

DAPHNE-3D: A NEW TRANSPORT SOLVER FOR UNSTRUCTURED TETRAHEDRAL MESHES

Evangelia Diamantopoulou¹, Daniele Sciannandrone¹

¹Service d'études des réacteurs et de mathématiques appliquées (SERMA)
CEA, Université Paris-Saclay, F-91191, Gif-sur-Yvette, France
evangelia.diamantopoulou@cea.fr, daniele.sciannandrone@cea.fr

ABSTRACT

A new Discrete Ordinates transport solver for unstructured tetrahedral meshes is presented. The solver uses the Discontinuous Galérkin Finite Element Method with linear or quadratic expansion of the flux within each cell. The solution of the one-group problem is obtained with non-preconditioned fixed-point or GMRES iterations. Precision and performances of the solver are evaluated on the 3D Radiation Transport Benchmark Problems proposed by Kobayashi, showing very good agreement with the reference and good computing times in serial execution.

KEYWORDS: Particle transport, Unstructured Meshes, Discontinuous Galérkin, Kobayashi benchmark

1. INTRODUCTION

The Service d'Études des Réacteurs et de Mathématiques Appliquées (SERMA) is currently developing a computing platform for radiation shielding applications, called OPERA. The aim is to provide an integrated environment containing tools for data pre- and post-processing, particle transport and material activation. OPERA already contains simplified transport solvers based on point-kernel, a 3D discrete ordinates solver for Cartesian geometries, and a reference Monte Carlo transport code. To extend OPERA capabilities, we are currently developing a new solver, designed to treat arbitrary 3D geometries. The new solver is called DAPHNE (for Deep Attenuation PHotons and NEutrons) and it solves the first-order form of the transport equation using the multi-group approximation for the energy and the discrete ordinates approximation for the angle. In its first implementation, the geometry representation is based on a tetrahedral mesh partitioning the space.

Solvers with similar characteristics are not new in the transport community. Among the codes that solve the first-order form of the transport equation we can quote Split-Cell [1], ATTILA [2], AHOT-C-UG [3] and AETIUS [4]. The first one uses a linear or exponential expansion of the flux within tetrahedral cells, and a short-characteristics closure. In a similar fashion, AHOT-C-UG uses the short-characteristics formulation but with a robust arbitrary polynomial expansion of the flux within each cell. The ATTILA code uses the Discontinuous Galérkin Finite Element Method (DGFEM) with linear expansion of the flux within tetrahedral and a trilinear expansion in hexahedral elements. The AETIUS code mimics ATTILA in that it uses linear DGFEM for tetrahedral meshes. The main difference between the characteristic formulation and DGFEM is in the choice of the closure relating boundary fluxes: in short-characteristics, this relation is retrieved from the

integral form of the transport equation, while DGFEM uses the “upwind” approximation. As a consequence, the characteristics formulation better approximates, in principle, problems with pure streaming with the disadvantage that the coupling coefficients involve the evaluation of computationally expensive exponential functions of the total cross-section. On the contrary, the DGFEM coefficients have a simple linear dependence on the total cross-section but pure transmission is not explicitly accounted for.

In the first implementation of DAPHNE-3D, we chose to explore the DGFEM formulation. In addition to the linear DGFEM that can be found in ATTILA and AETIUS, we also implemented a quadratic expansion of the flux within each tetrahedral cell. The objective is to explore whether a higher order formulation helps to achieve similar precision with reduced computational resources.

The paper is organized as follows: in next section we will briefly recall the DGFEM formulation as proposed by Wareing [2] and provide some algorithmic details on the construction of DGFEM matrices and on the solution of the one-group equation. The following section is dedicated to the results obtained on the benchmarks proposed by Kobayashi [5].

