An Efficient Parallel Algorithm of the Variational Nodal Method\for Heterogeneous Neutron Transport Problems

Authors: Yin Han, Liu Xiaojing, Zhang Tengfei, Zhang Tengfei

Date: 2024-02-13T00:00:00+00:00

Abstract

The heterogeneous variational nodal method (HVNM) has emerged as a potential approach for solving high-fidelity neutron transport problems. However, achieving accurate results with HVNM in large-scale problems using high-fidelity models has been challenging due to the prohibitive computational costs. This paper presents an efficient parallel algorithm tailored for HVNM based on the Message Passing Interface standard. The algorithm evenly distributes the response matrix sets among processors during the matrix formation process, thus enabling independent construction without communication. Once the formation tasks are completed, a collective operation merges and shares the matrix sets among the processors. For the solution process, the problem domain is decomposed into subdomains assigned to specific processors, and the red-black Gauss-Seidel iteration is employed within each subdomain to solve the response matrix equation. Point-to-point communication is conducted between adjacent subdomains to exchange data along the boundaries. The accuracy and efficiency of the parallel algorithm are verified using the KAIST and JRR-3 test cases. Numerical results obtained with multiple processors agree well with those obtained from Monte Carlo calculations. The parallelization of HVNM results in eigenvalue errors of 31 pcm/-90 pcm and fission rate RMS errors of 1.22%/0.66%, respectively, for the 3D KAIST problem and the 3D JRR-3 problem. In addition, the parallel algorithm significantly reduces computation time, with an efficiency of 68.51% using 36 processors in the KAIST problem and 77.14% using 144 processors in the JRR-3 problem.

Full Text

Preamble

An Efficient Parallel Algorithm of the Variational Nodal Method for Heterogeneous Neutron Transport Problems

Han Yin,[1] Xiao-Jing Liu,[1] and Teng-Fei Zhang[1],† [1]School of Nuclear Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China

The heterogeneous variational nodal method (HVNM) has emerged as a promising approach for solving high-fidelity neutron transport problems. However, achieving accurate results with HVNM in large-scale problems using high-fidelity models has been challenging due to prohibitive computational costs. This paper presents an efficient parallel algorithm tailored for HVNM based on the Message Passing Interface (MPI) standard. The algorithm evenly distributes response matrix sets among processors during the matrix formation process, enabling independent construction without communication. Once formation tasks are completed, a collective operation merges and shares the matrix sets among processors. For the solution process, the problem domain is decomposed into subdomains assigned to specific processors, and red-black Gauss-Seidel iteration is employed within each subdomain to solve the response matrix equation. Point-to-point communication is conducted between adjacent subdomains to exchange data along boundaries. The accuracy and efficiency of the parallel algorithm are verified using KAIST and JRR-3 test cases. Numerical results obtained with multiple processors agree well with Monte Carlo calculations. The parallelization of HVNM yields eigenvalue errors of 31 pcm/-90 pcm and fission rate RMS errors of 1.22%/0.66% for the 3D KAIST problem and the 3D JRR-3 problem, respectively. Additionally, the parallel algorithm significantly reduces computation time, achieving 68.51% efficiency with 36 processors for the KAIST problem and 77.14% efficiency with 144 processors for the JRR-3 problem.

Keywords: Neutron transport, Variational nodal method, Parallelization, KAIST, JRR-3

Introduction

The solution of the neutron transport equation plays a pivotal role in analyzing neutron distribution within nuclear systems. In recent years, advancements in computational resources have spurred growing interest in one-step neutron transport methods that eliminate homogenization. The Method of Characteristics (MOC) [1, 2] has been identified as a promising approach for one-step whole-core neutronics calculations. The fundamental idea behind MOC is to generate a set of parallel rays for each discretized angle and solve the one-dimensional neutron transport equation along these rays. However, direct application of MOC to three-dimensional whole-core domains leads to prohibitively high computational costs. Consequently, a common practice is to employ the two-dimensional/one-dimensional (2D/1D) approximation, known as 2D/1D-MOC [3–5]. In 2D/1D-

MOC, coupling 2D MOC calculations in the lateral plane with diffusion or transport calculations in the axial direction strikes an optimal balance between accuracy and computational cost. Several neutronic codes based on this method have been developed, including MPACT [3], PROTEUS-MOC [6], PANDAS-MOC [7], NECP-X [8, 9], and SHARK [10]. Nevertheless, 2D/1D-MOC still faces challenges, such as the complexity of coupling strategies between 2D and 1D calculations and potential convergence issues when refining the axial mesh [11, 12].

The variational nodal method (VNM) offers another option for one-step whole-core neutronics calculations. This method utilizes a functional for the second-order even-parity transport equation, with odd-parity Lagrange multipliers employed to enforce nodal balance. Response matrices (RMs) are obtained using a classical Ritz procedure. VNM was first proposed in the 1980s and initially applied to homogeneous node problems [13]. Over the years, VNM-based codes such as VARIANT [14, 15] and VITAS [16–24] have emerged, benefiting from its accuracy and adaptability to mesh geometry. Since 1997, VNM has expanded its capability to handle heterogeneous materials within nodes, enabling high-fidelity neutronics calculations. In 2017, a significant milestone was reached with the development of the 3D Heterogeneous Variational Nodal Method (HVNM), specifically designed for pin-resolved problems. This method, implemented in PANX [25, 26] and VITAS [17, 21], treats each pin cell as a single node and utilizes isoparametric finite elements to accurately represent pin-resolved geometry. Angular expansion is achieved using spherical harmonics, while radial and axial leakage expansions employ polynomials and piecewise constants, respectively. HVNM performs full 3D calculations directly without coupling calculations between 2D and 1D domains, as seen in 2D/1D-MOC. Therefore, HVNM avoids lateral integration and eliminates issues associated with negative leakage terms. Recent research [27] has compared the accuracy and efficiency of HVNM and 2D/1D-MOC in pin-resolved problems. It was reported that for the KAIST problem, the NuScale problem, and the BEAVRS problem, HVNM produces more accurate pin power distribution and superior computational efficiency compared to 2D/1D-MOC [27]. This demonstrates the significant potential of HVNM as an alternative to 2D/1D-MOC for one-step neutronics calculations.

