



A multithreaded parallel upwind sweep algorithm for the S_N transport equations discretized with discontinuous finite elements

Zhi-Wei Zong¹ · Mao-Song Cheng^{1,2} · Ying-Chi Yu^{1,2} · Zhi-Min Dai^{1,2}

Received: 23 April 2023 / Revised: 1 July 2023 / Accepted: 4 August 2023 / Published online: 11 December 2023

© The Author(s), under exclusive licence to China Science Publishing & Media Ltd. (Science Press), Shanghai Institute of Applied Physics, the Chinese Academy of Sciences, Chinese Nuclear Society 2023

Abstract

The complex structure and strong heterogeneity of advanced nuclear reactor systems pose challenges for high-fidelity neutron-shielding calculations. Unstructured meshes exhibit strong geometric adaptability and can overcome the deficiencies of conventionally structured meshes in complex geometry modeling. A multithreaded parallel upwind sweep algorithm for S_N transport was proposed to achieve a more accurate geometric description and improve the computational efficiency. The spatial variables were discretized using the standard discontinuous Galerkin finite-element method. The angular flux transmission between neighboring meshes was handled using an upwind scheme. In addition, a combination of a mesh transport sweep and angular iterations was realized using a multithreaded parallel technique. The algorithm was implemented in the 2D/3D S_N transport code ThorSNIPE, and numerical evaluations were conducted using three typical benchmark problems: IAEA, Kobayashi-3i, and VENUS-3. These numerical results indicate that the multithreaded parallel upwind sweep algorithm can achieve high computational efficiency. ThorSNIPE, with a multithreaded parallel upwind sweep algorithm, has good reliability, stability, and high efficiency, making it suitable for complex shielding calculations.

Keywords Shielding calculation · Discrete ordinates method · Discontinuous Galerkin finite element method · Unstructured meshes

1 Introduction

High-fidelity neutron-shielding calculations are an essential part of advanced nuclear reactor system design. The discrete ordinate (S_N) method is one of the most popular deterministic particle transport methods owing to its high angular resolution [1]. Reliable modeling and simulation of S_N transport depend on two key essential conditions and fundamental components: reasonable spatial discretization schemes and mesh distributions with sufficient modeling

fidelity. The two key factors interacted with and supported each other throughout the process.

Conventional Cartesian structured grids are widely used and implemented because of their regular arrangement and friendliness to finite-difference (FD) spatial discretization schemes [2–4]. However, there is no general robust meshing algorithm for acquiring exact descriptions of complex structures for realistic modeling and simulations [5]. Adaptive mesh refinement techniques retain the original mesh distribution features and mark cells with large errors for refinement [6]. Local error estimation techniques and geometric data structures play significant roles in determining the stabilities and computational efficiencies of these systems. Unstructured grids use piecewise flat or curved surfaces to provide complicated geometries with the highest possible flexibility. With the development of the finite element method (FEM), unstructured grids have been widely applied to reactor physical analyses [7], multiphysics simulations [8], and thermal-fluid dynamics [9]. Combining finite element numerical methods and the SN method with unstructured meshes has become a trend in nuclear shielding design.

This work was supported by the Strategic Pilot Science and Technology Project of the Chinese Academy of Sciences.

✉ Mao-Song Cheng
chengmaosong@sinap.ac.cn

¹ Shanghai Institute of Applied Physics, Chinese Academy of Sciences, Shanghai 201800, China

² University of Chinese Academy of Sciences, Beijing 100049, China

Unfortunately, the first-order SN equation cannot be solved stably using the continuous FEM [10]. However, while the solutions to this hyperbolic equation are smooth along the streamlines, they may be discontinuous perpendicular to the chosen direction. To obtain a robust solution, several Petrov–Galerkin (PG) approaches that update the test function as the sum of the trial functions and introduce an additional stabilization term [11–13]. It was observed that the free stabilization parameter affects the convergence of the S_N equation and raises uncertainty because it is usually considered as a matter of experience. Additionally, PGs cannot generally produce symmetric positive-definite coefficient matrices or inherit the best approximation property when the transport equations contain nonself-adjoint operators.

The discontinuous Galerkin finite element method (DGFEM) is a different category of stability improvement. This was pioneered by Reed and Hill for neutron transport problems in the early 1970s [14]. The DGFEM connects adjacent elements through the incoming interface fluxes from the upwind (upstream) meshes and applies the usual FEM procedure to a single mesh. However, the number of unknowns in the DGFEM is much larger than that in the continuous FEMs, which demands a matrix assembly and solution method. The standard matrix assembly procedure looks for a reliable iterative solver for a banded sparse system matrix solution and must handle the storage brought on by the increased freedom [15]. The matrix-free strategy stores the diagonal blocks and realizes the inversion of the transport operator by means of the matrix–vector product; however, a sum factorization technique to enable translation between vector entries and values or gradients in quadrature points is required [16]. Another commonly used technique is the conventional sweeping procedure, such as FDs, in which parallelization of the sweeping procedure is implemented to reduce the computing time, which may become important in a large SN order.

In terms of parallelization, complex grid connection relationships and irregular element sweep orders pose challenges to the algorithms. In 2000, Plimpton described an asynchronous, parallel message-passing algorithm that simultaneously performs sweeps from many directions across unstructured grids [17]. In 2002, Pautz used a low-complexity list-ordering heuristic to determine sweep ordering on a partitioned mesh [18]. The essence of these two algorithms is the conversion of the abstract element sweep order into a directed acyclic graph (DAG). Since 2010, several research institutions have conducted extensive studies on high-performance architectural applications [19, 20].

A multigroup 2D SN transport code, ThorSNIPE, with unstructured meshes was developed and validated [21]. This code was extended to 3D SN transport calculations and employed a multithreaded parallel upwind sweep algorithm to achieve a more accurate description and improve the

reliability and efficiency of transport calculations in complex geometries. The finite element method was implemented using an open-source finite element library deal.II-9.4.0 [22]. The deal.II is an open-source C++ program library targeted at the computational solution of partial differential equations using adaptive finite elements and focuses on generality, dimension-independent programming, parallelism, and extensibility. It includes many state-of-the-art programming techniques for solving partial differential equations, linear algebra problems, and computer science strategies and offers users a modern interface for complex data structures and algorithms. With the assistance of this library, ThorSNIPE can be made dimension-independent, supporting different mesh sizes and types of finite elements.

The remainder of this paper is organized as follows. In Sect. 2, the main theory and methodology of the ThorSNIPE code are described in detail, including the S_N weak-form equation based on DGFEM theory and the multithreaded parallel upwind sweep algorithm. The numerical results are presented and discussed in Sect. 3. The conclusions are summarized in Sect. 4.

2 Theory and methodology

2.1 Weak form of the discrete ordinates method equations

The S_N form was obtained by solving the Boltzmann transport equation along discrete directions and replacing the integrals over the unit sphere with quadrature sets. The steady-state multi-group S_N neutron transport equation can be written as

$$\Omega_m \cdot \nabla \psi_m^g(r) + \Sigma_t^g(r) \psi_m^g(r) = Q_m^g(r), \quad g = 1, \dots, G; m = 1, \dots, M; r \in D \quad (1)$$

where D is the domain. $\psi_m^g(r)$ is the angular flux in group g , direction Ω_m and position r . Σ_t^g is the macroscopic total cross section in group g . $Q_m^g(r)$ is a source term in group g , direction Ω_m and position r . For fixed source problems, it can be expressed as:

$$Q_m^g(r) = \sum_{g'=1}^G \sum_{n=0}^N \frac{2n+1}{4\pi} \Sigma_{s,n}^{g' \rightarrow g}(r) \sum_{k=-n}^n \phi_n^k(r) Y_n^k(\Omega_m) + q_g(r) \quad (2)$$

and for eigenvalue problems, it can be expressed as:

$$Q_m^g(r) = \sum_{g'=1}^G \sum_{n=0}^N \frac{2n+1}{4\pi} \Sigma_{s,n}^{g' \rightarrow g}(r) \sum_{k=-n}^n \phi_n^k(r) Y_n^k(\Omega_m) + \frac{\chi_g}{4\pi k_{\text{eff}}} \sum_{g'=1}^G v^{g'}(r) \Sigma_f^{g'}(r) \phi_{g'}(r), \quad (3)$$

$$\phi_g(r) = \sum_{m=1}^M \omega_m \psi_m^g(r), \tag{4}$$

where G and M are total groups and angles. $\Sigma_{s,n}^{g' \rightarrow g}(r)$ is the n th-order Legendre moment of scattering cross section from group g' to group g at position r . $\Sigma_f^{g'}(r)$ is the macroscopic fission cross section in group g' at position r . $\nu^{g'}(r)$ is the average number of neutrons produced per fission in group g' . k_{eff} is the eigenvalue. $\phi_n^k(r)$ represents k th, n th-order flux moment at position r . $Y_n^k(\Omega_m)$ is k th, n th-order spherical harmonic in direction Ω_m at position r . $q_g(r)$ is fixed-source in group g at position r . $\phi_g(r)$ is scalar flux in group g at position r . ω_m is weight factor of quadrature set.

