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

Authors: Zong, Zhiwei, Cheng, Maosong, Yu, Yingchi, Dai, Zhimin, Cheng, Maosong

Date: 2023-11-22T00:00:00+00:00

Abstract

The complex structure and strong heterogeneity of advanced nuclear reactor systems pose significant challenges for high-fidelity neutron shielding calculations. Unstructured meshes exhibit strong geometric adaptability and can overcome the deficiencies of conventional structured meshes in modeling complex geometries. To achieve more accurate geometric description and improve computational efficiency, we propose a multithreaded parallel upwind sweep algorithm for S_N transport calculations.

Full Text

Preamble

A Multithreaded Parallel Upwind Sweep Algorithm for the S_N Transport Equations Discretized with Discontinuous Finite Elements

Zhiwei Zong[1], Maosong Cheng[1,2,*], Yingchi Yu[1,2], Zhimin Dai[1,2]

[1]Shanghai Institute of Applied Physics, Chinese Academy of Sciences, Shanghai 201800, China
[2]University of Chinese Academy of Sciences, Beijing 100049, China

*Corresponding author. E-mail address: chengmaosong@sinap.ac.cn

Abstract: The complex structure and strong heterogeneity of advanced nuclear reactor systems pose significant challenges for high-fidelity neutron shielding calculations. Unstructured meshes exhibit strong geometric adaptability and can overcome the deficiencies of conventional structured meshes in modeling complex geometries. To achieve more accurate geometric description and improve

computational efficiency, we propose a multithreaded parallel upwind sweep algorithm for S_N transport calculations.

The spatial variables were discretized using the standard discontinuous Galerkin finite element method (DGFEM). Angular flux transmission between neighboring meshes was handled using an upwind scheme, and a combination of mesh transport sweeps and angular iterations was realized through multithreaded parallel techniques. 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 results demonstrate that the multithreaded parallel upwind sweep algorithm can achieve high computational efficiency. ThorSNIPE, equipped with this algorithm, exhibits good reliability, stability, and efficiency, making it suitable for complex shielding calculations.

Key words: Shielding calculation, Discrete ordinates method, Discontinuous Galerkin finite element method, Unstructured meshes

1 Introduction

High-fidelity neutron shielding calculations are essential for advanced nuclear reactor system design. The discrete ordinates (S_N) method is one of the most popular deterministic particle transport methods due to its high angular resolution [**?**]. Reliable modeling and simulation of S_N transport depend on two key components: reasonable spatial discretization schemes and mesh distributions with sufficient modeling fidelity. These two factors interact with and support each other throughout the computational process.

Conventional Cartesian structured grids are widely used because of their regular arrangement and compatibility with finite-difference (FD) spatial discretization schemes [**?**]. However, no general robust meshing algorithm exists for acquiring exact descriptions of complex structures in realistic modeling and simulations [**?**]. Adaptive mesh refinement techniques retain the original mesh distribution features and mark cells with large errors for refinement [**?**], where local error estimation techniques and geometric data structures play significant roles in determining system stability and computational efficiency. Unstructured grids use piecewise flat or curved surfaces to provide maximum flexibility for complex geometries. With the development of the finite element method (FEM), unstructured grids have been widely applied to reactor physics analyses [**?**], multiphysics simulations [**?**], and thermal-fluid dynamics [**?**]. Combining finite element numerical methods with the S_N method on unstructured meshes has become a trend in nuclear shielding design.

Unfortunately, the first-order S_N equation cannot be solved stably using continuous FEM [**?**]. While solutions to this hyperbolic equation are smooth along streamlines, they may be discontinuous perpendicular to the chosen direction. To obtain robust solutions, several Petrov-Galerkin (PG) approaches update the test function as the sum of trial functions and introduce additional stabilization

terms [**?**]. However, the free stabilization parameter affects the convergence of the S_N equation and introduces uncertainty, as it is usually determined empirically. Additionally, PG methods cannot generally produce symmetric positive-definite coefficient matrices or inherit the best approximation property when transport equations contain nonself-adjoint operators.