In our previous publication [17], we introduced and verified the high-fidelity modeling capability of HVNM using the C5G7 benchmark problem set. It is worth noting that previous verifications were limited to relatively small-scale pin-cell geometry cases. Therefore, further verification of HVNM is necessary to comprehensively investigate its feasibility for larger problems. Additionally, it is crucial to examine whether the method can be applied to problems with fine mesh sizes, such as plate-type assemblies with fuel plates at the millimeter scale. Unfortunately, the limited serial capability of HVNM has hindered its ability to achieve sufficient space-angle orders for desired accuracy when dealing with strongly heterogeneous problems or to calculate problems significantly larger than the C5G7 benchmark. Consequently, there is an urgent need for research on parallel algorithms for HVNM.

Prior to this work, significant efforts have been devoted to developing parallel algorithms for VNM. Several parallel strategies have been proposed, including a parallel approach based on the Message Passing Interface (MPI) standard implemented in VARIANT. However, the existing parallel implementation of VNM was only devoted to axial planes [28], limiting its applicability to 3D problems. Another parallel approach [29] based on non-overlapping domain decomposition has been investigated for solving the red-black algorithm; however, its restriction to regular-shaped finite elements hinders its effectiveness in addressing heterogeneous problems. Furthermore, a hybrid parallelization of HVNM for pin-resolved neutron transport calculations has been presented by Wang et al. [30]. However, that study lacks detailed analysis of parallel efficiency and is confined to Pressurized Water Reactors. These limitations highlight the research gap that still exists in developing a comprehensive and efficient parallel algorithm specifically tailored for HVNM, capable of addressing the challenges posed by intense heterogeneity and large-scale neutron transport problems. Therefore, this work aims to fill this research gap by proposing an efficient parallel algorithm for HVNM and conducting a thorough analysis of its parallel efficiency.

In this study, we propose a parallel formulation specifically tailored for HVNM within an MPI framework. Considering HVNM as a representative RM method, the HVNM procedure is divided into two steps: (a) constructing the RMs and (b) solving the resulting matrix equations. In step (a), each RM is constructed independently, which inherently allows for parallelism. Therefore, we employ a specialized parallel strategy, rather than domain decomposition, for RM formation to ensure optimal load balance. This approach evenly distributes the computational workload among MPI processors, optimizing parallel performance. The solution process is parallelized through non-overlapping domain decomposition. The entire space domain is divided into multiple subdomains, with each subdomain assigned to an MPI processor. The subdomains are coupled through interface nodes located along their boundaries.

The remainder of this paper is organized as follows. Section II introduces the theoretical models of HVNM and the parallel algorithm. In Section III, numerical results for representative heterogeneous neutron transport problems, KAIST and JRR-3, are obtained to verify the accuracy and performance of the parallel algorithm. The parallel performance is evaluated by comparing CPU time between serial and multi-core parallel computations. Finally, Section IV concludes the paper and discusses possible future improvements.

II. Theoretical Descriptions

A. Theoretical Models for Neutron Transport

This section provides the essential equations for HVNM, but for a comprehensive understanding of the derivation process and detailed matrix expressions, please refer to Ref. [25]. HVNM is based on the second-order neutron transport equation (NTE) with isotropic scattering approximation. The second-order NTE within the group takes the form of

$$-\Omega\cdot\nabla\Sigma_t^{-1}(r)\Omega\cdot\nabla\psi^+(r,\Omega)+\Sigma_t(r)\psi^+(r,\Omega)=\Sigma_s(r)\phi(r)+q(r),$$

where $\Sigma_t(r)$ and $\Sigma_s(r)$ are the macroscopic total and scattering cross-sections, respectively. $\psi^+(r,\Omega)$ is the even-parity angular flux at position r in direction Ω. $\phi(r)$ is the scalar flux satisfying $\phi(r)=\int\psi(r,\Omega)d\Omega$. $q(r)$ is the group source consisting of scattering and fission terms:

$$q(r)=\sum_{g'\neq g}\Sigma_{sgg'}(r)\phi_{g'}(r)+k_{eff}^{-1}\chi_g\sum_{g'}\nu\Sigma_{fg'}(r)\phi_{g'}(r),$$

In addition, the odd-parity angular flux $\psi^-(r,\Omega)$ is defined and satisfies

$$\psi^-(r,\Omega)=-\Sigma_t^{-1}(r)\Omega\cdot\nabla\psi^+(r,\Omega)$$

In HVNM, the second-order NTE is formulated as a variational principle in terms of a global functional $F[\psi^+,\psi^-]$

$$F[\psi^+,\psi^-]=\sum_v F_v[\psi^+,\psi^-],$$

which is a superposition of the functional for each node, $F_v[\psi^+,\psi^-]$:

$$F_v[\psi^+,\psi^-]=\int\int d\Omega[\Sigma_t^{-1}(\Omega\cdot\nabla\psi^+)^2+\Sigma_t\psi^{+2}]-\Sigma_s\phi^2-2\phi q]$$

$$-\int\int d\Omega n_p\cdot\Omega\psi^+\psi^-$$

$$+\int\int d\Omega(n_z^+\cdot\Omega\psi^+\psi^-|_{z^+}+n_z^-\cdot\Omega\psi^+\psi^-|_{z^-})$$

The remainder of this paper is organized as follows. Section II introduces the theoretical models of HVNM and the parallel algorithm. In Section III, numerical results for representative heterogeneous neutron transport problems, KAIST and JRR-3, are obtained to verify the accuracy and performance of the parallel algorithm. The parallel performance is evaluated by comparing the CPU time between serial and multi-core parallel computations. Finally, Section IV concludes the paper and discusses possible future improvements.

In summary, the spatial and angular independent variables r and Ω are suppressed. In local coordinates, $dV=dxdydz$ with $-\Delta x/2\le x\le\Delta x/2$,

$-\Delta y/2 \leq y \leq \Delta y/2$, $-\Delta z/2 \leq z \leq \Delta z/2$. n_p is the outward normal to the lateral interfaces extending over the periphery Γ, while n_z^+ and n_z^- are the outward normals to the top and bottom axial interfaces, respectively.

Within the node, the even-parity angular flux is expanded as

$$\psi^+(r,\Omega) = f^T(z) \otimes g^T(x,y)\psi(\Omega),$$

where $f(z)$ and $g(x,y)$ are vectors of orthonormal polynomials and continuous finite-element trial functions, respectively. $\otimes$ represents a tensor product. $\psi(\Omega)$ is a vector of expansion moments with respect to Ω. Correspondingly, the scalar flux is expanded as

$$\phi(r) = f^T(z) \otimes g^T(x,y)\phi,$$

where ϕ is a vector of scalar flux moments, satisfying $\phi = \int d\Omega\psi(\Omega)$. It is worth noting that the radial flux distribution within the node is represented by continuous, piecewise finite-element functions. This treatment allows for discontinuities in cross-sections at finite-element interfaces within each node, thereby eliminating the requirement for homogeneous nodes.

