accuracy in calculating the neutron flux within the nuclear reactor. The model offers a dependable strategy for computing the neutron flux distribution in nuclear reactors for advanced simulation techniques in the future, including reactor digital twinning. This approach is data-light, requires little to no training data, and still delivers remarkably precise output data.

Keywords Neutron diffusion equation · Physics informed neural network · Effective multiplication factor · Whale optimization algorithm

1 Introduction

Neutron flux distribution, which governs the performance and safety of nuclear reactors, delineates the spatial and temporal variations in neutron flux and neutron spectrum within the reactor core [1]. This distribution profoundly influences power generation, fuel consumption [2], reactivity control [3–5], and radiation damage to reactor components. Therefore, precise and efficient calculation of neutron flux distribution is essential for reactor design, analysis, and operation. However, calculating the neutron flux distribution is challenging due to the intricate and nonlinear nature of the neutron transport equation governing neutron behavior in

reactors [6]. The computational demands for solving this equation, especially for large-scale and high-fidelity models, necessitate substantial computational resources and time [7].

For neutron flux distribution calculations, neutron diffusion theory has emerged as a simplified neutron transport model that assumes isotropic neutron scattering and a constant energy spectrum within each energy group. Although this theory is adept at solving the neutron balance equation that links neutron flux to production and absorption rates in the reactor core, applying it to the entire reactor core is a complex and computationally intensive task [8, 9]. Numerous numerical techniques and approximations, such as nodal methods [10, 11], finite element methods [12, 13], and finite difference methods [14], have been developed to mitigate this complexity by discretizing the reactor core into smaller nodes. However, real-time neutron diffusion simulation remains resource-intensive and challenging for applications requiring rapid feedback to reactor operators or control systems. Real-time simulations demand temporal efficiency

✉ Peng-Cheng Zhao
zhaopengcheng1030@163.com

¹ School of Nuclear Science and Technology, University of South China, Hengyang 421001, China

(results must be generated within seconds or minutes) as well as high accuracy and reliability to ensure the validity and utility of the outcomes.

Online sensors can be used in reactors to simulate real-time core neutron diffusion [15]. They are capable of measuring various physical quantities within the reactor core, including temperature, pressure, flow rate, and neutron flux, and offer real-time data that can be employed as inputs or boundary conditions for neutron diffusion simulations. These sensors can facilitate real-time verification of simulation results by comparing the simulated and measured neutron fluxes, serving as a correction dataset for machine learning calculations. The neutron flux measurement systems employed in conventional second-generation and second-generation plus nuclear reactor cores are characterized by the periodic insertion of neutron detectors from the bottom of the pressure vessel to ascertain the current status of the reactor core. Regrettably, this methodology can only be used to perform intermittent measurements and requires perforations at the bottom of the pressure vessel [1]. Contrastingly, third-generation nuclear reactors, such as the HPR1000, employ a fundamentally distinct approach for performing neutron flux measurements [16]. They are principally designed around a measurement scheme centered around self-powered neutron detectors (SPNDs). The SPND-based system facilitates real-time acquisition of neutron flux distribution within the reactor core, presenting a marked departure from the sporadic measurements obtained using the second-generation reactors, which enable the transition of periodic and intermittent measurement practices to a continuous and comprehensive monitoring paradigm.

However, due to the limitations of the existing reactor monitoring system in capturing transient changes within the reactor, it is necessary to reconstruct the key parameters of the entire reactor based on limited data. Despite these computational methods being relatively mature, effectively integrating measured data into computational results is challenging, particularly for high-fidelity interactive nuclear reactor simulation techniques, such as nuclear reactor digital twins [17–19]. Recent scholarly endeavors within the field of nuclear reactor physics analysis have focused on the physics-informed neural network (PINN) algorithm, which integrates physical models with limited training data to establish models that adhere to physical constraints. As a hybrid data- and knowledge-driven model, PINN leverages data as well as the physical laws governing neutron flux distribution. The neural network in a PINN represents the solution to the neutron transport equation, which is trained by minimizing a loss function comprising residuals of the neutron transport equation, initial and boundary conditions, and observed data [20]. The advantages of a PINN over conventional methods and data-driven models include the incorporation of physical knowledge, utilization of data for additional constraints,

and the ability to solve inverse problems such as parameter estimation, model discovery, and optimal control [21].

Presently, data-driven deep learning models pose challenges related to interpretability and are often perceived as “black box models” [22–26]. In addition to harnessing existing measurement data to propel neural networks, incorporating well-established and reliable physical models expressed through partial differential equations is an alternative approach [27–31]. For example, the heat conduction equation and Navier–Stokes equation are used together to study thermal hydraulics of nuclear reactors [32], while the neutron transport equation and burnup equation are used for physics analysis [33]. Relying solely on either data or physical models may cause loss of valuable information. Recently, Houde et al. [34] juxtaposed data-driven and model-driven approaches to reveal that data-driven methods exhibit superior speed and accuracy, while model-driven methods excel in interpretability. The researchers advocate for a hybrid methodology that combines the strengths of both physical and data-driven models to improve the accuracy, reliability, and proficiency of existing models for prediction purposes. The primary objective of this study is to present a neural network model tailored for computing the neutron flux within the core of a nuclear reactor, thereby facilitating the reconstruction of neutron flux distribution. Engineered to assimilate physical laws, this neural network model integrates both model-driven and data-driven approaches, endowing the AI model with commendable accuracy and noticeable interpretability.

In recent years, methods such as PINN have received the attention of many scholars in the field of reactor physics [35–39]. Notably, Dong et al. [40] achieved outstanding precision by employing the PINN algorithm to solve the multidimensional neutron diffusion equation. Similarly, Jiangyu et al. [41] introduced a conservative PINN (CPINN) to address non-smoothness in the neutron diffusion equation in non-uniform media. Yu et al. [42, 43] proposed a data-driven PINN (DEPINN) to enhance the accuracy and efficiency associated with solving the neutron diffusion eigenvalue problem with limited prior experimental data. Prantikos et al. [44, 45] utilized PINN to solve the neutron diffusion equation in point kinetics, underscoring its advantages over traditional numerical methods for addressing irregular and high-dimensional problems.

Employing AI algorithms for nuclear reactor physics analyses is challenging with regard to accuracy and hyperparameters [46, 47]. Although accuracy is at par or lower compared to that of traditional numerical methods, hyperparameters are typically set using universal or empirical values that are rationalized through sensitivity analyses. Accordingly, this study pioneers an AI-based computing model tailored to the neutron diffusion equation. Grounded in the hp-variational PINN (VPINN) deep machine learning

technology [48], the model incorporates domain decomposition functions and various innovative technologies. Optimization of neural network hyperparameters is facilitated through the whale optimization algorithm (WOA) [49]. The model adeptly solves the neutron diffusion equation in single-energy single-medium, single-energy multi-medium, and two-energy group multi-medium scenarios, thus validating its versatile capabilities. Overall, this research introduces novel ideas and techniques by harnessing deep learning technology to facilitate intelligent solution of the neutron diffusion equation.

2 Neutron diffusion model

The neutron diffusion model is popularly used for nuclear reactor physics analyses and offers insights into the intricate behavior of neutrons within the reactor core. This model also meticulously considers various neutron interactions, including absorption, scattering, production, and transport. The model output facilitates the computing of the neutron flux and power distributions within the reactor core. The efficacy of the model for large-scale reactor calculations is underscored by its rapid computation speed and high efficiency.

At the heart of the neutron diffusion model lies the neutron transport equation, a framework that assumes isotropic neutron scattering angles to eliminate angular variables from the equation and thus streamline subsequent analyses. Consequently, the discretization and averaging of the continuous-energy spectrum ensue, culminating in the derivation of the neutron diffusion equation tailored for multi-energy group configurations. This equation is vital for understanding the neutron behavior within the reactor. In steady-state scenarios, the focal point invariably shifts toward the critical state of the nuclear reactor, a condition during which the stability and operational viability of the reactor must be studied. However, it is imperative to acknowledge that not all neutron reactions within the reactor attain criticality. Attaining critical conditions is contingent upon several factors, including reactor design, fuel composition, and control mechanisms. This observation underscores the intricate dynamics and multifaceted nature of factors influencing criticality, thus adding layers of complexity to our comprehension of reactor dynamics and safety considerations.

To refine the neutron diffusion equation for practical utility, attention is directed toward steady-state situations. The elimination of have garnered attention. Eliminating the time-dependent nonstationary term from the equation unveils a pivotal parameter known as the effective multiplication factor (k_{eff}). This factor assumes, which plays a central pivotal role in maintaining equilibrium within the equation, serving and serves as a yardstick an indicator to ascertain whether the reaction is critical. In the nuanced spectrum of

possibilities, a subcritical state emerges when k_{eff} is less than 1, a supercritical state manifests when k_{eff} exceeds 1, and criticality prevails when k_{eff} attains unity. The equation then becomes a eigenvalue equation is represented as follows:

$$-D_g \nabla^2 \phi_g + \Sigma_{r,g} \phi_g = \sum_{g'=1}^G (\Sigma_{s,g' \rightarrow g} \phi_{g'}) + \frac{\chi_g}{k_{\text{eff}}} \sum_{g'=1}^G (\nu_{g'} \Sigma_{f,g'} \phi_{g'}) + S_g, \quad (1)$$

where ϕ_g represents the neutron flux of the g th energy group. D_g represents the diffusion coefficient of the neutron in the g th energy group. $\Sigma_{r,g}$ is the removal cross section of the neutron in the g th energy group, which includes the absorption cross section and the scattering cross section from other energy groups. ν_g represents the number of neutrons produced by fission in the g th energy group, and S_g represents the external neutron source in the g th energy group.

Moreover, in the presence of monoenergetic energy and absence of an additional neutron source, the transient and steady-state neutron diffusion equations have been outlined in Eqs. (2) and (3), respectively.

$$\frac{1}{Dv} \frac{\partial \phi(r, t)}{\partial t} = \nabla^2 \phi(r, t) + \frac{k_{\infty} - 1}{L^2} \phi(r, t), \quad (2)$$

$$\nabla^2 \phi(r) + \frac{k_{\infty}/k_{\text{eff}} - 1}{L^2} \phi(r) = 0, \quad (3)$$

where L is the diffusion length, k_{∞} is the infinite multiplication factor, and v is the neutron speed. These equations can be categorized as second-order partial differential equations in the following unified form:

$$F[\mathbf{x}, \phi(\mathbf{x}), \nabla \phi(\mathbf{x}), \nabla^2 \phi(\mathbf{x})] = 0, \mathbf{x} \in \Omega. \quad (4)$$

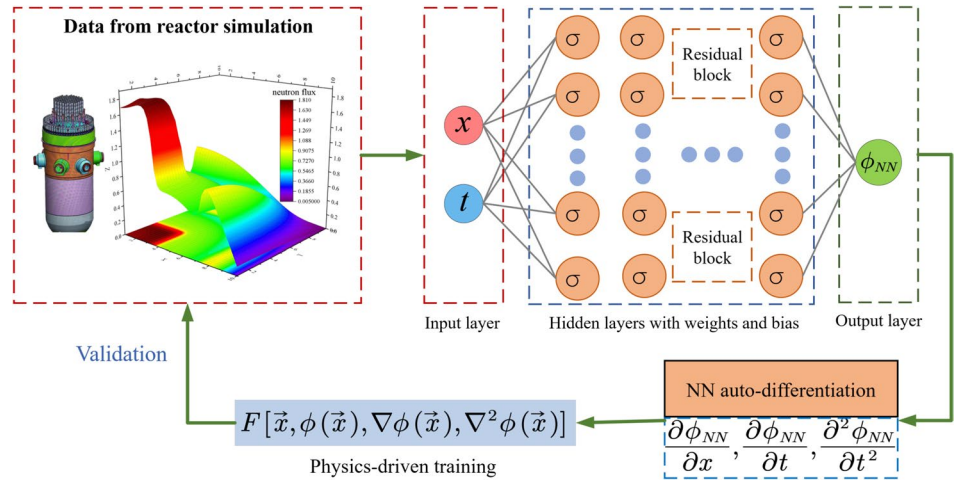
3 Framework of the hp-VPINN model for solving the neutron diffusion equation

3.1 Basic principle of PINN

PINNs are an advanced computational paradigm used to synthesize neural networks with fundamental physical principles to infer unknown physical fields. They incorporate partial differential equations (PDEs) that govern physical systems directly into the loss function, ensuring that the neural network inherently adheres to the prescribed physical laws and adeptly solves the associated PDEs. The intricate calculation flow is shown in Fig. 1 and provides a comprehensive representation of the PINN methodology.

Implementing a PINN involves the utilization of spatial and temporal coordinates as pivotal inputs for a neural

Fig. 1 (Color online) PINN calculation flowchart



network. These coordinates undergo a meticulous process of nonlinear mapping across multiple hidden layers, which results in the derivation of a continuous function denoted as ϕ_{NN} ; ϕ_{NN} represents the output of the neural network and provides solutions for the underlying physical system. Notably, ϕ_{NN} not only adheres to the stipulated physical constraints dictated by the PDEs, but also aligns seamlessly with any available experimental measurements. The intricate processing of spatial and temporal parameters through multiple layers underscores the nuanced nature of information transformation within the neural network.

