

Development and verification of the higher-order mode neutron flux calculation code HARMONY2.0

Er-Pin Zhang^{1,2} · Zi-Ning Ni^{1,2} · Nian-Biao Deng^{1,2} · Jin-Sen Xie^{1,2} · Yong Liu³ · Tao Yu^{1,2}

Received: 9 April 2024 / Revised: 9 July 2024 / Accepted: 28 July 2024 / Published online: 22 December 2024 © 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 2024

Abstract

Higher-order modes of the neutron diffusion/transport equation can be used to study the temporal behavior of nuclear reactors and can be applied in modal analysis, transient analysis, and online monitoring of the reactor core. Both the deterministic method and the Monte Carlo (MC) method can be used to solve the higher-order modes. However, MC method, compared to the deterministic method, faces challenges in terms of computational efficiency and α mode calculation stability, whereas the deterministic method encounters issues arising from homogenization-related geometric and energy spectra adaptation. Based on the higher-order mode diffusion calculation code HARMONY, we developed a new higher-order mode calculation code, HARMONY2.0, which retains the functionality of computing λ and α higher-order modes from HARMONY1.0, but enhances the ability to treat complex geometries and arbitrary energy spectra using the MC-deterministic hybrid two-step strategy. In HARMONY2.0, the mesh homogenized multigroup constants were obtained using OpenMC in the first step, and higher-order modes were then calculated with the mesh homogenized core diffusion model using the implicitly restarted Arnoldi method (IRAM), which was also adopted in the HARMONY1.0 code. In addition, to improve the calculation efficiency, particularly in large higher-order modes, event-driven parallelization/domain decomposition methods are embedded in the HARMONY2.0 code to accelerate the inner iteration of λ/α mode using OpenMP. Furthermore, the higher-order modes of complex geometric models, such as Hoogenboom and ATR reactors for λ mode and the MUSE-4 experiment facility for the prompt α mode, were computed using diffusion theory.

Keywords Neutron diffusion equation \cdot Higher-order modes \cdot Global homogenization \cdot Two-step method \cdot Domain decomposition

This work was supported by the National Natural Science Foundation of China (No. U2267207) and Science and Technology on Reactor System Design Technology Laboratory (No. KFKT-05-FWHTWU-2023004).

Jin-Sen Xie jinsen_xie@usc.edu.cn

⊠ Tao Yu yutao29@sina.com

¹ School of Nuclear Science and Technology, University of South China, Hengyang 421000, China

- ² Key Laboratory of Advanced Nuclear Energy Design and Safety, Ministry of Education, Hengyang 421000, China
- ³ Science and Technology on Reactor System Design Technology Laboratory, Nuclear Power Institute of China, Chengdu 610213, China

1 Introduction

The eigenvalue problem of the neutron transport/diffusion equations is one of the most problematic issues in reactor physics. By solving the λ eigenvalue equation, the effective multiplication factor k_{eff} and the neutron flux distribution (i.e., the fundamental λ mode) at the critical state of a reactor can be obtained. On the other hand, by solving the α eigenvalue equation, a reactor's asymptotic characteristics of neutron multiplication behavior with time can be obtained. In reactors deviating from criticality, apart from the fundamental mode, a higher-order mode neutron flux exists, which together (fundamental and higher-order modes) reflect the time-space characteristics of the neutron flux in the reactor. The application of higher-order modes has been extended to several research areas in reactor physics, including modal analysis [1], transient analysis [2], online monitoring [3],

higher-order reactor noise analysis [4], and higher-order perturbation theory [5].

Calculation methods for the higher-order modes of the neutron transport/diffusion equation are mainly divided into deterministic and Monte Carlo (MC) methods. In deterministic methods, subspace methods, including the implicitly restarted Arnoldi method (IRAM) [6] and subspace iteration (SSI) [7] algorithms, are typically used to solve multigroup transport/diffusion equations. The MC method is primarily based on the fission matrix method and its derivative methods, in which the fission matrix is obtained by the MC transport process, and the higher-order neutron flux is then calculated by solving the eigenvalue problem of the fission matrix [8, 9].

A major advantage of deterministic methods is the variety of numerical solvers available for large sparse matrices. These methods have led to many studies on the calculation of higher-order mode neutron fluxes, particularly for λ mode. However, research on solving higher-order λ modes based on deterministic methods has mainly focused on simple geometric problems or core problems with homogenized group constants of assemblies. For example, Wu et al. used the IRAM method to solve IAEA-3D and PHWR problems [6]. Bernal et al. used the Krylov-Schur method to solve the IAEA-3D and hexagonal lattice core VVER-1000 3D problems [10]. Morato et al. solved the BWR and C5G7 problems based on the Krylov-Schur method using homogenized fuel assemblies as the computational mesh and the finite difference and discrete coordinates method as an approximation [11]. To make the deterministic higher-order modes calculation applicable to complex cores, Abrate et al. used the "hybrid two-step method" of MC assembly homogenization and core deterministic higher-order mode calculation to obtain the λ higher-order modes of C5G7 and UAM benchmarks with the parallel SSI algorithm (PSSI) to speed up [7]. Compared with λ mode, there is relatively less research related to α mode, and the models of α mode calculations are usually similar to or simpler than those of the λ mode. For example, Verdú et al. solved the PHWR problem [12], and Avvakumov et al. solved the hexagonal lattice core VVER-1000 and HWR based on homogenized two-group constants [13]. In addition, the α -mode demonstrates lower computing efficiency than the λ -mode. In previous studies, to study the spatial effect induced by higher-order modes in an external neutron source-driven subcritical reactor, we developed the deterministic higher-order mode diffusion code HAR-MONY1.0, based on the IRAM algorithm [14].

In contrast, the MC method has significant advantages when dealing with complex geometries. Carney et al. obtained λ modes using MC models such as 2D-PWR, the Hoogenboom MC performance benchmark, fuel storage vault problem (FVP), and the advanced test reactor (ATR) model using the fission matrix method [15]. To speed up the calculation of MC, Gupta et al. implemented the SSI algorithm based on the MC method to solve higher-order λ modes [16]; however, it still required thousands of seconds of CPU time for one-dimensional problems. In their subsequent research, the solution time for the Takeda 3D LWR reached as high as 250,000 s of CPU time [17]. Additionally, a higher-order α - mode solution based on the MC method has also been studied. Owing to their low computational efficiency, MC calculations are typically used in relatively simple models, with a significant portion being one-dimensional. Betzler et al. used the transient rate matrix method (TRMM) to solve the forward/adjoint α mode in one-dimensional media [8]. Vitali et al. obtained the first few orders of α mode of a rod model and a continuous-energy transport model using α -k power iterations and the generalized iterated fission probability method [18]. Yamamoto employed the power iteration method to solve the one-dimensional five-region problem, and utilized a pulsed neutron method simulation to investigate the time characteristics of the first three α modes [19]. Betzler et al. utilized the TRMM method to solve the large hexagonal lattice core of the Fort St. Vrain (FSV) reactor and validated the numerical results with experimental α [20]. Vitali et al. implemented a matrixfilling Monte Carlo method in TRIPOLI-4 and applied it to solve the EOLE reactor problem [21]. The fission matrix method and its derivatives face challenges related to memory usage. Taking a 3D problem as an example, and assuming that there are 10^2 meshes for space, energy, and angle, the total number of meshes is 10^{10} . Consequently, the number of elements contained in the fission matrix would amount to 10²⁰. Given that each element is a double-precision floating-point number, both current and foreseeable computing resources would struggle to handle the memory overhead required for such a large amount of data [18].