2. DGFEM Formulation

For the derivation of the DGFEM equations, we start from the first-order form of the one-group, discrete ordinates transport equation assuming the standard Legendre expansion of the scattering cross-sections:

$$(\boldsymbol{\Omega}_n \cdot \boldsymbol{\nabla} + \sigma_t) \psi(\mathbf{r}, \boldsymbol{\Omega}_n) = \sum_l A_l(\boldsymbol{\Omega}_n) [\sigma_l \phi_l(\mathbf{r}) + Q_l(\mathbf{r})] = q(\mathbf{r}, \boldsymbol{\Omega}_n), \quad (1)$$

with ψ is the angular flux, q is the total angular source, A_l is the real Spherical Harmonic of order l , and ϕ_l is the angular moment of the flux with respect to the Spherical Harmonic of order l , computed with an angular quadrature formula. We have also assumed that the external source, Q_l , shares the same Spherical Harmonics expansion of the flux. Boundary conditions (BCs) are associated to Eq. (1). In our implementation we allow inhomogeneous BCs and reflective BCs. The latter are expressed as the mapping between the outgoing flux for a given direction and the incoming flux for the reflected one. The derivation of DGFEM equations is done following the formulation proposed by Wareing et al. [2]. For simplicity of the notation, we will omit the angular dependence and write the equations for one single angle. We start assuming that the spatial domain of the problem is partitioned into an unstructured mesh composed of K tetrahedral cells of volume V_k and surfaces Γ_{kf} for $k = 1, \dots, K$ and $f = 1, \dots, 4$. For each cell, we define a basis $\boldsymbol{\theta}_k(\mathbf{r})$ and write the expansion of the angular flux:

$$\psi(\mathbf{r}) = \boldsymbol{\theta}_k^T(\mathbf{r}) \boldsymbol{\psi}_k. \quad (2)$$

A similar expansion is assumed for the angular moments of the flux, the external source and the angular source. In our implementation we use basis of the Lagrangian type for linear (4-nodes) and quadratic elements (10-nodes). By projecting Eq. (1) onto the basis function, and by making use of the divergence theorem we obtain:

$$\sum_{f=1}^4 (\boldsymbol{\Omega}^T \mathbf{n}_{kf}) \int_{\Gamma_f} \boldsymbol{\theta}_k \psi_{kf} d\mathbf{r}_s - \sum_{i=1}^3 \Omega_i \int_{V_k} \frac{\partial}{\partial r_i} \boldsymbol{\theta}_k \boldsymbol{\theta}_k^T d\mathbf{r} \boldsymbol{\psi}_k + \sigma_t \int_{V_k} \boldsymbol{\theta}_k \boldsymbol{\theta}_k^T d\mathbf{r} \boldsymbol{\psi}_k = \int_{V_k} \boldsymbol{\theta}_k \boldsymbol{\theta}_k^T d\mathbf{r} \mathbf{q}_k. \quad (3)$$

Here, r_i is either x , y or z and Ω_i is the projection of Ω onto these axes. The symbol \mathbf{n}_{kf} represents the outgoing normal of face f of element k . Notice that for linear tetrahedral elements, this is constant for each face. The boundary fluxes, ψ_{kf} , are provided by the *upwind* closure:

$$\psi_{kf} = \begin{cases} \boldsymbol{\theta}_k^T \boldsymbol{\psi}_k & \Omega^T \mathbf{n}_{kf} \geq 0 \\ \boldsymbol{\theta}_{k^-}^T \boldsymbol{\psi}_{k^-} & \Omega^T \mathbf{n}_{kf} < 0 \end{cases}, \quad (4)$$

where with the symbol k^- we indicate the neighbor region through the face f . For the external boundaries, the incoming flux is given by BCs.

Eq. (3) is written in the global spatial coordinate system. As shown in [2], the calculation of DGFEM coefficients can be efficiently done in the local coordinate system of a unit tetrahedron of volume $\tilde{V} = 1/6$ with local coordinates $\tilde{\mathbf{r}} = \{\tilde{x}, \tilde{y}, \tilde{z}\}^T$ with $\tilde{x} \in [0, 1]$, $\tilde{y} \in [0, 1 - \tilde{x}]$, $\tilde{z} \in [0, 1 - \tilde{x} - \tilde{y}]$. By applying this transformation, the volume integrals in Eq. (3) become:

$$\int_{V_k} \frac{\partial}{\partial r_i} \boldsymbol{\theta}_k \boldsymbol{\theta}_k^T d\mathbf{r} = \det(\mathbf{J}_k) \sum_{j=1}^3 \mathbf{J}_{kij}^{-1} \int_{\tilde{V}} \frac{\partial}{\partial \tilde{r}_j} \tilde{\boldsymbol{\theta}} \tilde{\boldsymbol{\theta}}^T d\tilde{\mathbf{r}} = \det(\mathbf{J}_k) \sum_{j=1}^3 \mathbf{J}_{kij}^{-1} \mathbf{K}_j^*, \quad (5)$$

