Version of Record: https://www.sciencedirect.com/science/article/pii/S0021999119305327
Manuscript_ec8d2e21234976dfe7f3dcdc0ae6103a

Uncollided Flux Techniques for Arbitrary Finite Element Meshes

Milan Hanuš, Logan H. Harbour, Jean C. Ragusa[1], Michael P. Adams, Marvin L. Adams

Department of Nuclear Engineering, Texas A&M University, College Station, TX, USA

Abstract

The uncollided angular flux can be difficult to compute accurately in discrete-ordinate radiation transport codes, especially in weakly-scattering configurations with localized sources. It has long been recognized that an analytical or semi-analytical treatment of the uncollided flux, coupled with a discrete-ordinate solution for the collided flux, can yield dramatic improvements in solution accuracy and computational efficiency. In this paper, we present such an algorithm for the semi-analytical calculation of the uncollided flux. This algorithm is unique in several aspects: (1) it applies to arbitrary polyhedral cells (and can be thus coupled with collided flux solvers that support arbitrary polyhedral meshes without the need for explicit tetrahedral re-meshing), (2) it provides accurate uncollided solutions near sources, (3) it is devised with parallel implementation in mind, and (4) it minimizes the total number of traced rays and maintains a reasonable ray density on each local subdomain. This paper provides a complete derivation of the algorithm and demonstrates its important features on a set of simple examples and a standard transport benchmark. Assessment of its parallel performance will be the subject of a subsequent paper.

Keywords: uncollided flux, ray tracing, arbitrary finite elements, first-collision source, radiation transport

Email addresses: mhanus@tamu.edu (Milan Hanuš), loganharbour@tamu.edu (Logan H. Harbour), jean.ragusa@tamu.edu (Jean C. Ragusa), mpadams@tamu.edu (Michael P. Adams), mladams@tamu.edu (Marvin L. Adams)

[1]Corresponding author

1. Introduction

Many particle-transport problems of practical interest contain sources that are of small size relative to the spatial domain's dimensions. At a given spatial point away from such a small, localized source, the "uncollided" portion of the angular flux is non-zero **only** in a cone of directions. The size of this cone, expressed in units of steradians, is roughly the "visible" source area seen from that point divided by 4π times the square of the distance between the observation point and the source. Hence, if the visible source area is small or if the source and the observation point are far apart, the size of that cone can be quite small. It is extraordinarily difficult for standard angular discretization techniques, such as the discrete-ordinates method or the spherical-harmonics method, to compute this uncollided angular flux accurately when the size of this cone is smaller than $4\pi/N_{\text{angular}}$, where N_{angular} is the number of degrees of freedom in the chosen angular discretization scheme. As an example, consider a 1-cm^2 source area and an observation point 1 meter away; the cone size is about 8×10^{-6} ster and a uniform angular discretization scheme would require about 1.5 million directions. The need for an accurate treatment of the uncollided component of the angular flux is exacerbated in weak scattering media, where the uncollided flux contributes more significantly to the total (i.e., uncollided+collided) angular flux. Because of these difficulties, there has long been a need for algorithms that can calculate the uncollided flux analytically or semi-analytically (using ray-tracing). We note that the uncollided-flux approach is only one possible technique used to mitigate ray effects; other approaches include, for instance, P_n-equivalent fictitious-source methods [6]. Also note that both uncollided and first-collided fluxes may be singular [17]; in such instances, ray-effect mitigation techniques may have to be applied to both uncollided and first-collided angular fluxes.

Examples of transport codes that implement uncollided-flux algorithms include Partisn [1], Denovo [2], GRTUNCL3D [12], FNSUNCL3 [10, 11], IDT [26],

Attila [24], AETIUS [8]. These computer programs, with the exception of Attila and AETIUS, employ brick-type meshes to represent the problem geometry. However, not all problems can be easily represented using block grids (without a large increase in the number of mesh cells when curved surfaces need to be represented). For problems with complex geometrical shapes, one may have to use dedicated meshing software, resulting in arbitrary unstructured grids, such as the ones often encountered in finite element analysis. Furthermore, large transport problems are often carried out in parallel using many processors. For parallel-partitioned grids, having localized sources in some of the partitions can easily create a bottleneck, where partitions not owning a given source have to request uncollided flux solutions from other partitions, burdening source-owning partitions the most.

Arbitrary unstructured grids are often used with finite-element techniques, where a polynomial representation of the solution is sought in every mesh-cell. Hence, an uncollided-flux algorithm capable of handling finite-element grids should include the computation of spatial moments of the uncollided flux in every cell (to compute the first-collision source, for instance).

Thus, it is clear that enhancements to existing uncollided-flux algorithms are needed in order to handle: (1) finite-element spatial representations, (2) unstructured grids, and (3) distributed parallel grids. The method that we present here possesses many desirable features:

1. It is designed for arbitrary polyhedral-cell grids in 3D (thus not restricted to block-grids).
2. It is designed to be efficiently implementable on parallel computer architectures.
3. It eliminates rays that are nearly collinear (a situation often encountered when tracing from a source point to a cluster of far-distant cells) by introducing interpolations that add small, controllable errors.
4. It accurately treats complicated source shapes, consistently accounting for the non-point nature of physically realizable sources.

