Nuclear Engineering and Technology 58 (2026) 104074

Contents lists available at ScienceDirect

Nuclear Engineering and Technology

journal homepage: www.elsevier.com/locate/net

Original Article

Enhancing CMFD acceleration in MOC-based direct transport code

Fathurrahman Setiawan, Siarhei Dzianisau, Deokjung Lee *

Department of Nuclear Engineering, Ulsan National Institute of Science and Technology, 50, UNIST-gil, Ulsan, 44919, Republic of Korea

ARTICLE INFO

Keywords:
CMFD acceleration
STREAM
Neutron transport
odCMFD

ABSTRACT

The implementation of CMFD acceleration and recent enhancements in STREAM deterministic transport code are presented in this study. A comprehensive description of the two-level CMFD approach in STREAM is provided, including its parallelization strategies and iteration algorithms. The main performance improvement comes from adopting an improved odCMFD method, complemented with code optimization at the programming level. Numerical evaluations against various realistic core problems demonstrate that the adopted odCMFD significantly improve the efficiency and robustness of CMFD acceleration in STREAM, enabling stable and accurate simulations with coarser meshes and reduced computational time.

1. Introduction

STREAM is a deterministic transport code that employs a method of characteristics (MOC) to solve the multi-group neutron balance equation. STREAM is capable to work as a lattice code to generate macro-group cross section (MGXS) for nodal code in two-step framework [1] or direct solver utilizing the 3D MOC/diamond-difference (DD) scheme [2–4]. While the MOC method offers high-fidelity solutions by means of fine-mesh discretization, it costs thousands of source iterations to converge when simulating realistic core problems. This challenge can be resolved using acceleration methods such as the coarse-mesh finite difference method (CMFD).

First introduced in 1983 [5], CMFD has been widely adopted for transport acceleration. Its core idea is the use of correction factors to preserve the net current at the interfaces, allowing the lower-order solution to accurately duplicate the node-wise reaction rates of the higher-order solution. In this context, the MOC serves as the high-order solver with fine spatial resolution, while the CMFD method operates as the low-order solver with a coarser mesh. This correction-based coupling strategy grants CMFD flexibility to accelerate various high-order transport solvers, including Monte Carlo [6].

A major drawback of traditional CMFD is its inherent instability in certain problems, particularly when dealing with optically thick meshes. Several techniques have been proposed to address this issue. One straightforward approach is to perform multiple transport sweeps prior to applying CMFD acceleration [7,8]. While this method can improve stability, the high computational cost of transport solver may negate the benefits of CMFD acceleration. Another commonly used technique involves the application of relaxation factor to dampen the finite-difference diffusion coefficient [9] or the current correction factor [10]. The first one uses a factor over 1 or known as over-relaxation while the latter use a relaxing factor less than one (under-relaxation).

Modifying the flux prolongation scheme has also proven effective in improving stability. Traditional CMFD employs flat prolongation, where the prolongation is based on the ratio of updated CMFD flux to old CMFD flux. A more advanced approach which known as lpCMFD interpolates the flux differences linearly at the coarse-mesh cell edges between the high-order and low-order solution [11,12]. This approach has been found to be unconditionally stable, as supported by both theoretical and numerical analyses [13,14].

Another widely used stabilization method includes the use of the partial current-based CMFD (pCMFD) method [15]. Unlike the traditional CMFD, which applies a single correction factor to the net current, pCMFD uses two correction factors for the partial currents. This approach ensures unconditional stability regardless of optical thickness. Effort to generalize the traditional CMFD and pCMFD produce the optimally diffusive CMFD (odCMFD) method [16]. This method introduces an adjustable parameter into the diffusion coefficient, which is determined from a specific set of polynomials derived through Fourier analysis. The odCMFD approach has been shown to yield unconditionally stable solutions with minimal implementation complexity. Studies [17,18] have found that the optimal value of the adjustable factor varies depending on the transport method used. Zhou [18] proposed an improved version of the odCMFD method designed specifically for

* Corresponding author. Department of Nuclear Engineering, Ulsan National Institute of Science and Technology, Bldg. 112, room 501-650 UNIST-gil, Ulsan, 44919, Republic of Korea.
E-mail addresses: setiawanfrs@unist.ac.kr (F. Setiawan), siarheid@unist.ac.kr (S. Dzianisau), deokjung@unist.ac.kr (D. Lee).

https://doi.org/10.1016/j.net.2025.104074
Received 1 June 2025; Received in revised form 10 November 2025; Accepted 8 December 2025
Available online 9 December 2025

quasi-3D transport methods, aimed at solving 3D problems more efficiently. Since the 3D MOC/DD solver in STREAM can be considered a quasi-3D transport method, it is worth exploring the adoption of this improved odCMFD formulation to enhance the performance of STREAM.

The default stabilization techniques that have been implemented in STREAM are the odCMFD method which uses the original adjustable factor proposed by Zhu [16], along with a dynamic under-relaxation scheme applied to the current correction factor. The value of the damping factor in this scheme is determined empirically based on the residual of the fission rate [2]. Moreover, more transport iterations are performed at earlier stages to ensure the stability of CMFD.

This paper presents the detailed implementation of CMFD acceleration in STREAM direct transport code. The effectiveness of the improved odCMFD method is evaluated within the STREAM framework. The paper also discusses code-level optimization which contributes to further improvements in the efficiency of the CMFD acceleration process.

2. CMFD in STREAM

2.1. Two-level CMFD formulation

The formulation of traditional CMFD in STREAM follows the widely used formulation in Ref. [19]. It starts with the multigroup neutron diffusion formalism:

$$\nabla\cdot \mathrm{D_g}(\vec{\mathrm{r}})\nabla\phi_{\mathrm{g}}(\vec{\mathrm{r}}) + \Sigma_{\mathrm{t,g}}(\vec{\mathrm{r}})\phi_{\mathrm{g}}(\vec{\mathrm{r}}) = \sum_{g'}\Sigma_{\mathrm{s},g'g}(\vec{\mathrm{r}})\phi_{g'}(\vec{\mathrm{r}}) + \frac{\chi_g}{\mathrm{k_{eff}}}\sum_{g'}\nu\Sigma_{\mathrm{f},g'}(\vec{\mathrm{r}})\phi_{g'}(\vec{\mathrm{r}}) \tag{1}$$

where Σ_t, Σ_s and Σ_f are the coarse mesh cross-section for total, scattering and fission respectively.

Integrating over a coarse node indexed by m and applying Divergence Theorem to the streaming terms yields the node balance equation:

$$\sum_{n\in N(m)} J_{g,mn}A_{mn} + \Sigma_{r,g,m}\phi_{g,m}V_m = \sum_{g'\neq g}\Sigma_{s,g'g,m}\phi_{g',m}V_m + \frac{\chi_g}{k_{eff}}\sum_{g'}\nu\Sigma_{f,g',m}\phi_{g',m}V_m \tag{2}$$

where, $N(m)$ is the set of neighboring nodes to m, $J_{g,mn}$ is the net current from node m to node n for group g and A_{mn} is the area of node m-n interfaces. The group g refers to CMFD energy group. Σ_r refers to the removal cross-section which defined as the total cross section minus the within-group scattering cross section.