To improve the adaptability of deterministic higher-order modes calculation method in complex geometry reactors and further accelerate the calculation speed, we developed a code system HARMONY2.0 based on the concept of "hybrid two-step method" by combining the Monte Carlo and deterministic methods, utilizing OpenMC [22] coupled with HARMONY1.0. HARMONY2.0 retains the functionality of computing higher-order λ and α modes and their adjoint modes from HARMONY1.0 and, as well as the parallel acceleration of higher-order mode calculations through event-driven domain decomposition parallelization.

The remainder of the paper is organized as follows. Section 2 provides an overview of several key topics, including the theory and computational methods of higher-order harmonic neutron diffusion equations, the Monte Carlo homogenization method based on OpenMC, and the principles of parallel acceleration achieved using OpenMP. In addition, it presents an introduction to the HARMONY2.0 code. The validation results are presented in Sect. 3, including the verification of

....

parallel computing and the validation of HARMONY2.0 for higher-order mode solutions in complex geometric reactor cores.

2 Methods and methodologies

2.1 Eigenvalue problems of neutron diffusion equation

The neutron diffusion equation is expressed as follows:

$$\frac{1}{v}\frac{\partial\phi}{\partial t} = -L\phi + (1-\beta)\chi_{\rm p}F\phi + \sum_{i=1}^{0}\chi_{{\rm d},i}\lambda_iC_i,\tag{1}$$

$$\frac{\partial C_i}{\partial t} = \beta_i F \phi - \lambda_i C_i, i = 1, 2, \cdots, 6$$
⁽²⁾

where v represents the neutron velocity, ϕ denotes the neutron flux, t stands for time, β represents the total delayed neutron fraction, χ_p signifies the prompt neutron fission spectrum, $\chi_{d,i}$ represents the delayed neutron fission spectrum for the *i*th group, λ_i denotes the decay constant of the *i*th group precursor nucleus, c_i represents the concentration of the *i*th group precursor nucleus, and β_i represents the fraction of delayed neutrons for the *i*th group. The operators L and F are defined as follows:

$$L\phi = -\nabla \cdot D(r, E)\nabla\phi(r, E, t) + \Sigma_{t}(r, E)\phi(r, E, t) - \int_{0}^{\infty} \Sigma_{s}(r, E' \to E)\phi(r, E', t)dE'$$
(3)

$$F\phi = \int_0^\infty v\Sigma_{\rm f}(r, E')\phi(r, E', t){\rm d}E'$$
(4)

By introducing a balance factor of $1/\lambda$ and adjusting the right-hand side fission neutron source term to make the system pseudo-critical, the time derivative term is eliminated, resulting in a steady-state λ -mode neutron diffusion equation, which can be expressed using the following matrices:

$$L\phi = \frac{1}{\lambda}\chi F\phi \tag{5}$$

where $\chi = (1 - \beta)\chi_p + \sum_{6}^{i=1} \beta_i \chi_{d,i}$ is the total (or average) fission neutron spectrum. The neutron diffusion equation, which considers only prompt neutrons and disregards delayed neutrons, is as follows:

$$\frac{1}{v}\frac{\partial\phi}{\partial t} = -L\phi + (1-\beta)\chi_{\rm p}F\phi.$$
(6)

Supposing that the solution to Eq. (6) is

$$\phi_{\alpha,\mathrm{p}}(r,E)e^{\alpha t}.\tag{7}$$

By substituting Eq. (7) into Eq. (6), the neutron diffusion equation for the prompt α -mode can be obtained using Eq. (8):

$$\frac{\alpha}{\nu}\phi_{\alpha} = -L\phi_{\alpha} + (1-\beta)\chi_{\rm p}F\phi_{\alpha}.$$
(8)

The neutron diffusion equations of λ mode and prompt α mode can be solved mathematically by the eigenvalue problem in the form of $Ax = \lambda x$ or $Ax = \lambda Bx$, yielding a series of discrete eigenvalues $\lambda_1, \lambda_2, \dots, \lambda_n, \alpha_1, \alpha_2, \dots, \alpha_n$ along with the corresponding eigenvectors (modes) $\phi_{\lambda,1}, \phi_{\lambda,2}, \dots, \phi_{\lambda,n}$ $\phi_{\alpha,1}, \phi_{\alpha,2}, \dots, \phi_{\alpha,n}$. The largest eigenvalue is referred to as the fundamental eigenvalue. In λ mode, the fundamental eigenvalue carries the physical meaning of the effective multiplication factor $k_{\rm eff}$, and the associated fundamental eigenvector represents the neutron flux distribution in precise or pseudo-critical systems. The λ mode is typically used to characterize the reactivity and neutron flux distribution of reactors near criticality. In contrast, the α eigenvalue can be either a positive or negative real number, indicating that the reactor system is in a supercritical or subcritical state. The α -eigenvalue can also represent the evolution behavior of the unsteady neutron flux of the reactor over time [23]. Additionally, for noncritical systems, α and α modes are physically measurable quantities. (Typically, the fundamental eigenvalue of the prompt α mode is taken as the prompt neutron attenuation constant.)

The corresponding adjoint equations for the above λ and α eigenvalue equations are as follows:

$$L^{+}\phi_{\lambda}^{+} = \frac{1}{\lambda}(\chi F)^{+}\phi_{\lambda}^{+}$$
(9)

$$\frac{\alpha}{v}\phi_{a}^{+} = -L^{+}\phi_{a}^{+} + (1-\beta)(\chi_{p}F)^{+}\phi_{a}^{+}$$
(10)

The λ and prompt α adjoint equations have the same eigenvalues as the forward equations, where the λ adjoint flux also has the physical significance of "neutron importance" and can be used to calculate kinetic parameters such as the effective fraction of delayed neutrons [24, 25].

2.2 Methods for solving higher-order modes

In practice, when solving the higher-order modes of the neutron diffusion equation, matrix A in the eigenvalue problem ($Ax = \lambda x$) is often not directly constructed and solved. Instead, iterative methods such as fixed-source iterations combined with subspace iterations as outer iterative algorithms are employed to obtain higher-order modes. Taking IRAM as the outer algorithm, the calculation principle of the higher-order modes is as follows in Algorithm 1 [26, 27]. Here, operator A represents the matrix to be solved or a subfunction equivalent to that matrix.

Algorithm 1 Implicitly restarted Arnoldi method

Step A: Arnoldi process 1 initial guess vectors v_1 and $v_1 = r/||r||_2$ **2** for j=1 to m-1 do $z = Av_j$;// where Av_j represents the 3 fixed-source problem for *i=1 to j* do // modified Gram-Schmidt 4 process $h_{i,j} = (v_i, z) ;$ 5 $z = z - h_{i,j}v;$ 6 $h_{j+1,j} = ||z||_2;$ 7 if $h_{j+1,j} = 0$ then break; 8 else $v_{j+1} = z/h_{j+1,j};$ 9 10 end 11 end Step B: Implicit Restarting 12 following the k=m+p Arnoldi process, $AV_k = V_k H_k + f_k e_k^T$ 13 while $||Ax_i - \lambda x_i|| > tol, \forall i = 1 \cdots m \text{ do}$ compute the eigenvalue $\sigma(H_k)$ and select unwanted sets 14 $(\mu_1, \mu_2, \dots, \mu_p)$. for j=l to p do // remove unwanted sets to update $\sigma(H_m)$ by this process QR iterations with shifts of H_k using μ_i as the shift 15 value. $H_m \leftarrow Q^H H_m Q; V_m \leftarrow V_m Q;$ end 16 17 beginning with the new m-step Arnoldi process $AV_m = V_m H_m + f_m e_m^T;$ Compute the eigenpairs (λ_i, x_i) ; 18 if $||Ax_i - \lambda x_i|| < tol$ then break; 19 Apply p additional steps of the Arnoldi process to obtain 20 a new k=m+p steps the Arnoldi factorization $AV_k = V_k H_k + f_k e_k^T$ 21 end

The *m*th-order eigenvalues and eigenvectors obtained through the IRAM process are the *m*th-order modes required for the solution. This is described in Algorithm 1, the multiplication of matrices by subspace vectors (Av_j) corresponds to a fixed-source inner iteration. The pseudocodes for computing λ and prompt α modes are shown in Algorithms 2 and 3.