The odd-parity angular flux is expanded as

$$\psi_z^-(r,\Omega) = y_z^T(\Omega) \otimes h^T(x,y)\chi_z$$

$$\psi_\gamma^-(r,\Omega) = [f_\gamma^T(z) \otimes f_\gamma^T(\xi)] \otimes y_\gamma^T(\Omega)\chi_\gamma$$

on the axial and lateral interfaces, respectively. $h(x,y)$ denotes a piecewise constant vector, with each of its components equal to one over the domain of one or more finite elements and zero elsewhere. $y_z(\Omega)$ and $y_\gamma(\Omega)$ are vectors consisting of odd-parity spherical harmonics defined on the axial and lateral interfaces, respectively. χ_z and χ_γ are expansion moment vectors. The material interfaces within a single node can be explicitly described using these trial functions, ensuring that there is no smearing between materials at axial interfaces.

Inserting Eq. (6) through Eq. (9) into Eq. (5) results in the discretized functional in the form of

$$F_v[\psi(\Omega),\chi_\gamma,\chi_z] = \int d\Omega\psi^T(\Omega)A(\Omega)\psi(\Omega) - \phi^T I_z \otimes F_s\phi - 2\phi^T q$$

$$-\int d\Omega\psi^T(\Omega)E_\gamma(\Omega)\chi_\gamma$$

$$-\int d\Omega \psi^T(\Omega) E_z(\Omega) \chi_z$$

Requiring the discretized functional given in Eq. (10) to be stationary with respect to variation in $\psi(\Omega)$, χ_γ and χ_z, and employing the linear transformation of variables, finally results in the following equations:

$$j^+ = Bq + Rj^-$$

$$\phi = Vq - C(j^+ - j^-)$$

where j^+ and j^- stand for the vectors of the expansion moments of outgoing and incoming partial currents along the nodal surfaces, respectively. B, R, V, and C are the nodal RMs, which are coefficient matrices solely related to the nodal geometry and macroscopic cross-sections. Eq. (11) signifies the relationship between the neutron source within the node and the partial current on the node' s surface, while Eq. (12) represents neutron conservation within the node.

The numerical solution process in HVNM involves three levels of iteration [27]. The outermost iteration is the Fission-Source (FS) iteration, which utilizes the Power Method [31]. In each FS iteration, if up-scattering is present, the multigroup (MG) flux system is solved using the legacy Gauss-Seidel (GS) algorithm, referred to as the MG iteration. However, if there is no up-scattering, only a single sweep over the energy groups is required. Within each energy group, the within-group (WG) RM system, expressed by Eq. (11), is solved using the Red-Black Gauss-Seidel (RBGS) algorithm, referred to as WG iteration. The detailed solution process is presented in Algorithm 1.

Several techniques specifically tailored for HVNM, including the flat source region (FSR) acceleration method [25], partitioned matrix (PM) method [32], and quasi-reflected interface condition (QRIC) method [26], are employed to accelerate the solution process. These acceleration methods have been elaborated in our previous publications [21] and thus are not described in detail in this paper.

The FSR acceleration method aims to reduce the degrees of freedom within the node by partitioning the finite elements into FSRs. Within each FSR, the group source at each finite-element vertex is approximated as the average source within that FSR.

The PM method involves decomposing the response matrices into a low-order matrix corresponding to the surfaces of each node and a high-order spatial-angular matrix. The high-order terms are used to construct a correction source term for solving the low-order diffusion matrix equation during the iteration process.

The QRIC method aims to reduce the number of angular degrees of freedom on the interfaces by applying the reflective boundary condition (B.C.) to the high-order angular terms. This reduction leads to a smaller size of the response matrix, resulting in improved computational efficiency and reduced memory requirements.

B. Parallel Algorithm

The parallel algorithm tailored for HVNM is based on MPI. In the subsequent sections, we introduce the parallel algorithms for matrix formation and solution. Although HVNM incorporates acceleration methods such as PM, FSR, and QRIC, it is not necessary to consider the parallelization of these acceleration methods themselves. The parallel algorithm described in the following sections is fully compatible with these acceleration techniques.

1. Matrix Formation Parallel Algorithm According to the expressions of RMs (i.e., B, R, V and C designated as a matrix set), they are purely dependent on the node’ s geometry and macroscopic cross-sections. This implies that for a specific energy group, nodes with the same geometry, material, and finite element grid (categorized as a unique node) will have identical matrix sets. Therefore, the formation of matrix sets is an independent operation for each unique node and energy group; this independence allows for perfect scalability in a parallel computing environment using the MPI framework. Each MPI processor can construct matrix sets for a subset of unique nodes and energy groups simultaneously, without any communications.

The most straightforward and intuitive parallel scheme is to evenly assign the matrix formation tasks to all processors to achieve optimal load balance. Assuming there are N_G energy groups and N_U unique nodes, a total of $N_M = N_G \times N_U$ matrix sets need to be constructed. The formation of N_M matrix sets is partitioned by N_P processors so that each processor undertakes a part of the calculation simultaneously. If N_M is exactly divisible by N_P, the index of matrix sets to be calculated on processor p ($p \in [0, N_P - 1]$) can be defined as $i_p \in [p \cdot N_M/N_P, (p+1) \cdot N_M/N_P - 1]$. However, in cases where N_M cannot be evenly divided by N_P, the bounds of i_p need to be adjusted to allocate the remaining matrix sets to specific processors. Fig. 1 [Figure 1: see original paper] illustrates a partition example with $N_U = 2$ and $N_G = 4$. When $N_P = 2$, the matrix sets are evenly distributed among 2 processors with each processor being assigned 4 matrix sets. When $N_P = 3$, Processor 0 and Processor 1 are assigned 3 matrix sets while Processor 2 is assigned 2 matrix sets. The partition scheme enforces that the number of matrix sets assigned to each processor is as balanced as possible.

Each processor requires the corresponding set of RMs for its subdomain during the solution of the matrix equations presented in Eqs. (11) and (12). However, the distribution scheme of matrix sets may result in some processors not having the required response matrix sets locally. Instead, these matrix sets are allocated to other processors for construction. In such cases, communication between processors is necessary to ensure that each processor obtains the required response matrix sets for its subdomain. The communication scheme employed in this study involves transmitting the local matrix sets constructed by each processor to a designated processor, which performs the merging process to generate a global matrix set encompassing all unique nodes and energy groups. Finally, the global matrix set is dispatched to all other processors.