$$\int_{V_k} \boldsymbol{\theta}_k \boldsymbol{\theta}_k^T d\mathbf{r} = \det(\mathbf{J}_k) \int_{\tilde{V}} \tilde{\boldsymbol{\theta}} \tilde{\boldsymbol{\theta}}^T d\tilde{\mathbf{r}} = \det(\mathbf{J}_k) \mathbf{M}^*, \quad (6)$$

with \mathbf{J}_k the Jacobian matrix of the transformation. Matrix \mathbf{M}^* and the three \mathbf{K}_j^* can be computed once and used for all elements. As for volume integrals, the integral on surfaces in Eq. (3) is done using a local coordinate system mapping any triangular surface to the unit triangle of surface $\tilde{\Gamma}_s = 1/2$ with local coordinates $\tilde{\mathbf{r}}_s = \{\tilde{x}_s, \tilde{y}_s\}^T$ with $\tilde{x}_s \in [0, 1]$, $\tilde{y}_s \in [0, 1 - \tilde{x}_s]$. The integration is done by noticing that the trace of the Lagrangian basis function onto the surfaces is identically zero for the components that do not belong to the surface. We can thus write:

$$\tilde{\boldsymbol{\theta}}(\tilde{\mathbf{r}}_s) = \mathbf{P}_{kf} \tilde{\boldsymbol{\theta}}_s, \quad (7)$$

with \mathbf{P}_{kf} the rectangular permutation matrix $N_v \times N_s$, with N_v the number of volume nodes (4 for linear, 10 for quadratic elements) and N_s the number of surface nodes (3 for linear and 6 for quadratic). The permutation matrix is written for the face f and element k to take into account the local numbering of nodes of each region. The vector $\tilde{\boldsymbol{\theta}}_s$ contains the N_s non-zero entries of the trace of the basis. By inserting Eq. (4) and Eq. (7) into Eq. (3) we can finally obtain the surface integral as:

$$\int_{\Gamma_f} \boldsymbol{\theta}_k \boldsymbol{\theta}_{k'}^T d\mathbf{r}_s = |\mathbf{J}_f| \mathbf{P}_{kf} \int_{\tilde{\Gamma}_f} \tilde{\boldsymbol{\theta}}_s \tilde{\boldsymbol{\theta}}_s^T d\tilde{\mathbf{r}}_s \mathbf{P}_{k'f}^T = |\mathbf{J}_f| \mathbf{P}_{kf} \mathbf{U}^* \mathbf{P}_{k'f}^T, \quad (8)$$

with $|\mathbf{J}_f|$ being the magnitude of the surface Jacobian of the face f . Also note that we have distinguished the right- and left- permutation matrices to account for the cases of “outgoing” and “incoming” fluxes due to the *upwind* closure. In practice, since local numbering is known within each cell, the $\mathbf{U}_f^* = \mathbf{P}_{kf} \mathbf{U}^* \mathbf{P}_{k'f}^T$ is pre-computed, while for the incoming flux, the permutation is done on-the-fly using pre-computed \mathbf{U}^* . The final form of Eq. (3) is:

$$\left[\sum_{f_{out}} |\Omega^T \mathbf{n}_{kf}| |\mathbf{J}_f| \mathbf{U}_f^* - \det(\mathbf{J}_k) \sum_{i=j}^3 (\Omega^T \mathbf{J}_k^{-1})_j \mathbf{K}_j^* + \sigma_i \det(\mathbf{J}_k) \mathbf{M}_k^* \right] \boldsymbol{\psi}_k = \det(\mathbf{J}_k) \mathbf{M}_k^* \mathbf{q}_k + \sum_{f_{in}} |\Omega^T \mathbf{n}_{kf}| |\mathbf{J}_f| \mathbf{P}_{kf} \mathbf{U}^* \mathbf{P}_{k-f}^T \boldsymbol{\psi}_{k^-}. \quad (9)$$

For each region and angle, we build the matrix on the l.h.s of the equation on-the-fly and solve the problem using the Eigen library [6]. This strategy favors the reduction of the memory since the matrices do not have to be stored, but it can be computationally intensive especially for higher-order elements.