To ascertain the fulfillment of physical constraints, PINNs employ automatic differentiation techniques that involve the precise computation of derivatives inherent in the PDEs to reveal the residual of the PDEs. Concurrently, the neural network assesses its congruence with experimental data by computing the mean squared error between the predicted output (ϕ_{NN}) and empirically measured data. This dual-pronged evaluation comprehensively assesses the fidelity of the PINN model to both theoretical constraints and observed experimental data. The comprehensive PINN model orchestrates the construction of a nuanced loss function and integrates various components such as the weighted sum of PDE residuals and the mean squared error relative to experimental data. This intricately constructed loss function is minimized through the iterative training of the neural network, a process facilitated by the gradient descent algorithm. Consequently, the trained PINN model surfaces as a potent solver, adeptly addressing the specified PDEs.

In addition to traditional numerical methodologies such as finite element or finite difference methods, PINN offers distinct advantages for solving nonlinear PDEs due to its inherent ability to circumvent the necessity for explicit spatial and temporal discretization. Furthermore, owing to the highly nonlinear nature of neural networks, PINN shows enhanced capabilities in navigating and effectively addressing PDEs exhibiting

intricate physical phenomena, such as fluid flow [50] and simulation of time-domain electromagnetic fields [51].

After considering the neutron diffusion Eq. (4) over the computational domain Ω with a boundary denoted as $\partial\Omega$, we utilized Eq. (5) as the loss function in the PINN framework.

$$loss = w \cdot loss_f + w_b \cdot loss_b + w_d \cdot loss_d, \tag{5}$$

where $loss_f$, $loss_b$, and $loss_d$ are the loss terms associated with PDEs, boundary condition, and data, respectively; these parameters $\{w, w_b, w_d\}$ denote the weight coefficients in the loss function. They may be user-specified or tuned manually or automatically (e.g., they can be modified based on the numerical experiment in each problem); their optimal bound, however, is still an issue [52]. $loss_f$, $loss_b$, and $loss_d$ are defined as follows:

$$loss_f = \frac{1}{N_f} \sum_{i=1}^{N_f} r(\mathbf{x}_f^i)^2, \tag{6}$$

$$loss_b = \frac{1}{N_b} \sum_{i=1}^{N_b} r(\mathbf{x}_b^i)^2, \tag{7}$$

$$loss_d = \frac{1}{N_d} \sum_{i=1}^{N_d} r(\mathbf{x}_d^i)^2, \tag{8}$$

where N_f and N_b are the collocation points in their domains, while N_d is the number of experimental data points that are used; the residuals are given in Eq. (9–11):

$$r(\mathbf{x}_f) = F \left[\mathbf{x}_f, \phi_{NN}(\mathbf{x}_f), \nabla \phi_{NN}(\mathbf{x}_f), \nabla^2 \phi_{NN}(\mathbf{x}_f) \right], \forall \mathbf{x}_f \in \Omega, \tag{9}$$

$$r(\mathbf{x}_b) = \phi_{NN}(\mathbf{x}_b) - \phi(\mathbf{x}_b), \forall \mathbf{x}_b \in \partial\Omega, \tag{10}$$

$$r(\mathbf{x}_d) = \phi_{NN}(\mathbf{x}_d) - \phi(\mathbf{x}_d). \tag{11}$$

3.2 Basic principle of VPINN

Figure 2 illustrates the procedural intricacies encapsulated in the calculation flowchart of the VPINN, an advanced extension of the PINN framework. VPINN emanates from the assimilation of the Petrov-Galerkin method [53] into the foundational structure of PINN, marking a pivotal evolution in the computational architecture. This modification is based on the variational principle, a profound mathematical concept, which refines the neural network output by imposing an additional constraint layer. This principled refinement transforms the constraints embedded in the physical equations into a systematic process of minimizing a functional, thus representing a pivotal advancement in solution accuracy. The novelty of the VPINN lies in its ability to conform to the intricate constraints inherent in physical equations. The variational principle orchestrates the nuanced minimization process of a functional, thereby facilitating a more precise convergence toward solutions that faithfully adhere to the constraints. The optimization of this functional stands as a strategic cornerstone within the VPINN methodology, providing a method for refining the neural network output and enhancing the solution accuracy.

VPINN differs from the conventional Galerkin method due to its inherent flexibility in sourcing trial and weight functions from disparate function spaces. In the VPINN paradigm, an artificial neural network (denoted as ϕ_{NN}) assumes the role of the trial function and increases the network capacity to discern complex relationships within the dataset. The weight function within VPINN can manifest in various forms, including polynomial functions, trigonometric functions, Dirac delta functions, or even another neural network. This inherent versatility positions VPINN as an adaptive framework capable of addressing diverse physical phenomena with distinct characteristics. Therefore,

VPINN, which represents a harmonious fusion of sophisticated mathematical principles and neural network architecture, can solve intricate physical problems. Innovating the variational principle elevates the adaptability of the neural network, leading to improved accuracy and robustness for approximating solutions to complex physical equations. By choosing an appropriate weight function v , multiplying it with Eq. (4), and integrating the result over the entire solution domain, we obtain the variational residual:

$$R = \sum_i^n \int_{\Omega} F[\mathbf{x}_f, \phi_{NN}(\mathbf{x}_f), \nabla \phi_{NN}(\mathbf{x}_f), \nabla^2 \phi_{NN}(\mathbf{x}_f)] \cdot v_i dx, \tag{12}$$

$$v_n = P_n(x) - P_{n+2}(x), n = (0, 1, 2, \dots, K - 1). \tag{13}$$

In this study, Legendre polynomials have been used to construct a K -dimensional weight function space. The value of each order weight function at the boundary is zero, which ensures that the output results satisfy the boundary conditions. The loss function is defined as follows:

$$loss = w_r \cdot loss_r + w_b \cdot loss_b, \tag{14}$$

where

$$loss_r = \frac{1}{K} \sum_{i=0}^{K-1} |r_i|^2. \tag{15}$$

3.3 Basic principle of hp-VPINN

hp-VPINN, an evolution of VPINN, represents a nuanced augmentation that amalgamates the subdomain Petrov-Galerkin method with a discretization approach tailored for the solution domain. This modification facilitates the decomposition of the solution domain, thereby increasing the potential for h-refinement; it also introduces a pivotal

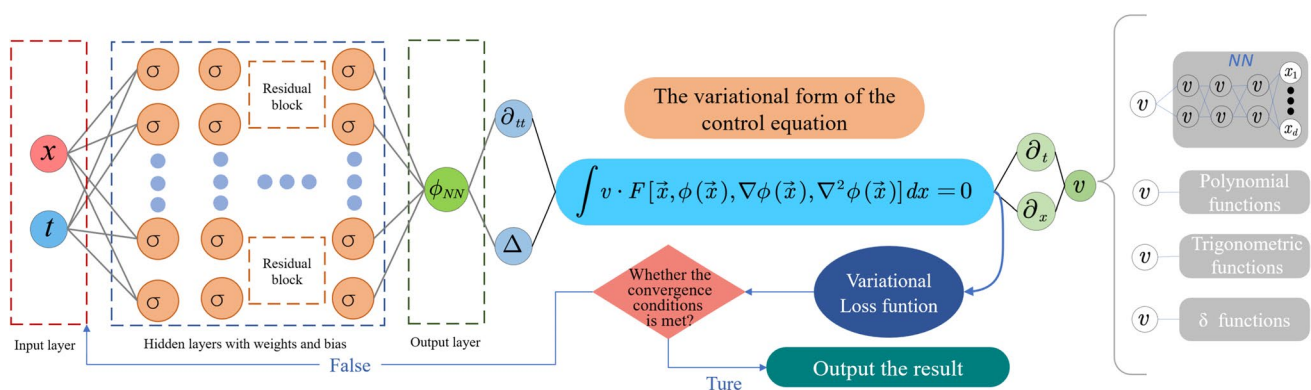


Fig. 2 (Color online) VPINN calculation flowchart

projection of the variational residual onto a p -dimensional space characterized by high-order polynomials. This strategy contributes to a more efficacious handling of partial differential equations, thereby substantively enhancing the precision of the resultant solutions. The intrinsic virtue of hp-VPINN lies in its ability to partition the solution domain into distinct subdomains, thereby accommodating h -refinement—a strategic approach to enhance the solution granularity. Additionally, the projection onto a high-dimensional polynomial space empowers the model to capture intricate nuances within the solution, which is particularly beneficial in scenarios demanding high-order accuracy. This deliberate interplay of h -refinement and p -dimensional projection corroborates the method's capacity for addressing a spectrum of physical problems with heightened accuracy.

In addition to its foundational advancements, hp-VPINN introduces a parallel computing dimension to its framework. By leveraging parallel computing techniques on the decomposed subdomains, hp-VPINN orchestrates a concurrent computation process that optimizes computational efficiency. This parallelization feature is particularly advantageous when addressing complex physical problems as it expedites the solution process through the simultaneous computation of multiple subdomains. Furthermore, hp-VPINN incorporates a weak integration formula to handle second-order PDEs within the loss term of the equation. By leveraging the integration by parts formula, this approach yields a refined formula that reduces the order of differentiation required for the neural network. Consequently, this reduction enhances the flexibility of the neural network during the training phase, rendering it more adaptable to diverse training scenarios.

The loss function of hp-VPINN can be derived from the loss function of VPINN. If the solution domain is divided into N subdomains, then the loss function of hp-VPINN is as follows:

$$\text{loss} = \text{loss}_R + w \cdot \text{loss}_b, \quad (16)$$

where

$$\text{loss}_R = \frac{1}{N \cdot (K-1)} \sum_{i=1}^N \sum_{j=0}^{K-1} \left[\int_{\Omega_i} v_{ij} \cdot F_{NN} dx \right]^2, \quad (17)$$

v_{ij} represents the j -th order weight function on the i -th subdomain Ω_i , which is a piecewise function that assumes a non-zero value only on its corresponding Ω_i and is equal to zero everywhere else. The steps will be demonstrated in the following example.

4 Innovative key technology proposed in the hp-VPINN framework for solving the neutron diffusion equation

4.1 Technology of parallel computing and convergence in multiple regions

To more effectively control and regulate nuclear reactors, with a paramount emphasis on augmenting the safety of nuclear power generation, a strategic paradigm that involves the meticulous partitioning of the actual core within a nuclear reactor is often employed. This conventional approach divides the core into distinct zones, namely: breeding zone and blanket zone; each of these zones are made of different materials according to their specific functions. Consequently, a discernible variance arises in the parameters of the materials deployed in these zones, which include critical attributes such as absorption cross section and fission cross section. These parameters play a pivotal role in formulating neutron diffusion equations, thus demanding nuanced considerations within their respective zones. The distinct characteristics of the breeding and blanket zones necessitate a tailored approach for delineating the parameters governing neutron diffusion. This intricacy engenders a spatially partitioned solution for comprehensively capturing the diverse dynamics manifesting within the core. The multifaceted nature of these nuclear processes, characterized by variations in materials and associated parameters, underscores the indispensability of a sophisticated computational framework capable of accommodating such intricacies, e.g., the hp-VPINN framework. This method enables the effective dissection of the solution domain and provides the model the capacity to discern and address the distinct characteristics inherent in each subdomain. Notably, the framework leverages the corresponding loss functions tailored to the unique attributes of individual subdomains, thus facilitating targeted and specialized training. This bespoke training regimen ensures that the model encapsulates the intricacies specific to each zone, crucial for comprehensively understanding the diverse nuclear processes occurring within the core.

An inherent advantage of the hp-VPINN framework lies in its ability to orchestrate multi-domain parallel convergence. Following the training phase within each subdomain, the framework amalgamates the individual loss functions into a unified computational scheme. This synergistic approach yields a model that has the capacity to seamlessly integrate the insights gained from different subdomains, thereby achieving a holistic understanding of the entire core. The culmination of these efforts translates into an advanced computational tool capable of addressing the challenges posed by the partitioned nature

of nuclear reactor cores, thereby contributing significantly to the advancement of nuclear reactor control and safety protocols.

4.2 Adaptive subdomain partitioning scheme

To study physical fields characterized by substantial gradients, employing multiple nonlinear neural networks is a viable strategy to furnish a more nuanced depiction of regions undergoing significant changes. Analogous to the meshing densification in specific areas of interest within finite element algorithms, this methodology aims to enhance the precision of computed outcomes in regions marked by pronounced alterations. It is noteworthy that this technique, while promising heightened accuracy, comes at the cost of increased computational demands and necessitates judicious resource management. To address the computational challenges posed by the utilization of multiple neural networks, an adaptive subdomain partitioning scheme has emerged as a prospective solution. This approach selectively refines the mesh in regions exhibiting substantial gradients within the physical field under consideration. By targeting areas showing pronounced change, the adaptive subdomain partitioning strikes an intricate equilibrium between computational resource allocation and accuracy. This judicious utilization of mesh refinement in gradient-rich zones ensures that computational resources are allocated where they are most needed, while the solution accuracy is optimized without incurring unnecessary computational overhead.