Equation (1) is a linear hyperbolic equation: It is necessary to transform this “strong form” differential-integral equation into “weak form” for solving. The spatial domain D is decomposed into K elements, which can be unstructured because the FEM is independent of the space dimension and grid choice. Consider ∂V_k as the surface of the k th element and V_k as the volume of the k th element, where $k = 1, \dots, K$. To obtain the weak form, Eq. (1) is multiplied by an arbitrary test function $\psi_m^*(r)$ and integrated into any element k . Equation (1) can be rewritten in monoenergetic form as

$$\int_{V_k} \Omega_m \cdot \nabla \psi_m(r) \psi_m^*(r) dV + \int_{V_k} \Sigma_t(r) \psi_m(r) \psi_m^*(r) dV = \int_{V_k} Q_m(r) \psi_m^*(r) dV. \tag{5}$$

The divergence theorem is then applied to Eq. (5), yielding:

$$\int_{\partial V_k} \Omega_m \cdot n(r) \psi_m^*(r) \psi_m(r) dS - \int_{V_k} \psi_m(r) \Omega_m \cdot \nabla \psi_m^*(r) dV + \int_{V_k} \Sigma_t(r) \psi_m(r) \psi_m^*(r) dV = \int_{V_k} Q_m(r) \psi_m^*(r) dV, \tag{6}$$

where $n(r)$ is the unit normal vector to the surface of cell ∂V_k at position r . Inner products on the volume and surface of any element are introduced to express the bilinear form [23].

$$\langle a, b \rangle_{V_k} = \int_{V_k} a(r) b(r) dV \tag{7}$$

$$(a, b)_{\partial V_k} = \int_{\partial V_k} |\Omega_m \cdot n(r)| a(r) b(r) dS \tag{8}$$

Substituting Eq. (7) and (8) in (6),

$$(\psi_m^*, \psi_m)_{\partial V_k} - \langle \psi_m^*, \Omega_m \nabla \psi_m \rangle_{V_k} + \langle \Sigma_t \psi_m^*, \psi_m \rangle_{V_k} = \langle Q_m, \psi_m^* \rangle_{V_k}. \tag{9}$$

The general boundary can be divided into three types based on its relative position: interior boundary, outflow domain boundary ($\partial V_k^+ = \{r \in \partial V_k, \Omega \cdot n(r) > 0\}$), and inflow domain boundary ($\partial V_k^- = \{r \in \partial V_k, \Omega \cdot n(r) < 0\}$). The inflow-domain boundary conditions were prescribed as either vacuum or reflective for the incoming directions. So, Eq. (9) can be rewritten as

$$(\psi_m^*, \psi_m^+)_{\partial V_k^+} - \langle \psi_m^*, \Omega_m \nabla \psi_m^* \rangle_{V_k} + \langle \Sigma_t \psi_m^*, \psi_m^* \rangle_{V_k} = \langle Q_m, \psi_m^* \rangle_{V_k} + (\psi_m^*, \psi_m^-)_{\partial V_k^-} + (\psi_m^*, \psi_m^{\text{in}})_{\partial V_k^{\text{in}}}, \tag{10}$$

where ψ_m^+ , ψ_m^- and ψ_m^{in} are the outflow angular, inflow angular, and interior face fluxes, respectively. Equation (10) represents the weak form of the S_N equation.

2.2 The discontinuous Galerkin finite element discretization scheme

The Galerkin method is theoretically based on the weighted residual method, which multiplies the original partial differential equation by a test function. Seeking a numerical approximation of the angular flux term using Lagrange polynomials of degree P

$$\psi_m = \sum_{j=1}^{J_p} \varphi_{m,j} b_j(r), \tag{11}$$

where $b_j(r)$ is the Lagrange shape function and $\varphi_{m,j}$ is an unsolved coefficient. The matrix form is convenient for solving Eq. (10), which expresses $\varphi_{m,j}$ and $b_j(r)$ in row vector notation, and Eq. (11) can be written as

$$\psi_m = [\varphi_{m,1}, \dots, \varphi_{m,J_p}] [b_1, \dots, b_{J_p}]^T = \varphi_m b^T. \tag{12}$$

In Galerkin discretization schemes, test functions share similar function space as basis functions,

$$\psi_m^* = [b_1, \dots, b_{J_p}]^T = b. \tag{13}$$

Substituting Eq. (12) and (13) in (10),

$$(b, b^T)_{\partial V_k^+} \varphi_m^+ - \langle b^T, \Omega_m \nabla b \rangle_{V_k} \varphi_m + \langle \Sigma_t b^T, b \rangle_{V_k} \varphi_m = \langle Q_m, \psi_m^* \rangle_{V_k} + (\psi_m^*, \psi_m^-)_{\partial V_k^-} + (\psi_m^*, \psi_m^{\text{in}})_{\partial V_k^{\text{in}}}. \tag{14}$$

The upwind scheme is the essence of the DGFEM. Notably, although the test functions on the edges are always taken from within the element, the primal functions on the edges are taken upwind with respect to the streaming direction. For outgoing edges, the primal functions are taken from within the element itself, whereas for incoming edges, the primal functions are taken from the upwind neighbor element.

2.3 Multithreaded parallel upwind sweep algorithm

2.3.1 Cell-based sweep algorithm

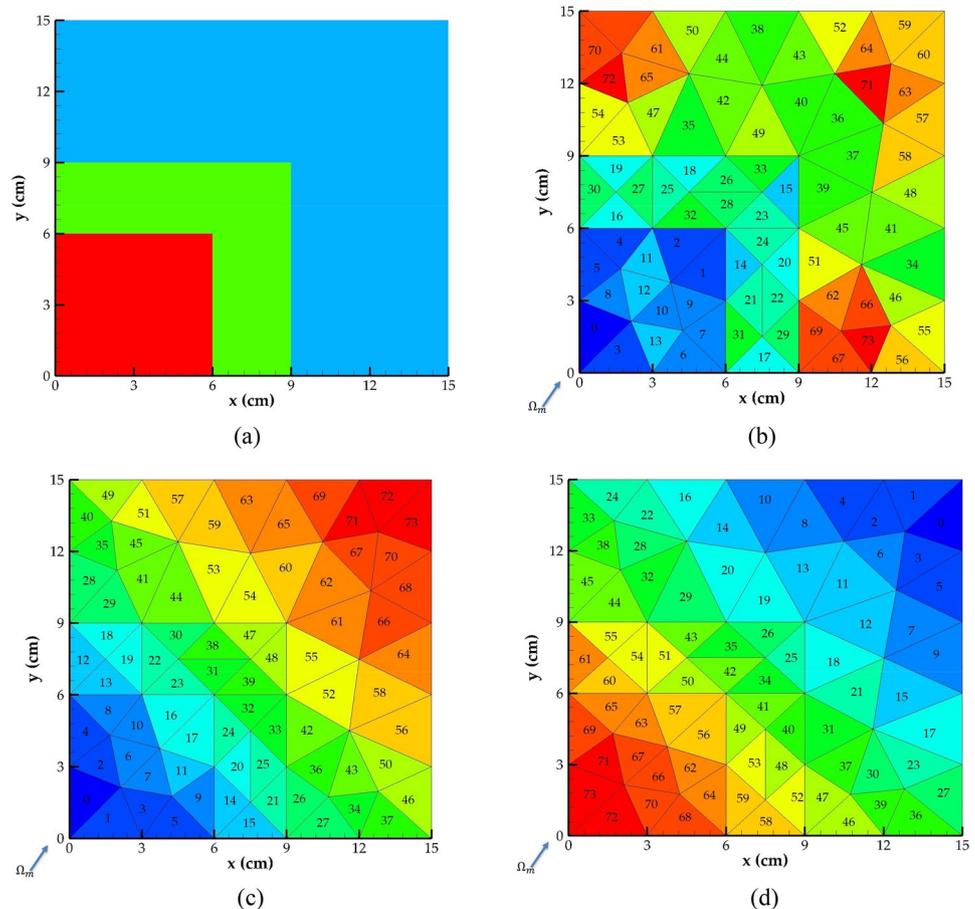
In the S_N method, a transport sweep was performed for all elements along each discrete direction. The ordering in which the elements and degrees of freedom (DOF) traverse affects the speed of convergence. As mentioned in Sect. 1, the DGFEM makes it possible to use a cell-based sweeping procedure in S_N methods rather than assembling a global matrix.