Void Limit

Some discretization schemes fail to provide the solution of transport in void regions. This is for example the case of schemes based on the second-order form of the transport equation [7], or when using cell-balance in the characteristics formulation [8]. The solution of Eq. (9) in void regions requires the matrix on the l.h.s. to be invertible for $\sigma_t = 0$. The analysis of \mathbf{U}_f^* and \mathbf{K}_j^* alone reveals that these matrices are singular for both linear and quadratic elements. However, it is easy to show that their kernels are orthogonal, thus insuring that the matrix resulting from their sum is always invertible. It must be noticed that for cells with high aspect ratio, the resulting matrix can be ill-conditioned.

One-Group Solution Algorithm

The solution of Eq. (1) is done by “sweeping” regions downstream for all the angles independently. This procedure does not allow an exact inversion of the one-group equation because of the presence of the scattering source and the reflective BCs. To solve Eq. (1) we implemented non-preconditioned fixed-point iterations and GMRES as proposed by Warsa et al. [9]. No modifications are done to their algorithm so details will not be given. The only difference worth noticing concerns the surface angular fluxes included in the Krylov space: while in [9] they include incoming and outgoing fluxes for reflective surfaces only, we chose to include outgoing fluxes for all the surfaces. This allows obtaining the outgoing flux for every surface without any additional sweep. If needed, incoming fluxes at reflective boundaries may be obtained by applying BCs.

3. Benchmark Results

The aim of the paper is to test the precision and the effectiveness of our DGFEM implementation for shielding applications regardless of the errors arising from the multi-group cross-section generation. To that end, we choose to test the solver on the Kobayashi Benchmark Problems [5]. They consist of three regions (source, void, and shield) with total cross section $\sigma_t = 0.1 \text{ cm}^{-1}$ for source and shield, and $\sigma_t = 1\text{E}-4 \text{ cm}^{-1}$ for the void region. Three geometry configurations are proposed, each one having two cases: Case i with no scattering, and Case ii with scattering ratio $\sigma_s/\sigma_t = 0.5$ for all regions. The benchmark problems are modelled and meshed using the GEOM and SMESH modules of SALOME-8 [10] respecting the initial geometry and by applying reflective boundary conditions.

For the calculation of the fluxes for Case i we have used fixed-point iterations while the results for Case ii occur by the use of GMRES with dimension of the Krylov space equal to 20. Triangular Tchebychev-Legendre angular quadrature formulas are used for the calculations. For a formula of order N , the number of angles is $4(N/2 + 1)(N/2)$. Calculations are done in double precision with convergence criterion of $1\text{E}-7$ on the relative error on average cell and surface fluxes for for

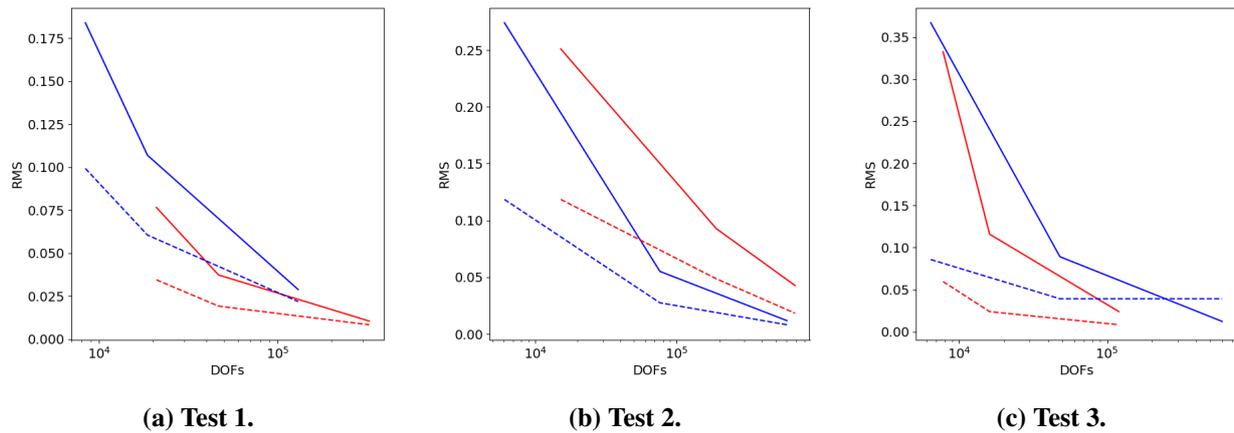


Figure 1: Root Mean Square vs number of Degrees of Freedom for linear (blue) and quadratic (red) DGFEM. Dashed: case with scattering. Solid: no scattering.