4.3 Intelligent k_{eff} search and inversion techniques

To perform numerical computations concerning reactor kinetics, source iteration methods are ideal for attaining approximate solutions. Nevertheless, when neural networks are introduced as an inner loop within a source iteration paradigm, iteratively training the network diminishes the efficiency of the solution process significantly. Thus, we assume that the transient diffusion equation will eventually reach a stable critical state and divide k_{∞} by a k_{eff} to make the adjustment to make the reaction critical, thus allowing the transient diffusion [Eq. (2)] to be written in the format of the steady-state diffusion [Eq. (3)], which is shown in Eq. (18), which is equivalent to Eq. (3) when the equation reaches a steady state, wherein the effective multiplication factor k_{eff} plays a central role.

$$\frac{1}{Dv} \frac{\partial \phi(r, t)}{\partial t} = \nabla^2 \phi(r, t) + \frac{k_{\infty}/k_{\text{eff}} - 1}{L^2} \phi(r, t) \quad (18)$$

As the nuclear reactor undergoes operational phases over a defined period denoted as τ , reaching a stable state becomes an inherent outcome. At this juncture, the neutron flux ϕ stabilizes and ceases to exhibit temporal variations. Equation (19) encapsulates this stabilized state, providing a

mathematical representation of the equilibrium achieved in the neutron flux within the reactor. This moment of post-operation stability (τ) is crucial for comprehensively understanding the reactor behavior and forms a foundational aspect of reactor kinetics analysis.

In essence, the path toward criticality in a nuclear reactor involves navigating through the intricacies of eigenvalue problems, with the effective multiplication factor serving as a key determinant. Although the introduction of neural networks into the solution methodology is promising, it necessitates careful consideration due to its impact on computational efficiency. The proposed intelligent search algorithm aims to enhance the computational efficacy, particularly in the context of solving the transient neutron diffusion equation, thereby providing a more robust understanding of nuclear reactor dynamics.

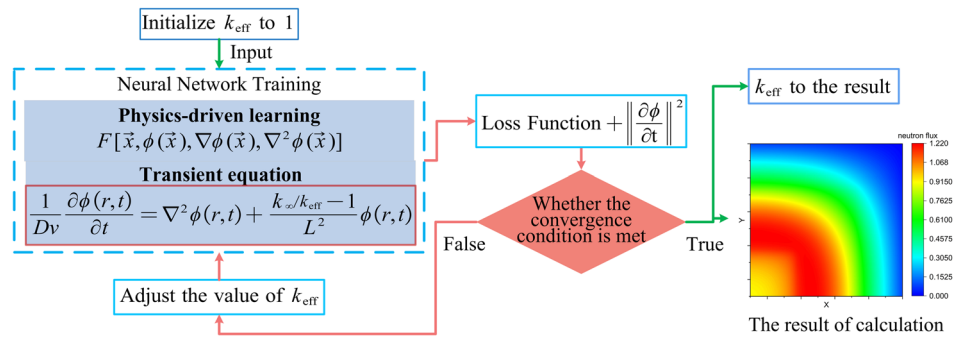
$$\phi_t(r, \tau) = 0 \quad (19)$$

To seamlessly integrate Eq. (19) into the loss function, we adopt a methodology akin to that employed for the boundary condition loss term in Eq. (10). The effective multiplication factor, k_{eff} , is determined through an astute evaluation of ϕ_t when t surpasses the operational period. τ During the iterative process, k_{eff} emerges as a dynamic variable within the neural network, subject to continuous adjustments. Notably, the neutron diffusion equation, characterized by multiple eigenvalues, designates k_{eff} as the eigenvalue that is typically close to unity. Consequently, k_{eff} is initially set to 1 within the neural network, orchestrating a focused search for the eigenvalue nearest to 1. The determination of k_{eff} is based on the evaluation of ϕ_t . In scenarios where ϕ_t is positive (reactor is in a supercritical state), the backpropagation algorithm inherent to the neural network automatically instigates a downward adjustment of k_{eff} . Conversely, when ϕ_t is negative (signifying a subcritical state), the adaptive mechanism directs an upward adjustment of k_{eff} . The culmination of this dynamic adjustment process aligns with the reactor's transition to a critical state, which is observed when ϕ_t approaches an approximate equilibrium around 0. At this juncture, k_{eff} manifests as the sought-after value since it encapsulates the reactor's criticality with precision.

The neural network's capacity to dynamically adapt in response to evolving reactor states represents a computational strategy attuned to the intricacies of nuclear reactor dynamics. Through these meticulously orchestrated adjustments, the neural network effectively navigates the multi-eigenvalue landscape of the neutron diffusion equation, precisely determines k_{eff} . During the criticality phase, and thus advances the computational understanding of nuclear reactor behavior. The calculation process is shown in Fig. 3.

To comprehend and control the dynamics of a nuclear reactor, intricacies of the steady-state neutron diffusion

Fig. 3 (Color online) k_{eff} intelligent search algorithm flowchart



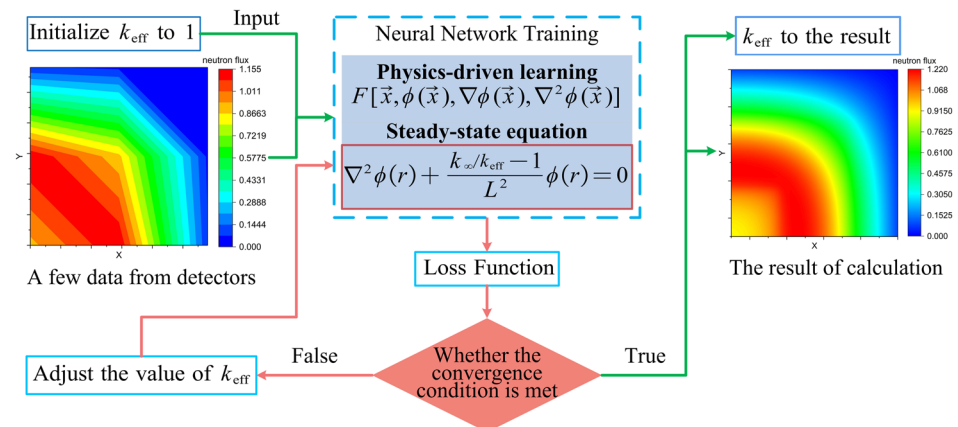
equation must be known [Eq. (3)]. In our pursuit of enhanced comprehension and predictive capabilities, we have proposed an inverse technique to determine the hitherto unknown parameter k_{eff} by training a neural network with a limited dataset. By employing sophisticated data-driven methodologies, the neural network is molded to adhere to the available data within specified regions. The crux of this method lies in its capacity to reconstruct a solution that not only aligns with the provided data, but also rigorously satisfies the neutron diffusion equation across the entire solution domain. Figure 4 illustrates the efficacy of this approach, showcasing the excellent applicability of the reconstructed solution for the governing equation. This reconstruction is not merely a localized approximation, but a holistic representation that resonates with the entire solution domain. To improve the practical applicability of nuclear reactors, the proposed technique is pivotal. By deriving a limited yet strategic dataset from neutron detectors within the reactor core, we can extrapolate the elusive k_{eff} value. The utilization of actual data, even in modest quantities, empowers the neural network to discern the critical state of the reactor core. This inference, which is rooted in a data-driven understanding of k_{eff} , becomes instrumental in determining the reactor’s criticality. Overall, the proposed methodology transcends the conventional paradigms of reactor analyses, offering a nuanced and adaptive means to infer crucial

parameters. The amalgamation of neural network training, inverse techniques, and data-driven adaptability represents a sophisticated computational strategy. It not only advances our ability to comprehend the behavior of nuclear reactors but also introduces a paradigm shift for leveraging limited data to unveil the characteristics of k_{eff} , thus contributing to the broader landscape of nuclear science and engineering (Fig. 4).

4.4 Parallel convergence technique for multi-group diffusion equation systems

Following the discretization of the continuous-energy neutron diffusion equation into G energy groups, a system comprising G partial differential equations emerges, G represents the number of energy groups used in the discretization of the continuous-energy neutron diffusion equation. The discretization introduces heightened intricacies into the solution process when employing neural networks. In response to this augmented complexity, a discerning strategy has been applied wherein each partial differential equation is treated as an individual loss term within the neural network. Subsequently, these disparate loss terms are combined, engendering a collective solution. By encapsulating each equation’s contribution within the network’s framework, this method empowers the neural network to address and resolve multiple

Fig. 4 (Color online) Flowchart depicting k_{eff} inversion techniques



interconnected equations. Suppose we need to solve a neutron diffusion equation containing G energy groups, then we need to construct a neural network with G outputs, that is to say, the last layer of this neural network we constructed has G neurons, and at this time each output of the neural network represents the neutron flux distribution function $\phi_g (g = 1, \dots, G)$ of an energy group, respectively. In the training process, we use the set of multi-energy group neutron diffusion equations as the loss function to constrain each output of the neural network, that is, the loss function is constructed as the form of adding the squares of the variational residuals of each group of neutron diffusion equations separately, so as to make the neural network able to output the results of the neutron flux distribution in line with each energy group of the set of multi-energy group neutron diffusion equations after the training is complete. The parallelism inherent in this approach enhances computational efficiency, thus facilitating the convergence of the neural network toward the solution associated with the intricate system of partial differential equations.

4.5 Optimizing neural network hyperparameters using the whale optimization algorithm (WOA)

Deep neural networks are sophisticated and characterized by several hyperparameters that intricately govern their architecture and performance. In particular, the depth, width, and learning rate play pivotal roles in shaping the neural network's capacity to extract complex features and learn intricate patterns. The neural network research is further enriched by the introduction of hp-VPINN, which extends the parameter landscape by incorporating additional elements. Notably, the weight assigned to the boundary condition loss term relative to the equation loss term emerges as a critical hyperparameter in hp-VPINN, introducing an additional layer of complexity. To optimize the dynamic interplay of these hyperparameters, a nuanced approach is proposed by Wang [52]. They advocate for the dynamic adjustment of weights assigned to various loss terms based on empirical observations. This adaptive weighting strategy ensures consistent gradient magnitudes throughout the training process, thereby fostering improved convergence. Despite the empirical success of such strategies, the current landscape lacks a mature and reliable theoretical framework for systematically determining these hyperparameters [54].

Recognizing the pivotal role of hyperparameters in influencing the training efficiency and accuracy of neural networks, this study embarks on a strategic exploration. By leveraging the WOA, an intelligent optimization methodology is introduced to navigate the multifaceted hyperparameter space. Inspired by the foraging behavior of whale populations, WOA intricately integrates the techniques of searching, encircling, and bubble-net attacking into its optimization processes. This

algorithmic approach is promising for refining the values of hyperparameters in neural networks, including those unique to hp-VPINN. The utilization of WOA in hyperparameter optimization significantly addresses the current gaps in theoretical understanding. By simulating the whale foraging behavior, WOA effectively navigates the hyperparameter landscape, surpassing conventional optimization algorithms including particle swarm optimization, differential evolution, gravitational search algorithm, and gray wolf optimization [55]. This superiority is evidenced in terms of solution accuracy and convergence speed, demonstrating the excellent efficacy of WOA for optimizing the intricate hyperparameter configurations inherent in deep neural networks, especially those augmented by hp-VPINN.

4.5.1 Encircling the prey

WOA considers the prey's position as the optimal or approximate optimal solution, and the other individuals in the population update their positions based on this reference. This process can be mathematically represented using the following formulas:

$$\begin{cases} d = |C \cdot X^*(t) - X(t)| \\ X(t+1) = X^*(t) - A \cdot d, \end{cases} \quad (20)$$

where $A = 2ar - a$; $C = 2r$. In these equations, t represents the iteration count, and C are coefficient vectors, X^* denotes the current best position of a whale, X represents the current position of a whale, and a linearly decreases from 2 to 0 during the iteration process. The variable r is a random number within the range $[0, 1]$.

4.5.2 Bubble-net attack strategies

Bubble-net attacks can be classified into two strategies: contraction encircling and spiral updating.

(1) Contraction encircling: This strategy is achieved through the parameter ' a ' in the formula. ' a ' linearly decreases from 2 to 0 during the iteration process, while ' A ' is a random number in the range $(-a, a)$, specifically within the interval $(-2, 2)$. When ' A ' is set between -1 and 1 , the new position of the whale can be defined as any position between its original position and the prey's position.

(2) Spiral updating: The distance between the whale and the prey is computed. Then, a spiral equation is established between these two positions to simulate the spiral motion of the whale:

$$\begin{cases} X(t+1) = d \cdot e^{bl} \cdot \cos(2\pi l) + X^*(t) \\ d = |X^*(t) - X(t)| \end{cases} \quad (21)$$

where ' d ' represents the distance between the whale and the prey, ' b ' denotes a constant for the spiral shape, and ' l ' is a

random number within the interval $[-1, 1]$. While encircling the prey, the whale simultaneously hunts the prey along a spiral trajectory. To simulate this concurrent behavior, a probability of 0.5 is assumed for either selecting contraction encircling or spiral updating to determine the whale’s current position. The mathematical model is as follows:

$$X(t + 1) = \begin{cases} X^*(t) - A \cdot d, p < 0.5 \\ d \cdot e^{bl} \cdot \cos(2\pi l) + X^*(t), p \geq 0.5 \end{cases} \quad (22)$$

Hunting Prey: The whales search for prey based on their positions. When $|A| > 1$, the whale’s position is updated through random selection. The mathematical model is given as follows:

$$\begin{cases} X(t + 1) = Xrand(t) - A \cdot d \\ d = |C \cdot Xrand(t) - X(t)| \end{cases} \quad (23)$$

where $Xrand(t)$ represents the position of a randomly selected whale.