The parallelization for matrix formation is outlined in Algorithm 2, which highlights the steps involved in distributing the matrix formation tasks and performing the necessary communication to generate the global matrix set. While no communication is required between processors during the calculation of matrix sets, load imbalances may occur if the number of matrix sets cannot be evenly distributed among the processors. In addition, the collective manipulations required to generate the global matrix set and transfer it to each processor introduce communication overhead, which can impact parallel performance. The communication overhead is mainly influenced by both the number of processors involved in the communication and the size of the matrices that need to be communicated. The communication overhead becomes more significant as the number of processors and the size of the matrices increase.

2. Solution Parallel Algorithm In the parallelization of the solution process in HVNM, non-overlapping domain decomposition is employed. The entire space domain is divided into multiple subdomains, with each subdomain assigned to an MPI processor. Examples of 3D non-overlapping domain decomposition are shown in Fig. 2 [Figure 2: see original paper]. The subdomains are coupled through interface nodes located along their edges. The primary challenge in parallelization lies in identifying the processes that require parallel communication and determining the effective way to implement it.

The iteration process of HVNM, mentioned in Section II A, reveals that the update of the eigenvalue in the FS iteration and the solution of the RM equation in the WG iteration require data transfer between subdomains. The eigenvalue updates can be parallelized through collective manipulation. The designated processor gathers the individual contributions to the total fission source from all processors, computes the next estimate of the eigenvalue, and broadcasts this value to all other processors. This ensures that all processors have consistent and updated eigenvalue estimates.

Conversely, the parallelization for the solution of the RM equation is more complex. When solving the RM equation, the global nodes are colored red and black, ensuring that adjacent nodes have different colors. Fig. 3 [Figure 3: see original paper] shows the red-black coloring scheme in a 2D domain. Based on the principle of continuity, it can be deduced that the incoming partial current on a surface of a red node is equal to the outgoing partial current on the same surface of the adjoining black node and vice versa. This equality relationship is applied to update the incoming partial current, while the RM equation is used to update the outgoing partial current. Obviously, data transfer between subdomains is necessary when updating the partial current defined across the boundaries of subdomains. A simple illustration of the data transfer is presented in Fig. 4 [Figure 4: see original paper].

In each subdomain, the partial currents are first updated through a loop over nodes in the order of red nodes followed by black nodes. Once a sweep of all red nodes or all black nodes is completed, the two adjacent subdomains engage in simultaneous point-to-point communication to exchange partial currents on each boundary, as illustrated in Fig. 4. In Fig. 4, the yellow arrows indicate the direction of data transfer after solving all red nodes, while the green arrows indicate the direction of data transfer after solving all black nodes. When the partial currents of all red nodes have been updated, each subdomain will engage in the exchange of updated partial currents of red nodes with its neighboring subdomains on the boundaries. Subsequently, each subdomain will utilize the received partial currents to update the partial currents of the black nodes on the boundaries. Likewise, parallel communication follows a similar process after updating all black nodes. The parallel algorithm for the solution process preserves the benefits of RBGS, ensuring that the incoming partial current used for updating the outgoing partial current on the subdomain' s boundaries is the most up-to-date. Thus, the parallel algorithm ensures satisfactory convergence speed when solving the WG response matrix equation. During each WG iteration, the number of point-to-point communications is equal to twice the number of adjoining subdomain boundaries. Taking Fig. 4 as an example, one WG iteration needs 2×4 point-to-point communications. At the end of all node sweeps, the designated processor gathers the individual contributions to the iteration error from all processors, computes the final iteration error, and broadcasts this value to all other processors. If the partial currents satisfy the convergence criterion, the WG iteration will be terminated. The detailed parallelization for WG solution is shown in Algorithm 3.

In the parallel algorithm for the solution process, there are three factors that can affect parallel performance. First, local workload imbalances may occur if the subdomains have an unequal number of nodes (referred to as local nodes), which can lead to load imbalances among processors. Second, the ratio of communication effort to local work also increases as the number of subdomains increases. In summary, the communication overhead becomes more significant compared to the computational workload, which can negatively impact the efficiency of the parallel algorithm. Third, communication imbalances may arise when subdomains have different numbers of communicated boundaries. Subdomains located in the middle of the problem typically have more adjacent subdomains to communicate with compared to those on the surfaces. This imbalance may also affect parallel performance.

III. Results and Discussion

The foregoing parallel algorithm has been implemented through a revision of the VITAS code. In this section, the accuracy and performance of the algorithm are evaluated using the KAIST problem [27] and the JRR-3 problem [33, 34]. These problems represent challenging scenarios in terms of computational requirements and spatial heterogeneity, making them suitable for assessing the performance of the parallel algorithm. It is worth noting that applying HVNM to plate-type assemblies in the JRR-3 problem involves modeling the internal structure of the reactor with mm-level grids, which poses significant challenges and represents the first attempt at applying this method to such reactors. Furthermore, the use of the JRR-3 problem for verification and analysis of the parallel algorithm underscores the appropriateness of the proposed parallel algorithm in tackling various types of heterogeneous transport problems.

The evaluation of parallel performance is measured using speedup (S) and efficiency (ε). Speedup is defined as the ratio of the sequential run time (T_s), estimated using the run time with one processor (T_1), to the parallel run time when using P processors (T_p). Efficiency measures the utilization of computational resources in a parallel computation. These metrics are calculated using Eq. (13).

All computations were performed on the PI 2.0 cluster supported by the Center for High Performance Computing at Shanghai Jiao Tong University. The PI 2.0 cluster consists of 654 compute nodes. Each compute node is equipped with two Intel Xeon Scalable Cascade Lake 6248 CPUs @ 2.5GHz, with each CPU having 20 cores.

A. KAIST Problems

The KAIST problem is derived from the MOX benchmark problem 2A proposed by KAIST in South Korea. It represents a simplified model of a light-water reactor with 52 fuel assemblies surrounded by a water reflector. The problem is simplified to a 1/8 core by applying reflective B.C. on the south, west, and bottom sides of the core to reduce computational complexity. The lateral geometry of the eighth core is illustrated in Fig. 5 [Figure 5: see original paper], including three types of fuel assemblies: UOX-1, UOX-2, and MOX-1. Each assembly consists of 289 pin cells arranged in a 17$×$17 pinlayout.TheUOX − 1andUOX − 2assembliescompriseUO_{2}$ pin cells with enrichments of 2.0% and 3.3%, respectively. The MOX-1 assembly contains three different types of MOX pin cells with enrichments of 4.3%, 7.0%, and 8.7%. The geometry of each pin cell is illustrated in the upper right corner of Fig. 5, where the circular area can represent fuel, moderator, or control rod, while the area between the circle and square represents moderator. The core height is 150 cm with 15 cm reflectors on the top.