The net current holds the high-order solution from transport calculation through the definition of current correction factor, $\widehat{D}$, as follows:

$$J_{g,mn} = -\widetilde{D}_{g,mn}\left(\phi_{g,n} - \phi_{g,m}\right) + \widehat{D}\left(\phi_{g,n} + \phi_{g,m}\right) \tag{3}$$

where, $\widetilde{D}$ is the node interface diffusion coefficient. From here, the definition of current correction factor can be written as:

$$\widehat{D} = \frac{J_{g,mn} + \widetilde{D}_{g,mn}\left(\phi_{g,n} - \phi_{g,m}\right)}{\phi_{g,n} + \phi_{g,m}} \tag{4}$$

This correction factor is the key to the formulation of CMFD as it ensures the CMFD solution to match the MOC transport solution. Note that, in the equation, the current and scalar flux are derived from the MOC transport solver, obtained through condensation from fine to coarse spatial and energy grids, prior to the construction of CMFD linear system.

The CMFD net currents are calculated as:

$$J_{g,mn} = \sum_{s\in mn} j_{out,g,s} \tag{5}$$

where, $j_{out,g,s}$ is the partial current on surface s for group g that is collected during the transport iteration. The surface s refers to six surfaces of the coarse mesh nodes.

The condensation of flux and group constants from fine mesh to CMFD mesh is given by:

$$\phi_{g,m} = \frac{\sum_{G\in gu\in m}\sum \phi_{G,F}V_u}{\sum_{u\in m} V_u} \tag{6}$$

$$\Sigma_{x,g,m} = \frac{\sum_{G\in gu\in m}\sum \Sigma_{x,G,u}\phi_{G,u}V_u}{\sum_{G\in gu\in m}\sum \phi_{G,u}V_u} \tag{7}$$

where, G is the MOC energy group, u is the index for MOC flat source regions or fine mesh cells and V_u is the volume of a fine mesh cell.

In STREAM, the MOC calculation uses 72 neutron energy groups, while CMFD acceleration is performed with eight energy groups. STREAM employs a two-level CMFD approach, where the pin-wise 8-group CMFD (L1CMFD) serves as the primary driver, further accelerated by the assembly-wise 8-group CMFD (L2CMD). The L1CMFD coarse-mesh size is equal to pin pitch (about 1–2 cm) while the axial height is similar with the thickness of MOC plane. For the L2CMFD, which is the accelerator of L1CMFD, the coarse-mesh size is equal to assembly pitch size in radial and ten times bigger than the L1CMFD in axial. Fig. 1 shows the flowchart of the two-level CMFD in STREAM.

The assembly-wise flux and group constants are generated from the pin-wise 8-group spectra via spatial condensation:

$$\phi_{g,M} = \frac{\sum_{m\in M}\phi_{G,m}V_m}{\sum_{m\in M} V_m} \tag{8}$$

$$\Sigma_{x,g,M} = \frac{\sum_{m\in M}\Sigma_{x,g,m}\phi_{g,m}V_m}{\sum_{m\in M}\phi_{g,m}V_m} \tag{9}$$

where M denotes the assembly-sized coarse mesh. Unlike the previous condensation, which is performed over both space and energy, this procedure condenses only over space.

For both pin- and assembly-level CMFD, the power iteration method is employed to solve the eigenvalue problem with the Gauss-Seidel iterative method serves as the inner solver. At the end of L1CMFD, the flux solution from CMFD is used to prolong the MOC scalar (ϕ) and angular (ψ) fluxes using flat prolongation approach:

$$\phi^{MOC} := \phi^{MOC}\frac{\phi_{m,g}^{CMFD_{new}}}{\phi_{m,g}^{CMFD_{old}}} \tag{10}$$

$$\psi^{MOC} := \psi^{MOC}\frac{\phi_{m,g}^{CMFD_{new}}}{\phi_{m,g}^{CMFD_{old}}} \tag{11}$$

A similar formulation is used to update the L1CMFD using the solution from L2CMFD.

2.2. Parallel CMFD via assembly decomposition

The CMFD acceleration module in STREAM employs hybrid MPI/OpenMP parallelization like other modules such as MOC transport solver that have been discussed in Ref. [2]. STREAM decomposes reactor core problems into several domains according to the number of MPI processes. The left-hand side of Fig. 2 illustrates an example where 13 assemblies are divided into four domains. The primary goal of domain decomposition is distributing memory allocation across multiple CPU nodes, which reduces the memory requirement per node. Each domain may have one or more assemblies distributed across the problem, aiming for an even distribution based on the computational load measured by the number of flat source regions. As a result, assemblies within a single domain may not always be physically adjacent. This approach confines