4.5.3 WOA-based neural network hyperparameter optimization model

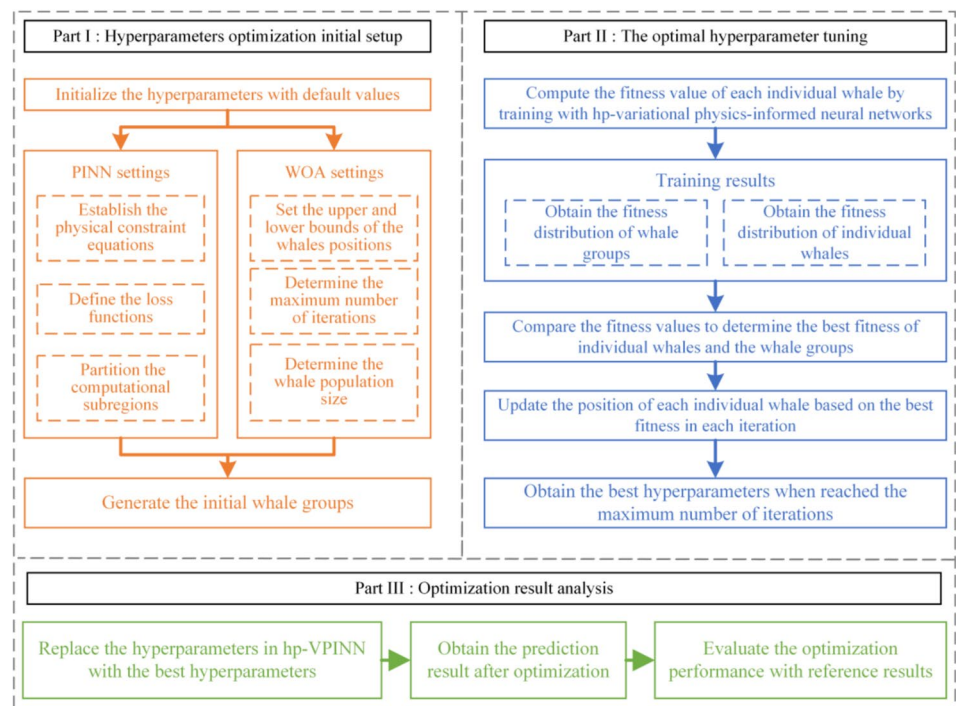
The instantiation of a neural network hyperparameter optimization model is articulated using the WOA (Fig. 5). This model is meticulously designed to discern optimal values for hyperparameters by undergoing iterative training on sample data. This pursuit is aimed at enhancing the overall

efficiency and efficacy of neural networks by tuning the hyperparameters governing their architecture and learning dynamics.

In this iterative process, the computed fitness value assumes a central role by serving as a metric for gauging the neural network performance with a specific set of hyperparameters. The most optimum result is obtained when the computed fitness value reaches its maximum within the ongoing iteration. This outcome is juxtaposed against the global best fitness value that encapsulates the most optimal solution discovered thus far. If the computed fitness value surpasses the global best fitness value, a transition occurs; the global best fitness value is consequently updated, incorporating the newfound superior solution. This strategic evolution reflects the algorithm’s capacity to dynamically adapt and refine its understanding of optimal hyperparameter configurations over successive iterations.

The amalgamation of nature-inspired heuristics enhances the algorithm’s ability to efficiently explore and exploit the hyperparameter landscape. By integrating the WOA into the hyperparameter optimization model, a systematic and adaptive approach emerges. The model is not merely a static entity but rather a dynamic system that continuously refines its understanding of optimal hyperparameter configurations. This iterative refinement process positions the WOA as a potent tool for optimizing neural network performance through judicious hyperparameter tuning.

Fig. 5 (Color online) Workflow diagram of neural network hyperparameter optimization based on WOA



5 Case study

In this section, the proposed computational framework and associated techniques have been validated through a series of cases exhibiting varying levels of complexity.

Section 5.1 delves into the solution of the neutron diffusion equation within a homogeneous medium at criticality. This specific case serves to meticulously validate the efficacy of our multi-region parallel convergence technique. Section 5.2 intricately examines the solution of the one-dimensional neutron diffusion equation formulated in spherical coordinates. This case is instrumental in scrutinizing the integrity of our region partitioning and refinement strategy. Section 5.3 extends the validation spectrum by conducting a comparative analysis between PINN and hp-VPINN. This evaluation is conducted by solving the two-dimensional neutron diffusion equation expressed in cylindrical coordinates. Section 5.4 scrutinizes the computational prowess of our framework for addressing the neutron diffusion equation in scenarios involving multiple media. Section 5.5 focuses on solving the one-dimensional transient neutron diffusion equation. This verification exercise serves to affirm the robustness of our intelligent search technique. Section 5.6 discusses the application of the inversion technique with WOA to solve the multi-group multi-media neutron diffusion equation within the 2D-TWIGL [56] benchmark problem. These case studies collectively enable a thorough assessment of the computational framework’s reliability and versatility across diverse neutron diffusion scenarios.

5.1 Solving the neutron diffusion equation for a homogeneous medium under critical conditions

Herein, we consider a one-dimensional neutron diffusion equation under critical conditions: $\nabla^2 \phi(x) + B_g^2 \cdot \phi(x) = 0$, where $x \in [-1, 1]$ and $\phi(\pm 1) = 0$. At criticality, $B_g^2 = \pi^2$ and $\phi(0) = 0.5$. The analytical solution for the one-dimensional neutron diffusion equation in this case is $\phi(x) = 0.5 \cos(\pi x)$. The loss function is as follows:

$$loss = loss_f + w_b \cdot loss_b + w_i \cdot loss_i, \tag{24}$$

where w_b and w_i represent the relative weights of their corresponding loss terms compared to the control equation loss term. The expressions for $loss_f$, $loss_b$, and $loss_i$ are as follows:

$$loss_f = \sum_{j=1}^k \int_{-1}^1 v_j \left[\nabla^2 \phi_{NN}(x) + B_g^2 \cdot \phi_{NN}(x) \right] dx, \tag{25}$$

$$loss_b = [\phi_{NN}(-1) - \phi(-1)]^2 + [\phi_{NN}(1) - \phi(1)]^2, \tag{26}$$

$$loss_i = [\phi_{NN}(0) - \phi(0)]^2. \tag{27}$$

If we denote the original strong-form equation loss term (24) as $loss_f^{(1)}$, then applying the integration by parts formula once or twice to Eq. (25) and combining it with the compact support property of the polynomials $P_n(x) - P_{n+2}(x)$, ($n = 0, 1, 2, \dots$) yields two alternative weak-form equations:

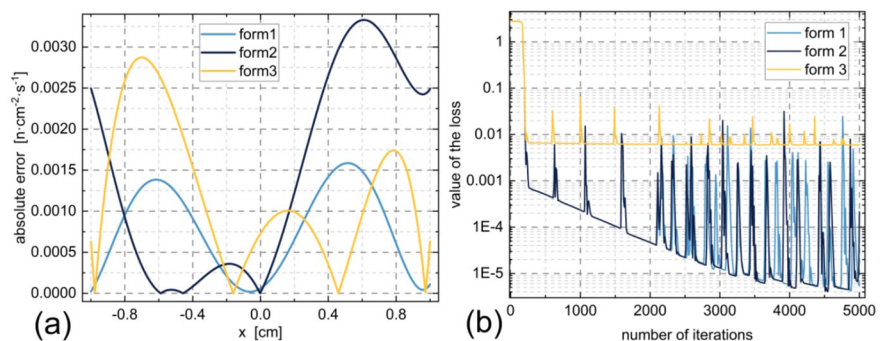
$$loss_f^{(2)} = \sum_{j=1}^n \int_{-1}^1 \left[B_g^2 \cdot \phi_{NN}(x) \cdot v_j - \phi'_{NN}(x) \cdot v'_j \right] dx, \tag{28}$$

$$loss_f^{(3)} = \sum_{i=1}^n \left(\int_{-1}^1 B_g^2 \cdot \phi_{NN}(x) \cdot v_j + \phi_{NN}(x) \cdot v''_j dx - \phi_{NN}(x) \cdot v'_j|_{-1}^1 \right). \tag{29}$$

In this computational investigation, all three configurations adopt a consistent network structure [1,80,80,80,80,1], a learning rate of 0.001, and 100 Gauss quadrature points, while 25th-order Legendre polynomials are employed as trial functions. These configurations are represented by the equation loss terms denoted as $loss_f^{(1)}$, $loss_f^{(2)}$, and $loss_f^{(3)}$, which encapsulate distinct formulations of the underlying mathematical model. The salient evaluation metric represents the distribution of absolute errors in the solutions (Fig. 6), revealing that all three configurations exhibit absolute errors below the threshold of 0.0035.

The mean square errors, a quantitative measure of solution accuracy, stand at 8.71×10^{-7} , 3.42×10^{-6} , and

Fig. 6 (Color online) Absolute error distribution and loss function decline trend. **a** Absolute error distribution for all three forms; **b** loss function decline trend for all three forms



2.16×10^{-6} for forms 1, 2, and 3, respectively. While form 1 attains the highest accuracy, forms 2 and 3 strategically reduce the order of derivatives in the governing equation. This reduction mitigates the computational burden imposed on the neural network when calculating higher-order derivatives through automatic differentiation, thus enhancing the training speed. Empirical evidence, which has been obtained through experimentation on an mx250 GPU that is equivalent in computing power to an NVIDIA GT1030 utilizing TensorFlow 1 on a Windows 11 system, substantiates the enhancement in training speed. The computational efficiency is demonstrated in the iteration durations for 5000 steps, revealing notable distinctions among forms 1, 2, and 3. The computational times are recorded as 51.65 s, 32.89 s, and 23.32 s, respectively. Notably, forms 2 and 3 exhibit an approximate twofold acceleration in iteration speed compared to form 1. The graphical representation of the decreasing trend of the loss functions for each form highlights the convergence behavior. Conversely, form 3 converges but displays a loss reduction plateau around the 350th iteration, suggesting a potential stability threshold.

To ensure a meticulous balance between computational efficiency and solution precision, form 1 is chosen as the preferred configuration for subsequent computational procedures because of its superior accuracy and its capacity to strike an optimal balance between precision and computational expediency, thereby aligning with the imperatives of our investigative pursuits.

To develop a refined computational strategy, we employ a domain decomposition approach to intricately partition the solution domain into two discrete subdomains for independent training. The weight functions governing each subdomain are meticulously crafted; they comprise rescaled Legendre polynomials exclusively confined to the specific subdomain as well as constant functions maintaining a zero-value constraint on the complementary subdomain. The strategic deployment of this domain decomposition technique aims to enhance the precision and efficiency of the training process.

Figure 7 depicts the effectiveness plot and residual distribution associated with this domain decomposition. The blue dashed line demarcates the boundary between the subdomains, thus emphasizing the distinct regions of focus. Each row within the figure shows the weight functions at various orders in the left plot, offering insights into their nuanced characteristics. The middle plot unveils the outcomes of the neural network training process, which leverages the hp-VPINN method with the weight functions depicted in the left plot. This visualization provides a comprehensive understanding of the solutions obtained through this intricate training paradigm. The right plot in

each row meticulously showcases the distribution of absolute errors corresponding to the respective weight functions and neural network solutions, offering a quantitative perspective on the accuracy and fidelity of the computational outcomes.

Such a domain decomposition strategy, which utilizes tailored weight functions and advanced neural network methodologies, proves to be a sophisticated and effective means of addressing complex computational challenges. This approach not only refines the granularity of the solution process but also contributes to the overall robustness and accuracy of the computational framework, as shown in Fig. 7.

5.2 Solving the neutron diffusion equation in one-dimensional spherical coordinates under critical conditions

For a bare spherical reactor with a radius of 1 (including extrapolation distance), when $\phi(0) = \pi$, the analytical solution of the neutron diffusion equation is as follows:

$$\phi(r) = \frac{\sin \pi r}{r}. \quad (30)$$

To enhance the computational fidelity of our methodology, we adopt a strategic domain decomposition strategy by segmenting the solution domain into three primary regions: $[-1, -2/3]$, $[-2/3, 2/3]$, and $[2/3, 1]$. To address localized complexities in regions with pronounced gradients, we implement additional local refinements to establish five subregions: $[-1, -1/3]$, $[-1/3, -2/3]$, $[-2/3, 1/3]$, $[1/3, 2/3]$, and $[2/3, 1]$. The underlying motivation for this approach lies in the nuanced adaptability of our computational framework toward intricacies inherent in diverse solution domains.