5. It accurately calculates the uncollided flux close to sources.
6. It accurately treats attenuation within source volumes.
7. It accurately computes spatial and spherical-harmonics moments of the uncollided flux, not just the cell-averaged scalar flux, in each cell in the problem domain. These quantities are needed for the computation of the "first-collision" source (FCS) using a finite-element spatial representation.
8. It provides exiting-flux information at specified points on the problem boundary, so that post-processing algorithms have all of the details necessary to (for example) compute detector response rates outside of the problem boundary.
9. It can be used to compute "last-collision" source to treat the streaming of particles after their last collisions.

The remainder of the paper is organized as follows. In Section 2, we review some details of the uncollided-flux technique. An overview of our algorithm is presented in Section 3. Detailed derivations related to unstructured polyhedral grids and to distributed meshes are given in Section 4. Results are presented in Section 5 and conclusions and further work are proposed in Section 6.

2. Uncollided Angular Flux Method

2.1. Generalities

Consider the following linear transport problem,

$$\begin{aligned}\boldsymbol{\Omega}\cdot\boldsymbol{\nabla}\psi(\boldsymbol{r},\boldsymbol{\Omega},E)+\sigma_t(\boldsymbol{r},E)\psi(\boldsymbol{r},\boldsymbol{\Omega},E)&=S_{\text{fixed}}(\boldsymbol{r},\boldsymbol{\Omega},E)\\&+\int_{4\pi}\mathrm{d}\Omega'\int_0^\infty \mathrm{d}E'\,\sigma_s(\boldsymbol{r},E'\to E,\boldsymbol{\Omega}'\to\boldsymbol{\Omega})\psi(\boldsymbol{r},\boldsymbol{\Omega}',E')\,,\quad\forall\boldsymbol{r}\in V\end{aligned}\tag{1}$$

with inflow boundary conditions:

$$\psi(\boldsymbol{r},\boldsymbol{\Omega},E)=\psi_{\text{inc}}(\boldsymbol{r},\boldsymbol{\Omega},E)\,,\quad\forall\boldsymbol{r}\in\partial V^-=\{\boldsymbol{r}\in\partial V,\text{ such that }\boldsymbol{\Omega}\cdot\boldsymbol{n}(\boldsymbol{r})<0\}\,.\tag{2}$$

In the above, $\boldsymbol{r},\boldsymbol{\Omega},E$ are the phase-space variables (position, direction, energy), ψ is the angular flux solution, σ_t and σ_s are the total and scattering cross

sections, V and ∂V and the problem's domain and its boundary, $\boldsymbol{n}(\boldsymbol{r})$ is the outward unit normal vector at point $\boldsymbol{r}$, and S_{fixed} is the extraneous fixed source. No fissile material has been included in the above equations but contributions due to fissions are straightforward to add and no loss of generality is incurred by omitting them here. For conciseness, the following operator notation will be used to represent the model described in Eqs. (1)-(2):

$$L\psi = H\psi + S\,, \tag{3}$$

where L is the streaming and interaction operator and H is the collision (scattering) operator. One can split the angular flux into its uncollided and collided parts, $\psi = \psi_u + \psi_c$. Then, solving Eq. (3) is formally equivalent to solving the following system of equations:

$$L\psi_u = S\,, \tag{4a}$$

and

$$L\psi_c = H\psi_c + H\psi_u\,. \tag{4b}$$

Eq. (4a) can be solved semi-analytically using ray-tracing (RT) techniques:

$$L^{\text{RT}}\psi_u = S\,. \tag{5a}$$

Once the uncollided angular flux is known, the first-collision source, $H\psi_u$, can be computed and serves as the source term to the collided flux equation, Eq. (4b), which can be solved using discrete-ordinates (SN) and source iteration (index ℓ):

$$L^{\text{SN}}\psi_c^{(\ell+1)} = H\psi_c^{(\ell)} + H\psi_u\,. \tag{5b}$$

It is usually much easier to solve the collided-flux problem using standard angular discretization techniques than to solve the original problem that includes the uncollided flux, because the collided flux arises from a more distributed source (the first-collision source) and is, therefore, more smoothly distributed in direction at any given spatial point. Upon convergence of the source iterations (superscript ∞), the total angular flux is given by

$$\psi = \psi_u + \psi_c^{\infty}\,. \tag{6}$$

Note that if angular discretization errors in the collided flux component are still unacceptably large, e.g., for detector response simulations, a last-collision source treatment can be employed, whereby the first-collision source, $H\psi_u$, and the SN-converged scattering source, $H\psi_c^{(\infty)}$, are used in one more ray-traced solution:

$$L^{\mathrm{RT}}\psi = H\psi_c^{(\infty)} + H\psi_u + S\,. \tag{7}$$

This ray-traced solution is typically only sought in a localized region of interest, hence ray tracing occurs from all regions of the computational domain towards the zone of interest. This is formally equivalent to invoking ray tracing for the uncollided component of the adjoint angular flux.

2.2. Current Issues Pertaining to Uncollided Flux On Unstructured Grids and Parallel-distributed Grids

In large-scale simulations of realistic radiation transport problems, it is not uncommon to employ unstructured meshes, i.e., the computational domain is represented with arbitrarily shaped tetrahedra, hexahedra, and more generally polyhedra. Such meshes are often encountered in finite-element discretizations due to the availability of powerful mesh generation software, such as Cubit [18], OpenCascade [14], TetGen [20], and GMSH [3]. Furthermore, the computational grid itself may be partitioned, or distributed, among several (many) processors for parallel computing. These two aspects (unstructured grids and parallel meshes) have a direct impact on existing uncollided-flux algorithms: first, to the best of our knowledge, there exist no such algorithms in the radiation transport community for arbitrary finite-element meshes; second, current algorithms were not designed with parallelism in mind, thus overburdening the processors owning localized sources with excessive ray-tracing.