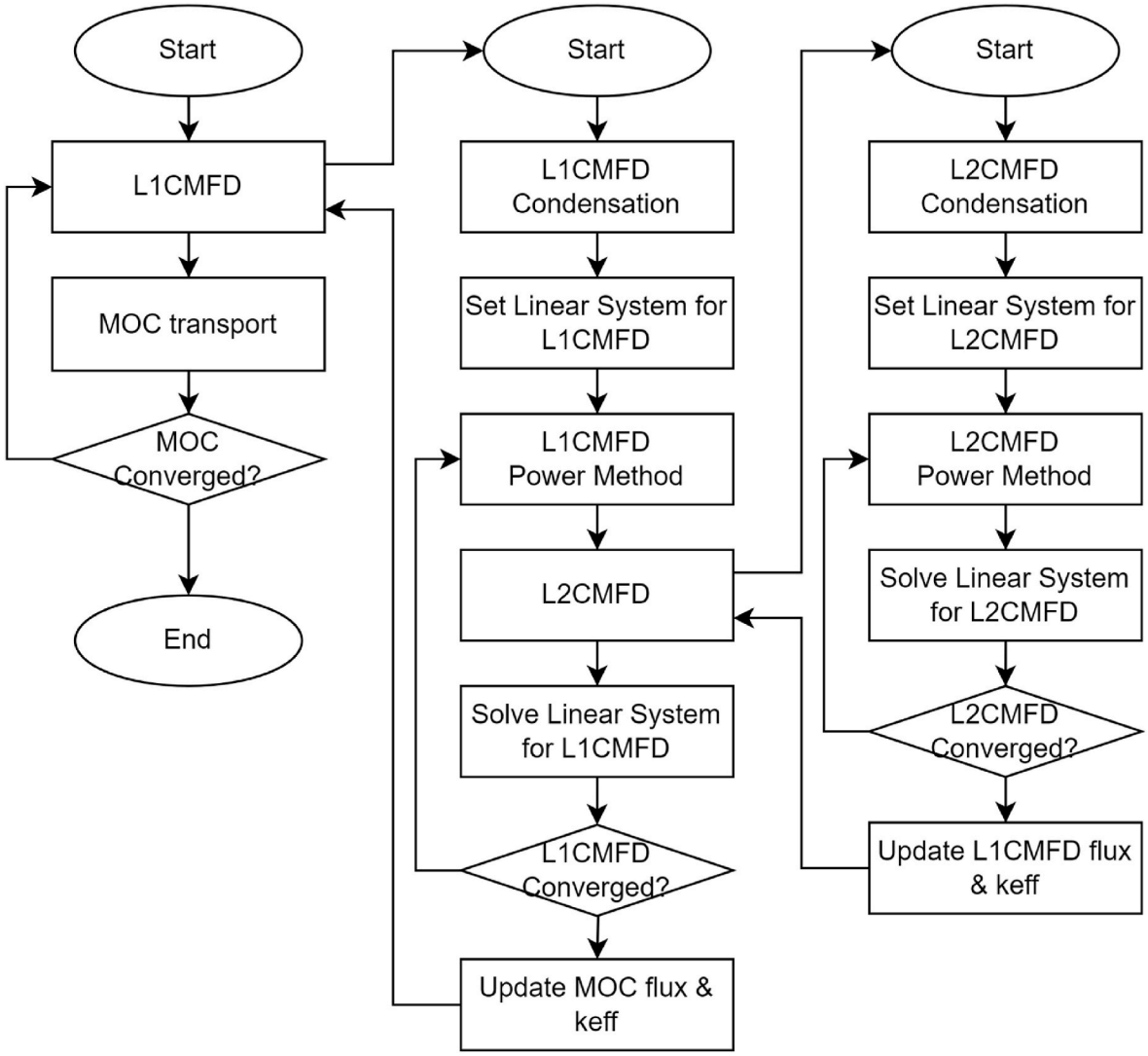

Fig. 1. Flowchart of two-level CMFD acceleration in STREAM.

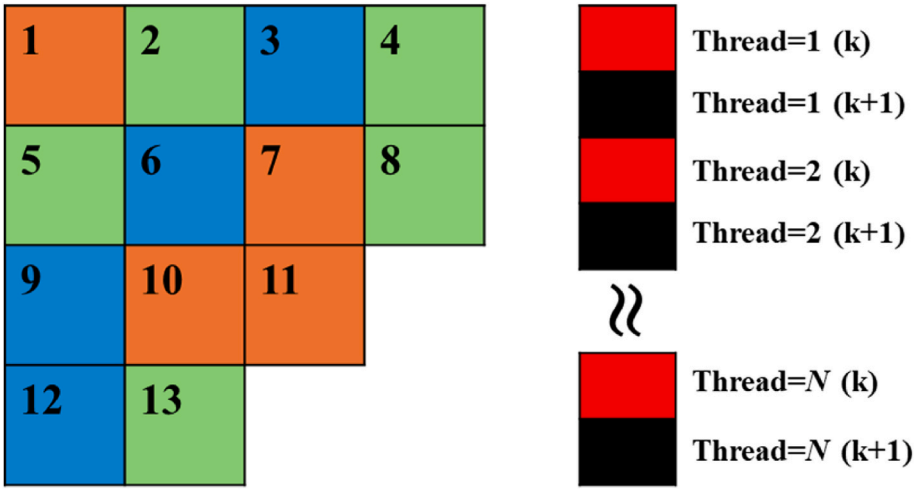

Fig. 2. Illustration of assembly-wise domain decomposition (left) and planar Red-Black parallelization (right).

the size of the problem or the size of the CMFD linear system grids to the dimensions of a single assembly. To ensure the continuity of the flux on each assembly interface, the MPI_Send and MPI_Recv procedures are utilized.

The work within each assembly is distributed through OpenMP threads with shared memory. In the CMFD module, the planar grid is the

target for this parallelization. As a result, the linear system solver that uses the Gauss-Seidel iterative method should incorporated the planar red-black ordering scheme to ensure the correctness of the solution. All red planes are updated first using data from the black planes at counter k, followed by the black planes using the updated data from the red planes at counter $k+1$ as shown in the right-hand side of Fig. 2. This approach allows half of total planes to be processed concurrently if the number of threads matches the required count. In practice, the number of threads is typically less than the number of simulated planes, so dynamic scheduling is used to ensure efficient workload distribution.

2.3. Iteration strategy

Algorithm 1 presents the inner iteration loop of pin-level CMFD in STREAM. Inter-assembly communication is performed at the beginning of the inner iteration loop then followed by the assembly loop that iterates over the list of assemblies within an MPI process. Inside this loop the linear system solver routine is invoked five times before proceeding to the calculation of the flux residual. This strategy was designed to reduce communication costs, with the value of five is determined empirically.

Algorithm 1: Inner iteration of CMFD

DO inner=1, 30
  Inter-assembly communication
  DO Assembly
    DO solver=1, 5
      DO red-black plane
        Compute total source, Q_i
        $\phi_i^{(k+1)} = \frac{1}{A_{ii}}\left(Q_i - \sum_{j=1}^{i-1} A_{ij}\phi_j^{(k+1)} - \sum_{j=i+1}^{n} A_{ij}\phi_j^{(k)}\right),\ i = 1,2,\ldots,n$
      END DO
    END DO
  END DO
  Check flux convergence
END DO

Fig. 3 shows the relative costs of inner iteration and its subcomponent for various number of linear solver invocations on C5G7 benchmark problem (unrodded case) [20]. As shown in the figure, increasing the number of solver calls per inner iteration significantly reduces the relative cost of communication and flux convergence checking, resulting in improved efficiency of total inner iteration runtime. The choice of five solver calls strikes a practical balance, as further increases yield diminishing returns in cost reduction.

The default global transport convergence threshold in STREAM is set to 1E-5 for both eigenvalue and fission source error. The convergence criteria for the CMFD outer iteration (power iteration) depends on the most recent transport fission source error. At the initial CMFD calculation – when MOC is not invoked yet – the threshold is set to 1E-2. For subsequent CMFD invocations, the threshold is updated to one-tenth of the previous MOC fission source error. The maximum number of CMFD outer iterations is limited to 200. For the CMFD inner iteration, the maximum number of linear solver iterations is set to 30, or it stops earlier if the flux residual is reduced to half the value of the CMFD fission source error, whichever occurs first. In practice, the flux usually converges within the 30-iteration limit.