The discontinuous Galerkin finite element method (DGFEM) represents a different category of stability improvement, pioneered by Reed and Hill for neutron transport problems in the early 1970s [**?**]. DGFEM connects adjacent elements through incoming interface fluxes from upwind (upstream) meshes and applies the standard FEM procedure to individual elements. However, DGFEM involves many more unknowns than continuous FEM, demanding efficient matrix assembly and solution methods. The standard matrix assembly procedure requires a reliable iterative solver for banded sparse systems and must handle the increased storage requirements [**?**]. The matrix-free strategy stores diagonal blocks and realizes inversion of the transport operator through matrix-vector products, though it requires sum factorization techniques to enable translation between vector entries and values or gradients at quadrature points [**?**]. Another commonly used technique is the conventional sweeping procedure, such as in FD methods, where parallelization of the sweeping procedure reduces computing time, which becomes important at large S_N orders.

Regarding parallelization, complex grid connectivity relationships and irregular element sweep orders pose algorithmic challenges. In 2000, Plimpton described an asynchronous, parallel message-passing algorithm that simultaneously performs sweeps from multiple directions across unstructured grids [**?**]. In 2002, Pautz used a low-complexity list-ordering heuristic to determine sweep ordering on partitioned meshes [**?**]. The essence of these algorithms is converting the abstract element sweep order into a directed acyclic graph (DAG). Since 2010, several research institutions have conducted extensive studies on high-performance computing applications [**?**].

A multigroup 2D S_N transport code, ThorSNIPE, with unstructured meshes was developed and validated [**?**]. This code was extended to 3D S_N transport calculations and employs a multithreaded parallel upwind sweep algorithm to achieve more accurate geometric description and improve the reliability and efficiency of transport calculations in complex geometries. The finite element method was implemented using the open-source finite element library deal.II-9.4.0 [**?**]. Deal.II is an open-source C++ program library for the computational solution of partial differential equations using adaptive finite elements, focusing on generality, dimension-independent programming, parallelism, and extensibility. It includes many state-of-the-art programming techniques for solving PDEs, linear algebra problems, and computer science strategies, offering users a modern interface for complex data structures and algorithms. With this library, ThorSNIPE can be made dimension-independent, supporting different mesh sizes and finite element types.

The remainder of this paper is organized as follows. Section 2 describes the

main theory and methodology of the ThorSNIPE code in detail, including the S_N weak-form equation based on DGFEM theory and the multithreaded parallel upwind sweep algorithm. Numerical results are presented and discussed in Section 3. Conclusions are summarized in Section 4.

2.1 Weak Form of the Discrete Ordinates Method Equations

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

$$\Omega_m \cdot \nabla \psi_g^m(\mathbf{r}) + \Sigma_t^g(\mathbf{r})\psi_g^m(\mathbf{r}) = Q_g^m(\mathbf{r}), \quad g = 1, \dots, G; \quad m = 1, \dots, M; \quad \mathbf{r} \in D$$

where D is the domain, Σ_t^g is the macroscopic total cross-section in group g, $\psi_g^m(\mathbf{r})$ is the angular flux in group g, direction m, and position $\mathbf{r}$, and $Q_g^m(\mathbf{r})$ is a source term in group g, direction m, and position $\mathbf{r}$.

For fixed source problems, the source term can be expressed as:

$$Q_g^m(\mathbf{r}) = q_g^m(\mathbf{r}) + \sum_{g'=1}^{G}\sum_{n=0}^{N}\frac{2n+1}{4\pi}\sum_{k=-n}^{n}\Sigma_{s,n}^{g'\to g}(\mathbf{r})Y_{nk}^m(\Omega_m)\phi_{nk}^{g'}(\mathbf{r})$$

where $q_g^m(\mathbf{r})$ is the fixed source in group g, direction m, and position $\mathbf{r}$, $\Sigma_{s,n}^{g'\to g}$ is the nth-order Legendre moment of scattering cross-section from group g' to group g at position $\mathbf{r}$, $Y_{nk}^m(\Omega_m)$ is the nth-order spherical harmonic in direction Ω_m, and $\phi_{nk}^{g'}(\mathbf{r})$ is the nth-order flux moment at position $\mathbf{r}$.