Note that, even for serial computations, current algorithms do not yield a uniform ray density throughout the computational grid: indeed, if rays are to be traced from a given source cell to all cells in the domain, neighboring cells that are located far away from the source cell will require almost co-linear rays to be spawned for the source, resulting in a higher ray density near the source.

Consider a 3D domain meshed with n^3 cells. For a single source cell, the number of rays to be traced is proportional to n^3 (if a quadrature rule is used to sample the cells, then $n_q^2 n^3$ rays would be traced, with n_q the number of 3D spatial quadrature points in a cell). In the algorithm we propose, the number of rays to be traced is only proportional to n^2 (by ray-tracing only to the meshed surfaces enclosing the volume). Hence, our method is significantly less demanding in terms of the number of ray-traces.

The concept of ray-tracing to some meshed surfaces enclosing a volume is extended to distributed grids, where these surfaces are employed to interpolate uncollided flux from upwind partitions, thus not requiring every single ray to be traced back all the way to the originating source cell. Obviously, interpolating uncollided flux can be prone to large errors due to the exponential attenuation. In practice, the interpolation is carried out on the argument of the exponential function, that is, on the optical thickness, not the flux itself.

The algorithm we describe in Section 3 works with arbitrary source/target cell types, while allowing for a more uniform ray density throughout the domain, and is written to be extensible to distributed meshes.

Finally, we note that the angular flux (be it the uncollided flux, from Eq. (5a), or the last-collided flux, from Eq. (7)) is only to be stored at the outer boundary of the computational domain, for subsequent simulations (e.g., detector response) or post-processing. However, what is needed inside the computational domain are space-angle moments of the uncollided angular flux. Angular moments (using spherical harmonic functions, for instance) are needed to compute the first-collision source (up to the anisotropy order in the scattering cross sections). Spatial moments are necessary to represent the first-collision source spatial distribution within a cell (i.e., the uncollided angular flux should be projected into the chosen finite-element representation). For instance, the space-angle moments of the uncollided angular flux in cell K due to a volumetric source in cell K' might be computed as follows:

$$M_{i,n}^g = \int_{K'} \mathrm{d}^3 r' \int_K \mathrm{d}^3 r f_{i,n}(\boldsymbol{r}, \boldsymbol{\Omega}_R) \frac{\exp\left(-\tau(\boldsymbol{r}, \boldsymbol{r}')\right)}{R^2} S_{\text{fixed}}^g(\boldsymbol{r}', \boldsymbol{\Omega}_R)\,, \qquad (8)$$

where $f_{i,n}(\boldsymbol{r}, \boldsymbol{\Omega}_R)$ is a space-angle basis function (e.g., the product of the i^{th} spatial basis function and the n^{th} spherical harmonic function), $\boldsymbol{R} = \boldsymbol{r} - \boldsymbol{r}'$ is the vector from the source point $\boldsymbol{r}'$ to the target point $\boldsymbol{r}$, $R = \|\boldsymbol{R}\|$, $\boldsymbol{\Omega}_R = \boldsymbol{R}/R$ is the unit direction vector from the source point to the target point. $\tau(\boldsymbol{r}_1, \boldsymbol{r}_2)$ is the number of mean free paths on a straight line between $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$. $S^g_{\text{fixed}}(\boldsymbol{r}', \boldsymbol{\Omega}_R)$ is the source emission rate density at source point $\boldsymbol{r}'$, in direction $\boldsymbol{\Omega}_R$, for energy group g. Here, we have used the Green's function expression for the uncollided flux.

If the two cells, the target cell K and the source cell K', are distinct, one might use Eq. (8) to directly evaluate the moments $M^g_{i,n}$, for example by taking a finite number of quadrature points in K and a finite number in K', together with their respective weights (equal to the fraction of cell volume associated to each point) so that

$$M^g_{i,n} = \sum_{j'} w'_j \sum_j w_j f_{i,n}(\boldsymbol{r}_j, \boldsymbol{\Omega}_{jj'}) \frac{\exp\left(-\tau(\boldsymbol{r}_j, \boldsymbol{r}_{j'})\right)}{R^2_{jj'}} S^g_{\text{fixed}}(\boldsymbol{r}_{j'}, \boldsymbol{\Omega}_{jj'})\,. \quad (9)$$

Even though this is likely the most straightforward approach to obtaining space-angle moments of the uncollided flux in an unstructured grid setting, we have already mentioned that the number of rays required in this approach can be excessive (see the discussion above). For example, in the Kobayashi benchmark presented in Section 5.2.4, a literature review revealed that the methods based on the direct quadrature needed to trace an order of 10^7 to 10^{10} rays to get results comparable to ours obtained with 10^6 rays. Moreover, accuracy of a quadrature-based approach may also be questionable for target cells near the source cell (when treating the distributed source as a collection of point sources is inaccurate). Furthermore, if the source cell is optically thick, most of the radiation leaving the source is emitted by the rim of the source cell and the accuracy of the uncollided solution using Eq. (9) depends on whether that rim zone has been adequately sampled by the numerical quadrature (typically requiring a high-order quadrature to ensure some quadrature points are located in the rim).