The element sweep order for each discrete direction in the ThorSNIFE code was developed using an upwind scheme based on the deal.II library. In our case, an array with all relevant cells is created and sorted in the upwind direction using a “dealii:: DoFRenummering:: CompareDownstream” object in the deal.II FEM library. A straightforward 2D model with three material regions and various sweep schemes is shown in Fig. 1. Cells in a conventional scheme are often categorized from edge to center to make it easier to facilitate the creation of a global matrix. The downwind system performs renumbering in the direction opposite to the discrete direction, whereas the upwind scheme manages

renumbering alongside it. The definitions of “upwind” and “downwind” are determined by multiplying the unit normal vector to the cell’s surface by the cosines of the corresponding discrete direction. In reality, the mesh location, discrete direction, and sweep define the order of each element. The sweep solution technique also complies with neutron transport laws. Not all pathways are mutually exclusive, and each distinct direction results in a unique order in which this solution technique can be applied.

From the perspective of solving the equations, various sweep schemes also affect the calculation of the incoming angular fluxes ψ_m^{in} in Eq. (14). Primal functions obtained from neighboring elements require numerical values with a certain precision. The calculation error was gradually reduced when the transport equation was solved along the direction of particle motion using the upwind scheme. However, other schemes must process nondirectional data, adapt, and learn based on the data received. This process can be laborious and error-prone, and the accumulation of errors inevitably affects the robustness of the numerical calculation method. Thus, the entire solution procedure required more iterations.

Fig. 1 (Color online) **a** Material region distribution; **b** Conventional scheme; **c** Upwind scheme; **d** Downwind scheme



Algorithm 1 displays the cell-based sweep algorithm for each group and angle once the sweep order has been determined. The local equation can be divided into two parts as follows: The transport matrix first makes contributions from the streaming and removal operators determined by the fixed mass matrix and total cross section. Second, the right-hand side (RHS) vector includes contributions from scattering source operations and fixed source or fission source operators. The in-group and cross-group couplings comprised the contributions of the scattering sources. Lastly, a “dealii::types:: global dof index” object translates the local solution to the global angular solution. This cell-based sweep technique delivers the DGFEM solution to all cells in a single sweep order without coupling integral terms, fully utilizing the key benefits of DGFEM approaches.

The following two steps were employed to compute the local contributions of an individual cell and RHS: First, the integral is transformed from an actual cell k into a unit/reference cell \hat{k} . For example, the streaming operator is transformed into

$$\int_{V_k} b^T(r)\Omega_m \nabla b(r) dV = \int_{V_{\hat{k}}} [J^{-1}(\hat{r})b^T(\hat{r})] [\Omega_m J^{-1}(\hat{r})\nabla b(\hat{r})] |\det J(\hat{r})| dV, \tag{15}$$

where the hat indicates the reference coordinates, and $J(\hat{r})$ is the Jacobian of the mapping $r = F_k(\hat{r})$. Second, this integral is approximated using the Gauss–Legendre quadrature. This yields the formula

Algorithm 1: cell-based sweep for each group and angle

```

Input:  $g, m, q_{fixed}(r)$ 
for upwind sorted cell do
  for DOF of cell  $i$  do
    for DOF of cell  $j$  do
       $transport\_matrix(i, j) = -\int_{V_k} b_i^T(r)\Omega_m \cdot \nabla b_j(r) dV + \int_{V_k} \Sigma_t(r)b_i(r)b_j^T(r) dV$ 
    for moment  $n$  do
      for moment  $k$  do
        for DOF of cell  $i$  do
           $source\_rhs(i) += \int_{V_k} b_i(r)(\Sigma_t(r)\phi_n^k(r) + q_{fixed}(r))Y_n^k(\Omega_m) dV$ 
        for faces on cell do
          if outflow face then
            for DOF of face  $i$  do
              for DOF of face  $j$  do
                 $boundary\_matrix(i, j) = \int_{\partial V_k^+} \Omega_m \cdot n(r)b_i(r)b_j^T(r) dS$ 
              else if inflow face then
                for DOF of face  $i$  do
                   $boundary\_rhs(i) = \begin{cases} \int_{\partial V_k^-} \Omega_m \cdot n(r)b_i(r)\psi_m^-(r) dS \\ \int_{\partial V_k^{in}} \Omega_m \cdot n(r)b_i(r)\psi_m^{in}(r) dS \end{cases}$ 
                 $cell\_rhs(i) = source\_rhs(i) + boundary\_rhs(i)$ 
                 $cell\_matrix(i, j) = transport\_matrix(i, j) + boundary\_matrix(i, j)$ 
                 $solution(i) = cell\_matrix(i, j)^{-1} cell\_rhs(i)$ 
              for DOF of cell  $i$  do
                 $angular\_solution[m](local\_dof\_indices[i]) = solution(i)$ 

```

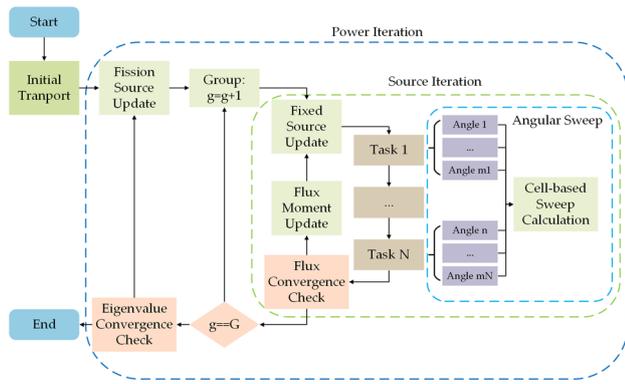


Fig. 2 Flowchart depicting multithreaded parallel strategy

$$\int_{V_k} b^T(r) \Omega_m \nabla b(r) dV = \int_{V_k} [J^{-1}(\hat{r}_q) b^T(\hat{r}_q)] [\Omega_m J^{-1}(\hat{r}_q) \hat{\nabla} b(\hat{r}_q)] \left| \det J(\hat{r}_q) \right| w_q, \tag{16}$$

where q is the index of the quadrature point, \hat{r}_q is the location of the reference cell, and w_q is the Gaussian quadrature weight. In this study, triangular and tetrahedral grids were used for the 2D and 3D cases, respectively.

2.3.2 Multithread parallel strategy

Parallel transport computation has gained popularity in the nuclear field because of the emergence of high-performance computing clusters [24]. How the processors schedule parallel jobs significantly affects the performance of code using parallel computing. The two basic techniques are spatial-domain decomposition (SDD) [25] and angular-domain decomposition (ADD) [26]. SDD techniques subdivide the spatial domain into subdomains, such that the common mesh-sweep method is performed concurrently in numerous subdomains at diverse angles.

However, parallelizing the S_N spatial sweep is more challenging because of the upwind-downwind dependencies between the domain boundaries.