The calculations employ seven-group macroscopic cross-sections, which can be found in Ref. [27]. Each pin cell is treated as one node radially, and the whole problem is evenly divided into 10 layers axially. Thus, there are 85 × 85 × 10 nodes in the problem. Each node has dimensions of 1.26 cm × 1.26 cm × 15 cm. The fuel pin cells are meshed using five radial rings for the fuel zone, one radial ring for the moderator zone, and eight azimuthal sectors. Each fuel pin cell comprises 48 quadratic finite elements, as shown in Fig. 6 [Figure 6: see original paper]. The meshing scheme for control rod pin cells and guide tube pin cells follows the same pattern as the fuel pin cells, with the only difference being the replacement of fuel material with the corresponding control rod material or guide tube material.

The FSR acceleration method is employed to accelerate the calculations, treating each finite element as one FSR. We specify 48 quadratic x-y finite elements in each node, using 2nd-order polynomials in the axial direction. On the lateral and axial interfaces, 2nd-order polynomials and 48 piecewise constants are employed, respectively. Angular integrals are evaluated utilizing a 25 × 25 Square Legendre-Chebyshev (SLC) cubature. On the nodal interfaces, PN_n expansions are employed where Pn represents the approximations on the interface after applying the QRIC method to eliminate high-order angular moments from n+1 through N. Table 1 summarizes detailed calculation settings for the KAIST problem, including the expansion orders, convergence tolerance, and applied acceleration methods. Sensitivity analysis for the spatial and angular expansion order indicates that this set of discretization schemes is adequate to eliminate errors associated with spatial and angular approximations. For brevity, the details of the sensitivity analysis are omitted in this paper.

1. Accuracy Comparison We performed both serial and parallel computations using multiple MPI processors to verify the accuracy of the parallel algorithm. As a comparison, the numerical results are compared with those obtained from MG Monte Carlo (MC) calculation, which served as the reference solution. In the MC calculation using MCNP, a simulation of 5 million particles per batch was performed, with a total of 500 batches, of which 300 batches were skipped. The large number of particles was sufficient for accurate simulation, and the statistical deviation of the eigenvalue was 3 pcm.

Table 2 presents the comparison results of the eigenvalue and axially integrated pin fission rate, including eigenvalue error, maximum fission rate percent error (MAX), average fission rate percent error (AVG), and root mean square (RMS) of the fission rate percent error. Table 2 only presents the results obtained from parallel computations using 48 processors, for the sake of brevity, as identical results were obtained with different numbers of parallel processors. From Table 2, it is observed that the parallel calculation yields the same eigenvalues and fission rates as the serial code: the eigenvalue error is 31 pcm, while the RMS of the fission rate percent error is 1.22%; this demonstrates the correct implementation of the parallel algorithm. The normalized fission rate distribution is depicted in Fig. 7 Figure 7: see original paper. It can be observed that sharp power gradients emerge throughout the core, with the power peak positioned at the interface between the MOX-1 and UOX-1 assemblies. Fig. 7(b) shows the percent error distribution of the fission rate.

2. Parallel Performance Analysis This section focuses on analyzing the parallel performance of the algorithm using two metrics: speedup and parallel efficiency, to assess its effectiveness. Speedup measures the extent to which parallel computation is faster than its sequential counterpart. Parallel efficiency assesses the utilization of computational resources in a parallel computation. These metrics are calculated using Eq. (13).

Table 3 compares the computation effort and parallel performance using 1, 4, 8, 12, 18, 25, and 36 processors. The computation time, speedup, and efficiency for response matrix formation (referred to as formation time/speedup/efficiency) and solution (referred to as solution time/speedup/efficiency) are provided. All parallel computations are performed using a single compute node to mitigate the impact of inter-node communication on parallel efficiency. Based on Table 3, it is evident that as the number of participating processors in parallel computation increases, the computation time significantly decreases, leading to an increase in speedup. The overall speedup with 36 processors exceeds 24.0. This demonstrates the effectiveness of parallelization in reducing total computational time. Furthermore, increasing the number of processors generally results in a decrease in parallel efficiency. When the number of processors increases from 4 to 36, the formation efficiency, solution efficiency, and overall efficiency decrease from 88.92%, 93.31%, and 93.31% to 43.83%, 70.50%, and 68.51%, respectively. This decrease is primarily attributed to the growing proportion of communication overhead compared to local work.

Regarding matrix formation, the workload assigned to each processor decreases as the number of processors increases because of the reduced number of local matrix sets. However, the communication overhead required to construct the global matrix sets becomes more significant with an increased number of processors, resulting in a larger portion of time spent on communication surpassing the time saved by distributing the workload across multiple processors. Consequently, the efficiency for matrix formation drops from nearly 90% with 4 processors to only 43.83% with 36 processors. Additionally, workload imbalance caused by the uneven distribution of matrix sets can also impact parallel efficiency. This workload imbalance may become more pronounced as the number of processors increases, further decreasing parallel efficiency.

During the solution process, the local workload can be measured by the number of local nodes, while communication overhead is associated with the number of communication boundaries between subdomains. Table 4 presents a comparison of communication and local work in the solution process, including the maximum and minimum numbers of local nodes and communication boundaries. A significant decrease in local nodes is observed as the number of processors increases, while the number of communication boundaries increases. This leads to a decrease in the ratio of local work to communication efforts. For instance, as the number of local nodes decreases from 18,490/17,640 to 2,250/1,960, the number of communication boundaries increases from 2/2 to 4/2. Furthermore, as mentioned in Section II B 2, subdomains situated in the middle of the problem generally exhibit a larger number of communication boundaries compared to those on the surfaces. This communication imbalance further reduces efficiency. The extent of communication imbalance can be estimated by calculating the relative difference between the maximum number of communication boundaries and the minimum number of communication boundaries. As indicated in Table 4, the number of communication boundaries is 2/2 with 4 processors, but it is 4/2 with 36 processors. Therefore, the communication imbalance may become more severe as the number of processors increases.