2.3. PWLD Finite-Element Representation

When solving Eq. (5b) for the collided-flux contribution on unstructured grids, a finite-element spatial discretization is often used. The spatial moments (finite-element representation) of the uncollided flux need to be obtained in order to compute the first-collision source. In most radiation transport codes capable of handling unstructured grids, mesh cells used are mainly tetrahedra or hexahedra. However, in recent years, finite-element discretizations for arbitrary polyhedral cells have been developed and used in radiation transport codes; see, for instance, the PieceWise Linear Discontinuous (PWLD) discretization devised by Adams [21, 22], and its quadratic extension [4]. On tetrahedral grids, the PWLD discretization devolves to the standard linear discontinuous (LD) representation. Some of the main advantages of the PWLD representation are as follows: (1) it is a discontinuous finite-element representation with the number of cell unknowns equal to the number of cell vertices; (2) it possesses the thick-diffusion limit; (3) it allows for a seamless transition between cells of different refinement levels in mesh adaptivity [16, 4]; (4) finally, PWLD allows for tetrahedral/hexahedral grids to be cut (e.g., with planes) to obtain non re-entrant spatial subdomains for parallel simulations without requiring that the cells being cut are of a specific type (the cut cells can be arbitrary polyhedra).

In this work, we seek spatial moments of the uncollided flux for arbitrary grids using the PWLD representation defined below. Consider a polyhedral cell K, with N_V vertices. For a given vertex i, we denote by F_i the set of faces of K containing vertex i, and by V_i the set of vertices sharing an edge with vertex i. Underlying the PWLD discretization is a set of tetrahedral cells. An example of such discretization and some of the underlying tetrahedra follows in Figure 1. Each tetrahedron is made of two adjacent vertices of the polyhedral cell (i.e., an edge of that cell), the face center for the face containing that edge, and the polyhedral cell center. However, it should be stressed that none of the face centers nor the cell center are associated with unknowns in the PWLD discretization. Thus, taking a cube as an example, the cube is decomposed into 24 tetrahedra to build the PWLD matrix, but the PWLD matrix system

only contains 8 unknowns for that cube. A PWLD representation for a given function f is given by :

$$f(\boldsymbol{r}) = \sum_{j=1}^{N_V} f_j b_j(\boldsymbol{r}).$$

Each basis function $b_j(\boldsymbol{r})$ is defined as

$$b_j(\boldsymbol{r}) = t_j^v(\boldsymbol{r}) + \sum_{k \in F_j} \beta_k t_k^{f,j}(\boldsymbol{r}) + \alpha t_c(\boldsymbol{r}). \tag{10}$$

The t_j^v functions are standard linear functions defined tetrahedron by tetrahedron. For example, t_j^v equals 1 at the j^{th} vertex and decreases linearly to zero on all other vertices of each edge that is connected to point j. t_c is unity at the polyhedral cell center and zero at each face center and each cell vertex. $t_k^{f,j}$ is unity at the face center (the face has vertex j as one of its vertices) and zero at the cell center and at each of the face vertices. The α and β_k are weights that yield the cell and face centers as weighted averages of the cell's vertices (for example, we have for a cube cell: $\beta_k = 1/4$ and $\alpha = 1/8$).

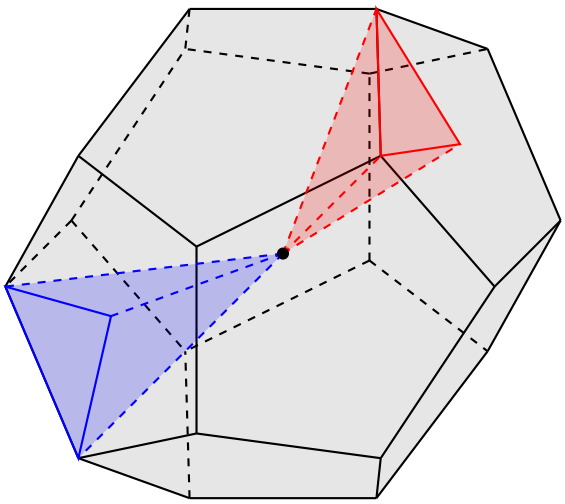


Figure 1: An arbitrary polyhedron highlighting two of the underlying tetrahedra. Each tetrahedron is made of two adjacent vertices of the polyhedral cell (i.e., an edge of that cell), the face center for the face containing that edge, and the polyhedral cell center.

3. Overview of the New Uncollided-Flux Algorithm

We have developed a new parallel uncollided-flux treatment for fixed-source problems with *arbitrary* polyhedral spatial cells. The goal is to treat the flux exiting from a source volume as nearly analytically as is practical, but in a way that is computationally efficient and portable on parallel architectures. In this section, we outline our algorithm.

3.1. Spatial Domain Decomposition

In keeping with common transport-code parallel strategies, our uncollided treatment employs a spatial domain decomposition that makes it straightforward to assign different portions of the spatial domain to different processing units. We define a set of "interpolation surfaces" (ISs) that divide the spatial domain into space-filling non-overlapping "interpolation volumes" (IVs). An IV is decomposed into polyhedral cells which are further subdivided into tetrahedral subcells to facilitate the ray-tracing. The centroid of a subcell that contains a fixed source is called a "source point" (SP) and serves as the origin of that subcell's uncollided rays. However, we stress that we always consider the non-point nature of the source volume; a SP should be understood as the source volume centroid. Treating every source cell as a volumetric source rather than as a point source (or collection of point sources) is discussed in detail in Sections 4.3.2 through 4.3.4.