ADD techniques are frequently considered more expandable and require less memory [27]. They entail a mesh sweep along a similar quadrant and are the same for all angles in the chosen quadrature [28]. All operations are statically scheduled to the participating processors without negatively affecting the load balance and are ideally suited for personal computers (PCs). In shared memory machines, the traditional approach to parallelism is to break code down into threads. The deal.II leverages the threading building blocks (TBB) library [29] as basic wrappers to build an object called “tasks”. TBB offers transferable interfaces across many different platforms and abstracts the specifics of run-state signals onto individual threads.

The entire procedure for the multithreaded parallel approach in the ThorSNIFE code is shown in Fig. 2. Both inner and outer iterations contain an angular loop. The number of CPU cores determines the total number of tasks in a PC system. The angular loops were divided into a specific number of subranges that were evenly distributed across the threads. A cell-based sweep sequence is used for each angular loop. A subrange is a task. The tasks are essentially independent during the iterations. They performed the calculations using the same mesh and equation structure as the respective quadrature set. However,

Table 1 Cross section of the IAEA 5-region fixed problem

Region	Σ_t (cm ⁻¹)	Σ_f (cm ⁻¹)	Σ_s (cm ⁻¹)	s (cm ⁻² s ⁻¹)
R1	0.60	0.079	0.53	1.0
R2	0.48	0.0	0.20	0.0
R3	0.70	0.0	0.66	1.0
R4	0.65	0.043	0.50	0.0
R5	0.90	0.0	0.83	0.0

Fig. 3 (Color online) Geometry of the IAEA 5-region fixed problem

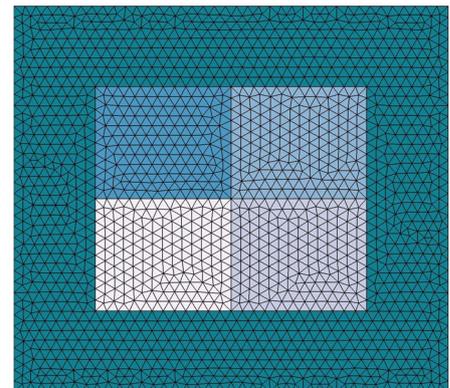
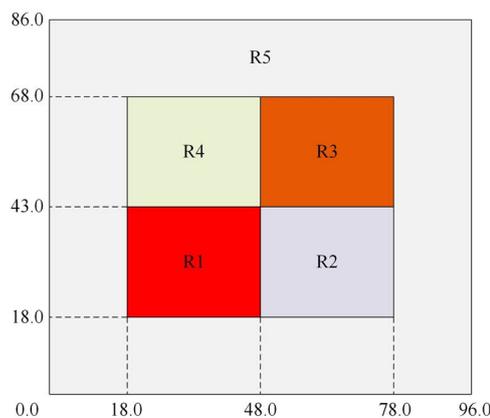


Table 2 Comparison of average flux of the IAEA benchmark

Reference	Average flux (Relative errors) $\text{cm}^{-2} \text{s}^{-1}$			
	Conventional	Downwind	Upwind	
R1	11.941	11.872 (0.58%)	11.874 (0.56%)	11.871 (0.59%)
R2	0.545	0.547 (0.37%)	0.547 (0.37%)	0.546 (0.18%)
R3	19.168	19.045 (0.64%)	19.051 (0.61%)	19.039 (0.67%)
R4	0.843	0.838 (0.60%)	0.840 (0.36%)	0.837 (0.71%)
R5	1.530	1.542 (0.78%)	1.553 (1.50%)	1.531 (0.07%)

Table 3 Comparison of average flux and k_{eff} of the IAEA benchmark

Reference	Average flux (Relative Errors) $\text{cm}^{-2} \text{s}^{-1}$			
	Conventional	Downwind	Upwind	
R1	0.01686	0.016371 (2.90%)	0.01661 (1.18%)	0.016643 (1.29%)
R2	0.000125	0.000131 (4.80%)	0.000128 (2.40%)	0.000128 (2.40%)
R3	0.000041	0.000040 (2.44%)	0.000041 (0.00%)	0.000041 (0.00%)
R4	0.000295	0.000295 (0.00%)	0.000293 (0.68%)	0.000293 (0.68%)
R5	0.000791	0.000799 (1.01%)	0.000788 (0.38%)	0.000789 (0.25%)
k_{eff}	1.0083	1.007076	1.008371	1.008305
pcm		-12	7	1

the storage of angular fluxes is based on the local DOF indices calculated using sweep schemes. The calculation results for different subranges were collected to update the scattering source. Following the execution of the subranges, threads keep themselves busy by stealing complete portions of the subranges from other threads if they have finished their work. Thus, the code can fully exploit the system thread resource, which requires frequent interruptions by the operating system to allow other threads to execute on the available processor cores.

3 Results and discussions

The multithreaded parallel upwind sweep algorithm is validated using three typical benchmark problems. The IAEA benchmark with five different regions was used to test the performances of the different sweep schemes. The Kobayashi-3i benchmark with cavities and bend ducts was adopted to evaluate the performance of the parallel computation for different mesh sizes and angular quadrature sets. The VENUS-3 benchmark was used to validate the algorithm's capabilities for realistic transport calculations. In all

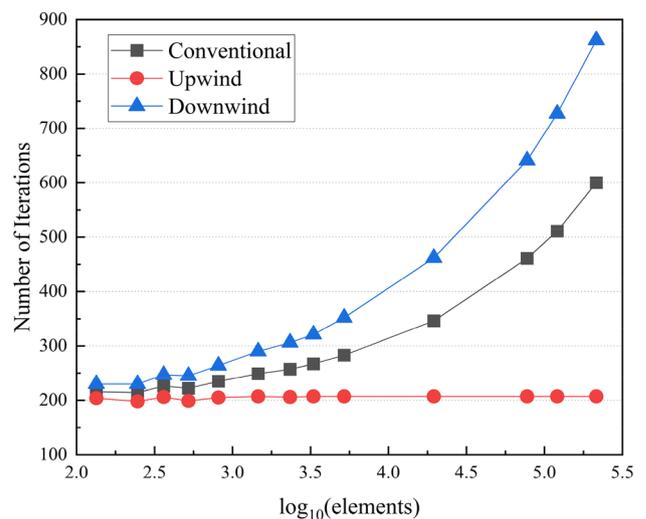


Fig. 4 Number of iterations with the change in the number of elements for different sweep schemes

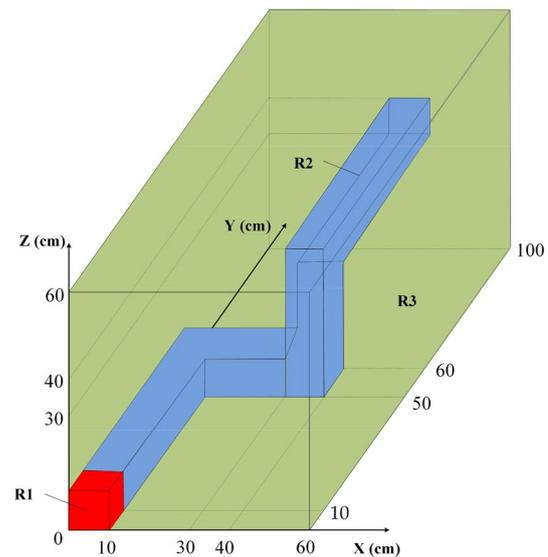


Fig. 5 (Color online) Geometry of the Kobayashi-3i benchmark

benchmarks, convergence criteria are 10^{-5} for the eigenvalue and 10^{-4} for the average flux in each space interval.

3.1 IAEA benchmark

The IAEA benchmark is a 2D five-region calculation problem consisting of two large source zones and two large absorber zones surrounded by light water [30] and is designed for IAEA advanced reactor neutron transport computation. There is a strong nonuniformity in the fuel region,

Table 4 Cross section of the Kobayashi-3i benchmark

Region	\sum_t (cm ⁻¹)	\sum_s (cm ⁻¹)	s (cm ⁻² s ⁻¹)
R1	0.1	0.05	1.0
R2	0.0001	0.00005	0.0
R3	0.1	0.05	0.0

as shown in Fig. 3. The cross sections are given in Table 1, and all boundary conditions are vacuum.