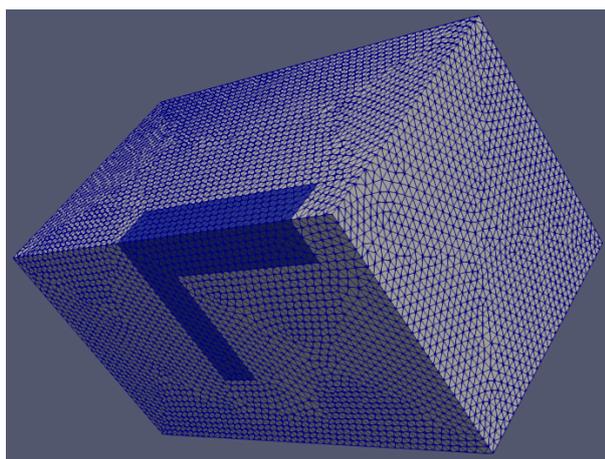
fixed-point iterations, and $1E-7$ residual in GMRES. Several tests have been performed for different mesh sizes and orders of angular quadrature aiming to the identification of the numerically-converged solution. Table 1 summarizes the chosen mesh and the angular quadratures, as well as the Root Mean Square (RMS), memory usage and computing time for a serial execution. The RMS here is computed on the calculation points required by the benchmark. Detailed results are given in Tables 2 and 3. Punctual values are obtained reconstructing the scalar flux in the desired point using the spatial basis function of the DGFEM approximation.

As a general comment, we see that both schemes have small relative error even for fluxes with attenuation of several orders of magnitude. Notice that the solution for Case ii (50% scattering) converges with lower order of quadrature as compared to Case i. This is due to reduced ray-effect in problems with scattering. Fig. 1 shows the RMS for the three tests for varying number of degrees of freedom. We notice that in tests 1 and 3, for comparable RMS, less degrees of freedom are required by quadratic elements with consequent reduced memory usage. This behavior is not observed in Test 2. However, in terms of computing time, the reduced number of degrees of freedom does not imply faster solution, as it can be seen from Table 1. As an example, Figure 2 shows a 3D plot of the mesh and of the scalar flux obtained for Case 3ii with linear DGFEM.

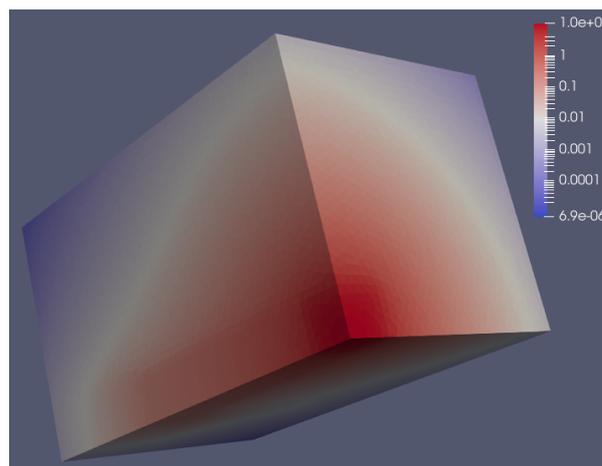
A comparison of our results with those published by AETIUS in [4] shows comparable precision. However, AETIUS results are computed using much coarser angular quadrature formula. This is probably due to the lack of the First Collision Source treatment in our solver. On the other hand, AETIUS solutions use much more refined spatial meshes. A comparison of our results with those provided by Ferrer with AHOT-C-UG [8] cannot be done since they model void regions using $\sigma_t = 0$, leading to higher fluxes. In addition, they provide results only for Case ii and for a subset of the points required by the benchmark. We can only notice that the convergence of the spatial mesh seems to be achieved with equal or coarser meshes. No information is provided about computing times in both references. Comparisons with the ATTILA code are not possible since results about Kobayashi benchmark have not been published.

Table 1: Summary of mesh size, quadrature, computing time and memory usage for linear and quadratic elements for the Kobayashi benchmark.