A key feature of our approach is a defined set of "interpolation surface points" (ISPs) on each IS at which we find the detailed uncollided flux solution for each source. A fractional area of an IS is associated to each ISP such that the full area of each IS is accounted for by its ISPs. This area subtends a cone of solid angle at the SP as shown in Figure 2.

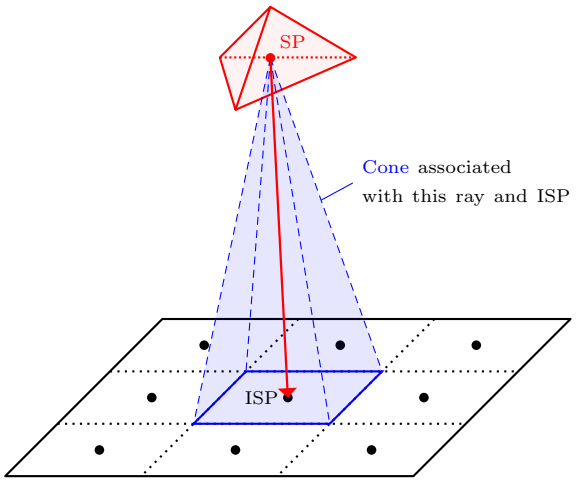


Figure 2: An example of the cone of solid angle associated with a single ray emanating from a source point to an ISP. For every SP there is one spatial cone for every ISP that is "outgoing" for that SP and that IV. The ISP areas fill the outgoing surface area and do not overlap, therefore the spatial cones they define fill the whole volumetric space and do not overlap. The cone of solid angle for a ray traced from an IS to another ISP is similar.

Tracing the full set of rays from each source point to each point in which the uncollided flux is desired would be prohibitive in a problem with high cell counts. It would also scale poorly if the processor that owned a source point had to initiate such a large number of rays emanating from its source point, while most processors that do not own sources would have far fewer rays to trace. Our algorithm solves this problem by creating a reasonable outgoing ray density near each source point, then using a carefully devised interpolation algorithm on each IS to spawn more rays as the distance from the source grows, with the goal of always maintaining a ray density that is appropriate for the local spatial mesh. The propagation of the rays throughout the domain and the interpolation algorithm is described in more detail in the following section. It is important to note there is a single ray for each SP-ISP pair, that is, the same ray is used to compute the uncollided flux in all cells it intersects between the source volume and the interpolation surface point.

3.2. Ray Propagation

The splitting of a global ray-tracing into separate traces throughout the subdomains (IVs) connected at common surfaces (ISs) is facilitated by two types of ray tracing. In a "forward trace," the starting and ending point of the ray in the direction of the particle flow are known and the ray can thus be traced directly, gathering the contributions to the PWLD representation of the uncollided flux in each mesh cell intercepted during the trace. In a "backward trace," a ray is traced from a given spatial point in the direction of the source point in order to determine the intersection with another surface in the mesh. This intersection will then serve as a starting point for the subsequent forward trace towards the given spatial point (utilizing the geometrical information about the cells visited during the backward trace).

The ray propagation from a given source point starts with a set of forward traces through the IV that contains the source point. The rays are traced from the source point to the ISPs on the enclosing ISs (seen as the rays in IV_1 in Figure 3). Once all traces from a single SP to all the ISPs that are lying on the inflow boundary of the neighboring IV are complete, the algorithm can obtain the information needed to spawn forward-tracing of additional rays in the neighboring IV via interpolation from the values stored for these ISPs (as described in more detail in Section 3.2.1). The points at which the values required for tracing the new rays will be interpolated are obtained by backward-tracing from the known ISPs on the outflow surface of the neighboring IV towards the inflow ISs. An example of these interpolated rays follows in Figure 3. A backward trace is first initiated from each outgoing ISP of IV_2 in the direction of the SP (this would occur for *every* SP in IV_1). The intersection of the line from a given outgoing ISP in IV_2 with the IS between IV_1 and IV_2 then serves as the starting point for the forward trace for a ray that contains the contribution from the SP to that outgoing ISP in IV_2.

This process is continued until rays have been propagated to the global domain boundary (ISPs on the domain boundary) for each source point in the domain. The values stored on boundary ISPs can be used to produce exiting-

flux information in post-processing. As rays propagate through the interior, their paths traversed through the spatial cells are utilized to obtain the PWLD representation of the uncollided flux.

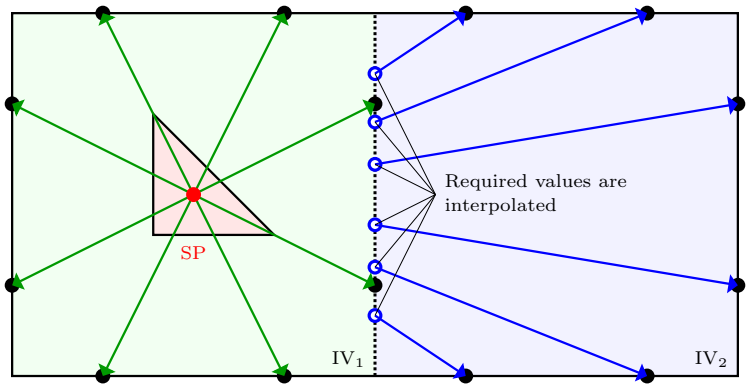


Figure 3: Two-dimensional example of ray propagation across an interpolation surface and to the boundary. Each filled point is an ISP. Rays are propagated to the outgoing ISPs of IV_1 from the SP. The rays in IV_2 do not align with the rays in IV_1, but intersect the common IS at different points. The averaged total cross section from the source volume exit to the intersection point is interpolated from the averaged values stored at the surrounding ISPs. Each point filled with white is interpolated. This is the only information needed to launch rays in IV_2.