For eigenvalue problems, the source term can be expressed as:

$$Q_g^m(\mathbf{r}) = \sum_{g'=1}^{G}\sum_{n=0}^{N}\frac{2n+1}{4\pi}\sum_{k=-n}^{n}\Sigma_{s,n}^{g'\to g}(\mathbf{r})Y_{nk}^m(\Omega_m)\phi_{nk}^{g'}(\mathbf{r}) + \frac{\chi_g}{4\pi k_{\text{eff}}}\sum_{g'=1}^{G}\nu\Sigma_f^{g'}(\mathbf{r})\phi_{g'}(\mathbf{r})$$

where χ_g is the average number of neutrons produced per fission in group g, $\Sigma_f^{g'}$ is the macroscopic fission cross-section in group g', k_{eff} is the eigenvalue, and $\phi_{g'}(\mathbf{r})$ is the scalar flux in group g' at position $\mathbf{r}$.

The scalar flux is defined as:

$$\phi_g(\mathbf{r}) = \sum_{m=1}^{M}\psi_g^m(\mathbf{r})\omega_m$$

where ω_m is the weight factor of quadrature set.

Equation (1) is a linear hyperbolic equation. To solve it, we must transform this "strong form" differential-integral equation into "weak form." The spatial domain is decomposed into unstructured elements because FEM is independent of space dimension and grid choice. Consider the monoenergetic form where Eq. (1) is multiplied by an arbitrary test function ψ^* and integrated over any element k:

$$\int_{V_k} \psi^*(\mathbf{r})\Omega_m \cdot \nabla\psi^m(\mathbf{r})\, dV + \int_{V_k} \psi^*(\mathbf{r})\Sigma_t\psi^m(\mathbf{r})\, dV = \int_{V_k} \psi^*(\mathbf{r})Q^m(\mathbf{r})\, dV$$

where V_k is the volume of the kth element. Applying the divergence theorem to Eq. (5) yields:

$$\int_{\partial V_k} \psi^*(\mathbf{r})\psi^m(\mathbf{r})\Omega_m\cdot\mathbf{n}\, dS - \int_{V_k} \nabla\psi^*(\mathbf{r})\cdot\Omega_m\psi^m(\mathbf{r})\, dV + \int_{V_k} \psi^*(\mathbf{r})\Sigma_t\psi^m(\mathbf{r})\, dV = \int_{V_k} \psi^*(\mathbf{r})Q^m(\mathbf{r})\, dV$$

where $\mathbf{n}$ is the unit normal vector to the surface of cell k. Inner products on the volume and surface of any element are introduced to express the bilinear form [**?**]:

$$(\psi^*, \psi^m)_{V_k} = \int_{V_k} \psi^*(\mathbf{r})\psi^m(\mathbf{r})\, dV$$

$$(\psi^*, \psi^m)_{\partial V_k} = \int_{\partial V_k} \psi^*(\mathbf{r})\psi^m(\mathbf{r})\Omega_m \cdot \mathbf{n}\, dS$$

Substituting Eq. (7) and (8) into (6) gives:

$$(\psi^*, \psi^m)_{\partial V_k} - (\nabla\psi^*\cdot\Omega_m, \psi^m)_{V_k} + (\psi^*, \Sigma_t\psi^m)_{V_k} = (\psi^*, Q^m)_{V_k}$$

The general boundary can be divided into three types based on its relative position: interior boundary, outflow domain boundary, and inflow domain boundary. The inflow-domain boundary conditions were prescribed as either vacuum or reflective for incoming directions. Therefore, Eq. (9) can be rewritten as:

$$(\psi^*, \psi^m)_{\partial V_k^+} + (\psi^*, \psi^m)_{\partial V_k^-} + (\psi^*, \psi^m)_{\partial V_k^{\text{in}}} - (\nabla\psi^*\cdot\Omega_m, \psi^m)_{V_k} + (\psi^*, \Sigma_t\psi^m)_{V_k} = (\psi^*, Q^m)_{V_k}$$

where ∂V_k^+, ∂V_k^-, and ∂V_k^{in} represent the outflow angular, inflow angular, and interior face fluxes, respectively. Eq. (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. We seek a numerical approximation of the angular flux term using Lagrange polynomials of degree P:

$$\psi_g^m(\mathbf{r}) \approx \sum_{j=1}^{J} b_j^m \varphi_j(\mathbf{r})$$

where $\varphi_j(\mathbf{r})$ is the Lagrange shape function and b_j^m is an unsolved coefficient. The matrix form is convenient in row vector notation, and Eq. (11) can be written as:

$$\psi_g^m(\mathbf{r}) = \mathbf{b}_g^{mT} \varphi(\mathbf{r}) = \varphi^T(\mathbf{r}) \mathbf{b}_g^m$$

where $\mathbf{b}_g^m = [b_1, b_2, \dots, b_J]^T$ and $\varphi(\mathbf{r}) = [\varphi_1, \varphi_2, \dots, \varphi_J]^T$.

In Galerkin discretization schemes, test functions share the same function space as basis functions:

$$\psi^*(\mathbf{r}) = \sum_{i=1}^{J} c_i \varphi_i(\mathbf{r}) = \mathbf{c}^T \varphi(\mathbf{r})$$

Substituting Eq. (12) and (13) into (10) yields:

$$\mathbf{c}^T \left[(\varphi, \varphi^T)_{\partial V_k^+} + (\varphi, \varphi^T)_{\partial V_k^-} + (\varphi, \varphi^T)_{\partial V_k^{\text{in}}} - (\nabla \varphi \cdot \Omega_m, \varphi^T)_{V_k} + (\varphi, \Sigma_t \varphi^T)_{V_k} \right] \mathbf{b}_g^m = \mathbf{c}^T (\varphi, Q_g^m)_{V_k}$$

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

2.3.1 Cell-Based Sweep Algorithm

In the S_N method, a transport sweep is performed for all elements along each discrete direction. The ordering in which elements and degrees of freedom (DOF) are traversed affects convergence speed. As mentioned in Section 1, DGFEM enables the use of 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 ThorSNIPE code was developed using an upwind scheme based on the deal.II library. In our implementation, an array containing all relevant cells is created and sorted in the upwind direction using a `dealii::DoFRenumbering::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 [Figure 1: see original paper]. Cells in a conventional scheme are often categorized from edge to center to facilitate creation of a global matrix. The downwind system performs renumbering 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 incoming angular fluxes in Eq. (14). Primal functions obtained from neighboring elements require numerical values with 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 received data. This process can be laborious and error-prone, and accumulated errors inevitably affect the robustness of the numerical method. Thus, the entire solution procedure requires more iterations.

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: First, the transport matrix makes contributions from 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 comprise the contributions of 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.

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

```
Input: for upwind sorted cell do
    for DOF of cell
        for DOF of cell
            for moment
                for moment
```

```
for DOF of cell
    for faces on cell do
        if outflow face then
            for DOF of face
                for DOF of face
        else if inflow face then
            for DOF of face
                for DOF of cell
```

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 into a unit/reference cell. For example, the streaming operator is transformed as:

$$\int_{V_k} \varphi_i(\mathbf{r})\Omega_m \cdot \nabla\varphi_j(\mathbf{r})\, dV = \int_{\hat{V}k} \varphi_i(\hat{\mathbf{r}})\Omega_m \cdot \nabla\varphi_j(\hat{\mathbf{r}}) \det(J)\, d\hat{V}$$

where the hat indicates reference coordinates and J is the Jacobian of the mapping $\mathbf{r} = \mathbf{F}_k(\hat{\mathbf{r}})$. Second, the integral is approximated using Gauss-Legendre quadrature, yielding:

$$\int_{\hat{V}k} \varphi_i(\hat{\mathbf{r}})\Omega_m \cdot \nabla\varphi_j(\hat{\mathbf{r}}) \det(J)\, d\hat{V} \approx \sum_{q=1}^{Q} \varphi_i(\hat{\mathbf{r}}_q)\Omega_m \cdot \nabla\varphi_j(\hat{\mathbf{r}}_q) \det(J(\hat{\mathbf{r}}_q))w_q$$

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

2.3.2 Multithread Parallel Strategy

Parallel transport computation has gained popularity in the nuclear field with the emergence of high-performance computing clusters [**?**]. Processor scheduling of parallel jobs significantly affects code performance. The two basic techniques are spatial-domain decomposition (SDD) [**?**] and angular-domain decomposition (ADD) [**?**]. SDD techniques subdivide the spatial domain into subdomains, enabling concurrent mesh-sweep operations in numerous subdomains at diverse angles. However, parallelizing the S_N spatial sweep is more challenging due to upwind-downwind dependencies between domain boundaries.