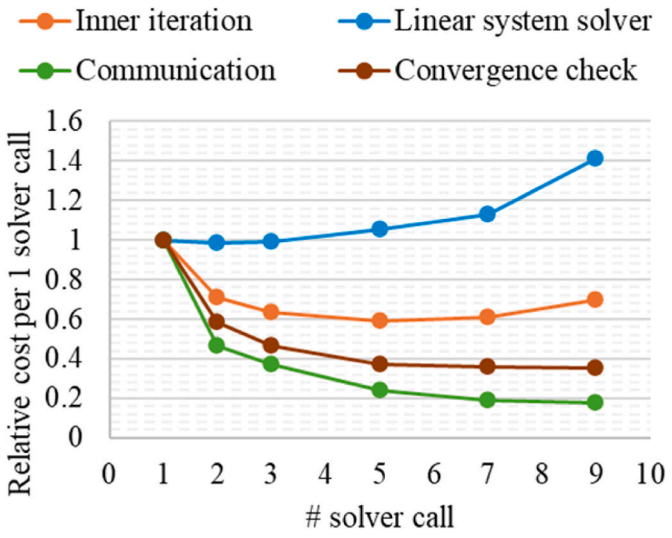

Fig. 3. Relative cost with varying numbers of solver calls on inner iteration loop.

3. Improvement of CMFD in STREAM

3.1. Adoption of improved odCMFD

The odCMFD was initially developed as the generalization for the traditional CMFD and pCMFD by introducing an arbitrary adjustable factor, θ_{od}, into the diffusion coefficient as expressed below:

$$D_{opt} = \frac{1}{3\Sigma_t} + \theta_{od}\Delta \tag{12}$$

where, D_{opt} is the optimal diffusion coefficient and Δ is the mesh size. When θ_{od} is set to 0, the diffusion coefficient reduces to that of the traditional CMFD. On the other hand, setting θ_{od} as 0.25 represents the pCMFD method.

From the developer perspective, this adjustment requires only minor modifications but can significantly enhance simulation performance. The value of θ_{od} itself is built from a specific set of polynomials derived from a Fourier analysis tailored to a particular problem. Initially, a monoenergetic infinite homogeneous problem was used to define the θ_{od} value. STREAM adopted this value.

Recently, work has been done to find more appropriate θ_{od} values for the quasi-3D MOC approach in STREAM. In the quasi-3D approach, the axial spatial distribution is integrated into the planar MOC equation. However, research has shown that the radial and axial neutron net current can impact CMFD acceleration differently in quasi-3D transport calculations [18]. Consequently, correction terms for the diffusion coefficient should be considered separately for the radial and axial directions, resulting in two sets of polynomials listed in Table 1. The a, b, c, and d is the fitting coefficient which can be found in Ref. [18].

This improved adjustable factor is expected to give computational gain by allowing larger axial coarse mesh to be used at the cost of slightly reduced accuracy. It is important to note that in STREAM, both the MOC and CMFD use identical axial meshes. This implies that optimizing the axial coarse mesh for CMFD could reduce the overall computational workload.

3.2. Relocation of angular flux prolongation

At the final stage of the CMFD acceleration, the converged multiplication factor and flux are transferred to the transport kernel or known as prolongation. STREAM uses flat prolongation as shown in Equation (10) and Equation (11) which involves the new and old CMFD flux ratio for scaling. This ratio is used to prolong the MOC scalar flux and outgoing angular flux. Notably, it has been observed that the prolongation

Table 1
Improved adjustable factor for quasi-3D MOC transport [18].

θ_{od}^{axial}	θ_{od}^{radial}
$0, \Sigma_t\Delta \le 0.45$	$0, \Sigma_t\Delta \le 1.75$
$\begin{cases} \sum_{i=0}^{6} a_i(\Sigma_t\Delta)^i, & 0.45 < \Sigma_t\Delta \le 11 \\ \max\left(0.1, \sum_{i=0}^{6} b_i(\Sigma_t\Delta)^i\right), & \Sigma_t\Delta > 11 \end{cases}$	$\begin{cases} \sum_{i=0}^{6} c_i(\Sigma_t\Delta)^i, & 1.75 < \Sigma_t\Delta \le 12.5 \\ \max\left(0.01, \sum_{i=0}^{5} d_i(\Sigma_t\Delta)^i\right), & \Sigma_t\Delta > 12.5 \end{cases}$

of the MOC outgoing angular flux is about ten times more computationally demanding than that of the scalar flux due to the large numbers of simulated rays.

Since the outgoing angular flux is required to update the incoming angular flux before the MOC/DD transport calculation begins, there is an opportunity to optimize performance by relocating the prolongation of angular flux from PROLONG_MOC routine in the CMFD module to the GET_PSI_IN_MPI_RB routine in the transport module. This target routine is responsible for updating the incoming angular fluxes for each assembly and for managing the MPI communication of these fluxes.

4. Numerical results

The numerical simulations are performed on a CPU node operating in a 64-bit Linux system with AMD EPYC 7543 processor (2.9 GHz, 2 × 32 cores) and 2 TB memory. All simulations are simulated with 32 MPI processes.

4.1. C5G7 benchmark

The first test problem is the well-known 3D C5G7 benchmark [20]. There are three sets of simulated configurations: unrodded case, rodded A case, and rodded B case. The coarse mesh size in the radial direction corresponds to the size of a pin cell, which measures 1.26 cm × 1.26 cm for this problem. The MOC option is set to 0.03 cm, 64 azimuthal, and six polar angles. The axial mesh is equally divided into heights of 1.19 cm, 3.57 cm, and 7.14 cm, resulting on 108, 36, and 18 planes respectively, for each configuration. The configuration with the largest number of planes serves as the baseline model for performance and accuracy comparison. The whole-core problem is simulated in this study, with vacuum boundary conditions set in all directions.

A test run without CMFD acceleration is first conducted for the C5G7 unrodded case with 108 axial planes. Without acceleration, the number of transport iterations required to get the convergence solution is 341, resulting in simulation time up to 23 h. In contrast, simulation with CMFD acceleration, only needs 8 iterations with total runtime of about 40 min, which is 33 times more efficient. This comparison clearly demonstrates the importance of CMFD acceleration in enhancing the convergence rate in STREAM code.

Table 2 lists the performance benefits when using the improved odCMFD. The number of transport iterations and total execution time are metrics that are evaluated. For the baseline model of C5G7 problem, the number of iterations required to solve the problem is larger when using improved odCMFD. However, as the size of axial mesh is increased, the advantage of the improved odCMFD became evident. While the original odCMFD exhibited divergence for coarser meshes, the improved odCMFD remained stable and successfully converged. Despite the increase in iteration count when using coarser mesh, the code still achieves a speedup of up to 2.5 times.

Table 3 compares the C5G7 numerical results between the original and improved odCMFD in terms of the eigenvalue (keff) and root-mean-square (RMS) of relative pin power difference. For baseline models, both methods deliver comparable results. By reducing the number of simulated planes, it results in deviations of up to 72 pcm and 0.172 % for the eigenvalue and pin power, respectively.

Table 3
Numerical results with original odCMFD and improved odCMFD for C5G7 problems.

Problem	Keff odCMFD	Keff Improved odCMFD	Keff difference[a]	RMS of pin relative difference[a]
C5G7 unrodded	1.14294	1.14295	1 pcm	0.024 %
	–	1.14306	12 pcm	0.153 %
	–	1.14281	−13 pcm	0.156 %
C5G7 rodded A	1.12801	1.12802	1 pcm	0.023 %
	–	1.12812	11 pcm	0.146 %
	–	1.12764	−37 pcm	0.172 %
C5G7 rodded B	1.07959	1.07959	0	0.035 %
	–	1.07953	−6 pcm	0.100 %
	–	1.07887	−72 pcm	0.169 %

[a] Compared against baseline model of original odCMFD.