3.2.1. Ray Data Interpolation

When a forward ray intersects an IS at an ISP, the averaged total cross section from the exiting source volume surface is stored at that ISP (for each SP in that IV). On a given ISP, the only necessary pieces of information to spawn rays in the neighboring IV on the opposite side of the IS are:

- the coordinate of the source point,
- the source strength, and
- the averaged total cross section.

These rays emulate rays originating from the source point and traversing through the IV without the need to trace them.

The averaged total cross section is the number of mean-free paths (accumulated segment by segment as the ray is traced) divided by the accumulated total segment length:

$$\langle \sigma_t \rangle_i = \frac{\tau_i}{\|\mathbf{ISP}_i - \mathbf{S}_{ex,i}\|}\,, \tag{11}$$

where i denotes the index of the ISP and $\mathbf{S}_{ex,i}$ the coordinates of its corresponding source surface exit point. Consider a given IV and let $\{\mathbf{ISP}_i\}$ be the set of ISPs on its ISs that form the inflow boundary of the neighboring IV. Once the ISPs from this set have been traced to, backward traces are spawned from the outflow ISPs in the neighboring IV to determine the starting points for the new rays. Take one such point P. The averaged total cross section at P is obtained by a simple inverse-distance-weighted interpolation [19]:

$$\langle \sigma_t \rangle_P = \begin{cases} \dfrac{\sum_{i=1}^{r} w_i \langle \sigma_t \rangle_i}{\sum_{i=1}^{r} w_i}\,, & \text{if } \|\mathbf{ISP}_i - \mathbf{P}\| \neq 0 \ \text{ for any } i, \\ \langle \sigma_t \rangle_i\,, & \text{if } \|\mathbf{ISP}_i - \mathbf{P}\| = 0 \ \text{ for some } i, \end{cases}$$

where the weights are

$$w_i = \frac{1}{\|\mathbf{ISP}_i - \mathbf{P}\|^p}\,,$$

p is a parameter (set to 2 in our experiments presented in Section 5), and $\{\mathbf{ISP}_i\}_{i=1}^{r}$ are the r nearest neighbors of point $\mathbf{P}$ (equal to 4 for quadrangular surface meshes, for instance).

The interpolation scheme interpolates *only* this averaged total cross section at the incident point of an IS of an IV. Neither the flux nor the number of mean-free paths are interpolated. The average cross section is much more slowly varying ray-to-ray than these quantities, and thus its interpolation introduces less error. The number of mean-free paths and the angular flux at the incident point can be reconstructed from Eq. (11) and known source information.

The interpolation scheme assumes that the averaged total cross section between the source exit point and any spatial point on the ray is the same for all directions in the entire cone of solid angle associated with the ray. This is not always perfectly true, but is exact if the material covered by the cone is homogeneous, for instance.

3.2.2. Ray Tracing Dependencies

The interpolation process described above is the basis to propagating rays throughout the entire domain from each SP. We further elaborate on the example shown in Figure 3 by adding more IVs to describe in more detail the process of tracing rays from a given spatial point (source volume exit or point on an IS) to an ISP, including the inherent dependencies.

As in Figure 3, rays are first propagated to the outgoing ISPs of IV_1 (which contains the SP). Refer now to Figure 4. First, rays are spawned from the SP to all ISPs on the ISs that define IV_1 and the appropriate information stored on each IS (source strengths and averaged cross sections to the single SP). Next, backward traces are initiated from the outgoing ISPs of IV_2 and IV_3 in the direction of the SP, which intersect at $\mathrm{IS}_{1,2}$ and $\mathrm{IS}_{1,3}$, respectively. $\mathrm{IS}_{n,m}$ denotes the interpolation surface separating interpolation volumes n and m. These tasks are independent. Once an intersection is found for a backward trace, a forward trace is initiated to imitate a ray from the SP to each outgoing ISP (while actually starting some distance away from the SP on either $\mathrm{IS}_{1,2}$ or $\mathrm{IS}_{1,3}$ in this case). Lastly, once tracing is completed to $\mathrm{IS}_{2,4}$ and $\mathrm{IS}_{3,4}$, backward traces are again initiated from the outgoing ISs of IV_4 towards the SP, and forward rays then generated from the intersecting points on $\mathrm{IS}_{2,4}$ and $\mathrm{IS}_{3,4}$. This process is illustrated in a task dependence graph in Figure 5.