Algorithm 2 λ mode

Data: coefficient of finite difference a_a, b_a, \ldots **Input:** subspace vector of outer iteration $V_{input} = v_j$ **Output:** updated subspace vector $Av_j = V_{output}$ 1 the fission source calculation, $FS = \chi_g \sum_{i=1}^G (\nu \Sigma_f)_{g'} V_{\text{input}};$ 2 initial interiteration vector ϕ_{in} , and initial residual RES; **3 while** *RES* > *tolerance* **do** scatter source calculation $SS = \sum_{g'=1}^{G} \Sigma_{g' \to g} \phi_{\text{in}};$ 4 inner iteration $\phi'_{in} \leftarrow \phi_{in}$ with FS, SS, a_g, b_g, \ldots ; 5 if *RES* < *tolerance* then break; 6 7 else update inner iteration vector $\phi_{in} \leftarrow \phi'_{in}$; 8 end return $V_{\text{output}} = \phi_{\text{in}}$ 9

Algorithm 3 α mode

	Data: coefficient of finite difference a_g, b_g, \ldots						
	Input: subspace vector of outer iteration $V_{input} = v_j$						
	Output: updated subspace vector $Av_j = V_{\text{output}}$						
1	Alpha source calculation: $AS = 1/vV_{input}$. initial						
	inter-iteration vector ϕ_{in} , and initial residual RES ;						
2	while <i>RES</i> > <i>tolerance</i> do						
3	scatter source calculation $SS = \sum_{g'=1}^{G} \Sigma_{g' \to g} \phi_{in};$						
4	prompt fission source calculation:						
	$PFS = (1 - \beta)\chi_g \sum_{g'=1}^G (\nu \Sigma_f)_{g'} \phi_{\text{in}};$						
5	inner iteration $\phi'_{in} \leftarrow \phi_{in}$ with $AS, PFS, SS, a_g, b_g, \ldots$						
	;						
6	if <i>RES</i> < <i>tolerance</i> then break;						
7	else update inner iteration vector $\phi_{in} \leftarrow \phi'_{in}$;						
8	end						
9	return $V_{ m output} = \phi_{ m in}^{'}$						

In HARMONY2.0 code, the inner iteration is solved using the finite difference method. In Algorithms 2 and 3, a_g , b_g ,... represent the difference coefficients of the leakage terms after discretization. For example, in the *x*-direction, the inner node formula is given by Eq. (11).

$$a_{g}(i,j,k) = b_{g}(i-1,j,k) = -\frac{2D_{g}(i,j,k)D_{g}(i-1,j,k)\Delta y(j)\Delta z(k)}{D_{g}(i,j,k)\Delta x(i-1) + D_{g}(i-1,j,k)\Delta x(i)}$$
(11)

where a_g and b_g represent the x^- and x^+ directions, respectively. At the outer boundary with zero-flux boundary conditions, the corresponding formulas are given by Eq. (12).

$$a'_{g}(i,j,k) = -\frac{2D_{g}(i,j,k)}{\Delta x(i)} \Delta y(j) \Delta z(k)$$
(12)

For the y and z-directions, the subscripts j, k, \dots in Eq (11) and (12) are changed accordingly.

The discretized central difference equations of the iteration (in line 5 of Algorithm 2 and line 6 of Algorithm 3) are expressed in Eq. (13). Common numerical methods such as the Jacobi iteration and successive over-relaxation iteration can be used to solve these equations, although specific formulations are not provided here.

$$a_{g}(i,j,k)\phi_{g}(i-1,j,k) + b_{g}(i,j,k)\phi_{g}(i+1,j,k) + \dots + e_{g}(i,j,k)\phi_{g}(i,j,k) = S_{g}(i,j,k)$$
(13)

where $e_g(i, j, k)$ denotes the removal rate as shown in Eq. (14):

$$e_g(i,j,k) = \sum_{R,g}(i,j,k)\Delta x(i)\Delta y(j)\Delta z(k) - a_g(i,j,k) - b_g(i,j,k) - \dots$$
(14)

where $S_g(i, j, k)$ denotes the total number of source terms. The λ and prompt α modes have different expressions, as shown in Eq. (15). The corresponding formulas for *FS*, *SS*, *PFS*, and *AS* are presented in Algorithms 2 and 3.

$$S_{g}(i,j,k) = \begin{cases} FS_{g}(i,j,k) + SS_{g}(i,j,k) \\ AS_{g}(i,j,k) - PFS_{g}(i,j,k) - SS_{g}(i,j,k) \end{cases}$$
(15)

2.3 Homogenization by OpenMC

This subsection introduces the MC homogenization method used to obtain the space- and spectrum-homogenized group constants of the reactor.

Traditional two-step methods for reactor core calculations face challenges in accurately describing complex geometric configurations owing to spatial discretization limitations. Additionally, homogenization processes entail addressing intricate issues such as resonance treatment and leakage corrections. To address these concerns, deterministic methods rely on assumptions and approximations. In contrast, MC homogenization methods utilize tallying functions to calculate neutron reaction rates statistically based on continuous-energy cross-sectional data. The group constants are then obtained using principles such as the conservation of reaction rates. This method not only allows for an accurate description of various geometrical configurations but also avoids challenges such as resonance treatment. Several MC codes have functionalities developed to generate group constants for coupling with downstream deterministic codes, including RMC [28], Serpent [29], OpenMC [30, 31], McCard [32], and MVP [33].

In spatial homogenization using traditional approaches, the reactor core is treated as single assemblies or individual fuel rod lattices. Reflective boundary conditions were assumed during homogenization [34]. Because the traditional assembly or lattice homogenization approach neglects the actual boundary conditions of the assembly or lattice, necessary correction methods or special treatments are inevitable when using these homogenized group constants for core calculations [35]. In contrast, due to its geometric flexibility, MC homogenization can not only implement assembly homogenization (infinite lattice homogenization [36]) with reflective boundary conditions but also achieve global homogenization with "real" boundary conditions [37] and "mesh" homogenization for generating group constants based on a mesh with "real" boundary conditions, among other types of homogenization.

Infinite lattice homogenization using the MC method, similar to deterministic assembly homogenization, involves modeling individual assemblies and then calculating the homogenized group constants using the full reflective boundary conditions of each type of assembly, as shown in Fig. 1a. However, global homogenization establishes a model around the assembly or even the entire core to obtain the homogenized group constants for individual assemblies, as illustrated in Fig. 1b. Mesh homogenization, however, directly divides the core space into meshes, calculating the homogenized constants for each mesh under "real" boundary conditions. Given the flexibility in adjusting the mesh parameters, mesh homogenization offers greater applicability in geometric processing. The difference in the steps of the full-core calculation between the infinite lattice or global homogenization and mesh homogenization is demonstrated in Fig. 2.

By utilizing mesh homogenization, the critical calculation, tallying, and cross-sectional data treatment required for subsequent computations can be efficiently achieved in a single run of the MC calculation, without the need to perform homogenization calculations for each assembly separately. Furthermore, the group constant data obtained through mesh homogenization can be stored in accordance with the mesh index, which facilitates coupling with downstream deterministic core calculation programs.