Fig. 8 [Figure 8: see original paper] depicts a visual representation of the parallel performance with varying numbers of processors. Fig. 8 illustrates that the overall parallel performance is predominantly influenced by the parallel performance of the solution phase. This is due to the relatively insignificant contribution of the formation time compared to the solution time. For instance, considering the results obtained using a single processor, the ratio of solution time to formation time is 20.0. This implies that the response matrix formation is more susceptible to the escalating communication overhead resulting from increased processors, compared to the solution phase. As illustrated in Fig. 8(a), the solution speedup exhibits a nearly linear growth trend as the number of processors increases, while the formation speedup progresses at a relatively slower pace. This discrepancy becomes particularly noticeable when the number of processors rises from 12 to 18, where the formation speedup remains almost unchanged. Conversely, the efficiency of matrix formation experiences a more pronounced decline with an increasing number of processors compared to the solution efficiency, as depicted in Fig. 8(b).

It is worth noting, as observed from Fig. 8(b), that the solution efficiency actually increases when the number of processors transitions from 18 to 25. This improvement in efficiency is likely attributed to load balancing. As shown in Table 4, when utilizing 18 processors, a significant load imbalance exists, with a maximum/minimum number of local nodes of 4206/3920. This imbalance leads to some processors being underutilized while others are overloaded. However, when the number of processors is adjusted to 25, each processor is assigned a subdomain with an equal number of local nodes. The equal distribution of local nodes among processors promotes load balance in the solution process, ultimately contributing to enhanced parallel efficiency.

B. JRR-3 Problems

We model the JRR-3 problem to demonstrate the applicability of the parallel algorithm to a spatial domain with a more complex geometric structure. This problem is constructed based on the Japan Research Reactor No.3 (JRR-3) [33, 34] designed by the Japan Atomic Energy Research Institute (JAERI). JRR-3 is a water-cooled research reactor using plate-type fuels. The geometric representation of the JRR-3 reactor is illustrated in Fig. 9 [Figure 9: see original paper]. The reactor core is composed of 26 standard fuel assemblies, 6 follow fuel assemblies with neutron absorber, and 5 glory hole assemblies. Surrounding the core is a baffle with a thickness of 1 cm and an inner radius of 30.0 cm. Furthermore, there is a 30 cm axial reflector located at the top and bottom of the reactor. The lateral geometry of typical assemblies is illustrated in Fig. 10 [Figure 10: see original paper]. All assemblies have dimensions of 7.72 cm × 7.72 cm. The standard fuel assembly comprises 20 evenly arranged fuel plates, each with a thickness of 0.076 cm and a length of 6.16 cm. The follow fuel assembly consists of 16 fuel plates, also with a thickness of 0.076 cm, but a shorter length of 4.9 cm. The absorber assembly incorporates an absorber material with a thickness of 0.5 cm. Further detailed parameters of assemblies can be found in Ref. [34]. The calculations employ seven-group macroscopic cross-sections, which are provided in Ref. [34].

The reference solutions for all cases in this problem were obtained from the MC code RMC [35-37]. In the MC calculation using RMC, a simulation of 10 million particles per batch was performed, with a total of 800 batches, of which 300 batches were skipped. The statistical deviation of the eigenvalue was 1 pcm.

1. Accuracy Comparison (1) Assembly Cases

Adhering to a progressive approach, we initially perform calculations for 2D fuel assemblies. We divide the standard fuel assembly into nodes of 7 × 20, while dividing the follow fuel assembly into nodes of 6 × 18, to facilitate comparison of the fission rates of fuel plates, as illustrated in Fig. 11 [Figure 11: see original paper]. Given the intricate composition of assemblies in the JRR-3 problem, a more refined finite element grid is necessitated to accurately represent the geometry within each unique node, in contrast to the grids employed in the KAIST problem. The size of the finite element grids is even smaller than 0.05 cm, as illustrated in Fig. 12 [Figure 12: see original paper]. During the calculation, P11_3 spherical harmonics and 2nd-order polynomials are employed on the lateral interfaces, while retaining the remaining calculation parameters identical to those employed in the KAIST problem. Concerning the parallel calculation, the fuel assembly is decomposed into 2 × 8 subdomains, with each subdomain assigned to an individual processor.

Table 5 presents the comparison results of the eigenvalue and plate fission rates for the standard fuel assembly and follow fuel assembly. It can be observed that using 16 processors for computation yields results that closely align with the reference results. The eigenvalue error is below 50 pcm, and the RMS of plate fission rate percent error is less than 0.1%. This not only demonstrates the feasibility of HVNM in dealing with plate-type fuel assemblies but also confirms the correctness of the parallel algorithm.

Fig. 13 [Figure 13: see original paper] illustrates the fission rate distribution of the fuel plates for the standard fuel assembly and the follow fuel assembly. In the standard fuel plates, the fission rates are homogenized into 5 sections, while in the follow fuel plates, they are homogenized into 4 sections. It can be found that the fuel plate segments located near the periphery of the assembly exhibit higher fission rates compared to those located at the center of the assembly.

(2) Whole-Core Cases

In the whole-core calculation, the entire reactor is divided into 9 × 9 assemblies, with each assembly further subdivided into 7 × 20 nodes. Both 2D and 3D whole-core cases are examined. In the 3D calculation, the axial direction is divided into 45 layers, each with a height of 3 cm. The spatial and angular expansion schemes for the whole-core calculation are listed in Table 6 . All other calculation parameters remain the same as those used in the KAIST problem.

Table 7 presents the comparison results for the eigenvalue and axially integrated assembly fission rates. In the 2D and 3D whole-core cases, the eigenvalue errors are -56 pcm/-90 pcm, and the RMS of fission rate percent error are 0.54%/0.65%, respectively. These results indicate the high accuracy achieved through the parallelization of HVNM.

Fig. 14 [Figure 14: see original paper] illustrates the normalized fission rate distribution of the assemblies, excluding the assemblies in the reflector region. It can be observed that the fuel assemblies positioned at the central region of the core display higher fission rates compared to the assemblies located at the periphery of the core. Fig. 15 [Figure 15: see original paper] illustrates the error distribution of the assembly fission rates, with an error range of -0.94% to 1.72%. The maximum error is observed in the assembly near the reflector region.