Consistency in methodology is maintained by preserving the same set of parameters and network architecture employed in the preceding case study. This ensures a seamless comparison and validation of the enhancements introduced. Figure 8 provides a visual representation of the outcomes derived from this refined methodology. Notably, the localized refinement of regions characterized by heightened gradients significantly reduces the absolute error. By quantitatively assessing the improvements, the mean square errors demonstrate a notable decrease from 1.1×10^{-7} before refinement to 5.9×10^{-8} after refinement. This empirical evidence substantiates the efficacy of our methodology in ameliorating computational accuracy through targeted local refinements. The computational efficiency of the methodology is another interesting facet. Comparative training times for 5000 steps serve as a crucial metric. The temporal dynamics reveal a nuanced picture, with the training time increasing from 61.64 s before refinement to 103.55 s after refinement. This temporal variation underscores the interplay between

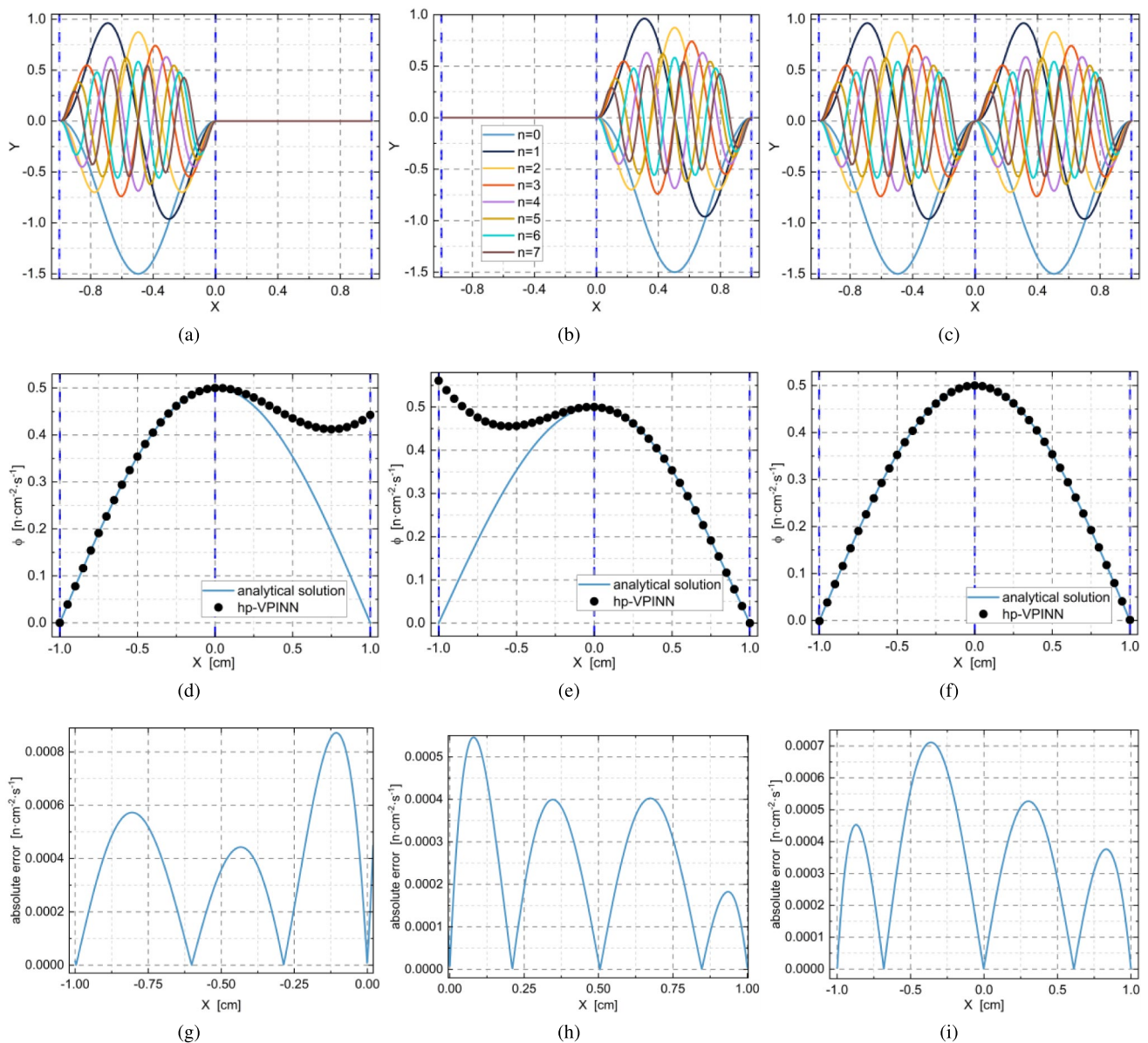


Fig. 7 (Color online) Solution results obtained using Legendre polynomial as a weight function in the subdomains. **a** The weight function of Case 1; **b** the weight function of Case 2; **c** the weight function of

Case 3; **d** the results of Case 1; **e** the results of Case 2; **f** the results of Case 3; **g** the absolute error in Case 1; **h** the absolute error in Case 2; **i** the absolute error in Case 3

computational precision and efficiency, necessitating a careful consideration of the trade-offs involved in localized refinement strategies within our computational framework.

5.3 Solving the neutron diffusion equation in two-dimensional cylindrical coordinates

The critical state of the neutron diffusion equation in two-dimensional cylindrical coordinates has been elucidated through the application of advanced computational methodologies, specifically hp-VPINN and PINN. This investigation delves into the nuanced aspects of solving complex

neutron diffusion problems by scrutinizing the efficacy and computational accuracy of these methodologies. The experiment is initiated by setting an initial value of $\phi(0, 0) = 1$, which establishes the foundation for subsequent computational exploration. The solution domain and the distribution of absolute errors are depicted in Fig. 9, providing a visual representation of the computational outcomes.

When implementing the hp-VPINN method, a judicious selection of computational parameters is employed. One hundred Gaussian-Legendre integration points are distributed within each domain, demonstrating the meticulous attention to detail. To further enhance accuracy, the Latin

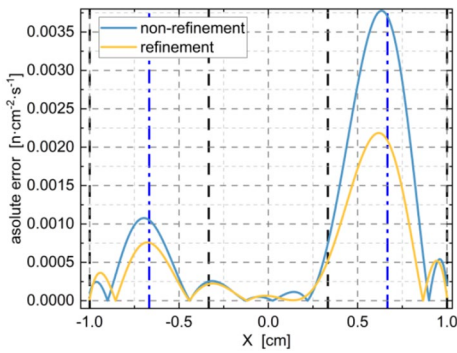


Fig. 8 (Color online) Absolute error distribution before and after mesh refinement in the solution domain

hypercube sampling method [57] has been applied to sample 80 boundary points for each boundary, thus contributing to a comprehensive training dataset. This meticulous approach results in 420 training points for a single domain and 1220 training points distributed across nine domains. The calculated mean square errors for this method are 1.02×10^{-5} and 8.66×10^{-6} , which corroborate its precision and reliability in addressing the critical state of the neutron diffusion equation.

Figure 9d–f unveils the outcomes derived from employing the PINN method, while maintaining an equivalent number of training points for performing a comparative analysis. The mean square errors for the PINN method are 1.84×10^{-5} and 1.75×10^{-5} , showcasing a commendable level of accuracy. However, a discerning observation is the differential computational efficiency demonstrated by the two methods. The computational times for four distinct cases, each involving 1000 iterations, are 9.61 s, 65.30 s, 15.84 s, and 31.66 s. These temporal dynamics underscore the computational efficiency achieved by the hp-VPINN method, positioning it as a promising avenue for addressing complex neutron diffusion scenarios.

An intriguing insight derived from this comparative analysis is that despite maintaining an equivalent training cost, the hp-VPINN method consistently demonstrated heightened computational accuracy compared to the PINN method. This substantiates the efficacy of the hp-VPINN approach for determining the critical state of the neutron diffusion equation, thus showcasing its potential as a robust computational tool for nuclear engineering and computational physics.

Figure 10 visualizes the convergence dynamics observed in the hp-VPINN model (featuring nine meticulously defined subdomains) and the PINN model (featuring an extensive training dataset of 1220 points). This

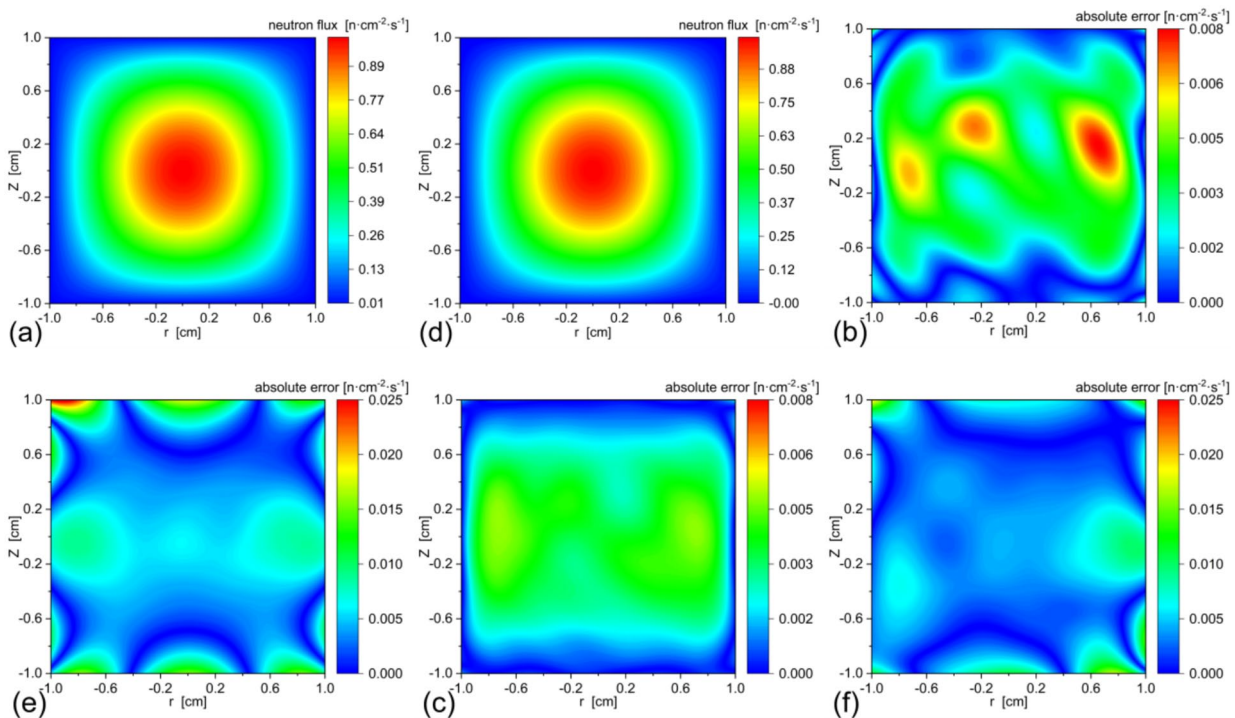


Fig. 9 (Color online) Comparison of hp-VPINN and PINN for solving two-dimensional cylindrical coordinate neutron diffusion equation. **a** solution of hp-VPINN with one subdomain; **b** absolute error of hp-VPINN with one subdomain; **c** absolute error of hp-VPINN

with nine subdomains; **d** solution of PINN with 420 training points; **e** absolute error of PINN with 420 training points; **f** absolute error of PINN with 1220 training points

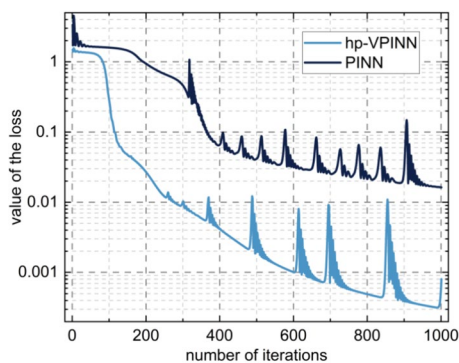


Fig. 10 (Color online) Loss function decline trend of hp-VPINN and PINN

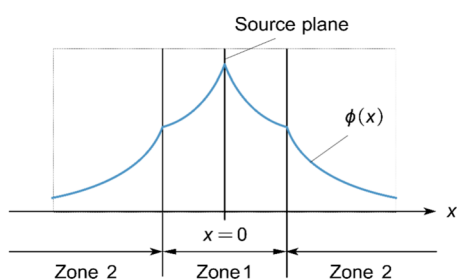


Fig. 11 (Color online) Schematic representing the one-dimensional single-group two-medium problem with a constant source

comparative analysis provides a pivotal insight into the performance disparities between the two methodologies when solving the critical state of the neutron diffusion equation. The hp-VPINN model, with its adept incorporation of subdomains, demonstrates a conspicuous acceleration in convergence rate when juxtaposed with the PINN model. The nuanced visualization in Fig. 10 illustrates the expedited pace at which the hp-VPINN model refines its solution toward convergence. Additionally, a discernible reduction in loss, relative to the PINN model, underscores the efficacy of the hp-VPINN methodology for navigating the intricate landscape of the neutron diffusion equation.

5.4 Solving the neutron diffusion equation in heterogeneous media

For the one-dimensional single-group two-material problem with a constant source, the one-dimensional space is divided into three regions of three different materials. At $x = 0$, a steady point source S emits neutrons in both directions (Fig. 11).

The corresponding neutron diffusion equation is as follows:

$$\begin{cases} \frac{d^2\varphi_1}{dx^2} - \frac{1}{L_1^2}\varphi_1 = 0, 0 < |x| < \frac{|a|}{2} \\ [2ex] \frac{d^2\varphi_2}{dx^2} - \frac{1}{L_2^2}\varphi_2 = 0, |x| > \frac{|a|}{2} \end{cases} \quad (31)$$

The boundary conditions are given below.