2.4 Parallel using OpenMP

In this subsection, we discuss the inner iteration acceleration process by using event-driven parallelization and domain decomposition based on OpenMP [38]. The differences between the two types of parallelization were also discussed.

In the context of higher-order mode calculations or subsequent research, improving the computational speed is very important. In the process from MC homogenization to higher-order mode core calculation, several steps can be parallelized for acceleration, such as critical calculation, inner flux iteration, and the last step for eigenpairs solution. Taking the example of solving the prompt α modes of a PHWR model with a mesh division of $18 \times 18 \times 10$ and utilizing HARMONY1.0 for serial computation under two-group conditions with a subspace number of 50, the inner iteration process requires approximately 760 s of CPU time. With 200 subspace vectors and an order of 100, it took more than 4000 s. Such rates are unacceptable for studies requiring higher-order mode expansion and reconstruction analyses requiring large number of higher-order modes. (All the calculations in this study were performed on a Linux system equipped with an Intel i5-13600K CPU and 16 GB of RAM.)

2.4.1 Event-driven parallelization

One approach to parallelization that is easily achieved using OpenMP is to parallelize individual code blocks (events). For instance, when writing serial code, it is necessary to nest multilayer FOR loop code blocks to calculate the scattering





Fig. 2 (Color online) Core calculation with different assembly homogenized group constants. **a** Infinite lattice homogenization; **b** mesh homogenization

source terms during the inner iteration. However, in a serial execution, each layer loop must be traversed sequentially. In reality, the computation of terms (such as the scatter source) in individual meshes has no inherent dependency, and can be performed simultaneously. This presents a natural opportunity for parallelization.

Similarly, parallel acceleration can be achieved for events with nested loops such as total source calculations and flux iterations. The events and "communication" nodes of the parallel computation achieved using this principle-based approach (hereafter referred to as event-driven parallelization) are depicted as Fig. 3a.

The event-driven approach requires a certain amount of time for thread-task communication and waiting between multiple parallel computation steps. Thread waiting primarily occurs because of disparities in workload and performance variations among threads, resulting in a load

🖄 Springer

imbalance [39]. The time consumed by thread-task communication is primarily attributed to the distribution of computational tasks among threads and the aggregation of computational results from different threads. As the number of threads increases, the time required for thread communication inevitably increases and the load imbalance becomes more severe. Both factors contribute to the reduced parallel efficiency. In certain cases of event-driven parallelization, this can result in slower performance compared to serial execution.

2.4.2 Domain decomposition parallelization

Event-driven parallelization divides parallel regions based on each computation event because of the existence of dependencies between different computation events. For instance, the total source calculation must wait for the



Fig. 3 Two types of parallelization. a Event-driven; b domain decomposition

completion of the scatter source calculation before proceeding, necessitating a thread waiting between these two computational events. However, although dependencies exist between different computational events, they exist only in the same mesh. For example, the total source calculation at the mesh point (1, 1, 1) must wait for its scatter source calculation to be completed, but the total source at the mesh point (3, 3, 3) does not depend on the completion of the scatter source at (1, 1, 1). This gives rise to the "domain decomposition" parallel method [6, 40].

The principle and schematic diagram of domain decomposition parallel computation is as follows: Consider all spatial meshes (or energy groups) as the "all region" that needs iterative computation. Based on this, divide the "all region" into N subdomains, where N threads are responsible for the iterative computation in each subdomain (as Fig. 4). This approach ensures that thread-task communication occurs only at the beginning and after each thread completes its task [41], thereby significantly reducing its frequency. The events and "communication" nodes of the domains decomposition are as Fig. 3b.

2.5 The HARMONY2.0 code

Based on the above principles, we developed an MCdeterministic hybrid two-step method for the higher-order mode calculation code HARMONY 2.0, building upon HARMONY1.0. The outer iteration employs the IRAM algorithm, whereas the inner iteration module is recoded and uses Python modules to couple with the mgxs API of OpenMC. The program modules and overall computational process are outlined in Fig. 5.

The complete computational process is as follows: begin by constructing the reactor model using OpenMC and defining the parameters for mesh homogenization, such as mesh structure, energy groups, and cross-sectional types. Subsequently, execute the critical calculation to generate homogenized cross-sectional data. Afterward, use the Python coupling modules to connect with Fortran modules, which include calculations for difference coefficients, result processing, and the inner iteration module. Finally, compile the Fortran modules for computing higher-order modes. Notably, Python modules require the reading of a simple text file containing information about the geometry mesh/ energy bins. Furthermore, a few modules can be invoked



Fig. 4 (Color online) Schematic of domain decomposition

to generate HARMONY2.0's inner iteration modules for parallel computation or to choose the solution mode (λ or prompt α mode; forward, adjoint calculation). Table 1 lists the names and functions of the Python modules.

3 Results and analysis

3.1 Comparison of serial and parallel computing

The α mode of the PHWR-3D-2 G problem was chosen for computational testing to compare the results and acceleration effects of serial and parallel computations. The number of subspace vectors was set as 200 to compute the first 100 higher-order modes. The PHWR-3D model ($19 \times 19 \times 10$) shown in Fig. 6 and the first 10 order's prompt α values are listed in Table 2. A time comparison between the event-driven domain decomposition and serial computations is presented in Table 3. In the case of parallel computation using domain decomposition, the subdomain partitions were as follows: two threads were



Fig. 5 (Color online) Schematic diagram of HARMONY 2.0 code system structure and the overall calculation flow

		Version	Function		
]	MESH_MXS_Former	λ /prompt α	Process mesh homogenized group constants		
	COE_Former		Generate difference coefficient calculation code		
2	Solver_Former	λ /prompt α ; forward/ adjoint; parallel	Generate iterative computing modules in different modes		
J	Editor		Generate eigenpairs post-processing files		



Fig. 6 (Color online) PHWR reactor. a xy direction; b xz direction

Table 2 Prompt α mode result of PHWR (first 10)

Order	α (s ⁻¹)						
	Ref. [42]	Serial	Event-driven	Domain decomposition			
1	-3.5650	-3.3872	-3.3872	-3.3872			
2	-1.8335×10^{1}	-1.8297×10^{1}	-1.8297×10^{1}	-1.8297×10^{1}			
3	-1.8335×10^{1}	-1.8297×10^{1}	-1.8297×10^{1}	-1.8297×10^{1}			
4	-3.8560×10^{1}	-3.8537×10^{1}	-3.8537×10^{1}	-3.8537×10^{1}			
5		-4.3540×10^{1}	-4.3540×10^{1}	-4.3540×10^{1}			
6		-4.5691×10^{1}	-4.5691×10^{1}	-4.5691×10^{1}			
7		-5.2802×10^{1}	-5.2802×10^{1}	-5.2802×10^{1}			
8		-5.2802×10^{1}	-5.2802×10^{1}	-5.2802×10^{1}			
9		-7.7179×10^{1}	-7.7179×10^{1}	-7.7179×10^{1}			
10		-7.8292×10^{1}	-7.8292×10^{1}	-7.8292×10^{1}			

allocated with $9 \times 18 \times 10$, four threads with $9 \times 9 \times 10$, eight threads with $9 \times 9 \times 5$, and 16 threads with $9 \times 9 \times 2$ or 3 (*z*-direction).

The results indicate that serial and parallel computations for both methods yielded the same values. The acceleration ratio of event-driven parallelization reached a maximum of 1.95 with five threads but decreased beyond five threads. In contrast, domain decomposition parallelization outperforms event-driven parallelization significantly, with a maximum acceleration ratio of 6.16, which was achieved using 16 threads.