2. Parallel Performance Analysis The 3D whole-core case is more capable of demonstrating the superiority of the parallel algorithm due to the significantly larger computational workload compared to the assembly cases and 2D whole-core cases; this is evident from the results presented in Table ??. As shown in Table ??, the total number of spatial-angular degrees of freedom for the 3D whole-core case exceeds 10 million, posing a significant challenge for computational resources and indicating the necessity of employing a parallel algorithm for 3D whole-core calculations. Consequently, this section focuses on analyzing the performance of the parallel algorithm specifically for the 3D whole-core case. Furthermore, considering the computational memory and time constraints associated with performing 3D whole-core calculations using a single processor, the results with 36 processors are taken as the baseline for evaluating parallel performance. Accordingly, the definition of speedup and efficiency is adjusted to $S_p = T_{36}/T_p$, where T_{36} represents run time with 36 processors.

The comparative results regarding computational effort and parallel performance, employing 36, 60, and 144 processors, are presented in Table 8 . In all cases, the calculations are performed using an equal number of compute nodes, specifically 12, to minimize the impact of inter-node communication on parallel performance. It is observed that the parallel algorithm demonstrates efficient acceleration and parallel efficiency. Additionally, the parallel performance exhibits a similar trend to that observed in the KAIST problem as the number of processors increases. The total computation time decreases from 1.24 h using 36 processors to 0.40 h using 144 processors, yielding a speedup of 3.09. The overall efficiency stands at 95.42% and 77.14% using 60 and 144 processors, respectively. The underlying reasons for this trend have been extensively discussed in Section III A 2 and will not be reiterated here. However, in contrast to the KAIST problem, the JRR-3 problem has a lower proportion of solution time. For instance, with 36 processors, the formation time accounts for 93% of the total time in the KAIST problem, while in the JRR-3 problem, it constitutes only 65%. Consequently, the solution efficiency of the JRR-3 problem exerts a comparatively less dominant influence on the overall efficiency when compared to the KAIST problem.

IV. Conclusions

In this paper, we introduce an efficient parallel algorithm for HVNM within an MPI framework. The accuracy and efficiency of the parallel algorithm for HVNM are verified using the representative heterogeneous neutron transport problems, KAIST and JRR-3. In the KAIST problem, which encompasses a large spatial domain, the numerical results using multiple processors align perfectly with those obtained from the serial calculation, thus confirming the accuracy of the parallel algorithm. Meanwhile, a significant reduction in total computation time is achieved utilizing the parallel algorithm, decreasing from 16.64 h using a single processor to only 0.67 h using 36 processors, resulting in a speedup of 24.66. The efficiency achieved with 36 processors amounts to 68.51%. In the 3D whole-core case of the JRR-3 problem, the parallelization of HVNM results in an eigenvalue error of -90 pcm and an RMS error of the fission rate of 0.66% compared to the results obtained from the MC MG calculation; this signifies the effectiveness of HVNM in addressing neutron transport problems involving mm-level finite element grids. Additionally, the parallel calculation using 144 processors achieves an overall speedup of 3.09 and an overall efficiency of 77.14% compared with the results obtained with 36 processors, thus verifying the efficient acceleration and efficiency of the parallel algorithm.

Currently, the parallel algorithm has not achieved the desired scaling. Future endeavors will concentrate on improving the parallel efficiency of the algorithm. For matrix formation, one potential approach is to have each MPI processor store only the matrix sets corresponding to its subdomain, rather than storing the global matrix sets. This approach not only reduces the size of communication data and the amount of communication, resulting in decreased communication time, but also minimizes unnecessary memory consumption. Additionally, separating the matrix formation and solution segments of the process to allow for different numbers of processors in each segment could be considered in future investigations. Furthermore, a performance analysis of the parallel algorithm will be performed for transport problems that incorporate burnup, where each node in the problem domain represents a unique node. This analysis will provide insights into the algorithm’ s efficiency in handling such problems.

References

[1] M. Dai, M. Cheng, Application of material-mesh algebraic collapsing acceleration technique in method of characteristics-based neutron transport code. Nucl. Sci. Tech. 32, 87 (2021). doi: 10.1007/s41365-021-00923-w

[2] L. Liu, C. Hao, Y. Xu, Equivalent low-order angular flux non-linear finite difference equation of MOC transport calculation. Nucl. Sci. Tech. 31, 125 (2020). doi: 10.1007/s41365-020-00815-8

[3] B. Collins, S. Stimpson, B. W. Kelley et al., Stability and accuracy of 3D neutron transport simulations using the 2D/1D method in MPACT. J. Comput. Phys. 326, 612–628 (2016). doi: 10.1016/j.jcp.2016.08.022

[4] B. W. Kelley, E. W. Larsen, A consistent 2D/1D approximation to the 3D neutron transport equation. Nucl. Eng. Des. 295, 598–614 (2015). doi: 10.1016/j.nucengdes.2015.07.026

[5] G. S. Lee, N. Z. Cho, 2D/1D fusion method solutions of the three-dimensional transport OECD benchmark problem C5G7 MOX. Prog. Nucl. Energ. 48, 410-423 (2006). doi: 10.1016/j.pnucene.2006.01.010

[6] A. Hsieh, G. Zhang, W.S. Yang, Consistent Transport Transient Solvers of the High-Fidelity Transport Code PROTEUS-MOC. Nucl. Sci. Eng. 194, 508-540 (2020). doi: 10.1080/00295639.2020.1746619

[7] S. Tao, Y. Xu, Neutron transport analysis of C5G7-TD benchmark with PANDAS-MOC. Ann. Nucl. Energy 169, 108966 (2022). doi: 10.1016/j.anucene.2022.108966

[8] J. Chen, Z. Liu, C. Zhao et al., A new high-fidelity neutronics code NECP-X. Ann. Nucl. Energy 116, 417–428 (2018). doi: 10.1016/j.anucene.2018.02.049

[9] Z. Shen, Q. Sun, D. He et al., Comparison and verification of NECP-X and OpenMC using high-fidelity BEAVRS benchmark models. Nucl. Tech. 45, 71–79 (2022). doi: 10.11889/j.0253-3219.2022.hjs.45.010602 (in Chinese)

[10] C. Zhao, X. Peng, H. Zhang et al., A new numerical nuclear reactor neutronics code SHARK. Front. Energy Res. 9 (2021). doi: 10.3389/fenrg.2021.784247

[11] M. Dai, A. Zhang, M. Cheng, 3D MOC/DD neutron transport method with thin axial meshes. Ann. Nucl. Energy 185,109731 (2023). doi: 10.1016/j.anucene.2023.109731

[12] C. Zhao, X. Peng, H. Zhang et al., Analysis and comparison of the 2D/1D and quasi-3D methods with the direct transport code SHARK. Nucl. Eng. Technol. 54, 19-29 (2022). doi: 10.1016/j.net.2021.07.038

[13] C. B. Carrico, E. E. Lewis, G. Palmiotti, Three-dimensional variational nodal transport methods for cartesian, triangular, and hexagonal criticality calculations. Nucl. Sci. Eng. 111, 168-179 (1992). doi: 10.13182/NSE92-1

[14] E. E. Lewis, C. B. Carrico, G. Palmiotti, Variational nodal formulation for the spherical harmonics equations. Nucl. Sci. Eng. 122, 194-203 (1996). doi: 10.13182/NSE96-1

[15] G. Palmiotti, E. E. Lewis, C. B. Carrico, VARIANT: VARIational Anisotropic Nodal Transport for multidimensional cartesian and hexadgonal geometry calculation. Argonne, IL: Argonne National Laboratory, 1995.