(1) Symmetric surface neutron source:

$$\lim_{x \rightarrow 0} J_1(x) = S/2. \quad (32)$$

(2) As x approaches infinity, $\varphi_2 = 0$:

$$\lim_{x \rightarrow \infty} \varphi_2(x) = 0. \quad (33)$$

(3) Neutron flux continuity at the interface between materials:

$$\varphi_1\left(\frac{a^-}{2}\right) = \varphi_2\left(\frac{a^+}{2}\right), \quad (34)$$

$$\varphi_1\left(-\frac{a^-}{2}\right) = \varphi_2\left(-\frac{a^+}{2}\right). \quad (35)$$

(4) Neutron current density continuity at the interface between materials:

$$D_1 \frac{d\varphi_1\left(\frac{a^-}{2}\right)}{dx} = D_2 \frac{d\varphi_2\left(\frac{a^+}{2}\right)}{dx}, \quad (36)$$

$$D_2 \frac{d\varphi_2\left(-\frac{a^-}{2}\right)}{dx} = D_1 \frac{d\varphi_1\left(-\frac{a^+}{2}\right)}{dx}, \quad (37)$$

when $x > 0$, the analytical solution to the diffusion equation is as follows:

$$\begin{cases} \varphi_1 = A_1 \cosh\left(\frac{x}{L_1}\right) + C_1 \sinh\left(\frac{x}{L_1}\right) \\ \varphi_2 = A_2 \exp\left(-\frac{x}{L_2}\right) + C_2 \exp\left(\frac{x}{L_2}\right) \end{cases} \quad (38)$$

where

$$\begin{cases} C_1 = \frac{SL_1}{2D_1} \\ C_2 = 0 \\ A_1 = \frac{SL_1}{2D_1} \frac{D_1L_2 \cosh\left(\frac{a}{2L_1}\right) + D_2L_1 \sinh\left(\frac{a}{2L_1}\right)}{D_2L_1 \cosh\left(\frac{a}{2L_1}\right) + D_1L_2 \sinh\left(\frac{a}{2L_1}\right)} \\ A_2 = e^{\frac{a}{2L_2}} \left(A_1 \cosh\left(\frac{a}{2L_1}\right) - \frac{SL_1}{2D_1} \sinh\left(\frac{a}{2L_1}\right) \right) \end{cases} \quad (39)$$

We set the solution domain as and divide it into two regions: $[0, a/2]$ and $[a/2, 1]$. To solve this problem using the hp-VPINN method, we need to construct appropriate loss

functions. For boundary condition (1), we construct $loss_{b1}$, as shown in Eq. (40):

$$loss_{b1} = \left[D_1 \frac{d\varphi_{NN}(10^{-6})}{dx} - S \right]^2. \tag{40}$$

For boundary condition (2), we consider the distance at $x = 10^3$ to be approximately infinity and then construct $loss_{b2}$, as shown in Eq. (41):

$$loss_{b2} = [\varphi_{NN}(10^3) - 0]^2. \tag{41}$$

For boundary condition (3), since we use the Tanh function as the activation function for the fully connected neural network, the output of the neural network itself acts as a continuous function; thus, there is no need to set a separate loss term for this.

For boundary condition (4), we set the $loss_{b3}$ value, as shown in Eq. (42):

$$loss_{b3} = \left[D_1 \frac{d\varphi_{NN}(\frac{a}{2} - 10^{-6})}{dx} - D_2 \frac{d\varphi_{NN}(\frac{a}{2} + 10^{-6})}{dx} \right]^2. \tag{42}$$

As delineated in Sect. 3, determining the loss term in the partial differential equation is pivotal for the computational methodology under consideration because it influences the precision and fidelity of the computational framework employed for solving the neutron diffusion equation.

To validate the efficacy of the established method, a systematic exploration is conducted by manipulating specific parameters. Notably, a (1 cm), uniform diffusion coefficients

($D_1 = D_2 = 1$ cm), and material curvature ($L_1 = 1$ cm) are fixed across all experimental scenarios. The variable of interest, denoted as L_2 , is intentionally set at 0.5 cm, 1.0 cm, and 1.5 cm for zones 1, 2, and 3, respectively. These meticulously chosen L_2 values allow for a comprehensive assessment of the method’s performance under diverse conditions.

The computational outcomes of these carefully configured scenarios are shown in Fig. 12. The graphical depiction elucidates the solution landscape and the corresponding absolute errors for the three conditions. The mean squared errors are 2.12×10^{-6} , 2.64×10^{-8} , and 6.76×10^{-8} for conditions 1, 2, and 3, respectively. This quantitative assessment serves as a robust indicator of the computational framework’s proficiency for addressing the critical state of the neutron diffusion equation under varying conditions. By maintaining methodological consistency, it is ensured that the fundamental aspects of the neural network structure and computational parameters are constant throughout these experimental iterations. This stability ensures a fair and unbiased evaluation of the method’s response to the variable L_2 , fortifying the reliability and scientific rigor of the findings.

Furthermore, the temporal dynamics of the computational process are quantified through the runtime required for 3000 iterations. The runtime is measured as 22.12 s and provides valuable insights into the computational efficiency achieved within the stipulated iterations. This temporal metric and the quantitative assessment of mean squared errors collectively offer a comprehensive evaluation of the method’s computational accuracy and efficiency when addressing the critical state of the neutron diffusion equation under varying conditions.

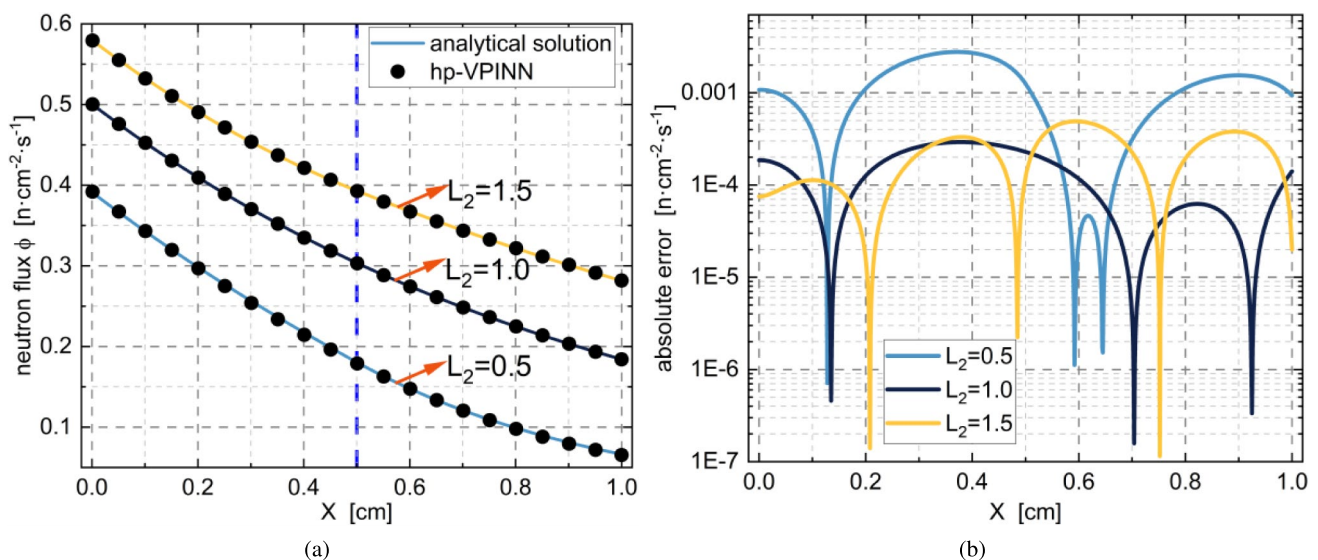


Fig. 12 (Color online) Solution results and absolute errors. **a** Solution results; **b** absolute errors

5.5 Solving the one-dimensional transient neutron diffusion equation

The transient one-dimensional neutron diffusion equation made for rectangular coordinates can be written in the form of Eq. (2) and its analytical solution is as follows:

$$\phi(x, t) = \sum_{n=1}^{\infty} \left[\frac{a}{2} \int_{-\frac{a}{2}}^{\frac{a}{2}} \phi_0(x') \cos \frac{(2n-1)\pi}{a} x' dx' \right] \cdot \cos \frac{(2n-1)\pi}{a} x e^{\frac{(\lambda_n-1)t}{\lambda_n}} \tag{43}$$

The resolution of the eigenvalue equation (Sect. 4.3) stands as a formidable computational challenge for neutron diffusion. Dong et al. [31] proposed a parallel search technique based on binary search to ascertain the value at the critical state. Despite its efficacy, this method necessitates multiple iterations of neural network training and imposes substantial demands on computational memory, which are particularly evident when managing several parallel tasks.

In response to the computational complexities posed by the aforementioned method, this study introduces a novel search technique, which strategically leverages the physical characteristics exhibited at the critical state. The crux of this methodology lies in the initialization of to 1, which is followed by continuous adjustments throughout the iterative cycles of the neural network. A key premise underlying this approach is rooted in the fact that when the reaction achieves criticality at a specific time (denoted as τ), the temporal derivative of the neutron flux distribution ($\phi_t(r, t) = 0$)

assumes a zero value for subsequent instances; this indicates the absence of reaction propagation. Therefore, we construct the corresponding loss term as follows:

$$loss_k = \frac{1}{n_r \cdot n_t} \sum_{i=0}^{i=n_r} \sum_{j=0}^{i=n_t} \left(\frac{\partial \phi(r_i, t_j)}{\partial t} \right)^2, \tag{44}$$

where $t_j > \tau, j = 1, 2, 3 \dots n_t$, and n_r is the number of spatial sampling points at $t = t_j$. Two test cases are set to validate this method, with the equation parameters being $v = 2.2 \times 10^3$ m/s, $D = 0.211 \times 10^{-2}$ m, $L^2 = 2.1037 \times 10^3$ m², and $a = 2$ m; the other conditions are shown in Table 1.

The outcomes of our computational endeavor are unveiled in Fig. 13, which offers a visual representation of the solution results. The mean squared errors, which are in the order of 10^{-6} , bear testament to the precision and fidelity of our computational framework for addressing the complexities inherent in the neutron diffusion domain. Compared to the reference values of 0.9995811 and 1.0035790 for the two test cases, our technique yields absolute errors of 0.02 pcm and 0.01 pcm. This attests to the efficacy of our search methodology, which achieves results that closely align with the expected reference values, thus signifying a high degree of computational accuracy in predicting the effective multiplication factor.

Furthermore, the temporal dynamics of our computational process, quantified by the runtime required for 8000 iterations, reveal that the machinery operates within a temporal framework of 352.88 s; this reflects the nuanced interplay between computational accuracy and efficiency when determining neutron diffusion solutions.

Table 1 Example parameters and initial condition

	Case 1	Case 2
k_{∞}	1.0041	1.001
Initial condition	$\cos(\pi x/a) - 0.4(\cos(2\pi x/a) + 1)$	$\cos(2\pi x/a) + 1$

5.6 Solving the neutron diffusion equations in multi-energy group and multi-media systems

The 2D-TWIGL benchmark problem, a notable illustration within the domain of nuclear reactor simulations [58],

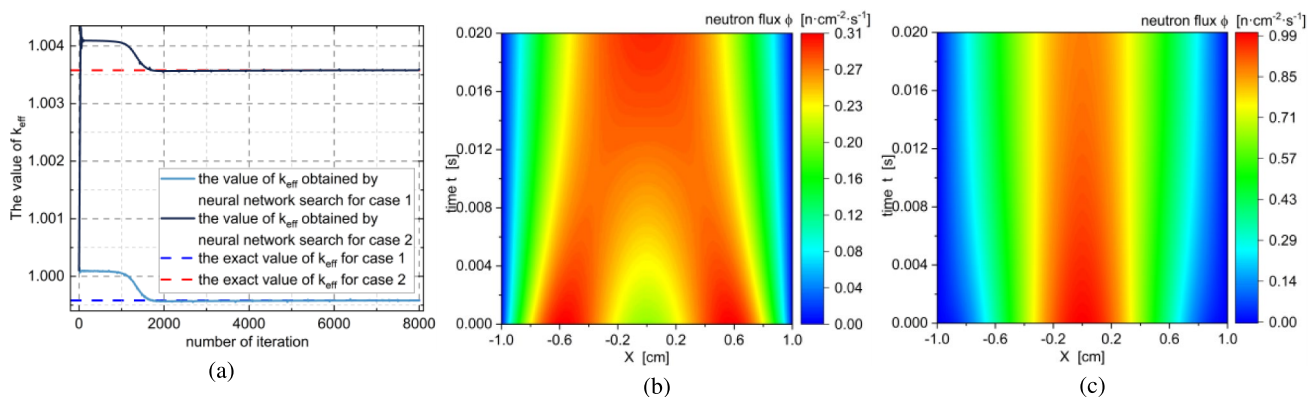


Fig. 13 (Color online) Solution results and absolute errors. a Search results for k_{eff} ; b solution results for case 1; c search results for case 2

encapsulates a simplified two-dimensional rectangular core configuration that is categorized into distinct breeding and blanket regions. The core, exhibiting a square geometry with a side length of 160 cm, is partitioned into three discernible regions. Symmetry considerations guide our computational approach, which focuses exclusively on a 1/4th portion of the core as the designated computational domain. The intricacies of the core configuration and its defining parameters are elucidated in Fig. 14, while Table 2 provides a meticulous tabulation of these critical parameters.