Computational experiments were conducted using a relaxation factor of 1.0. However, adjusting this factor could potentially further accelerate the computation rates. For instance, in the case of domain decomposition with 16 threads, changing the relaxation factor to 1.05 reduced the



 Table 3 Computation time of PHWR (first 100 modes)

Threads	Time of inner iteration (s)						
	Event-driven	Speed up ratio	Domain decomposition	Speed up ratio			
1	1908.79						
2	1158.54	1.65	883.39	2.16			
3	1126.03	1.70					
4	1135.15	1.68	516.85	3.69			
5	976.67	1.95					
6	995.08	1.92					
7	1194.73	1.60					
8	1133.10	1.68	376.98	5.06			
16	1464.02	1.30	310.07	6.16			

computational time from 310.07 to 298.62 s. In addition, the HARMONY code was rewritten and ported to the Linux platform. This reduced the time for serial computations from over 4000 s to 1908.79 s, compared with HARMONY1.0. Therefore, with the parallelization implemented in HARMONY2.0, the computational speed increases appreciably.

3.2 Validation of C5G7 model

As shown in Table 2, the first four-order results of PHWR agree well with the reference values. However, because the validation of PHWR was obtained directly through deterministic calculations, it is still necessary to validate the hybrid two-step method used in this study before further discussion. Given the limited number of related benchmarks, we selected the C5G7 model for validation.

In the homogenization calculations, the criticality calculation parameters were set as follows: 20,000 generations

with the first 500 inactive and 100,000 particles per cycle for C5G7-2D and 500,000 particles per cycle for C5G7-3D. During the homogenization and mode calculations, the mesh divisions for C5G7-2D is 51×51 , and the mesh division for C5G7-3D is $51 \times 51 \times 10$. The models used in OpenMC were 1/4 or 1/8 symmetric, therefore, reflective boundary conditions were employed in the mode calculations. The k_{eff} values of OpenMC were 1.18640(\pm 5.6 pcm) and 1.18378(\pm 0.8 pcm), respectively. The C5G7 model and simplified calculation process are shown in Fig. 7.

The calculation results are presented in Table 4, where the reference values were obtained from Gupta et al. and calculated using the SSI-based MC method [17]. Here, we observed a difference between the HARMNOY2.0 computed values and reference values. The difference was 267 pcm for the fundamental eigenvalue of C5G7-2D and 253 pcm for C5G7-3D. As the order increases, the differences gradually increase, which could be due to several reasons such as differences in algorithms and discrepancies between diffusion and transport theories. One reason for this is that the settings of the number of meshes lead to differences in the eigenvalue solutions, particularly for higher orders. This will be discussed in the following subsections.

3.3 Verification of complex geometry computation

To validate the applicability of HARMONY2.0, to complex geometric reactor cores, we selected the Hoogenboom problem [43], Advanced Test Reactor (ATR) [22, 46], and MUSE-4 experimental facility [47] as test cases. This subsection discusses the computational results for the three

Table 4	λ mode	results	of	C5G7
---------	----------------	---------	----	------

Order	C5G7-2D		C5G7-3D		
	Ref	HARMONY2.0	Ref	HARMONY2.0	
1	1.18637	1.18370	1.18328	1.18075	
2	0.91623	0.90220	1.15969	1.16073	
3	0.87625	0.86260	1.11468	1.12420	
4	0.72468	0.70179	1.05192	1.07698	
5	0.59157	0.56272	0.97627	1.02564	
6	0.59088	0.56207	0.91421	0.97632	
7	0.49873	0.46282	0.89875	0.93382	
8	0.49651	0.46461	0.89274	0.90061	
9	0.36976	0.33206	0.87448	0.90159	
10	0.36860	0.33114	0.86528	0.88743	

models. Models of the Hoogenboom and ATR are shown in Fig. 8, and the MUSE-4 model is illustrated in Fig. 13.

3.3.1 Hoogenboom

The Hoogenboom problem, proposed by Hoogenboom et al., is an MC performance benchmark comprising 241 fuel assemblies with the same enrichment level. Each assembly consisted of fuel rods arranged in a 17×17 mesh and included 25 guide tubes. The axial zone of the core is divided into cold and hot regions [43]. In this study, Hoogenboom problem is chosen as a typical square lattice core model for solution.

The computational results for the different spatial mesh configurations and energy group divisions are presented in Table 5. Radially, the model was divided into one or four meshes per assembly. The energy boundaries of the



Fig. 7 C5G7 model and calculation flow



Fig. 8 (Color online) Hoogenboom a and ATR b model

2-group energy structure were (0, 0.625, 1.96403×10⁷) in eV, and for 6-group, they were (0, 4.53999×10², 9.11822×10³, 6.73795×10⁴, 4.97871×10⁵, 2.23130×10⁶, and 1.96403×10⁷). For the 8-group configuration, the CASMO 8-group [44] energy structure was employed for the eight-group configuration. The MC critical calculation parameters were set to 1,000,000 particles per cycle, totaling 1150 cycles, with the first 150 cycles being inactive. The results showed that the differences between the fundamental eigenvalues and k_{eff} obtained from OpenMC were within 200 pcm. Specifically, the fundamental eigenvalue calculation for the 34 × 34 × 8 8-group configuration is 1.00126, differing by only 1 pcm from OpenMC.s. Furthermore, we obtained the fission matrix using RMC3.5

Table 5 λ mode results of Hoogenboom

and solved for higher-order modes using SciPy [45]. Table 5 presents the results along with those from [15].

As mentioned in the previous subsection, the division of meshes affects the calculation results for higher-order modes (similar to the fission matrix method [15]). The discrepancies in the results for the higher-order modes were partly due to differences in the diffusion calculations under different meshes. In addition, from a mathematical perspective, the calculation of higher-order modes is equivalent to an eigenvalue problem. Different mesh divisions affect the order of the matrix A, thereby altering the total number of corresponding eigenvalues. This also resulted in larger discrepancies in the higher-order eigenvalue results as the order increased across different mesh divisions. As shown in Table 5, the comparison indicates that the lack of an energy

Order	OpenMC	HARMONY2.0					RMC&SciPy	Results of [15].
		19×19×8 &2 g	19×19×8 &6 g	19×19×8 &8 g	34×34×8 &8 g	36×36×8 &8 g	34×34×8	42×42×20
1	1.00127	1.00293	1.00093	1.00031	1.00126	1.00059	1.00193	0.99919
2	±8.5 pcm	0.99402	0.99051	0.98961	0.99078	0.98991	0.98687	0.98483
3		0.99323	0.98861	0.98791	0.98808	0.98840	0.98659	0.98362
4		0.99263	0.98817	0.98766	0.98844	0.98816	0.97444	0.98469
5		0.98437	0.97792	0.97717	0.97763	0.97768	0.96766	0.96956
6		0.98353	0.97773	0.97679	0.97701	0.97729	0.96664	0.96950
7		0.98183	0.97482	0.97377	0.97474	0.97406	0.96011	0.96693
8		0.98032	0.97420	0.97324	0.97250	0.97302	0.95975	0.96591
9		0.97797	0.97138	0.97097	0.97093	0.97241	0.95954	0.96043
10		0.97553	0.96834	0.96706	0.96593	0.96740	0.94412	0.95671

group dimension leads to progressively larger discrepancies among the higher eigenvalues. By the 10th order, the fission matrix method results from RMC3.5 &SciPy are 0.94412, whereas those from HARMONY 2.0 are all above 0.96.

Figure 9 illustrates the mode distribution of Hoogenboom with a $34 \times 34 \times 8$ -mesh. The eigenvectors of different groups of fundamental modes exhibited similar shapes. Therefore, only the distributions of the eighth group at z = 4 for the first ten orders are displayed. As depicted in the Fig. 9, the first, seventh, and tenth orders exhibit fully symmetric distributions, whereas the second, third, fourth, fifth, and sixth orders exhibit a 1/2 symmetric distribution. The eighth and ninth orders displayed a 1/4 symmetric distribution.