ADD techniques are frequently considered more scalable and require less memory [**?**]. They entail mesh sweeps along similar quadrants and are identical for all angles in the chosen quadrature set [**?**]. All operations are statically scheduled to participating processors without negatively affecting load balance, making

them ideally suited for personal computers (PCs). In shared memory machines, the traditional approach to parallelism is to break code into threads. Deal.II leverages the Threading Building Blocks (TBB) library [**?**] as basic wrappers to build objects called "tasks." TBB offers portable interfaces across many platforms and abstracts the specifics of run-state signals onto individual threads.

The entire procedure for the multithreaded parallel approach in the ThorSNIPE code is shown in Fig. 2 [Figure 2: see original paper]. 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 specific subranges evenly distributed across threads. A cell-based sweep sequence is used for each angular loop, where each subrange constitutes a task. The tasks are essentially independent during iterations, performing calculations using the same mesh and equation structure as the respective quadrature set. However, angular flux storage is based on local DOF indices calculated using sweep schemes. Calculation results for different subranges are collected to update the scattering source.

Following execution of the subranges, threads keep themselves busy by stealing complete portions of subranges from other threads if they have finished their work. Thus, the code can fully exploit system thread resources, which requires frequent interruptions by the operating system to allow other threads to execute on available processor cores.

3 Results and Discussions

The multithreaded parallel upwind sweep algorithm was validated using three typical benchmark problems. The IAEA benchmark with five different regions tested the performance of different sweep schemes. The Kobayashi-3i benchmark with cavities and bend ducts evaluated parallel computation performance for different mesh sizes and angular quadrature sets. The VENUS-3 benchmark validated the algorithm's capabilities for realistic transport calculations. In all benchmarks, convergence criteria were 10^{-5} for eigenvalue and 10^{-4} for 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 [**?**], designed for IAEA advanced reactor neutron transport computation. There is strong nonuniformity in the fuel region, as shown in Fig. 3 [Figure 3: see original paper]. Cross-sections are given in Table 1 , and all boundary conditions are vacuum.

This benchmark problem primarily analyzes the effect of element sweep order on cell-based algorithm efficiency and demonstrates the superiority of the upwind sweep scheme. Under the PNTN-S8 angular approximation, comparisons

of numerical results for fixed source and eigenvalue cases simulated with 2338 elements are presented in Tables 2 and 3, respectively. Relative deviations in average flux were within 2.50% for the fixed-source case and within 5.00% for the eigenvalue case. All numerical results show excellent agreement with references, demonstrating the accuracy and validity of the algorithm and modeling approach for such strongly heterogeneous problems. The change in element sweep order does not significantly affect results because transport sweep operations were performed individually on each element and results were added. Since matrix addition is commutative and operators in the equation can be described as applying a sum of matrices, the orders of various components do not impact the final result.

As shown in Fig. 4 [Figure 4: see original paper], the number of iterations was compared with changes in element number for fixed-source calculations. Numerical results indicated that the upwind sweep scheme had stable convergence speed regardless of element count. The relationship between iteration count and element number grows exponentially in both sweep schemes. Primary functions on edges are determined based on upwind edges with regard to streaming direction, and transport problems involve directional information flow. The upwind scheme fully utilizes initial boundary conditions and provides spatial flux moments for neighboring elements. Other sweep schemes increased source and power iterations required to meet convergence criteria in subsequent matrix calculations to compensate for lack of constraints. Additionally, no overhead cost for assembling and solving the stiffness matrix was observed for the three sweep schemes. Therefore, the upwind sweep algorithm can save significant computational costs with fewer iterations and is suitable for DGFEM simulations.

3.2 Kobayashi-3i Benchmark

The Kobayashi-3i benchmark [**?**] is a bend duct-type problem with void regions in a highly absorbing medium, as shown in Fig. 5 [Figure 5: see original paper]. Cross sections used in this benchmark are listed in Table 4 . This problem was designed to analyze features of spatial and angular discretization and demonstrate parallel efficiency of the multithreaded strategy.