The computational resolution of the 2D-TWIGL benchmark problem is undertaken through the systematic integration of hp-VPINN and the parameter inversion technique. By employing the KOMODO program, formerly recognized as the three-dimensional core neutronics code ADPRES [59], we navigate the intricacies of this benchmark problem. The KOMODO program provides indispensable neutron flux data for a 40 × 40 core configuration, establishing a foundational dataset for subsequent computational methodologies. To optimize the neural network’s input, a Latin hypercube sampling approach is applied to selectively choose 1% of the entire core dataset. The neural network undergoes a robust training regimen, encompassing control equations, boundary conditions, and additional pertinent physical information. This meticulous training ensures not only accuracy but also the convergence of the neural network, which is crucial in addressing the intricate nature of the governing equations and zoning arrangements.

The hyperparameters in hp-VPINN are optimized by implementing WOA. Key hyperparameters, including the number of artificial whales (set to 5), the maximum iteration count for the optimization algorithm (set to 10), and the maximum iteration count for the neural network (set to 10,000), are systematically tuned. The convergence criteria, which are an equation loss threshold below 10⁻⁴ and an absolute k_{eff} error of less than 20 pcm, dictate the conclusion

Table 2 Zonal cross-section parameters

Zone	Energy group	D_g (cm)	$\Sigma_{a,g}$ (cm ⁻¹)	$\nu\Sigma_{f,g}$ (cm ⁻¹)	$\Sigma_{1\rightarrow 2}$ (cm ⁻¹)
1,2	1	1.4	0.01	0.007	0.01
1,2	2	0.4	0.015	0.2	
3	1	1.3	0.008	0.003	0.01
3	2	0.5	0.05	0.06	

of the training process. The fitness evaluation in WOA incorporates three vital metrics: the algorithm’s fitness, mean squared error in hp-VPINN neutron flux calculations compared to KOMODO, and the mean squared error of k_{eff} . These metrics are inversely proportional, and their combined impact is encapsulated in an overall fitness metric. Figure 15 depicts the iterative process, showcasing the variation in the maximum normalized fitness during the execution of WOA.

The optimal hyperparameters obtained are a learning rate of 4.9×10^{-4} and weightings of 8 and 9 for the data loss term and boundary condition loss term relative to the equation loss term, respectively. The neural network structure comprises six hidden layers with specific node counts [56, 40, 62, 89, 97, 79]. The computational time required for this comprehensive computational endeavor is 1142.23 s.

The computational analysis results are presented in Fig. 16 and mirror the findings documented in literature [60]. The computed results are meticulously juxtaposed against the reference outcomes in Fig. 17. The inverted value, a pivotal metric in neutron diffusion analyses, is calculated as 0.913222. This value exhibits a remarkable degree of concordance with the reference value of 0.913214. The absolute error of a mere 0.8 pcm underscores the precision and fidelity of the computational framework in approximating the k_{eff} parameter. This fidelity indicates the computational robustness and high accuracy of the model in the solution landscape of the neutron diffusion equation.

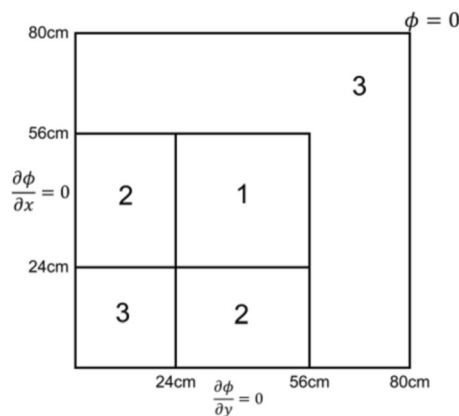


Fig. 14 Schematic of the 2D-TWIGL benchmark problem

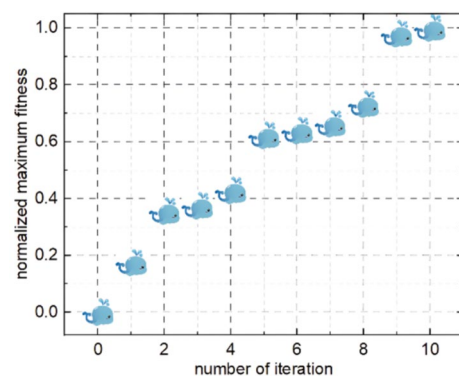


Fig. 15 (Color online) Variation of fitness with iteration times

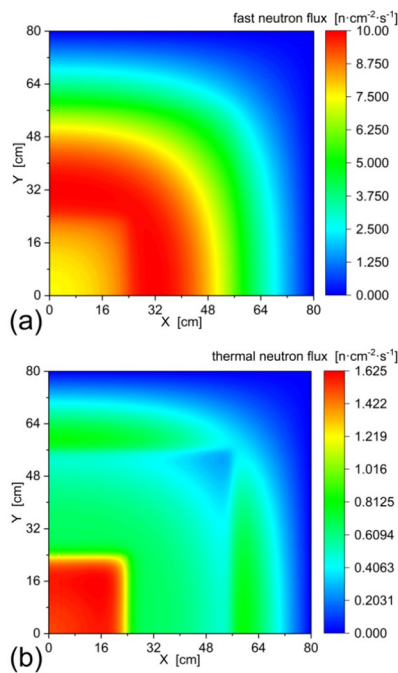


Fig. 16 (Color online) Solution results of neutron flux. **a** The solution for the fast neutron flux; **b** the solution for the thermal neutron flux

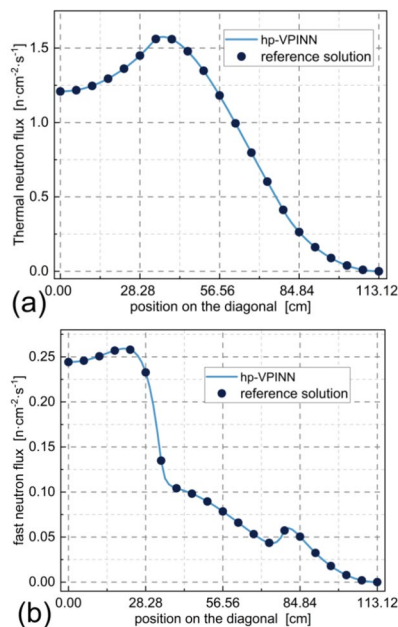


Fig. 17 (Color online) Diagonal distribution of normalized solution results and reference solution. **a** Thermal neutron flux; **b** Fast neutron flux

The obtained outcome underscores the efficacy of the computational model, which is based on 1% of the entire core neutron flux data and refined through WOA. This meticulously designed model exhibits a remarkable capacity to

produce high-precision solutions for the neutron flux distribution throughout the reactor core. Within the realm of the reactor physics analysis, this model introduces an innovative and interactive method for digital twinning of a nuclear reactor. The novelty of this approach lies in its capability to invert the complete neutron flux distribution across the core, while leveraging only a limited subset of actual data. This inversion process is pivotal since it enables a comprehensive understanding of the neutron behavior in the reactor core based on a constrained dataset.

One of the distinctive attributes of this model is its pronounced reduction in the requisite input data, while ensuring high accuracy. This reduction in input data is achieved without compromising the reliability of the results, thereby offering an efficient means to infer crucial safety-related information. This is noteworthy as it is difficult to measure certain safety-related parameters within the reactor environment.

Moreover, the model's ability to derive valuable safety-related data from a limited set of detectable information within the reactor underscores its practical utility. The significance of this capability is magnified in scenarios where obtaining extensive data directly from the reactor is logistically challenging or resource intensive. The model's adeptness in extracting crucial safety-related insights from a modest set of observable data makes it a valuable tool for nuclear reactor safety analyses. The overarching advantage of this computational model lies in its ability to facilitate real-time core neutron diffusion simulations for nuclear reactor safety analysis. By providing a nuanced understanding of neutron behavior and flux distribution, this model contributes to enhancing the accuracy and efficiency of safety assessments, thereby advancing the domain of nuclear reactor analysis and simulation.

6 Conclusion

This study leverages the hp-VPINN technique as a computational tool for tackling the neutron diffusion equation, while addressing both the physical intricacies and practical challenges inherent in its formulation. The employed techniques encompass multi-region parallel convergence, region partitioning refinement, intelligent k_{eff} search, and k_{eff} inversion. These techniques collectively provide a nuanced understanding of the neutron diffusion equation and offer effective solutions to real-world complexities.

A salient aspect of this investigation involves the application of the WOA to automatically optimize the hyperparameters of the neural network. This strategic utilization enhances the efficiency and convergence of the neural network, thereby augmenting the overall computational efficacy of the hp-VPINN technique.

The empirical findings drawn from rigorous verification procedures across multiple cases elucidate the robustness and versatility of the hp-VPINN model:

1. The hp-VPINN model, employed for forward solving of the neutron diffusion equation in scenarios involving single-energy single-medium and single-energy multi-medium configurations, exhibits commendable accuracy solely based on physical data. Moreover, its generality across different coordinate systems underscores its versatility.
2. Compared to the PINN model, the hp-VPINN model, under equivalent training costs, emerges as a superior choice since it showcases heightened accuracy for solving the neutron diffusion equation. This finding emphasizes the advantageous computational characteristics embedded in the hp-VPINN framework.
3. To inversely solve the neutron diffusion equation for multi-group multi-medium scenarios, the hp-VPINN model deploys WOA to navigate the hyperparameter landscape. This meticulous optimization process facilitates the convergence of the neural network, enabling precise inversion of k_{eff} . The model achieves a remarkable absolute error of less than 10^{-5} in the inversion results, leveraging only 1% of the data from the entire core. This outcome underscores the efficiency and precision of the proposed method.

In summary, the proposed hp-VPINN framework harnesses the power of deep learning to elucidate the neutron flux distribution in reactors. Notably, this approach stands out because it delivers high-precision solutions with minimal training data, making it particularly advantageous for advanced reactor simulation techniques such as digital twinning of reactors.

Author contributions All authors contributed to the study conception and design. Material preparation, data collection and analysis were performed by Fu-Lin Zeng, Xiao-Long Zhang, Peng-Cheng Zhao and Zi-Jing Liu. The first draft of the manuscript was written by Fu-Lin Zeng, and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

Data availability The data that support the findings of this study are openly available in Science Data Bank at <https://cstr.cn/31253.11.sciencedb.28488> and <https://doi.org/10.57760/sciencedb.28488>.

Declarations

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

References

1. B.G. Park, B.J. Jun, G.D. Kang et al., Measurement of the neutron flux distribution in a research reactor by neutron activation analysis. *J. Radioanal. Nucl. Chem.* **330**, 501–512 (2021). <https://doi.org/10.1007/s10967-021-07837-2>
2. T.X. Wang, T.F. Zhang, H.C. Wu et al., Modeling and analysis of depletion experiment benchmark based on assembly calculation. *Nucl. Tech.* (in Chinese) **43**(06), 060003 (2020). <https://doi.org/10.11889/j.0253-3219.2020.hjs.43.060003>
3. Z.Q. Wu, J.S. Xie, L. Lou et al., Control rod strategy for long-term small rod-controlled pressurized water reactors. *Nucl. Tech.* (in Chinese) **47**(03), 030604 (2024). <https://doi.org/10.11889/j.0253-3219.2024.hjs.47.030604>
4. C. Ming, Z. LYU, Y.W. Jia, Simulation study on the characteristics of reactor control drum based on 3KEYMASTER platform. *Nucl. Tech.* (in Chinese) **43**(06), 060007 (2020). <https://doi.org/10.11889/j.0253-3219.2020.hjs.43.060007>
5. C. Wang, H. Xiao, Z.J. Liu et al., Geometric configuration of fuel assembly for small lightweight lead-bismuth reactor. *Nucl. Tech.* (in Chinese) **46**(09), 090606 (2023). <https://doi.org/10.11889/j.0253-3219.2023.hjs.46.090606>
6. M.A. Shafii, W.W. Yunanda, D. Fitriyani et al., Neutron flux distribution calculation for various spatial mesh of finite slab geometry using one-dimensional diffusion equation. *AIP Conf. Proc.* **2180**(1), 020001 (2019)
7. A. Gupta, Methods and applications of 3-D SN neutron transport code ATES3. *Life. Cycle. Reliab. Saf. Eng.* **12**, 175–186 (2023). <https://doi.org/10.1007/s41872-023-00229-3>
8. R. Gross, D. Tomatis, E. Gilad, High-accuracy neutron diffusion calculations based on integral transport theory. *Eur. Phys. J. E* **135**(2), 235 (2020). <https://doi.org/10.1140/epjp/s13360-020-00216-y>
9. Z.C. Xiang, Q.F. Chen, P.C. Zhao et al., Apply implicitly restarted Arnoldi method to solving eigenvalue problem and reducing dimensionality in neutron diffusion. *Nucl. Tech.* (in Chinese) **47**, 020604 (2024). <https://doi.org/10.11889/j.0253-3219.2024.hjs.47.020604>
10. D. Wang, F. Li, J. Guo et al., Improved nodal expansion method for solving neutron diffusion equation in cylindrical geometry. *Nucl. Eng. Des.* **240**(8), 1997–2004 (2010). <https://doi.org/10.1016/j.nucengdes.2010.04.021>
11. H. Yin, X.J. Liu, T.F. Zhang, An efficient parallel algorithm of variational nodal method for heterogeneous neutron transport problems. *Nucl. Sci. Tech.* **35**, 69 (2024). <https://doi.org/10.1007/s41365-024-01430-4>
12. Y.Z. Chun, C. Gong, Fast solution of neutron diffusion problem by reduced basis finite element method. *Ann. Nucl. Energy* **111**, 702–708 (2018). <https://doi.org/10.1016/j.anucene.2017.09.044>
13. Z.W. Zong, M.S. Cheng, Y.C. Yu et al., A multithreaded parallel upwind sweep algorithm for the SN transport equations discretized with discontinuous finite elements. *Nucl. Sci. Tech.* **34**, 200 (2023). <https://doi.org/10.1007/s41365-023-01355-4>
14. A. Zhu, M. Jarrett, Y. Xu et al., An optimally diffusive Coarse Mesh Finite Difference method to accelerate neutron transport calculations. *Ann. Nucl. Energy* **95**, 116–124 (2016). <https://doi.org/10.1016/j.anucene.2016.05.004>
15. Y. Zhou, L.Z. Chao, Z.Y. Liu et al., Research on numerical method and experimental verification of the vanadium SPND gamma effect based on current component separation model. *Nucl. Tech.* (in Chinese) **47**, 090403 (2024). <https://doi.org/10.11889/j.0253-3219.2024.hjs.47.090403>
16. X. Huang, W. Zong, T. Wang et al., Symmetry energy from nuclear multifragmentation. *Front. Energy Res.* **8**, 127 (2020). <https://doi.org/10.3389/fenrg.2020.00127>
17. E. Zhu, T. Li, J. Xiong et al., A super-real-time three-dimension computing method of digital twins in space nuclear power. *Comput. Methods Appl. Mech. Eng.* **417**, 116444 (2023). <https://doi.org/10.1016/j.cma.2023.116444>