4.2. Realistic core problems

Three representative large core problems are simulated to evaluate the implementation of the improved odCMFD method in STREAM code. The first two problems involve two Korean reactors, OPR-1000 and APR-1400. Both reactors utilize 16 × 16 assembly lattices with gadolinia as the burnable absorber. The OPR-1000 core model employed in this study is adapted from Ref. [3], with marginal modification to the supporting structure. Simulation is conducted under hot full power (HFP) conditions at the beginning of cycle (BOC), with thermal hydraulic and equilibrium xenon feedback. The radial and axial configurations of the OPR-1000 core are provided in Fig. 4.

For APR-1400, the model is developed to closely follow the specifications outlined in benchmark [21]. Specifically, the simulation represents the HFP condition at BOC, corresponding to the APR04V06 case within the benchmark documentation. Neither thermal-hydraulic nor xenon feedback is considered in this simulation.

In addition to the Korean reactors, the BEAVRS benchmark core [22] is simulated at the BOC state. This benchmark employs a 17 × 17 assembly lattice, using Pyrex as the burnable absorber material. The boron concentration in moderator is 975 ppm. Two calculation conditions are considered for this core: hot zero power (HZP) and hot full power (HFP). In the HZP scenario, both thermal-hydraulic and equilibrium xenon feedback are disabled, while these feedbacks are enabled in the HFP scenario.

All reactor problems analyzed in this study had been previously simulated using the STREAM code with the original odCMFD formulation and were either verified against Monte Carlo reference calculation or validated with measurement data. For instance, the OPR-1000 depletion simulation using STREAM was validated against measurement data in Ref. [3], while the BEAVRS benchmark had been verified and validated in Refs. [2,3] respectively. Verification of STREAM results

Table 2
Performance of original odCMFD and improved odCMFD for C5G7 problems.

Problem	Plane height [cm]	# of planes	odCMFD		Improved odCMFD	
			Iteration	Time [min]	Iteration	Time [min]
C5G7 unrodded	1.19	108	8	40.4	11	49.1
	3.57	36	Diverged	–	19	29.6
	7.14	18	Diverged	–	17	15.0
C5G7 rodded A	1.19	108	8	40.3	11	49.0
	3.57	36	Diverged	–	19	28.8
	7.14	18	Diverged	–	17	15.5
C5G7 rodded B	1.19	108	8	40.3	11	49.1
	3.57	36	Diverged	–	17	28.0
	7.14	18	Diverged	–	25	22.5

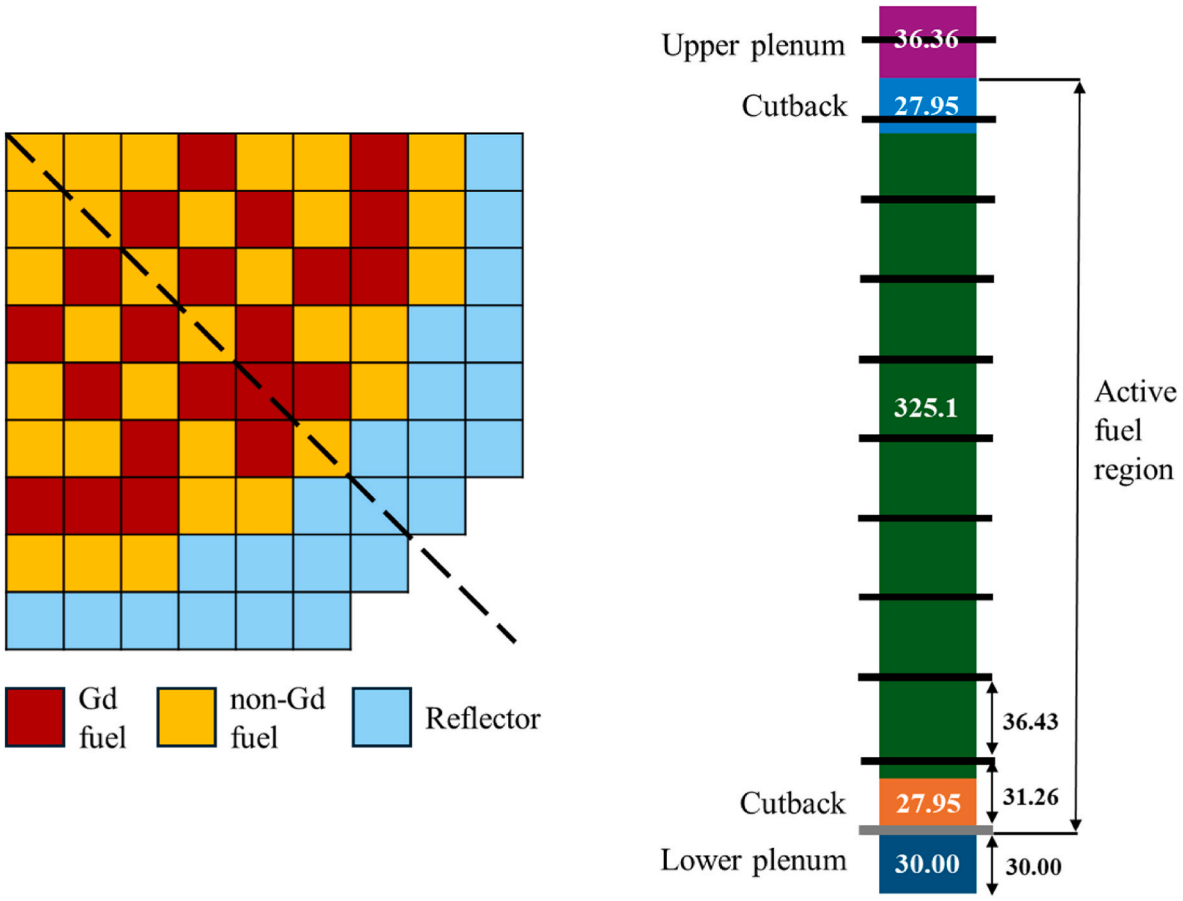

Fig. 4. Core configuration of OPR-1000 reactor: radial view (left) and axial view (right).

for the APR-1400 benchmark is available in Ref. [23]. These prior verifications and validations demonstrate that the reactor models employed in this work are representative, providing a reliable foundation for assessing the improved odCMFD in STREAM.

Table 4 lists the calculation parameters used for the baseline models of all three reactors. To examine the benefits of the improved odCMFD approach, two additional models are created for each core, using maximum axial plane heights of 6 cm and 10 cm. Three cases for each core problem are executed using STREAM with original odCFMD and STREAM with improved odCMFD methods.

Table 5 summarizes the computational performance of all cases. Several key observations can be drawn.