Angular flux distributions over the full domain were modeled and simulated with mesh sizes of 5 cm, 4 cm, 3 cm, 2.5 cm, and 2 cm. Quadrature set quality was quantified by taking the root mean square (RMS) of relative scalar flux for all N point detectors:

$$\mathrm{RMS} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\frac{\phi_i^{\mathrm{cal}} - \phi_i^{\mathrm{ref}}}{\phi_i^{\mathrm{ref}}}\right)^2}$$

where reference solutions were obtained using an analytical method [**?**]. The RMS of relative errors in scalar flux using different quadrature sets versus number of ordinates is presented in Fig. 6a [Figure 6: see original paper]. Vari-

ous approximations in DGFEM with different mesh sizes converged at variable speeds and began to flatten as ordinate number increased under S20 angular approximations.

The ThorSNIPE code runs on a PC with 8 cores/16 threads (one 11th Gen Intel(R) Core(TM) i7-11700F CPU at 2.5 GHz) to demonstrate multithread technology performance. Speedup ratio versus quadrature set order is presented in Fig. 6b [Figure 6: see original paper], where parallel implementation efficiency exceeds 100% when S12 angular approximations are attained with mesh size less than 3 cm. Comparative results show the proposed method achieves better acceleration as mesh number increases. Speedup ratio increases more slowly when S20 angular approximations are attained, as each thread has over 30 parallel tasks and memory access becomes a bottleneck in matrix solvers for partial differential equations. Numerical results are shown in Fig. 7 [Figure 7: see original paper], and neutron fluxes at key points are listed in Table 5 . Compared with MCNP [**?**], ThorSNIPE effectively reduces angular discretization errors and fulfills transport computational requirements to a certain degree.

3.3 VENUS-3 Benchmark

The VENUS-3 benchmark was proposed to validate calculation methodologies and cross-section libraries for predicting fluence rates in reactor pressure vessels (RPVs), featuring clean structural geometry representing standard PWR pressure vessel conditions [**?**]. Fig. 8 [Figure 8: see original paper] illustrates the geometric configuration and material distribution of the VENUS-3 facility. The model includes two fuel rod types (PLSA), core baffle, water reflector, barrel, Pyrex control rods, neutron pad, three grid types, bottom support, reactor vessel, and jacket inner walls. Detailed descriptions and qualifications were obtained from the benchmark document. The benchmark arranges 386 dosimeters: 244 ^{58}Ni(n,p)^{58}Co dosimeters, 104 ^{115}In(n,n')^{115m}In dosimeters, and 38 ^{27}Al(n,a)^{24}Na dosimeters placed at 268 spatial locations. Experimental rate sets are provided as ratios of calculated reaction rate to average dosimeter cross-section, termed equivalent fission flux.

Numerical results shown in Fig. 9 [Figure 9: see original paper] were modeled using PNTN-S8 angular discretization, 199 KASHIL-E70 neutron group multigroup cross-sections [**?**], and 407,819 tetrahedral elements. Fig. 10 [Figure 10: see original paper] shows C/E comparisons of equivalent fission fluxes at three detector positions. For ^{115}In(n,n')^{115m}In dosimeters, almost all equivalent fission flux deviations are within \$±\$10%, except at detector position No. 31 (located in core barrel at 0.7° angle) where numerical values are affected by reflective boundary conditions. For ^{27}Al(n,a)^{24}Na dosimeters, 92% of deviations were within \$±\$10%, with three dosimeters overestimating experimental values by approximately 11.9%. For ^{58}Ni(n,p)^{58}Co dosimeters, 95% of deviations are limited to \$±\$10%, except at 11 detector positions where numerical values overestimated experimental values by 10.1%-20.7%, concentrated at the junction of PLSA region, outer baffle, and core barrel. Scalar flux distribution is not

smooth due to strong spatial nonuniformity, imposing higher requirements for local mesh refinement that significantly degrades simulation accuracy in these regions. Overall agreement between ThorSNIPE results and experimental data was very good, demonstrating high reliability and stability for complex shielding calculations.