	Case 1i	Case 1ii	Case 2i	Case 2ii	Case 3i	Case 3ii
	Linear					
Mesh size/Quad	32k/S80	32k/S40	150k/S80	150k/S60	150k/S80	150k/S60
RMS [%]	2.89	2.19	1.18	0.81	1.23	0.44
Time [min]	6.72	7.42	42.81	102.17	45.33	100.96
Memory [Gb]	8.4	4.4	37.5	38.4	37.9	40.2
	Quadratic					
Mesh size/Quad	4k/S80	4k/S40	68k/S80	68k/S60	12k/S80	12k/S60
RMS [%]	3.74	1.93	4.14	1.83	2.40	0.83
Time [min]	6.47	6.22	103.41	216.95	17.95	38.66
Memory [Gb]	1.6	1.6	18.6	25.7	4.1	3.3



(a) Mesh and materials.



(b) Average scalar flux (log-scale).

Figure 2: Mesh and solution for Case 3ii with linear DGFEM.

4. Conclusions and Perspectives

DAPHNE-3D is a new discrete ordinates transport solver for three-dimensional unstructured tetrahedral meshes. In this work we have implemented a Discontinuous Galérkin Finite Element Method for space discretization using linear and quadratic expansions of the angular flux. Inhomogeneous and reflective boundary conditions are currently available. The one-group solution algorithm uses non-preconditioned fixed-point iterations or GMRES. The precision and the performance of the DGFEM solver have been tested on the 3D Radiation Transport Benchmark Problems proposed by Kobayashi for shielding applications. Results show very good agreement between the

Table 2: Results for Kobayashi problem 1.

Coord	Ref	Case i		Ref	Case ii	
		Ratio (Lin)	Ratio (Quad)		Ratio (Lin)	Ratio (Quad)
5, 5, 5	5.95659E+00	0.9741	1.0000	8.29260E+00	0.9696	0.9995
5, 15, 5	1.37185E+00	1.0620	0.9818	1.87028E+00	1.0650	0.9870
5, 25, 5	5.00871E-01	0.9697	1.0018	7.13986E-01	0.9704	1.0040
5, 35, 5	2.52429E-01	0.9672	0.9863	3.84685E-01	0.9694	0.9891
5, 45, 5	1.50260E-01	1.0101	1.0274	2.53984E-01	1.0061	1.0222
5, 55, 5	5.95286E-02	1.0243	0.9936	1.37220E-01	1.0063	0.9944
5, 65, 5	1.53283E-02	1.0146	1.0188	4.65913E-02	1.0076	1.0072
5, 75, 5	4.17689E-03	1.0082	1.0572	1.58766E-02	1.0023	1.0224
5, 85, 5	1.18533E-03	1.0012	1.0289	5.47036E-03	1.0020	1.0076
5, 95, 5	3.46846E-04	1.0123	1.0804	1.85082E-03	0.9863	1.0082
5, 5, 5	5.95659E+00	0.9741	1.0000	8.29260E+00	0.9696	0.9995
15, 15, 15	4.70754E-01	1.0401	1.0194	6.63233E-01	1.0414	1.0200
25, 25, 25	1.69968E-01	1.0232	1.0263	2.68828E-01	1.0205	1.0204
35, 35, 35	8.68334E-02	1.0292	1.0049	1.56683E-01	1.0225	1.0036
45, 45, 45	5.25132E-02	0.9997	1.0229	1.04405E-01	1.0016	1.0156
55, 55, 55	1.33378E-02	0.9576	0.9392	3.02145E-02	0.9763	0.9608
65, 65, 65	1.45867E-03	0.9688	1.1070	4.06555E-03	0.9878	1.0665
75, 75, 75	1.75364E-04	0.9643	1.0068	5.86124E-04	0.9855	1.0075
85, 85, 85	2.24607E-05	0.9711	1.0065	8.66059E-05	0.9853	1.0080
95, 95, 95	3.01032E-06	0.9914	1.0176	1.12892E-05	0.9893	1.0089
5, 55, 5	5.95286E-02	1.0243	0.9936	1.37220E-01	1.0063	0.9944
15, 55, 5	5.50247E-02	1.0091	0.9727	1.27890E-01	1.0072	0.9816
25, 55, 5	4.80754E-02	0.9912	1.0159	1.13582E-01	0.9990	1.0074
35, 55, 5	3.96765E-02	0.9823	1.0104	9.59578E-02	0.9939	1.0072
45, 55, 5	3.16366E-02	1.0392	1.0577	7.82701E-02	1.0230	1.0365
55, 55, 5	2.35303E-02	0.9737	1.0042	5.67030E-02	0.9845	0.9978
65, 55, 5	5.83721E-03	0.9604	1.0388	1.88631E-02	0.9733	1.0151
75, 55, 5	1.56731E-03	1.0247	1.0448	6.46624E-03	1.0078	1.0143
85, 55, 5	4.53113E-04	0.9861	1.0228	2.28099E-03	1.0003	1.0073
95, 55, 5	1.37079E-04	0.9435	1.0637	7.93924E-04	0.9744	1.0101