1) For the baseline cases, the computational cost of both methods is comparable across all problems.
2) When the axial resolution is reduced, the original odCMFD fails to converge due to divergence in CMFD acceleration.

Table 4
Calculation parameters for realistic core problems.

Parameter	OPR-1000	APR-1400	BEAVRS
Core symmetry	Octant	Octant	Octant
# of assemblies	41	48	43
Ray spacing/azi. angle/pol. angle	0.05/48/6	0.05/48/6	0.05/48/6
Max. plane height	3 cm	3 cm	3 cm
# of axial planes	197	182	181
# of axial planes in active region	169	136	145
Fuel pin (rings/sector)	2/8	3/8	3/8
Gd pin (rings/sector)	10/8	10/8	–
Total FSRs	68,686,808	92,189,370	88,820,139

Table 5
Performance of original odCMFD and improved odCMFD for realistic core problems.

Problem	Max. plane height [cm]	# of planes	odCMFD		Improved odCMFD	
			Iteration	Time [hr]	Iteration	Time [hr]
OPR-1000[a]	3	197	23	8.5	23	8.6
	6	106	Diverged	–	22	4.9
	10	65	Diverged	–	22	3.4
APR-1400	3	182	10	5.7	10	5.3
	6	114	Diverged	–	12	3.9
	10	65	Diverged	–	20	3.7
BEAVRS HZP	3	181	10	4.6	10	4.3
	6	115	Diverged	–	12	3.2
	10	68	Diverged	–	24	4.0
BEAVRS HFP[a]	3	181	23	11.4	24	11.0
	6	115	Diverged	–	23	7.1
	10	68	Diverged	–	24	5.0

[a] Thermal-hydraulic and equilibrium xenon feedback is applied.

3) In problems with cross-section feedback, coarse mesh simulations with improved odCMFD require a similar number of iterations as the corresponding baseline cases. As a result, computational speedup is nearly proportional to the reduction in the number of planes, leading to substantial savings in computational time.
4) In problems without cross-section feedback, coarse mesh simulations with improved odCMFD require more iterations to converge. This limits the overall runtime speedup compared to cases with cross-section feedback, though improvements are still evident.

Taken together, these results demonstrate that the improved

odCMFD method achieves robust convergence across all cases while maintaining computational performance that is either comparable to or better than the original odCMFD. The efficiency benefit is especially notable in cases with cross-section feedback, where the reduction in the number of axial planes translates almost directly into computational speedup.

Table 6 compares the numerical results between the original odCMFD and improved odCMFD in terms of the eigenvalue (keff) and RMS of relative pin power difference. For cases employing fine axial meshes, both methods show comparable results, with eigenvalue difference within 3 pcm and RMS difference below 0.03 %. Slight discrepancies are observed when the axial resolution is reduced, with maximum eigenvalue difference of 12 pcm and RMS difference of about 0.1 %.

Figs. 5 and 6 provide a more detailed comparison of pin power differences for two Korean reactor cores and the BEAVRS core problems, respectively. These figures depict the relative differences per pin, calculated as (coarsest case with improved odCMFD/finest case with original odCMFD) – 1. This means that a positive value indicates the coarse mesh model overpredicts the fine mesh model, while a negative value indicates underprediction. Problems without cross-section feedback display more visible differences with an in-out tilt pattern, while cases with cross-section feedback appear more flattened with more localized differences in assemblies containing burnable absorber. This behavior stems from the coarser axial mesh, which affects axial streaming in the MOC/DD transport as both CMFD and MOC rely on the same axial discretization. Nevertheless, given the computational advantages, these slight differences are justified.

Overall, these results demonstrate that the implementation of improved odCMFD in STREAM is more robust than the original, as it can handle coarser mesh sizes in the axial direction. More importantly, it allows high-fidelity simulations with cross-section feedback to achieve a significant speedup without compromising numerical accuracy.

4.2.1. *Impact of relocating angular flux prolongation*

To assess the impact of relocating angular flux prolongation kernel, timers are set to the PROLONG_MOC and GET_PSI_IN_MPI_RB routines. Two problems are considered, OPR-1000 and BEAVRS HFP, each modeled with a maximum height of 10 cm. Table 7 shows the runtime of each routine before and after implementing the relocation. All simulations are run twice.

Relocating the angular flux prolongation kernel leads to a substantial reduction in the runtime of the PROLONG_MOC routine, confirming that the angular flux prolongation is the dominant contributor to its computational cost. These improvements remain consistent between runs. On the other hand, the runtime of the GET_PSI_IN_MPI_RB routine exhibits noticeable variations between runs. This variability arises due to the algorithm implemented in STREAM when decomposing the reactor core. In STREAM, the number of domains is equal to number of MPI processes, with minimum size of domain is an assembly. STREAM algorithms allow a domain to have assemblies that separated each other. By allowing domains to consist of separate, not necessarily adjacent assemblies, the combinatorial possibilities for balanced decomposition increase, making it easier to optimize the load distribution. This algorithm makes the location of assembly within domains may differ between runs.

Prior to relocation, the PROLONG_MOC routine accounted for about 6 % of the total CMFD runtime in the OPR-1000 case and 5 % in the BEAVRS HFP case. These values are relatively small compared to the CMFD solver routine, which occupies up to 75 % of the total CMFD runtime. After relocation, considering only the improvements within the CMFD module, the runtime is reduced by approximately 14 s for the OPR-1000 problem and 19 s for the BEAVRS HFP problem, lowering the contribution of PROLONG_MOC to around 0.6 % for both cases.

For realistic core problems such as OPR-1000, APR-1400, and BEAVRS simulated in this study, CMFD itself represents only about 3–4 % of the total simulation time, with over 80 % of the computational cost associated with the MOC/DD transport. However, recent work has shown that GPU acceleration of the MOC/DD transport significantly increases the relative share of the CMFD module in the global runtime [24]. Furthermore, the CMFD solver runtime can also be substantially reduced through GPU optimization [24]. In that sense, minimizing the PROLONG_MOC runtime within the CMFD module becomes increasingly valuable.

Overall, this optimization provides measurable improvement to the PROLONG_MOC routine while introducing only a minimal, if any, overhead to the GET_PSI_IN_MPI_RB routine. Importantly, this refinement introduces no numerical deviations, since no additional mathematical operations are involved in relocating the angular flux prolongation.

Table 6
Numerical results with original odCMFD and improved odCMFD for realistic core problems.

Problem	# of planes	Keff odCMFD	Keff Improved odCMFD	Keff difference[a]	RMS of pin relative difference[a]
OPR-1000[b]	197	0.96930	0.96930	0	0.012 %
	106	–	0.96925	−4 pcm	0.035 %
	65	–	0.96920	−10 pcm	0.055 %
APR-1400	182	1.00970	1.00967	−3 pcm	0.032 %
	114	–	1.00963	−7 pcm	0.068 %
	65	–	1.00958	−12 pcm	0.109 %
BEAVRS HZP	181	0.99925	0.99925	0	0.010 %
	115	–	0.99922	−3 pcm	0.061 %
	68	–	0.99920	−5 pcm	0.110 %
BEAVRS HFP[b]	181	0.95862	0.95865	3 pcm	0.030 %
	115	–	0.95858	−4 pcm	0.024 %
	68	–	0.95854	−8 pcm	0.050 %

[a] Compared against baseline model of original odCMFD.
[b] Thermal-hydraulic and equilibrium xenon feedback is applied.