Numerical results from TORT-3.2 with 1,004,295 hexahedral grids and 26 BUGLE-96 neutron energy groups [**?**] were selected as comparative references. For uniformly distributed 3.3% fuel regions, TORT-3.2 with hexahedral structured meshes achieved better results. ThorSNIPE with tetrahedral unstructured meshes can decrease numerical oscillation in complex regions. For ^{27}Al(n,a)^{24}Na dosimeters, ThorSNIPE results tended to be lower than TORT-3.2 in the core barrel, possibly due to errors in average dosimeter cross-section.

Fig. 11 [Figure 11: see original paper] shows speedup ratios for three quadrature sets with P3 Legendre expansion order and four Legendre expansion orders with PNTN-S8 angular discretization. Test conditions were consistent with TORT-3.2 code. Overall, algorithm behavior was similar to the Kobayashi-3i benchmark. For PNTN-S12, acceleration ratio was 6.47 with 407,819 elements. Comparing different expansion orders, the P5 expansion order showed the steepest upward trend, achieving maximum speedup ratio of 6.98. Numerical results indicate the proposed algorithm achieves favorable speedup ratio as mesh scale increases, as expected.

4 Conclusions

This study proposes and implements a multithreaded parallel upwind algorithm for solving the first-order Boltzmann neutron equation 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 stable solution by applying DGFEM spatial discretization. The multithread parallel strategy automatically manages angular sweep stacks for each thread and improves ThorSNIPE code quality and efficiency.

Algorithm performance and accuracy were tested using several typical benchmark problems. In the IAEA benchmark, the upwind sweep algorithm proved optimal for DGFEM calculations and can save significant computational costs. Results show ThorSNIPE agrees well with references, with 1 pcm deviation in eigenvalue and within 2.50% for flux distribution. In the Kobayashi-3i benchmark, angular flux simulations over the full domain with different mesh sizes and quadrature set orders indicate the multithreaded parallel algorithm has potential extensibility. In the VENUS-3 benchmark, 96% of equivalent fission flux deviations are limited to $±$10%, except at 15 detector positions where numerical values overestimated experimental values by 10.1%-20.7%. Maximum speedup ratio was 6.98 with 407,819 elements. VENUS-3 results demonstrate this robust algorithm can reliably support practical engineering applications and

improve computational efficiency.

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

Author Contributions: All authors contributed to study conception and design. Material preparation, data collection, and analysis were performed by Zhiwei Zong and Maosong Cheng. The first draft was written by Zhiwei Zong and all authors commented on previous versions. All authors read and approved the final manuscript.

Data Availability: Data supporting 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: The authors declare no competing interests.

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

References

1. E.E. Lewis, W.F. Miller, *Computational Methods of Neutron Transport* (American Nuclear Society, Illinois, 1981)
2. W.A. Rhoades, Jr., W.W. Engle, 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. Zhone et al., Reconstruction and parallelization of 3D SN 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 unstructured triangular meshes, *Comput. Phys.* (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/s41365-021-00934-7
8. V. Labouré, Y. Wang, J. Ortensi et al., Multischeme equivalence procedure for neutron transport finite element methods, *Ann. Nucl. Energy* 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 SN method for solving first-order neutron transport equation, *Ann. Nucl. Energy* 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, UK, 2005), pp. 14-17
16. Y. Wang, S. Schunert, J. Ortensi et al., Rattlesnake: A MOOSE-based multiphysics multischeme radiation transport application, *Nucl. Tech.* 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, *Proc. SC2000 Conf.* (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, 111-136 (2002)
19. J.Y. Moller, J.J. Lautard, Minaret, a deterministic neutron transport solver for nuclear core calculations, 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 of Conferences* 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, 77-85 (2023). https://doi.org/10.11889/j.0253-3219.2023.hjs.46.020602
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. (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. Supercomp.* 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. Energy* 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. Energy* 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, *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, USA, 2015), pp. 307-339
30. J. Stepanek, T. Auerbach, W. Hälg, 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. Energy* 39, 229-244 (2001). https://doi.org/10.1016/S0149-1970(01)00007-5
32. M. Pescarini, R. Orsi, 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 Phy. Soc.* 59, 1199-1202 (2011). https://doi.org/10.3938/jkps.59.1199

Note: Figure translations are in progress. See original paper for figures.