3.3.2 ATR

The ATR is one of the few large research reactors worldwide and is characterized by high flux, numerous irradiation channels, and reactivity control through multiple drums [22, 46]. It utilizes special curved-plate fuel elements, which make it challenging to calculate higher-order modes using a deterministic method. Hence, we chose ATR as the complex geometry core model for HARMONY2.0 test.

Table 6 presents ATR's first ten eigenvalue calculation results with a $50 \times 50 \times 1$ mesh, equivalent to a mesh size of $2 \text{ cm} \times 2 \text{ cm} \times 125 \text{ cm}$. The group energy structure follows CASMO 8-group. The OpenMC critical calculation parameters were 1,000,000 particles for 1150 cycles and using the first 150 cycles were inactive. The computed k_{eff} for ATR was 1.02293. A comparison revealed an error of only 226 pcm for ATR. Additionally, no approximation corrections were applied during MC homogenization, and further adjustments to parameters such as the energy group structure were not made. Figure 10 illustrates the differences in the fundamental flux distributions among the 8th groups in ATR with a mesh division of $50 \times 50 \times 1$.

The fundamental neutron flux distribution of the ATR conformed to the characteristic neutron flux distribution of a thermal reactor. This distribution entails fast neutrons generated from fission and predominantly concentrated in the fuel region, whereas the distribution of epithermal neutrons tends to be relatively flat owing to diffusion. By contrast, thermal neutrons exhibit a lower flux in the fuel region because of absorption, which reflects the absorption effect of the fuel material. Notably, this distribution also highlights the neutron absorption by structural materials, as evidenced by the lower neutron flux in regions containing absorbers, such as the surrounding drums.

The 2nd to 9th orders of ATR's first and 8th group modes are shown in Fig. 11, exhibiting symmetrical distributions similar to 1/2 and 1/4 across the various orders.

The corresponding adjoint mode distributions for the 1st and 8th groups are shown in Fig. 12. A comparison revealed that the adjoint fundamental mode distribution

Nev	ATR			
	OpenMC	HARMONY2.0		
1	1.02293	1.02519		
2	±8.3 pcm	0.85252		
3		0.86857		
4		0.77553		
5		0.62682		
6		0.54278		
7		0.53784		
8		0.54650		
9		0.46770		
10		0.45797		



Fig. 9 (Color online) λ mode of Hoogenboom (g = 8, z = 4, order 1 - 10)



Fig. 10 (Color online) Fundamental λ mode of ATR (g = 1 - 8)



Fig. 11 (Color online) λ modes of ATR (group 1 &8, order 2–9)

Fig. 12 (Color online) adjoint λ modes of ATR (group 1 &8, order 1–8)

of the 8th group exhibits a pattern "opposite" to that of the forward fundamental distribution in the fuel region. In other words, the adjoint neutron flux is larger in the fuel region, reflecting the physical significance of the "neutron importance" associated with the adjoint neutron flux.

3.3.3 MUSE-4

Compared with the λ mode, the calculation results of the higher-order modes in the prompt α mode are limited

because of fewer experiments compared with critical benchmarks and lower computational efficiency. Therefore, we chose the MUSE-4 experimental facility for validation, which provides experimental data for prompt neutron attenuation constants [47].

The MUSE-4 subcritical experimental apparatus is a modification of the zero-power MASURCA facility that is expanded to accommodate a wider range of experiments. At its core, it utilizes MOX fuels in the active region, with a central aluminum tube extending toward the accelerator. This configuration serves as an external neutron source for simulating the target of an accelerator-driven system (ADS) reactor. Surrounding this central feature, the lead components replicate the spallation target of the ADS, coupled with a reflector layer composed of a sodium-stainless steel blend. The outermost layer consists of stainless steel shielding, which provides radiation protection. This setup allows control of the reactor's subcriticality by adjusting the loading of the fuel assemblies. When loaded with 976 fuel assemblies, its $k_{\rm eff}$ corresponds to 0.97 (Fig. 13).

The parameters for the OpenMC homogenization calculation were as follows: 500,000 particles, 100 inactive cycles, and 1000 active cycles. The computed k_{eff} is 0.96817. The parameters for the mesh homogenization are defined with a 6-group energy structure with energy boundaries of (0, 4.53999×10^2 , 9.11822×10^3 , 6.73795×10^4 , 4.97871×10^5 , 2.23130×10^6 , and 1.96403×10^7) in eV. The mesh spacing in the *xy* direction was 5.3 cm, whereas in the *z*-direction, it was divided into 10.3/12 cm. The numbers of mesh points were $28 \times 30 \times 14$ and $28 \times 30 \times 12$ for the two configurations, respectively. The computed eigenvalues for both the λ and prompt α modes for the first ten orders are presented in Table 7

The results indicate that the computed fundamental eigenvalues for both λ modes under the two different mesh configurations differ from the OpenMC calculation by less than 200 pcm. Similarly, the fundamental prompt α eigenvalue closely aligns with the experimental values. Figure 14 shows the computed first-order eigenvectors of λ and the corresponding α eigenvectors for a mesh configuration of $28 \times 30 \times 12$. The shapes of the fundamental distributions for the same energy group are similar for the λ and prompt α modes, differing only in the numerical values.

The computed results for the 2–9 orders of the first group's λ modes and the corresponding prompt α modes are shown in Fig. 15. In the λ -mode, the distributions for each order adhere approximately to symmetric distribution patterns, such as 1/2 and 1/4. Although the prompt α -mode still approximately satisfies the symmetric distribution patterns, its symmetry is less pronounced. In addition to the similarity in the fundamental distribution shapes, there are significant differences in the shapes of the higher-order modes between the λ and prompt α modes.

3.4 Computation time evaluation

Based on the cases discussed in the previous section, the execution times for each part of HARMONY2.0 are shown in Fig. 16. In these cases, event-driven parallel examples utilized six threads, whereas domain decomposition employed 16 threads.

The most time-consuming part of these cases was the MC homogenization calculation. (All cases in Fig. 16 uses 1,000,000 particles with 12 threads.) In our tests, some cases using 100,000 particles yielded the same fundamental eigenvalues as those using 1,000,000 particles. For example, the Hoogenboom problem, with a mesh division of 34×34×8 and 8 groups, required only 2710.89 s of MC calculation with 100,000 particles (1,000,000 required 25837.21 s). However, the use of fewer particles could lead to asymmetrical flux results because of the characteristics of MC calculations. Additionally, because MC homogenization is based on tally functions, if the particle numbers are not sufficiently large, the cross sections of some meshes might exhibit increased errors owing to insufficient particle counts or even result in a zero cross-sectional value. These issues may be resolved using variance-reduction techniques.

Fig. 13 (Color online) MUSE-4 model in OpenMC (with mesh) **a** x - y direction; **b** x - z direction



(a)



Table 7 MUSE-4 eigenvalue results (first 10) 100	OpenMC	Experiment value/(s ⁻¹) [47]	HARMONY2.0			
results (lifst 10)			$28 \times 30 \times 14$		$28 \times 30 \times 12$	
			λ	α/s^{-1}	λ	α/s^{-1}
	$k_{\rm eff} = 0.96817 \pm 3.2 \rm pcm$	$\alpha = -46308 (k_{\rm eff} = 0.97)$	0.96963	- 43430	0.96717	- 45561
			0.69617	- 72951	0.69353	- 73006
			0.60302	- 73376	0.60063	- 73452
			0.60091	- 74452	0.59986	- 74516
			0.46344	- 75652	0.46018	- 75803
			0.46288	- 75964	0.46095	- 76078
			0.44332	- 77311	0.44054	- 77471
			0.39024	- 77433	0.38549	- 77650
			0.34971	- 77657	0.35699	- 77784
			0.34700	- 81859	0.34327	- 81871



Fig. 14 (Color online) Fundamental mode of MUSE-4 (group 1–6, z = 7). a λ mode; b prompt α mode

In addition to the MC calculations, the most timeconsuming part is the inner iteration when calculating the first 10 modes with 20 subspace vectors. Taking the C5G7-3D tests as examples, the ratio of the time spent on inner iterations to that spent on other parts (mainly the DNEUPD function [26] and its associated calculations) was approximately 11.85:1. Our tests showed that at the 500th mode with 600 subspace vectors, the inner iteration takes approximately 1446.98 s, while the time for other parts increases to 701.54 s, bringing the ratio to 2.06:1.