This benchmark problem is mainly used to analyze the effect of the element sweep order on the cell-based algorithm efficiency and assert the superiority of the upwind sweep scheme. Under the $P_N T_N S_8$ angular approximation, comparisons of the numerical results for the fixed source and eigenvalue cases simulated with 2338 elements are presented in Tables 2 and 3, respectively. The relative deviations in the average flux were within 2.50% in the fixed-source case and within 5.00% in the eigenvalue case. All the numerical results show excellent agreement with the references, demonstrating the accuracy and validity of the algorithm and modeling approach for such strongly heterogeneous problems. The change in the element sweep order does not make a large difference to the results. This is because the transport

sweep operations were performed individually on each element, and the results were added. Because matrix addition is commutative and these operators in the equation can be described as applying a sum of matrices, the orders of various components will not impact the final result.

As shown in Fig. 4, the number of iterations was compared with the change in the number of elements in the fixed-source calculation. The numerical results indicated that the upwind sweep scheme had a stable convergence speed, regardless of the number of elements. The relationship between the number of iterations and elements grows exponentially in both types of sweep schemes. The primary functions on the edges are determined based on the upwind edges with regard to the streaming direction, and transport problems involve a directional flow of information. The upwind scheme fully uses the initial boundary conditions and provides spatial flux moments for the neighboring elements. The other sweep schemes increased the number of source and power iterations required to meet the convergence criteria in the subsequent matrix calculations to compensate for the lack of constraints. In addition, no overhead cost of assembling and solving the stiffness matrix in the current framework was observed for the three sweep schemes mentioned above. Therefore, the upwind sweep algorithm can

Fig. 6 a Root-mean-square of Scalar flux L2-error versus the order of quadrature sets for the Kobayashi-3i benchmark; **b** speedup ratio versus the order of quadrature sets for the Kobayashi-3i benchmark

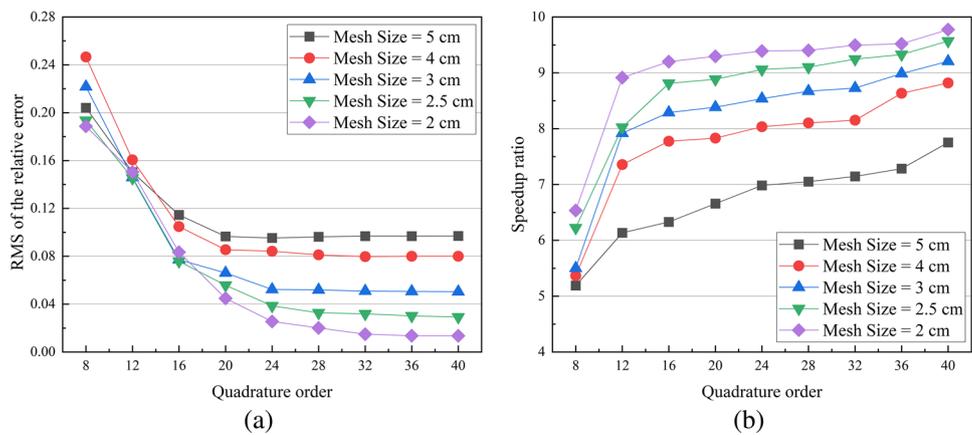


Fig. 7 (Color online) Mesh distribution and scalar flux of the Kobayashi-3i benchmark

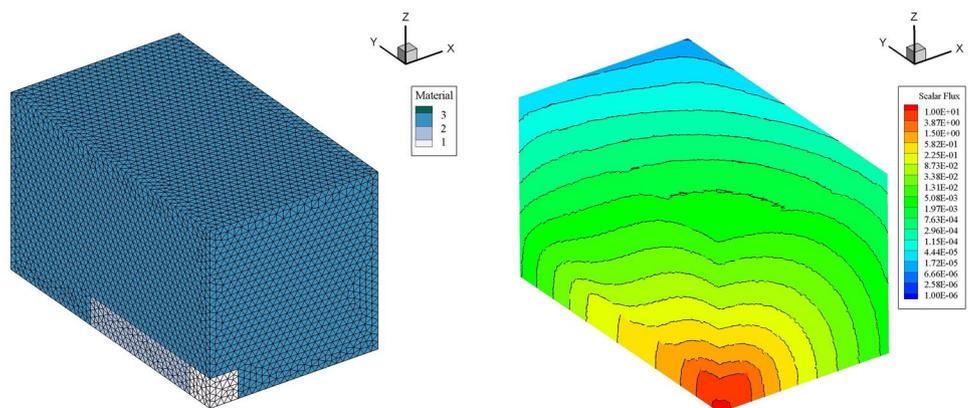


Table 5 Total flux comparison in the Kobayashi-3i benchmark

Case	Co-ordinates	MCNP ($\text{cm}^{-2} \text{s}^{-1}$)	ThorSNIPE ($\text{cm}^{-2} \text{s}^{-1}$)	Relative errors (%)
3A	(5,5,5)	8.61578×10^{-0}	8.61432×10^{-0}	-0.02
	(5,15,5)	2.16130×10^{-0}	2.13602×10^{-0}	-1.17
	(5,25,5)	8.93784×10^{-1}	8.93551×10^{-1}	-0.03
	(5,35,5)	4.78052×10^{-1}	4.72878×10^{-1}	-1.08
	(5,45,5)	2.89424×10^{-1}	2.87933×10^{-1}	-0.52
	(5,55,5)	1.92698×10^{-1}	1.91669×10^{-1}	-0.53
	(5,65,5)	1.04982×10^{-1}	1.04268×10^{-1}	-0.68
	(5,75,5)	3.37544×10^{-2}	3.43534×10^{-2}	1.77
	(5,85,5)	1.08158×10^{-2}	1.08608×10^{-2}	0.42
3B	(5,95,5)	3.39632×10^{-3}	3.47318×10^{-3}	2.26
	(15,55,5)	6.72147×10^{-2}	6.74001×10^{-2}	0.28
	(25,55,5)	2.21799×10^{-2}	2.20850×10^{-2}	-0.43
	(35,55,5)	9.90646×10^{-3}	9.74980×10^{-3}	-1.58
	(45,55,5)	3.39066×10^{-3}	3.43623×10^{-3}	1.34
	(55,55,5)	1.05629×10^{-3}	1.04113×10^{-3}	-1.44
3C	(5,95,5)	3.44804×10^{-4}	3.45115×10^{-4}	0.09
	(15,95,5)	2.91825×10^{-4}	2.98723×10^{-4}	2.36
	(25,95,5)	2.05793×10^{-4}	2.07437×10^{-4}	0.80
	(35,95,5)	2.62086×10^{-4}	2.63330×10^{-4}	0.47
	(45,95,5)	1.05367×10^{-4}	1.07620×10^{-4}	2.14
	(55,95,5)	4.44962×10^{-5}	4.46921×10^{-5}	0.44

save significant computational costs with fewer iterations and is suitable for DGFEM simulations.

3.2 Kobayashi-3i benchmark

The Kobayashi-3i benchmark [31] is a bend duct-type problem with void regions in a highly absorbing medium, as shown in Fig. 5. The cross sections used in this benchmark problem are listed in Table 4. This problem was set up to

analyze the features of spatial and angular discretization and demonstrate the parallel efficiency of the multithreaded strategy.

Angular flux distribution over full domain with mesh sizes of 5 cm, 4 cm, 3 cm, 2.5 cm, and 2 cm are modeled and simulated. The quality of the quadrature sets was quantified by taking the root-mean-square (RMS) of the relative scalar flux of all N point detectors.