5. Conclusion

This paper revisits the implementation of CMFD acceleration in STREAM and the recent improvements of it. The formulation and workflow of the two-level CMFD approach are detailed. The radial decomposition in STREAM allows CMFD linear system to be solved per assembly. The flux solution within an assembly is obtained using the Gauss-Seidel iterative method, which is parallelized via planar red-black scheme with OpenMP threads. To maintain solution continuity across the entire core, flux information is exchanged between assemblies using MPI communication. To mitigate communication overhead, a strategy involving multiple linear solver calls within a single iteration loop is employed.

The improved CMFD method has been implemented in STREAM to enhance the performance and robustness of CMFD acceleration. This improvement is complemented with fine-tuning at the programming level by relocating the prolongation of angular flux from CMFD module to transport module.

The numerical results demonstrate significant improvements in the computational efficiency of STREAM. The improved odCMFD method enables stable and accurate simulations even with coarser axial meshes, providing substantial speedups without compromising numerical accuracy. Relocating the angular flux prolongation kernel further improve the CMFD runtime without affecting numerical accuracy. Collectively, these improvements make STREAM more efficient for large-scale reactor simulations, while ensuring high-fidelity solution.

CRediT authorship contribution statement

Fathurrahman Setiawan: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology,

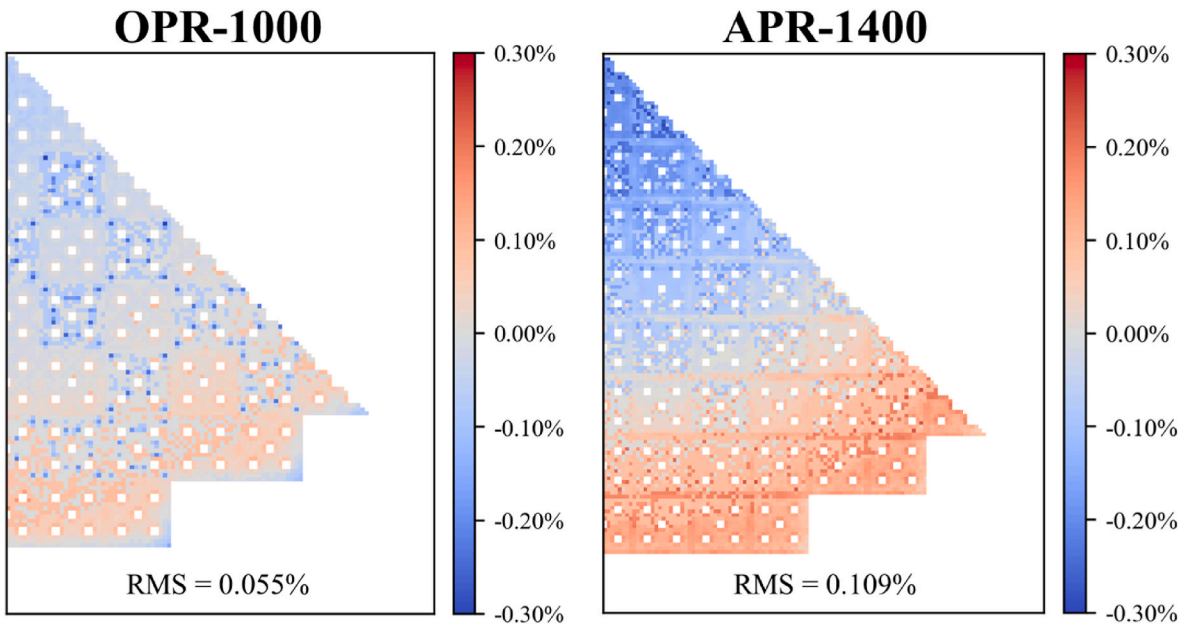

Fig. 5. Relative pin power difference between original and improved odCMFD for Korean reactors.

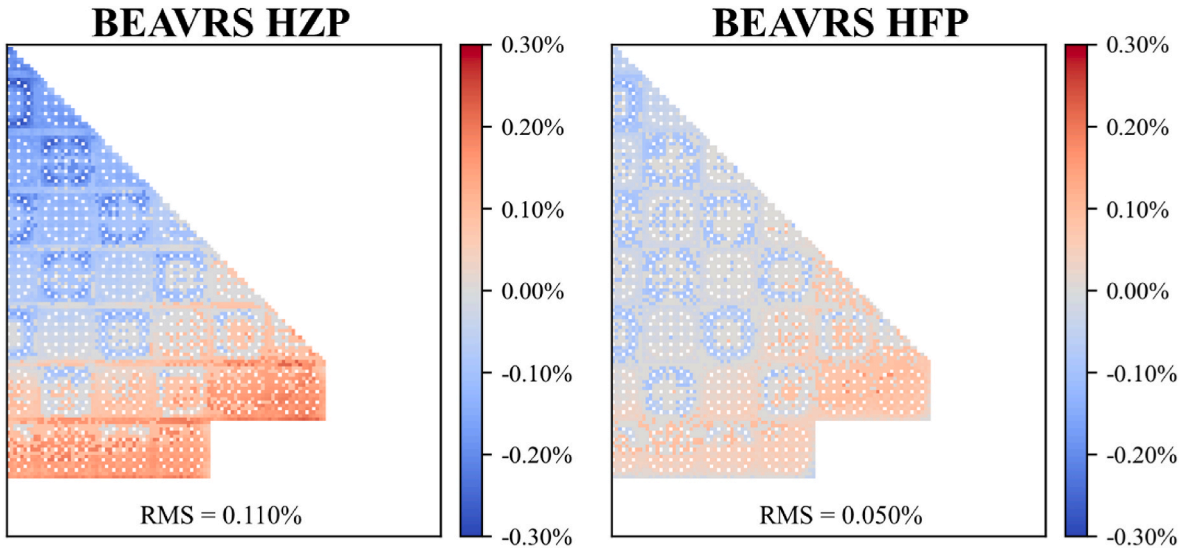

Fig. 6. Relative pin power difference between original and improved odCMFD for BEAVRS core problems.

Table 7
Comparison of runtime before and after relocation of angular flux prolongation.