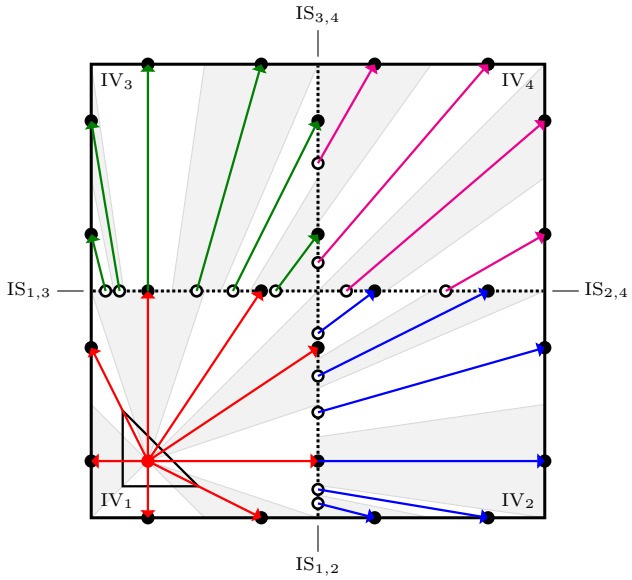


Figure 4: Two-dimensional example of ray propagation across multiple IVs for a single SP. Each filled point is an ISP and each point filled partially with white is a point where the average cross section at the given IS is interpolated. The shading between each ray distinguishes the cone of solid angle between rays. The internal ISs are labeled by the IVs that they define. For example, $IS_{1,3}$ is the IS that defines a surface on both IV_1 and IV_3. The boundary ISs are not labeled.

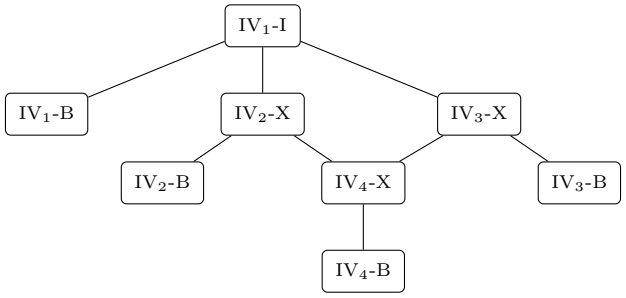


Figure 5: A graph that represents the expansion of the tasks required for the single source case in Figure Figure 4. There are three types of tasks: Type "I" (internal task) is the work an IV does for internal source points. This task includes tracing rays to all the ISPs on the outgoing boundaries of the IV. Type "X" (external task) is the work an IV does for a source point originating in a different IV. Type "B" (boundary task) writes the uncollided flux records for ISPs on global boundaries.

3.3. Summary of the Ray-Tracing Algorithm

To summarize, our algorithm for ray-tracing in unstructured, distributed grids consists of a hierarchy of tasks with different levels of parallelism:

1. For each source volume (i.e., a polyhedral cell containing a non-zero source), the source volume is decomposed into tetrahedral subcells, and the centroid of each such tetrahedron is labeled as a source point (SP). Contributions from each SP can be computed independently and thus the high-level ray-tracing tasks from SP's are embarrassingly parallel.
2. Each task spawned from a given SP is further divided into ray-tracing to all of the interpolation surfaces (ISs). While tracing to each IS for an interpolation volume (IV) containing the SP can be done in parallel, dependencies between the inflow and outflow ISs must be respected when propagating the rays to the downstream IVs. Once the inflow ISs of an IV have been traced to, tracing to each outflow IS of that IV can again proceed in parallel.
3. On the next level (ray-tracing to a given IS), ray-tracing to all of the interpolation surface points (ISPs) of a given IS is another embarrassingly parallel operation (all of these rays are independent).
4. Finally, for a single ray (the lowest-level task in our algorithm), the sequential nature of the uncollided particle propagation must be respected and thus mesh cells in the path of the ray have to be traversed serially (in both the forward and backward trace phases).

In the next Section, we describe the mathematical derivations related to ray tracing between an SP and an ISP or between two ISPs. Recall that, even though we have labeled these quantities as "points", we actually take into account the source volume for each SP, the area surrounding each ISP, and the "visible area" (solid angle cone) associated with each (SP,ISP) pair.

4. Mathematical Derivations

Here, we seek a representation of the uncollided angular flux as an expansion in spatial basis functions in each polyhedral cell. Furthermore, we assume that the only angular-distribution information needed inside the computational domain is a set of spherical-harmonics moments – exactly those needed to generate the first- and last-collision source. Having selected our space-angle basis for the uncollided flux, we seek the space-angle coefficients via a least-square fit method, with the requirement that the only information that can be used is the one we have obtained along the rays that the algorithm generates.

The expansion for the g-th group and n'-th moment of the uncollided flux in a single cell K is

$$\psi_g^{u,K}(\boldsymbol{r},\boldsymbol{\Omega}) = \sum_S \sum_{n'} C_{n'} \sum_{j=1}^{N_{V_K}} \phi_{j,g,n'}^{K,S} b_j^K(\boldsymbol{r})\, Y_{n'}(\boldsymbol{\Omega})\,, \tag{12}$$

where

$C_{n'} \equiv$ a normalization constant for the n'-th spherical-harmonics function,

$N_{V_K} \equiv$ the number of vertices of cell K,

$\phi_{j,g,n'}^{K,S} \equiv$ the coefficient of the j-th spatial basis function, for the n'-th spherical-harmonics function, for the g-th group, due to the S-th source point, to be calculated,

$b_j^K(\boldsymbol{r}) \equiv$ the j-th spatial basis function,

$Y_{n'}(\boldsymbol{\Omega}) \equiv$ the n'-th spherical-harmonics function.

We use a single-integer indexing in the spherical moment functions to represent their order and degree, for notational simplicity.