Table 3: Results for Kobayashi problems 2 and 3.

Coord	Ref	Case i		Ref	Case ii	
		Ratio (Lin)	Ratio (Quad)		Ratio (Lin)	Ratio (Quad)
Problem 2						
5, 5, 5	5.95659E+00	0.9981	0.9985	8.61696E+00	0.9977	0.9988
5, 15, 5	1.37185E+00	0.9975	1.0060	2.16123E+00	0.9973	1.0041
5, 25, 5	5.00871E-01	1.0143	1.0034	8.93437E-01	1.0094	1.0016
5, 35, 5	2.52429E-01	1.0027	1.0022	4.77452E-01	1.0005	1.0008
5, 45, 5	1.50260E-01	1.0004	1.0018	2.88719E-01	0.9992	0.9999
5, 55, 5	9.91726E-02	1.0063	1.0001	1.88959E-01	1.0030	0.9991
5, 65, 5	7.01791E-02	0.9973	0.9968	1.31026E-01	0.9969	0.9948
5, 75, 5	5.22062E-02	1.0053	0.9969	9.49890E-02	1.0027	0.9887
5, 85, 5	4.03188E-02	1.0051	0.9948	7.12403E-02	1.0026	0.9807
5, 95, 5	3.20574E-02	1.0032	0.9959	5.44807E-02	1.0121	0.9999
5, 95, 5	3.20574E-02	1.0032	0.9959	5.44807E-02	1.0121	0.9999
15, 95, 5	1.70541E-03	1.0432	1.1642	6.58233E-03	1.0235	1.0690
25, 95, 5	1.40557E-04	1.0028	0.9894	1.28002E-03	1.0001	1.0005
35, 95, 5	3.27058E-05	0.9986	0.9988	4.13414E-04	1.0010	1.0055
45, 95, 5	1.08505E-05	0.9984	0.9907	1.55548E-04	1.0007	1.0037
55, 95, 5	4.14132E-06	0.9976	0.9917	6.02771E-05	0.9909	0.9960
Problem 3						
5, 5, 5	5.95659E+00	1.0027	1.0108	8.61578E+00	1.0026	1.0090
5, 15, 5	1.37185E+00	1.0025	0.9869	2.16130E+00	1.0013	0.9881
5, 25, 5	5.00871E-01	1.0063	1.0150	8.93784E-01	1.0036	1.0101
5, 35, 5	2.52429E-01	1.0077	1.0048	4.78052E-01	1.0045	1.0037
5, 45, 5	1.50260E-01	1.0116	1.0010	2.89424E-01	1.0078	1.0016
5, 55, 5	9.91726E-02	0.9953	1.0111	1.92698E-01	0.9964	1.0078
5, 65, 5	4.22623E-02	0.9929	0.9965	1.04982E-01	0.9962	1.0048
5, 75, 5	1.14703E-02	1.0133	1.0113	3.37544E-02	1.0073	1.0049
5, 85, 5	3.24662E-03	1.0058	0.9890	1.08158E-02	1.0000	0.9986
5, 95, 5	9.48324E-04	1.0058	0.9967	3.39632E-03	1.0034	1.0043
5, 55, 5	9.91726E-02	0.9953	1.0111	1.92698E-01	0.9964	1.0078
15, 55, 5	2.45041E-02	0.9992	0.9464	6.72147E-02	0.9989	0.9719
25, 55, 5	4.54477E-03	1.0104	0.9656	2.21799E-02	1.0006	0.9933
35, 55, 5	1.42960E-03	0.9865	0.9934	9.90646E-03	0.9959	1.0000
45, 55, 5	2.64846E-04	1.0005	1.0178	3.39066E-03	0.9951	1.0006
55, 55, 5	9.14210E-05	0.9976	0.9950	1.05629E-03	1.0019	1.0002
5, 95, 35	3.27058E-05	0.9981	1.0060	3.44804E-04	1.0051	1.0072
15, 95, 35	2.68415E-05	1.0042	1.0010	2.91825E-04	0.9910	0.9922
25, 95, 35	1.70019E-05	0.9907	0.9868	2.05793E-04	0.9975	0.9969
35, 95, 35	3.37981E-05	0.9561	0.9553	2.62086E-04	0.9934	0.9975
45, 95, 35	6.04893E-06	1.0198	0.9330	1.05367E-04	1.0030	0.9946
55, 95, 35	3.36460E-06	1.0018	0.9792	4.44962E-05	0.9970	0.9980