Fig. 8 (Color online) Geometry of the Venus-3 benchmark

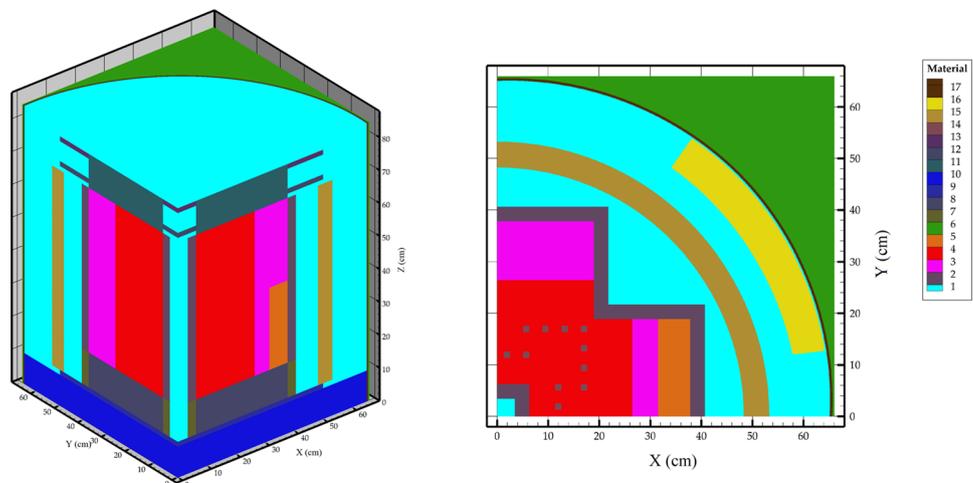
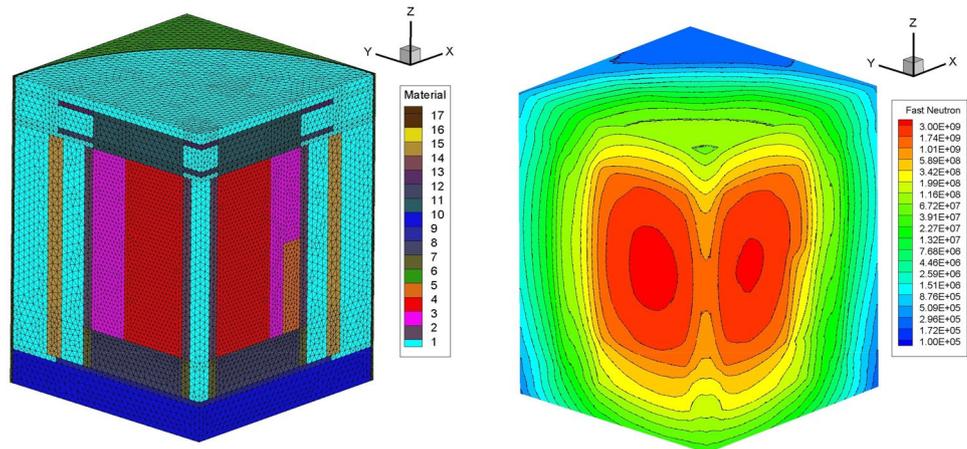


Fig. 9 (Color online) Mesh distribution and fast neutron fluxes of the VENUS-3



$$\text{RMS} = \sqrt{\frac{1}{N} \sum_{i=1}^N \left(\frac{\phi_{\text{cal}} - \phi_{\text{ref}}}{\phi_{\text{ref}}} \right)^2} \quad (17)$$

where the reference solutions ϕ_{ref} were obtained using an analytical method [31]. The RMS of the relative errors in the scalar flux using different quadrature sets versus the number of ordinates is presented in Fig. 6a. It should be noted that the various approximations in the DGFEM with different mesh sizes converged at variable speeds and started to flatten as the number of ordinates increased under S_{20} angular approximations.

The ThorSNIPE code runs on a PC with 8 cores/16 threads (one 11th Gen Intel(R) Core(TM) i7-11700F CPU running at 2.5 GHz) to demonstrate the performance and gain of the multithread technology. The speedup ratio with respect to serial mode versus the order of quadrature sets is presented in Fig. 6b, where the parallel implementation efficiency exceeds 100% when the S_{12} angular approximations are attained, and the mesh size is less than 3 cm. The comparative results show that the proposed method can achieve a better acceleration effect as the number of meshes increases. It can also be observed that the speedup ratio increases more slowly when the S_{20} angular approximations are attained. In this case, each thread has over 30 parallel tasks, and memory access becomes a bottleneck in the matrix solvers for partial differential equations. The numerical results are shown in Fig. 7, and the neutron fluxes at the key points are listed in Table 5. Compared with MCNP [31], the ThorSNIPE code effectively reduces angular discretization errors and fulfills the transport computational requirement to a certain degree.

3.3 VENUS-3 benchmark

The VENUS-3 benchmark was proposed to validate the capabilities of the calculation methodologies and

cross-section libraries for predicting fluence rates in RPVs because it has a very clean structural geometry representing standard PWR pressure vessel conditions [32]. Figure 8 illustrates the geometric configuration and material distribution of the VENUS-3 facility. The VENUS-3 model includes two types of fuel rods: the PLSA, core baffle, water reflector, barrel, Pyrex control rods, barrel, neutron pad, three types of grids, bottom support, reactor vessel, and jacket inner walls. Detailed descriptions and qualifications were obtained from the benchmark document. The benchmark arranges 386 dosimeters, with 244 $^{58}\text{Ni}(n, p)^{58}\text{Co}$ dosimeters, 104 $^{115}\text{In}(n, n')^{115\text{m}}\text{In}$ dosimeters, and 38 $^{27}\text{Al}(n, a)^{24}\text{Na}$ dosimeters placed in 268 different spatial locations. The experimental rate sets are provided as a ratio of the calculated reaction rate to the average dosimeter cross section, named the equivalent fission flux.

The numerical results shown in Fig. 9 were modeled and simulated using the $P_N T_N S_8$ angular discretization, 199 KASHIL-E70 neutron groups multigroup cross sections [33], and 407,819 tetrahedral elements. Figure 10 shows a C/E comparison of the equivalent fission fluxes at the three detector positions. For the $^{115}\text{In}(n, n')^{115\text{m}}\text{In}$ dosimeters, almost all the equivalent fission flux deviations are included within $\pm 10\%$, except for a detector position (No. 31 located in the core barrel with 0.7° angle) where the equivalent fission flux numerical value is affected by reflective boundary. For the $^{27}\text{Al}(n, a)^{24}\text{Na}$ dosimeters, 92% of the equivalent fission flux deviations were within $\pm 10\%$, and the remaining three dosimeters overestimated the corresponding experimental values by approximately 11.9%. For the $^{58}\text{Ni}(n, p)^{58}\text{Co}$ dosimeters, 95% of the equivalent fission flux deviations are limited within $\pm 10\%$, except for 11 detector positions where the equivalent fission flux numerical values overestimated the corresponding experimental values by about 10.1%–20.7%. They are concentrated at the junction of the PLSA region, outer baffle, and core barrel. The scalar flux distribution is not smooth because of the strong spatial