Figure 16 also shows that domain decomposition achieved considerable acceleration in all test cases. Furthermore, the calculations in α mode are significantly slower than those in λ mode. In the case of the MUSE-4 model, despite using domain decomposition, the inner iterations for the first 10 modes required 1518.40 s (approximately 60.35 times that of the λ mode), and the time for the other parts increased to 51.87 seconds. Thus, it is evident that acceleration in the α mode is necessary.

4 Conclusion

Based on a hybrid two-step method of MC homogenization combined with deterministic higher-order mode calculations, we developed the HARMONY2.0 code with enhanced capabilities for handling complex geometric cores. Validation was performed by solving C5G7, Hoogenboom, ATR, and MUSE-4 cores, demonstrating that HARMONY2.0 can compute higher-order modes in both λ and prompt α modes for complex geometric cores. The computed fundamental eigenvalues for λ mode closely matched the results obtained from OpenMC, and the fundamental prompt α eigenvalues





(a)



(b)

for the MUSE-4 core aligned well with the experimental values.

Furthermore, to enhance the calculation speed of deterministic higher-order modes, we employed two acceleration methods: event-driven parallelization and domain decomposition parallelization based on OpenMP. The verification confirmed that the serial and parallel computations obtained the same results. Both methods achieve acceleration in solving higher-order modes, with domain decomposition significantly outperforming event-driven parallelization. In the case of event-driven parallelization, the computational speed initially increases with the number of threads but decreases below serial execution when the thread count exceeds a certain threshold, resulting in an overall acceleration ratio significantly lower than expected. For domain decomposition parallelization, the acceleration ratio continued to increase with the number of threads, up to 16.

The computational results also validated the feasibility of using the MC-deterministic hybrid two-step method to solve higher-order modes in complex geometric cores. Additionally, the development of the HARMONY2.0 code, based on this method could provide support for studies on the spatial effects induced by higher-order modes in reactors, as well as research requiring a large number of higherorder modes, such as mode expansions and reconstruction analyses. Moreover, HARMONY2.0 code offers a feasible approach for calculating the prompt neutron attenuation constants of reactors.





Author Contributions All authors contributed to the study conception and design. Material preparation, data collection, and analysis were performed by Er-Pin Zhang, Zi-Ning Ni, Nian-Biao Deng, Jin-Sen Xie, Yong Liu, and Tao Yu. The first draft of the manuscript was written by Er-Pin Zhang, 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.j00186.00406 and https://doi.org/10.57760/sciencedb.j00186.00406.

Declarations

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

References

- G. Verdú, D. Ginestar, Modal decomposition method for the BWR stability analysis using alpha modes. Ann. Nucl. Energy 67, 31–40 (2014). https://doi.org/10.1016/j.anucene.2013.07.035
- A. Carreño, A. Vidal-Ferràndiz, D. Ginestar et al., Modal methods for the neutron diffusion equation using different spatial modes. Prog. Nucl. Energy 115, 181–193 (2019). https://doi.org/10. 1016/j.pnucene.2019.03.040
- D. Ginestar, R. Miró, G. Verdú et al., Modal processing of the Local Power Range Monitors the signals in the BWR NPP. Ann. Nucl. Energy 38(11), 2441–2455 (2011). https://doi.org/10. 1016/j.anucene.2011.07.010
- S.M. Ayyoubzadeh, S.A. Hosseini, N. Vosoughi, Higher-order power reactor noise analysis: multigroup diffusion model. Ann. Nucl. Energy 111, 354–370 (2018). https://doi.org/10.1016/j. anucene.2017.09.003

- T. Sauzedde, P. Archier, F. Nguyen, Strengths and weaknesses of the modal expansion method for perturbation calculations in nuclear reactor physics. Ann. Nucl. Energy **193**, 110024 (2023). https://doi.org/10.1016/j.anucene.2023.110024
- W.B. Wu, Y.R. Yu, Q. Luo et al., Calculation of higher eigenmodes of forward and adjoint neutron diffusion equations using IRAM algorithm based on domain decomposition. Ann. Nucl. Energy 143, 107463 (2020). https://doi.org/10.1016/j.anucene. 2020.107463
- N. Abrate, G. Bruna, S. Dulla et al., Assessment of numerical methods for evaluating higher-order harmonics in the diffusion theory. Ann. Nucl. Energy 128, 455–470 (2019). https://doi.org/ 10.1016/j.anucene.2019.01.011
- S. Carney, F. Brown, B. Kiedrowski et al., Theory and applications of fission matrix method for continuous-energy Monte Carlo. Ann. Nucl. Energy 73, 423–431 (2014). https://doi.org/ 10.1016/j.anucene.2014.07.020
- B.R. Betzler, W.R. Martin, B.C. Kiedrowski et al., Calculate α -eigenvalues of one-dimensional media using the Monte Carlo method. J. Comput. Theor. Trans. 43(1-7), 38-49 (2014). https://doi.org/10.1080/00411450.2014.909851
- Á. Bernal, A. Hébert, J.E. Roman et al., A Krylov–Schur solution of the eigenvalue problem of the neutron diffusion equation was discretized using the Raviart-Thomas method. J. Nucl. Sci. Technol. 54(10), 1085–1094 (2017). https://doi.org/10.1080/00223131.2017.1344577
- S. Morató, A. Bernal, R. Miró et al., Calculation of λ modes of the multigroup neutron transport equation using discrete ordinates and the Finite Difference Method. Ann. Nucl. Energy 137, 107077 (2020). https://doi.org/10.1016/j.anucene.2019.107077
- G. Verdú, D. Ginestar, J. Román et al., 3D alpha modes of the nuclear power reactor. J. Nucl. Sci. Technol. 47(5), 501–514 (2010). https://doi.org/10.1080/18811248.2010.9711641
- A.V. Avvakumov, V.F. Strizhov, P.N. Vabishchevich et al., Spectral properties of the dynamic processes in nuclear reactors. Ann. Nucl. Energy (2017). https://doi.org/10.1016/j.anucene. 2016.09.021