[16] Q. Sun, W. Xiao, X. Li et al., A variational nodal formulation for multidimensional unstructured neutron diffusion problems. Nucl. Eng. Technol. 55, 2172-2194 (2023). doi: 10.1016/j.net.2023.02.021

[17] W. Xiao, H. Yin, X. Liu et al., On the transient models of the VITAS code: applications to the C5G7-TD pin-resolved benchmark problem. Nucl. Sci. Tech. 34, 20 (2023). DOI: 10.1007/s41365-023-01170-x

[18] M. Yang, Q. Sun, C. Luo et al., Development and verification of a neutronics-thermal hydraulics coupling code with unstructured meshes neutron transport model. Nucl. Tech. 46, 030601 (2023). doi: 10.11889/j.0253-3219.2023.hjs.46.030601 (in Chinese)

[19] W. Xia, T. Zhang, X. Liu et al., Preliminary study on the fast solution strategy of hexagonal nodal neutron transport calculation program. Nucl. Tech. 43, 89-94 (2020). doi: 10.11889/j.0253-3219.2020.hjs.43.020603 (in Chinese)

[20] H. Yin, T. Zhang, D. He et al., A quasi-transport integral variational nodal method for homogeneous nodes based on the 2D/1D method. Comput. Phys. Commun. 274, 108290 (2022). doi: 10.1016/j.cpc.2022.108290

[21] T. Zhang, W. Xiao, H. Yin et al., VITAS: a multi-purpose simulation code for the solution of neutron transport problems based on variational nodal methods. Ann. Nucl. Energy 178, 109335 (2022). doi: 10.1016/j.anucene.2022.109335

[22] T. Zhang, J. Xiong, L. Liu et al., Development and implementation of an integral variational nodal method to the hexagonal geometry nuclear reactors. Ann. Nucl. Energy 131, 210–220 (2019). doi: 10.1016/j.anucene.2019.03.031

[23] T. Zhang, H. Wu, L. Cao et al., An improved variational nodal method for the solution of the three-dimensional steady-state multi-group neutron transport equation. Nucl. Eng. Des. 337, 419–427 (2018). doi: 10.1016/j.nucengdes.2018.07.009

[24] H. Yin, B. Zhang, X. Liu et al., Study on hexagonal integral variational nodal method and acceleration method. Nucl. Tech. 43, 48-54 (2020). doi: 10.11889/j.0253-3219.2020.hjs.43.060008 (in Chinese)

[25] T. Zhang, Y. Wang, E. E. Lewis et al., A Three-Dimensional Variational Nodal Method for Pin-Resolved Neutron Transport Analysis of Pressurized Water Reactors. Nucl. Sci. Eng. 188, 160–174 (2017). doi: 10.1080/00295639.2017.1350002

[26] T. Zhang, E. E. Lewis, M. A. Smith et al., A Variational Nodal Approach to 2D/1D Pin Resolved Neutron Transport for Pressurized Water Reactors. Nuc. Sci. Eng. 186, 120–133 (2017). doi: 10.1080/00295639.2016.1273023

[27] Y. Wang, C. Zhao, H. Wu et al., Comparison of the performance of heterogeneous variational nodal method and 2D/1D-MOC on pin-resolved neutron transport calculations of PWR. Ann. Nucl. Energy. 138, 107227 (2020). doi: 10.1016/j.anucene.2019.107227

[28] U. R. Hanebutte, H. S. Khalil, G. Palmiotti et al., Development of a parallelization strategy for the VARIANT code. Argonne National Lab., IL (United States). Reactor Analysis Div., 1996. doi: 10.2172/465204

[29] Y. Wang, C. Rabiti, G. Palmiotti, Parallelization of the Red-Black Algorithm on Solving the Second-Order PN Transport Equation with the Hybrid Finite Element Method. Paper presented at ANS Summer Meeting, Hollywood, Florida, 26-30 June 2011.

[30] Y. Wang, T. Zhang, E. E. Lewis et al., Three-dimensional variational nodal method parallelization for pin resolved neutron transport calculations. Prog. Nucl. Energ. 117, 102991 (2019). doi: 10.1016/j.pnucene.2019.03.007

[31] E. E. Lewis, W. F. Miller, Computational methods of neutron transport. (1984).

[32] T. Zhang, J. Xiong, A hybrid acceleration method to pin-by-pin calculations using the heterogeneous variational nodal method. Ann. Nucl. Energy 132, 723–733 (2019). doi: 10.1016/j.anucene.2019.06.052

[33] H. Tsuruta, H. Ichikawa, J. Iwasaki, Neutronics design of upgraded JRR-3 research reactor. JAERI-M-84-099. Japan (2008)

[34] C. Zhao, X. Peng, W. Zhao et al., Verification of the direct transport code SHARK with the JRR-3M macro benchmark. Ann. Nucl. Energy 177, 109294 (2022). doi: 10.1016/j.anucene.2022.109294

[35] Z. Li, K. Wang, Y. Guo et al., Forced propagation method for Monte Carlo fission source convergence acceleration in the RMC. Nucl. Sci. Tech. 32, 27 (2021). doi: 10.1007/s41365-021-00923-w

[36] Q. Pan, T. Zhang, X. Liu et al., SP3-coupled global variance reduction method based on RMC code. Nucl. Sci. Tech. 32, 122 (2021). doi: 10.1007/s41365-021-00973-0

[37] T. Huang, Z. Li, K. Wang et al. Hybrid windowed networks for on-the-fly Doppler broadening in RMC code. Nucl. Sci. Tech. 32, 62 (2021). doi: 10.1007/s41365-021-00901-2