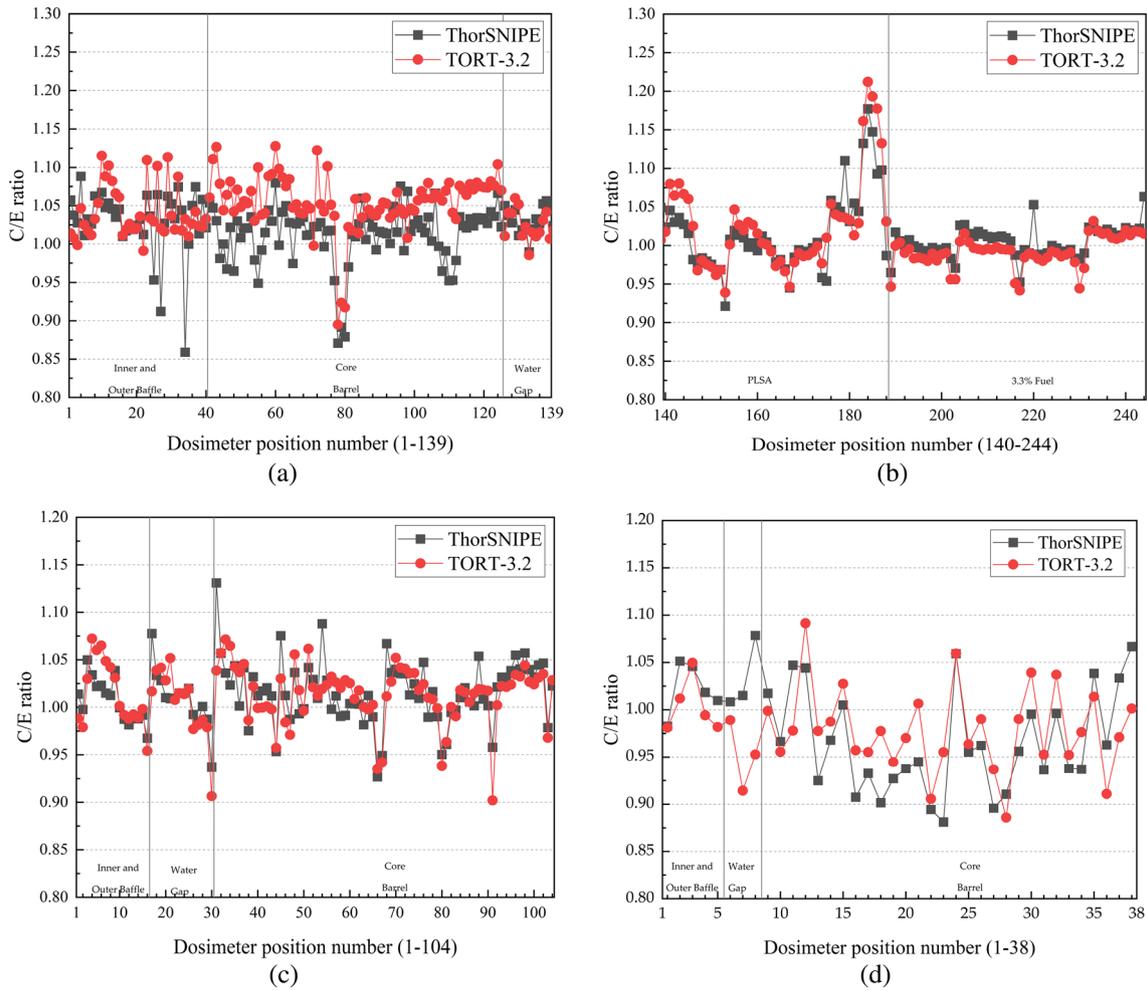


Fig. 10 C/E comparison of equivalent fission fluxes at detector positions: **a** $^{58}\text{Ni}(n,p)^{58}\text{Co}$ from No. 1 to 139; **b** $^{58}\text{Ni}(n,p)^{58}\text{Co}$ from No. 140 to 244; **c** $^{115}\text{In}(n,n')^{115m}\text{In}$; **d** $^{27}\text{Al}(n,\alpha)^{24}\text{Na}$

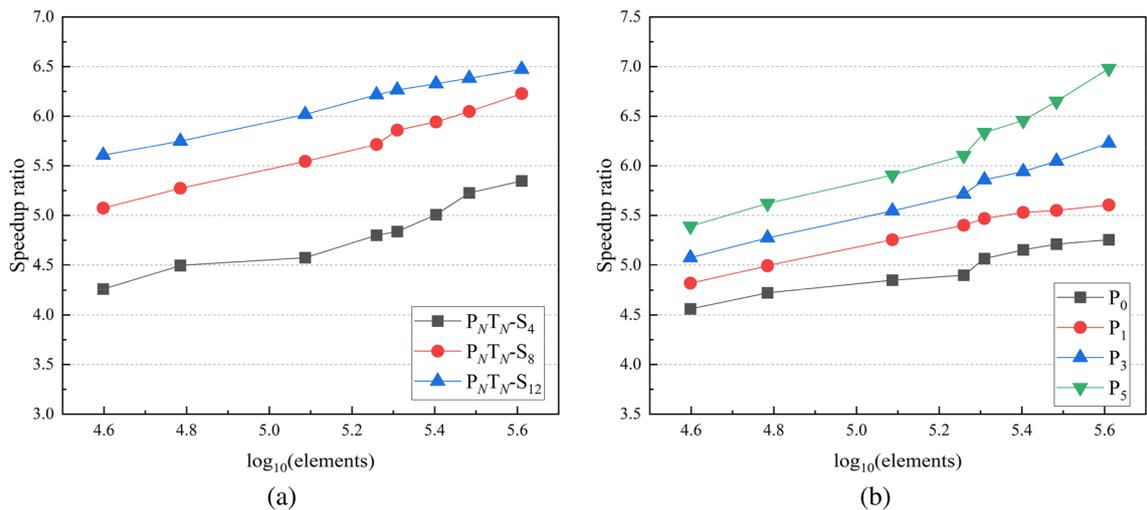


Fig. 11 Speedup ratio with the change in the **a** order of quadrature sets; **b** order of Legendre expansion

nonuniformity of the geometry model, and higher requirements are proposed for local mesh refinement, which significantly degrades the simulated accuracy of these regions. The overall agreement between the numerical results from the ThorSNIPE code and the experimental results was very good, which shows that the ThorSNIPE code is highly reliable and stable for complex shielding calculations.

The numerical results [32] from TORT-3.2 with 1,004,295 hexahedral grids and 26 BUGLE-96 neutron energy groups were selected as comparative references. For the uniformly distributed 3.3% fuel regions, TORT-3.2 with hexahedral structured meshes achieved better results. The ThorSNIPE code with tetrahedral unstructured meshes can decrease the numerical oscillation for complex regions. For the $^{27}\text{Al}(n, a)^{24}\text{Na}$ dosimeters, the numerical results of ThorSNIPE tended to be lower than those of TORT-3.2 in the core barrel. This underestimation might have been caused by an error in the average dosimeter cross section.

In Fig. 11, the speedup ratios of the three quadrature sets with the P_3 Legendre expansion order and the four Legendre expansion orders with $P_N T_N S_8$ angular discretization are shown. The test conditions were consistent with TORT-3.2 code. Overall, the behavior of the proposed algorithm was similar to that of the Kobayashi-3i benchmark. For $P_N T_N S_{12}$, the acceleration ratio was 6.47 with 407,819 elements. Comparing the results of different expansion orders, the P_5 expansion order has the steepest upward trend, and the maximum speedup ratio achieved was 6.98. The numerical results indicate that the proposed algorithm achieves a favorable speedup ratio as the scale of the mesh increases, as expected.

4 Conclusion

In this study, a multithreaded parallel upwind algorithm to solve the first-order Boltzmann neutron equation is proposed and implemented in the multigroup 2D/3D S_N transport code ThorSNIPE. The cell-based upwind sweep algorithm realizes angular flux transmission between neighboring meshes and achieves a stable solution by applying DGFEM spatial discretization. The multithread parallel strategy automatically manages the angular sweep stack of each thread and improves the quality and efficiency of the ThorSNIPE code.

The performance and accuracy of the proposed algorithm were tested using several typical benchmark problems. In the IAEA benchmark, the upwind sweep algorithm was proven to be the optimal method for DGFEM calculations and can save significant computational costs. The results show that the ThorSNIPE code agrees well with the reference, with a deviation of 1 pcm in eigenvalue and within 2.50% for flux distribution. In the Kobayashi-3i benchmark, the results for angular flux simulations over the full domain in different

mesh sizes and quadrature set orders indicate that the multithreaded parallel algorithm has potential extensibility. In the VENUS-3 benchmark, 96% of the equivalent fission flux deviations are limited within $\pm 10\%$, except for 15 detector positions where the equivalent fission flux numerical values overestimated the corresponding experimental values by about 10.1%–20.7%. The maximum speedup ratio was 6.98 with 407,819 elements. The results of the VENUS-3 benchmark show that this robust algorithm can play a reliable role in practical engineering applications and improve computational efficiency.

In the future, angular and algebraic multigrid methods planned to solve discretized equations to overcome the reduced convergence speed of the iteration method will be investigated. Furthermore, we plan to address more complex problems to assess the accuracy and effectiveness of the proposed method and present a rigorous mathematical convergence analysis of the new scheme.

Author contributions All authors contributed to the study conception and design. Material preparation, data collection, and analysis were performed by Zhi-Wei Zong and Mao-Song Cheng. The first draft of the manuscript was written by Zhi-Wei Zong and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

Data availability The data that support the findings of this study are openly available in Science Data Bank at <https://doi.org/10.57760/sciencedb.13607> and <https://cstr.cn/31253.11.sciencedb.13607>.