Problem	Before relocation			After relocation			Time saving[b] [s]
	PROLONG_MOC [s]	GET_PSI_IN_MPI_RB [s]	Total[a] [s]	PROLONG_MOC [s]	GET_PSI_IN_MPI_RB [s]	Total[a] [s]	
OPR-1000	16.5	242.4	258.9	1.8	258.6	260.4	−1.5
OPR-1000 (rerun)	16.6	238.8	255.4	1.7	242.7	244.4	11
BEAVRS HFP	21.2	298.7	319.9	2.5	302.9	305.4	14.5
BEAVRS HFP (rerun)	21.0	291.7	312.7	2.5	285.5	288	24.7

[a] Total is PROLONG_MOC + GET_PSI_IN_MPI_RB.
[b] Time saving is Total before relocation – Total after relocation.

Investigation, Formal analysis, Conceptualization. **Siarhei Dzianisau:** Writing – review & editing, Methodology, Conceptualization. **Deokjung Lee:** Writing – review & editing, Supervision, Resources, Project administration, Funding acquisition.

Conflict of interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was supported by the Innovative Small Modular Reactor Development Agency grant funded by the Korea Government (MOTIE) (No.RS-2024-00407975).

References

[1] J. Choe, S. Choi, P. Zhang, J. Park, W. Kim, H.C. Shin, H.S. Lee, J. Jung, D. Lee, Verification and validation of STREAM/RAST-K for PWR analysis, Nucl. Eng. Technol. 51 (2019) 356–368, https://doi.org/10.1016/j.net.2018.10.004.
[2] S. Choi, D. Lee, Three-dimensional method of characteristics/diamond-difference transport analysis method in STREAM for whole-core neutron transport calculation, Comput. Phys. Commun. 260 (2021), https://doi.org/10.1016/j.cpc.2020.107332.
[3] S. Choi, W. Kim, J. Choe, W. Lee, H. Kim, B. Ebiwonjumi, E. Jeong, K. Kim, D. Yun, H. Lee, D. Lee, Development of high-fidelity neutron transport code STREAM, Comput. Phys. Commun. 264 (2021), https://doi.org/10.1016/j.cpc.2021.107915.
[4] A. Rahman, H.C. Lee, D. Lee, High fidelity transient solver in STREAM based on multigroup coarse mesh finite difference method, Nucl. Eng. Technol. 55 (2023) 3301–3312, https://doi.org/10.1016/j.net.2023.05.019.
[5] K.S. Smith, Nodal method storage reduction by nonlinear iteration, Trans. Am. Nucl. Soc. 44 (1983) 265–266.
[6] J. Park, P. Zhang, H. Lee, S. Choi, J. Yu, D. Lee, Performance evaluation of CMFD on inter-cycle correlation reduction of monte carlo simulation, Comput. Phys. Commun. 235 (2019) 111–119, https://doi.org/10.1016/j.cpc.2018.09.014.
[7] M. Jarret, B. Kochunas, A. Zhu, T. Downar, Analysis of stabilization techniques for CMFD acceleration of neutron transport problems, Nucl. Sci. Eng. 184 (2016) 208–227, https://doi.org/10.13182/NSE16-51.
[8] L. Li, K. Smith, B. Forget, Techniques for stabilizing coarse-mesh finite difference (CMFD) in methods of characteristics (MOC), in: Int. Conf. Math. Comput. (M&C) 2015, Nashville, Tennessee, 2015.
[9] A. Yamamoto, Generalized coarse-mesh rebalance method for acceleration of neutron transport calculations, Nucl. Sci. Eng. 151 (2005) 274–282, https://doi.org/10.13182/NSE151-274.
[10] D. Lee, Convergence analysis of coarse mesh finite difference method applied to two-group three-dimensional neutron diffusion problem, J. Nucl. Sci. Technol. 49 (2012) 926–936, https://doi.org/10.1080/00223131.2012.712478.
[11] D. Wang, S. Xiao, A linear prolongation approach to stabilizing CMFD, Nucl. Sci. Eng. 190 (2018) 45–55, https://doi.org/10.1080/00295639.2017.1417347.
[12] Q. Shen, Y. Xu, T. Downar, Stability analysis of the CMFD scheme with linear prolongation, Ann. Nucl. Energy 129 (2019) 298–307, https://doi.org/10.1016/j.anucene.2019.02.011.
[13] Y. Chan, S. Xiao, Convergence study of variants of CMFD acceleration schemes for fixed source neutron transport problems in 2D Cartesian geometry with fourier analysis, Ann. Nucl. Energy 134 (2019) 273–283, https://doi.org/10.1016/j.anucene.2019.06.021.
[14] J. Li, Y. Xu, D. Wang, Q. Shen, B. Kochunas, T. Downar, Demonstration of A linear prolongation CMFD method on MOC, in: PHYSOR 2020, 2020. Cambridge, UK.
[15] N.-Z. Cho, G.S. Lee, C.J. Park, Partial current-based CMFD acceleration of the 2D/1D fusion method for 3D whole-core transport calculations, Trans. Am. Nucl. Soc. 88 (2003), 594–594.
[16] A. Zhu, M. Jarrett, Y. Xu, B. Kochunas, E. Larsen, T. Downar, An optimally diffusive Coarse Mesh finite difference method to accelerate neutron transport calculations, Ann. Nucl. Energy 95 (2016) 116–124, https://doi.org/10.1016/j.anucene.2016.05.004.
[17] X. Zhou, Z. Liu, L. Cao, H. Wu, The improved odCMFD method based on the 2D/1D fourier analysis, Ann. Nucl. Energy 186 (2023), https://doi.org/10.1016/j.anucene.2023.109775.
[18] X. Zhou, H. Zhang, C. Zhao, B. Wang, W. Zhao, Z. Chen, Z. Gong, The improved odCMFD method accelerated for quasi-3D method based on fourier analysis, Ann. Nucl. Energy 206 (2024), https://doi.org/10.1016/j.anucene.2024.110651.
[19] K. Smith, J. Rhodes, Full-Core, 2-D, LWR core calculations with CASMO-4E, in: PHYSOR 2002, 2002. Seoul, Korea.
[20] M.A. Smith, E.E. Lewis, B.C. Na, Benchmark on deterministic 3-D MOX fuel assembly transport calculations without spatial homogenization, Prog. Nucl. Energy 48 (2006) 383–393, https://doi.org/10.1016/j.pnucene.2006.01.002.
[21] S. Yuk, APR1400 Reactor Core Benchmark Problem, RPL-INERI-CA-004, KAERI, Daejeon, South Korea, 2019.
[22] N. Horelik, B. Herman, B. Forget, K. Smith, Benchmark for evaluation and validation of reactor simulations (BEAVRS), in: Int. Conf. Math. Comput. Meth. Appl. Nucl. Sci. Eng. (M&C) 2013, 2013. Sun Valley, ID.
[23] R. Lee, Verification of STREAM3D for APR1400 Reactor Core Benchmark, Ulsan National Institute of Science and Technology (UNIST), Department of Nuclear Engineering, 2022. Master Thesis.
[24] F. Setiawan, S. Dzianisau, D. Lee, Introduction of GPU-enabled CMFD acceleration for performance enhancement of neutron transport code STREAM3D-GPU, in: M&C 2025, April 27-30, 2025. Denver, CO, USA.