- Z.N. Ni, J.S. Xie, M.A. Wasaye et al., Effect of higher-order Harmonics on the steady-state neutron flux in accelerator-driven subcritical reactors. Int. J. Energy Res. 2023, 14 (2023). https:// doi.org/10.1155/2023/4114886
- S. Carney, F. Brown, B. Kiedrowski et al., Fission matrix capability for MCNP Monte Carlo simulations (United States (US), 2012). https://doi.org/10.2172/1050495
- A. Gupta, R.S. Modak, Monte Carlo solution of k-eigenvalue problem using subspace iteration method. Nucl. Sci. Eng. 194(2), 87–103 (2020). https://doi.org/10.1080/00295639.2019. 1668655
- A.K. Mallick, A. Gupta, Subspace iteration-based Monte Carlo scheme for simultaneous computation of multiple dominant k-eigenmodes and acceleration of fission source iterations. Ann. Nucl. Energy 166, 108783 (2022). https://doi.org/10.1016/j.anuce ne.2021.108783
- V. Vitali, F. Chevallier, A. Jinaphanh et al., Spectral analysis using the direct and adjoint Monte Carlo methods. Ann. Nucl. Energy 137, 107033 (2020). https://doi.org/10.1016/j.anucene.2019. 107033
- T. Yamamoto, Higher-order α-mode eigenvalue calculation using Monte Carlo power iteration. PNST 2, 826–835 (2011). https:// doi.org/10.15669/pnst.2.826
- B.R. Betzler, B.C. Kiedrowski, W.R. Martin et al., Calculate the alpha eigenvalues and eigenfunctions using the Markov transition rate matrix Monte Carlo method. Nucl. Sci. Eng. **192**(2), 115–152 (2018). https://doi.org/10.1080/00295639.2018.1497397
- V. Vitali, F. Chevallier, A. Jinaphanh et al., Eigenvalue separation and eigenmode analysis using matrix-filling Monte Carlo method. Ann. Nucl. Energy 164, 108563 (2021). https://doi.org/10.1016/j. anucene.2021.108563
- P.K. Romano, N.E. Horelik, B.R. Herman et al., OpenMC: stateof-the-art Monte Carlo code for research and development. Ann. Nucl. Energy 82, 90–97 (2015). https://doi.org/10.1016/j.anuce ne.2014.07.048
- D. Brockway, P. Soran, P. Whalen, Monte Carlo eigenvalue calculations. Monte Carlo methods and applications in Neutronics, Photonics and statistical Physics, Cadarache Castle, Provence, France April 22–26, (Springer, Berlin, 1985), 378–387. https:// doi.org/10.1007/BFb0049064
- J. Lewins, Importance: The Adjoint function. Physical basis of variational and perturbation theories in transport and diffusion problems (1965). https://www.osti.gov/biblio/4752764
- Y. Nagaya, G. Chiba, T. Mori et al., Comparison of Monte Carlo calculation methods for effective delayed neutron fractions. Ann. Nucl. Energy 37(10), 1308–1315 (2010). https://doi.org/10.1016/j. anucene.2010.05.017
- R.B. Lehoucq, D.C. Sorensen, C. Yang, ARPACK user guide: solution to large-scale eigenvalue problems with implicitly restarted Arnoldi methods. Soc. Ind. Appl. Math. (1998). https:// doi.org/10.1137/1.9780898719628
- 27. L. Hogben, *Handbook of Linear Algebra*, 2nd edn. (Chapman and Hall/CRC, Boca Raton, 2006). https://doi.org/10.1201/b16113
- K. Wang, Z.G. Li, D. She et al., RMC-A Monte Carlo code for reactor core analysis. Ann. Nucl. Energy 82, 121–129 (2015). https://doi.org/10.1016/j.anucene.2014.08.048
- E. Fridman, J. Leppänen, The Serpent Monte Carlo code is used for few-group cross-section generation. Ann. Nucl. Energy 38(6), 1399–1405 (2011). https://doi.org/10.1016/j.anucene.2011.01.032
- Z. Liu, K. Smith, B. Forget et al., Cumulative migration method for computing rigorous diffusion coefficients and transport crosssections using the Monte Carlo method. Ann. Nucl. Energy 112, 507–516 (2018). https://doi.org/10.1016/j.anucene.2017.10.039
- 31. W. Boyd, A. Nelson, P.K. Romano et al., Multigroup cross-section generation using the OpenMC Monte Carlo particle transport

code. Nucl. Technol. **205**, 928–944 (2019). https://doi.org/10. 1080/00295450.2019.1571828

- H.J. Park, H.J. Shim, H.G. Joo et al., Generation of a few group diffusion theory constants using the Monte Carlo code McCARD. Nucl. Sci. Eng. **172**, 66–77 (2008). https://doi.org/10.13182/ NSE11-22
- Y. Nagaya, K. Okumura, T. Sakurai et al., MVP/GMVP Version 3. General-purpose Monte Carlo codes for neutron and photon transport calculations are based on continuous energy and multigroup methods. JAEA (2017). https://doi.org/10.11484/ jaea-data-code-2016-019
- S.C. Liu, G.B. Wang, J.G. Liang et al., Burnup-dependent core neutronics analysis of plate-type research reactor using deterministic and stochastic methods. Ann. Nucl. Energy 85, 830–836 (2015). https://doi.org/10.1016/j.anucene.2015.06.041
- S.C. Liu, G.B. Wang, G.C. Wu et al., Neutronics comparative analysis of the neutronics of plate-type research reactors using deterministic and stochastic methods. Ann. Nucl. Energy 79, 133–142 (2015). https://doi.org/10.1016/j.anucene.2015.01.027
- W.R.D. Boyd III, Reactor agnostic multigroup cross-section generation for fine-mesh deterministic neutron transport simulations. Diss. Massachusetts Institute of Technology, (2017). http://hdl. handle.net/1721.1/112525
- Q.Q. Pan, K. Wang, One-step Monte Carlo global homogenization based on RMC code. Nucl. Eng. Technol. 5, 1209–1217 (2019). https://doi.org/10.1016/j.net.2019.04.001
- OpenMP Architecture Review Board. OpenMP Application Programming Interface Version 4.0 [Fortran], OpenMP (2013). https://www.openmp.org/wp-content/uploads/OpenMP-4.0-Fortr an.pdf
- X. Zhang, S.C. Liu, Y.X. Chen, Development of Monte Carlo hybrid parallel algorithm and its application in high-resolution spatial energy flux distribution calculations. Nucl. Eng. Des. 413, 112500 (2023). https://doi.org/10.1016/j.nucengdes.2023.112500
- H.B. Zhang, H.C. Wu, L.Z. Cao, The acceleration technique for 2D MOC is based on Krylov subspace and domain decomposition methods. Ann. Nucl. Energy 38, 2742–2751 (2011). https://doi. org/10.1016/j.anucene.2011.08.015
- S.C. Liu, J.G. Liang, K. Wang et al., Development of an integrated parallelism strategy for large-scale depletion calculations in Monte Carlo code RMC. Ann. Nucl. Energy 135, 106941 (2020). https://doi.org/10.1016/j.anucene.2019.106941
- K.P. Singh, S.B. Degweker, R.S. Modak et al., Iterative method for obtaining the prompt and delayed alpha modes of the diffusion equation. Ann. Nucl. Energy 38(9), 1996–2004 (2011). https://doi. org/10.1016/j.anucene.2011.04.018
- J. E. Hoogenboom, W. Martin, B. Petrovic, Monte-Carlo performance benchmark test-aims, specifications, and first results. (2011) https://api.semanticscholar.org/CorpusID:62028541
- 44. VTT Technical Research Centre of Finland. CASMO 8-group Structure. Serpent Wiki, 29 Nov. 2017. https://serpent.vtt.fi/media wiki/index.php/CASMO_8-group_structure
- P. Virtanen, R. Gommers, T.E. Oliphant et al., SciPy 1.0: fundamental algorithms for scientific computing in Python. Nat. Methods 17(3), 261–272 (2020)
- 46. F.M. Marshall, Advanced test reactor—testing capabilities and plans and advanced test reactor national scientific user facility partnerships and networks. United States (2008). https://www.osti. gov/biblio/936621/
- Organisation for Economic Co-operation and Development— Nuclear Energy Agency. Benchmark for computer simulations of MASURCA critical and Subcritical experiments. ISBN 92-64-01086-6, (2005). https://www.oecd-nea.org/upload/docs/appli cation/pdf/2019-12/nsc-doc2005-23.pdf

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.