Declarations

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

References

1. E.E. Lewis, W.F. Miller, *Computational Methods of Neutron Transport* (American Nuclear Society, Illinois, 1981)
2. W.A. Rhoades, W.W. Engle Jr., *New Weighted-Difference Formulation for Discrete-Ordinates Calculations*. Tech. Rep., Oak Ridge National Laboratory, CONF-771109-53 (1977)
3. Y. Chen, B. Zhang, L. Zhang et al., ARES: a parallel discrete ordinates transport code for radiation shielding applications and reactor physics analysis. *Sci. Tech. Nucl. Install.* **2017**, 2596727 (2017). <https://doi.org/10.1155/2017/2596727>
4. T. Cheng, L. Wei, B. Zhong et al., Reconstruction and parallelization of 3D S_N program for neutron/photon transport. *Nucl. Power Eng.* **35**, 147–150 (2014). <https://doi.org/10.13832/j.jnpe.2014.S2.0147>
5. M.A. Puso, J. Solberg, A stabilized nodally integrated tetrahedral. *Int. J. Numer. Methods Eng.* **67**, 841–867 (2006). <https://doi.org/10.1002/nme.1651>
6. Y. Wang, J.C. Ragusa, Standard and goal-oriented adaptive mesh refinement applied to radiation transport on 2D unstructured triangular meshes. *J. Comput. Phys.* **230**, 763–788 (2011). <https://doi.org/10.1016/j.jcp.2010.10.018>

7. N. Dai, B. Zhang, Y. Chen et al., Adaptive discontinuous finite element quadrature sets over an icosahedron for discrete ordinates method. *Nucl. Sci. Tech.* **32**, 98 (2021). <https://doi.org/10.1007/s41365-021-00934-7V>
8. V. Labouré, Y. Wang, J. Ortensi et al., Multischeme equivalence procedure for neutron transport finite element methods. *Ann. Nucl. Energ.* **166**, 108712 (2022). <https://doi.org/10.1016/j.anucene.2021.108712>
9. B.Q. Li, *Discontinuous Finite Elements in Fluid Dynamics and Heat Transfer* (Springer, London, 2006)
10. D. Kuzmin, *A Guide to Numerical Methods for Transport Equations* (Thesis, University of Erlangen-Nuremberg, 2010)
11. H. Guo, W. Chen, X. Jiang et al., A new continuous finite element S_N method for solving first-order neutron transport equation. *Ann. Nucl. Energ.* **175**, 109237 (2022). <https://doi.org/10.1016/j.anucene.2022.109237>
12. C.C. Pain, M.D. Eaton, R.P. Smedley-Stevenson et al., Streamline upwind Petrov–Galerkin methods for the steady-state Boltzmann transport equation. *Comput. Methods Appl. Mech. Eng.* **195**, 4448–4472 (2006). <https://doi.org/10.1016/j.cma.2005.09.004>
13. T.A. Manteuffel, K.J. Ressel, Least-squares finite-element solution of the neutron transport equation in diffusive regimes. *SIAM J. Numer. Anal.* **35**, 806–835 (1998). <https://doi.org/10.1137/S003614299629970>
14. W.H. Reed, T.R. Hill, *Triangular Mesh Methods for the Neutron Transport Equation*. Tech. Rep., Los Alamos National Laboratory, LA-UR-73-479 (1973)
15. O.C. Zienkiewicz, R.L. Taylor, J.Z. Zhu, *The Finite Element Method: Its Basis and Fundamentals*, 6th edn. (Elsevier, Butterworth Heinemann, Oxford, 2005), pp.14–17
16. Y. Wang, S. Schunert, J. Ortensi et al., Rattlesnake: a MOOSE-based multiphysics multischeme radiation transport application. *Nucl. Technol.* **207**, 1047–1072 (2021). <https://doi.org/10.1080/00295450.2020.1843348>
17. S. Plimpton, B. Hendrickson, S. Burns et al., Parallel algorithms for radiation transport on unstructured grids. in *Proceedings of the 2000 ACM/IEEE Conference* (Dallas, Texas, USA, 2000). <https://doi.org/10.1109/SC.2000.10030>
18. S.D. Pautz, An algorithm for parallel SN sweeps on unstructured meshes. *Nucl. Sci. Eng.* **140**, 11–136 (2002)
19. J.Y. Moller, J.J. Lautard, Minaret, a deterministic neutron transport solver for nuclear core calculations. in *International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering* (Rio de Janeiro, Brazil, 2011)
20. W.K. Jong, O.L. Young, A deep penetration problem calculation using AETIUS: an easy modeling discrete ordinates transport code using unstructured tetrahedral mesh, shared memory parallel. *EPJ Web Conf.* **153**, 06025 (2017). <https://doi.org/10.1051/epjconf/201715306025>
21. Z. Zong, M. Cheng, A two-dimensional finite element shielding calculation code with mass-matrix lumping technique and unstructured meshes. *Nucl. Tech.* **46**, 020602 (2023). <https://doi.org/10.11889/j.0253-3219.2023.hjs.46.020602>. (in Chinese)
22. D. Arndt, W. Bangerth, M. Feder et al., The deal. II library, version 9.4. *J. Numer. Math.* **30**, 231–246 (2022). <https://doi.org/10.1515/jnma-2022-0054>
23. V. Labouré, R.G. McClarren, Y. Wang, Globally conservative, hybrid self-adjoint angular flux and least-squares method compatible with voids. *Nucl. Sci. Eng.* **185**, 294–306 (2017). <https://doi.org/10.1080/00295639.2016.1272374>
24. Y.Y. Azmy, Performance and performance modeling of a parallel algorithm for solving the neutron transport equation. *J. Supercomput.* **6**, 211–235 (1992). <https://doi.org/10.1007/BF00155800>
25. G. Zhang, J. Liu, L. Cao et al., Neutronic calculations of the China dual-functional lithium–lead test blanket module with the parallel discrete ordinates code hydra. *Nucl. Sci. Tech.* **31**, 74 (2020). <https://doi.org/10.1007/s41365-020-00789-4>
26. Z. Hong, G. Yuan, A parallel algorithm with interface prediction and correction for spherical geometric transport equation. *Prog. Nucl. Energ.* **51**, 268–273 (2009). <https://doi.org/10.1016/j.pnucene.2008.09.003>
27. J.W. Fisher, Y.Y. Azmy, Comparison via parallel performance models of angular and spatial domain decompositions for solving neutral particle transport problems. *Prog. Nucl. Energ.* **47**, 37–60 (2007). <https://doi.org/10.1016/j.pnucene.2006.08.003>
28. Y.Y. Azmy, On the adequacy of message-passing parallel supercomputers for solving neutron transport problems. in *Proceedings of the 1991 ACM/IEEE Conference on Supercomputing* (New York, USA, 12–16 November 1991). <https://doi.org/10.1109/SUPERC.1990.130088>
29. V. Alessandrini, *Shared Memory Application Programming Concepts and Strategies in Multicore Application Programming* (Morgan Kaufmann, Boston, 2015), pp.307–339
30. J. Stepanek, T. Auerbach, W. Hälgl, *Calculation of Four Thermal Reactor Benchmark Problems in X-Y Geometry*. Tech. Rep. ANS, USA, EIR-464 (1982)
31. K. Kobayashi, N. Sugimura, Y. Nagaya, 3D radiation transport benchmark problems and results for simple geometries with void region. *Prog. Nucl. Energ.* **39**, 119–144 (2001). [https://doi.org/10.1016/S0149-1970\(01\)00007-5](https://doi.org/10.1016/S0149-1970(01)00007-5)
32. M. Pescarini, R. Orisi, M. G. Borgia et al., *ENEA Nuclear Data Center Neutron Transport Analysis of the VENUS-3 Shielding Benchmark Experiment*. Tech. Rep. ENEA-Bologna, Italy, KT-SCG-00013 (2001)
33. D.H. Kim, C.S. Gil, Y.O. Lee, Validation of an ENDF/B-VII.0-based neutron and photon shielding library in MATXS-format. *J. Korean Phys. Soc.* **59**, 1199–1202 (2011). <https://doi.org/10.3938/jkps.59.1199>

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