18. H. Song, M. Song, X. Liu, Online autonomous calibration of digital twins using machine learning with application to nuclear power plants. *Appl. Energy* **326**, 119995 (2022). <https://doi.org/10.1016/j.apenergy.2022.119995>
19. M.Y. Hu, X.Y. Zhang, C.T. Peng et al., Current status of digital twin architecture and application in nuclear energy field. *Ann. Nucl. Energy* **202**, 110491 (2024). <https://doi.org/10.1016/j.anucene.2024.110491>
20. M. Raissi, P. Perdikaris, G.E. Karniadakis, Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.* **378**, 686–707 (2019). <https://doi.org/10.1016/j.jcp.2018.10.045>
21. S. Cuomo, V.S. Di Cola, F. Giampaolo et al., Scientific machine learning through physics-informed neural networks: where we are and what's next. *J. Comput. Phys.* **88**, 92 (2022). <https://doi.org/10.1007/s10915-022-01939-z>
22. S. Tan, T. Li, Y. Liu et al., Thoughts on the application of artificial intelligence in nuclear energy field. *Nucl. Power Eng. (in Chinese)* **44**(2), 1–8 (2023). <https://doi.org/10.13832/j.jnpe.2023.02.0001>
23. H. Zhang, X. Lv, D. Liu et al., Nuclear power AI applications: status, challenges and opportunities. *Nucl. Power Eng. (in Chinese)* **44**(1), 1–8 (2023). <https://doi.org/10.13832/j.jnpe.2023.01.0001>
24. Y.H. Zhou, W.T. Xu, L. Liu et al., Prediction of interfacial area concentration based on interpretable neural network. *Nucl. Tech. (in Chinese)* **47**(08), 080502 (2024). <https://doi.org/10.11889/j.0253-3219.2024.hjs.47.080502>
25. S.Q. Chun, H. Feng, A.N. Zhang et al., Method of predicting transient thermal hydraulic parameters of the core based on the gated recurrent unit model of soft attention. *Nucl. Tech. (in Chinese)* **47**(01), 010603 (2024). <https://doi.org/10.11889/j.0253-3219.2024.hjs.47.010603>
26. J.Y. Chen, X.Y. Liu, P.C. Zhao et al., Prediction method of reactor neutron flux and keff based on the optimized extreme learning machine model. *Nucl. Tech. (in Chinese)* **47**(10), 181–190 (2024). <https://doi.org/10.11889/j.0253-3219.2024.hjs.47.100604>
27. Z. Shi, C. Gao, Z. Dou et al., Dynamic stall modeling of wind turbine blade sections based on a data-knowledge fusion method. *Energy* **305**, 132234 (2024). <https://doi.org/10.1016/j.energy.2024.132234>
28. W.W. Zhang, X.H. Peng, J.Q. KOU et al., Heterogeneous data-driven aerodynamic modeling based on physical feature embedding. *Chin. J. Aeronaut.* **37**(3), 1–6 (2023). <https://doi.org/10.1016/j.cja.2023.11.010>
29. M. Du, Y. Chen, L. Nie et al., Physics-constrained robust learning of open-form partial differential equations from limited and noisy data. *Phys. Fluids* **36**, 057123 (2024). <https://doi.org/10.1063/5.0204187>
30. J. Hu, Z. Dou, W. Zhang, Fast fluid-structure interaction simulation method based on deep learning flow field modeling. *Phys. Fluids* **36**, 045106 (2024). <https://doi.org/10.1063/5.0200188>
31. A.Y. Cheng, H. Cai, S. Chen et al., Application of deep learning methods combined with physical background in wide field of view imaging atmospheric Cherenkov telescopes. *Nucl. Sci. Tech.* **35**, 84 (2024). <https://doi.org/10.1007/s41365-024-01448-8>
32. T.S. Wang, X. Chai, C.R. Guan et al., Numerical and theoretical investigations of heat transfer characteristics in helium–xenon cooled microreactor core. *Nucl. Sci. Tech.* **34**, 162 (2023). <https://doi.org/10.1007/s41365-023-01311-2>
33. H. Guo, Y.W. Wu, Q.F. Song et al., Development of multi-group Monte-Carlo transport and depletion coupling calculation method and verification with metal-fueled fast reactor. *Nucl. Sci. Tech.* **34**, 163 (2023). <https://doi.org/10.1007/s41365-023-01310-3>
34. H. Song, X. Liu, M. Song, Comparative study of data-driven and model-driven approaches in prediction of nuclear power plants operating parameters. *Appl. Energy* **341**, 121077 (2023). <https://doi.org/10.1016/j.apenergy.2023.121077>
35. Q.H. Yang, Y. Yang, Y.T. Deng et al., Physics-constrained neural network for solving discontinuous interface K-eigenvalue problem with application to reactor physics. *Nucl. Sci. Tech.* **34**, 161 (2023). <https://doi.org/10.1007/s41365-023-01313-0>
36. Y.C. Xie, Y. Ma, Y.H. Wang, Solving steady-state mono-energy neutron diffusion solution set with parameterized physics-informed neural network. *At. Energy Sci. Tech. (in Chinese)* **58**(6), 1242–1249 (2023). <https://doi.org/10.7538/yzk.2023.youxian.0765>
37. M.H. Elhareef, Z. Wu, Physics-informed neural network method and application to nuclear reactor calculations: a pilot study. *Nucl. Sci. Eng.* **197**(4), 601–622 (2022). <https://doi.org/10.1080/00295639.2022.2123211>
38. E. Schiassi, M. Florio, B.D. Ganapol. De et al., Physics-informed neural networks for the point kinetics equations for nuclear reactor dynamics. *Ann. Nucl. Energy* **167**, 108833 (2022). <https://doi.org/10.1016/j.anucene.2021.108833>
39. Y.C. Xie, Y.H. Wang, Y. Ma, Boundary dependent physics-informed neural network for solving neutron transport equation. *Ann. Nucl. Energy* **195**, 110181 (2024). <https://doi.org/10.1016/j.anucene.2023.110181>
40. D. Liu, Q. Luo, L. Tang et al., Solving multi-dimensional neutron diffusion equation using deep machine learning technology based on PINN model. *Nucl. Power Eng. (in Chinese)* **43**(2), 1–8 (2022). <https://doi.org/10.13832/j.jnpe.2022.02.0001>
41. J. Wang, X. Peng, Z. Chen et al., Surrogate modeling for neutron diffusion problems based on conservative physics-informed neural networks with boundary conditions enforcement. *Ann. Nucl. Energy* **176**, 109234 (2022). <https://doi.org/10.1016/j.anucene.2022.109234>
42. Y. Yang, H. Gong, S. Zhang et al., A data-enabled physics-informed neural network with comprehensive numerical study on solving neutron diffusion eigenvalue problems. *Ann. Nucl. Energy* **183**, 109656 (2023). <https://doi.org/10.1016/j.anucene.2022.109656>
43. Y. Yang, H.L. Gong, Q.L. He et al., On the uncertainty analysis of the data-enabled physics-informed neural network for solving neutron diffusion eigenvalue problem. *Nucl. Sci. Eng.* **198**(5), 1075–1096 (2024). <https://doi.org/10.1080/00295639.2023.2236840>
44. K. Prantikos, L.H. Tsoukalas, A. Heifetz, Physics-informed neural network solution of point kinetics equations for a nuclear reactor digital twin. *Energies* **15**(20), 7697 (2022). <https://doi.org/10.3390/en15207697>
45. K. Prantikos, S. Chatzidakis, L.H. Tsoukalas et al., Physics-informed neural network with transfer learning (TL-PINN) based on domain similarity measure for prediction of nuclear reactor transients. *Sci. Rep.* **13**(1), 16840 (2023). <https://doi.org/10.1038/s41598-023-43325-1>
46. T. Wang, Z.J. Liu, P.C. Zhao et al., Design method of high-flux lead-bismuth cooled reactor neutron flux maximization based on BP neural network. *Nucl. Tech. (in Chinese)* **47**(10), 100602 (2024). <https://doi.org/10.11889/j.0253-3219.2024.hjs.47.100602>
47. Y. Wang, J.M. Lu, J.Y. Yao et al., Optimization of subchannel analysis for lead-bismuth reactor fuel assemblies. *Nucl. Tech. (in Chinese)* **47**(07), 070603 (2024). <https://doi.org/10.11889/j.0253-3219.2024.hjs.47.070603>
48. E. Kharazmi, Z. Zhang, G.E. Karniadakis, hp-VPINNs: variational physics-informed neural networks with domain decomposition. *Comput. Methods Appl. M* **374**, 113547 (2021). <https://doi.org/10.1016/j.cma.2020.113547>

49. S. Mirjalili, A. Lewis, The whale optimization algorithm. *Adv. Eng. Softw.* **95**, 51–67 (2016). <https://doi.org/10.1016/j.advengsoft.2016.01.008>
50. C. Cheng, G.T. Zhang, Deep learning method based on physics informed neural network with resnet block for solving fluid flow problems. *Water* **13**(4), 423 (2021). <https://doi.org/10.3390/w13040423>
51. P. Zhang, Y. Hu, Y. Jin et al., A Maxwell's equations based deep learning method for time domain electromagnetic simulations. *IEEE J. Multiscale Multiphys. Comput. Tech.* **6**, 35–40 (2021). <https://doi.org/10.1109/JMMCT.2021.3057793>
52. S. Wang, Y. Teng, P. Perdikaris, Understanding and mitigating gradient flow pathologies in physics-informed neural networks. *SIAM J. Sci. Comput.* **43**(5), A3055–A3081 (2021). <https://doi.org/10.1137/20M1318043>
53. S.N. Atluri, T. Zhu, A new meshless local Petrov–Galerkin (MLPG) approach in computational mechanics. *Comput. Mech.* **22**(2), 117–127 (1998). <https://doi.org/10.1007/s004660050346>
54. D. Mishkin, N. Sergievskiy, J. Matas, Systematic evaluation of convolution neural network advances on the imagenet. *Comput. Vis. Image Underst.* **161**, 11–19 (2017). <https://doi.org/10.1016/j.cviu.2017.05.007>
55. S. Mirjalili, S.M. Mirjalili, A. Lewis, Grey wolf optimizer. *Adv. Eng. Softw.* **69**, 46–61 (2014). <https://doi.org/10.1016/j.advengsoft.2013.12.007>
56. L.A. Hageman, J.B. Yasinsky, Comparison of alternating-direction time-differencing methods with other implicit methods for the solution of the neutron group-diffusion equations. *Nucl. Sci. Eng.* **38**(1), 8–32 (1969). <https://doi.org/10.13182/NSE38-8>
57. E.J. Pebesma, B.M.H. Gerard, Latin hypercube sampling of Gaussian random fields. *Technometrics* **41**(4), 303–312 (1999). <https://doi.org/10.1080/00401706.1999.10485930>
58. M. Imron, Development and verification of open reactor simulator ADPRES. *Ann. Nucl. Energy* **133**, 580–588 (2019). <https://doi.org/10.1016/j.anucene.2019.06.049>
59. J. Alnaqbi, D. Hartanto, R. Alnuaimi et al., Static and transient analyses of Advanced Power Reactor 1400 (APR1400) initial core using open-source nodal core simulator KOMODO. *Nucl. Eng. Tech.* **54**(2), 764–769 (2022). <https://doi.org/10.1016/j.net.2021.08.012>
60. J. Ge, D. Zhang, W. Tian et al., Steady and transient solutions of neutronics problems based on finite volume method (FVM) with a CFD code. *Prog. Nucl. Energy* **85**, 366–374 (2015). <https://doi.org/10.1016/j.pnucene.2015.07.012>

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.