numerically converged DGFEM solution and the reference values provided by the benchmark. Comparisons between linear and quadratic elements show that the higher-order expansion helps reducing the total number of degrees of freedom, with consequent reduction of the memory. However, the use of higher-order elements does not always provide a reasonable reduction of computing time. Results also confirmed that a very large quadrature order is required to eliminate the ray-effect arising in this kind of configuration.

As a perspective here, we are interested in the investigation and comparisons of additional spatial discretization methods such as those proposed by [1] and [3], in particular for shielding applications. Future work will be also dedicated to the construction of suitable low-order preconditioner of the one-group equation for accelerating the fixed-point and GMRES iterative schemes. The implementation of ray-effect mitigation techniques such as the First Collision Source method is also a necessary step for the effective solution of shielding problems. Finally, since the solver is currently implemented using only serial execution, significant reduction of the computing time can be obtained by applying shared-memory parallelism.

REFERENCES

- [1] Kirk A. Mathews, Rodney L. Miller, Charles R. Brennan, “Split-Cell, Linear Characteristics Transport Method for Unstructured Tetrahedral Meshes”, *Nuclear Science and Engineering*, **136**, pp. 178-201 (2000).
- [2] Todd A. Wareing, John M. McGhee, Jim E. Morel, and Shawn D. Pautz, “Discontinuous Finite Element SN Methods on Three-Dimensional Unstructured Grids”, *Nuclear Science and Engineering*, **138**, pp. 256-268 (2001).
- [3] Rodolfo M. Ferrer, Yousry Y. Azmy, “A Robust Arbitrarily High-Order Transport Method of the Characteristics Type for Unstructured Grids”, *Nuclear Science and Engineering*, **172**, pp. 33-51 (2012).
- [4] John Woon Kim, Young-Ouk Lee, “AETIUS solutions for Kobayashi 3D benchmarks with the first collision source method on the volume source and unstructured tetrahedral mesh”, *Annals of Nuclear Energy*, **113**, pp. 446-469 (2018).
- [5] Keisuke Kobayashi, and Naoki Sugimura, “3D Radiation Transport Benchmark Problems and Results for Simple Geometries with Void Region”, *Progress in Nuclear Energy*, **39**, pp. 119-144 (2001).
- [6] Gaël Guennebaud, Benoît Jacob and others, “Eigen v3” eigen.tuxfamily.org (2010).
- [7] J. E. Morel, and J. M. McGhee, “A Self-Adjoint Angular Flux Equation”, *Nuclear Science and Engineering*, **132**, pp. 312-325 (1999).
- [8] Rodolfo M. Ferrer, “An Arbitrarily High Order Transport Method of the Characteristic Type for Unstructured Tetrahedral Grids”, Ph.D. Thesis, The Pennsylvania State University, (2010).
- [9] James S. Warsa, Todd A. Wareing, and Jim E. Morel, “Krylov Iterative Methods and the Degraded Effectiveness of Diffusion Synthetic Acceleration for Multidimensional SN Calculations in Problems with Material Discontinuities”, *Nuclear Science and Engineering*, **147**, pp. 218-248 (2004).
- [10] “SALOME: The Open Source Integration Platform for Numerical Simulation”, salome-platform.org (2019).