Consider the following functional which is the difference, in cell K, of the uncollided flux due to a source volume S and its finite-element representation:

$$\text{functional} = \int_{V_K} \mathrm{d}^3 r \int_{4\pi} \mathrm{d}\Omega \left[\psi_g^{u,K,S}(\boldsymbol{r},\boldsymbol{\Omega}) - \sum_{n'} C_{n'} \sum_{j=1}^{N_{V_K}} \phi_{j,g,n'}^{K,S} b_j^K(\boldsymbol{r}) Y_{n'}(\boldsymbol{\Omega}) \right]^2 \tag{13}$$

where V_K is the volume of cell K. We use a least-square process to minimize that functional and determine the expansion coefficients $\{\phi_{j,g,n'}^{K,S}\}$ of the finite-element representation. Minimization of this functional yields :

$$0 = 2\int_{V_K} \mathrm{d}^3 r \int_{4\pi} \mathrm{d}\Omega\, b_i^K(\boldsymbol{r}) Y_n(\boldsymbol{\Omega}) \left[\psi_g^{u,K,S}(\boldsymbol{r},\boldsymbol{\Omega}) - \sum_{n'} C_{n'} \sum_{j=1}^{N_{V_K}} \phi_{j,g,n'}^{K,S} b_j^K(\boldsymbol{r}) Y_{n'}(\boldsymbol{\Omega}) \right] ,$$
$$i = 1, \ldots, N_{V_K} , \quad \forall n , \tag{14}$$

which can be arranged into the following matrix equation :

$$\mathbf{A}^{K,S} \boldsymbol{\phi}_{n,g}^{K,S} = \mathbf{y}_{n,g}^{K,S} . \tag{15}$$

The $\mathbf{A}$ matrix is a mass-matrix and therefore independent of the angular moment n. The size of the system is $N_{V_K} \times N_{V_K}$ where N_{V_K} is the number of vertices in cell K. The elements of $\mathbf{A}$ and $\mathbf{y}$ are :

$$\left[\mathbf{A}^{K,S}\right]_{i,j} = \int_{V_K} \mathrm{d}^3 r\, b_i^K(\boldsymbol{r})\, b_j^K(\boldsymbol{r}) \tag{16a}$$

$$\left[\mathbf{y}_{n,g}^{K,S}\right]_i = \int_{V_K} \mathrm{d}^3 r\, b_i^K(\boldsymbol{r}) \int_{4\pi} \mathrm{d}\Omega\, Y_n(\boldsymbol{\Omega})\, \psi_g^{u,K,S}(\boldsymbol{r},\boldsymbol{\Omega}) \tag{16b}$$

Note that we used the orthogonality property of the spherical harmonic functions, $\int_{4\pi} \mathrm{d}\Omega Y_{n'}(\boldsymbol{\Omega}) Y_n(\boldsymbol{\Omega}) = C_n^{-1}\delta_{nn'}$, in Eq. (16a).

Recall that our polyhedral cells are subdivided into tetrahedra; for instance, a cubic cell containing a source is actually treated as 24 tetrahedra, thus as 24 source volumes, and there are therefore 24 SPs in such a cell. For increased accuracy, each source tetrahedron can be further subdivided into 12 sub-tetrahedra, as described in Section 5.1.2. The space-angle moments of the uncollided flux in a given cell K are obtained simply by accumulating the contributions from all source volumes:

$$\boldsymbol{\phi}_{n,g}^{K} = \sum_{S} \boldsymbol{\phi}_{n,g}^{K,S} . \tag{17}$$

The remainder of this section outlines the integration techniques utilized in both space and angle to compute the entries of $\mathbf{A}$ and $\mathbf{y}$ appearing in Eqs. (16). We will consider one given source volume and one energy group and drop the indices S and g when they are not needed to prevent ambiguity.

4.1. Ray Tracing of a Given Ray

We consider a ray, indexed by m, originating from the source point (SP) – the geometric centroid of the given source volume – and ending at the given interpolation surface point (ISP). The ISP represents a fraction of the interpolation surface and has therefore an associated area. Hence, each ray is defined by an (SP,ISP) pair and has an associated solid angle $\Delta\omega_m$; we use the formula of van Oosterom and Strackee [23] to compute that solid angle for an arbitrary planar face and point.

The ray tracing procedure determines the various polyhedral cells traversed from a given SP to a given ISP. Tracing the ray through each cell involves traversing some of its underlying tetrahedra. The entry/exit points of ray m for any cell K (and its tetrahedra $T \in K$) are determined by the algorithm described in Appendix A.

In the following subsections, we derive the formulas describing the uncollided flux attenuation outside of the source volume. The uncollided flux attenuation within the source volume will be handled by a separate method described later in Section 4.3.4.

4.2. Computing $\mathbf{A}$

Given the fact that we rely only on ray-tracing quantities to compute $\mathbf{y}$, we wish to use the same data (and make the same approximation) when calculating $\mathbf{A}$. Therefore, we do not integrate the mass matrix $\mathbf{A}$ using standard finite-element techniques. Rather, we introduce the substitution $\mathrm{d}^3r = s^2\,\mathrm{d}s\,\mathrm{d}\omega$ for every ray m that intersects cell K (and its underlying tetrahedra, denoted hereafter by T) :

$$\left[\mathbf{A}^K\right]_{i,j} = \sum_{m\in K} \Delta\omega_m \sum_{T\in K} \int_{s_{in}^{T,m}}^{s_{out}^{T,m}} \mathrm{d}s\, s^2 b_i(\boldsymbol{r}(s)) b_j(\boldsymbol{r}(s)) \;, \tag{18}$$

where b_i and b_j are the PWLD basis functions in cell K defined in Eq. (10) and evaluated on each tetrahedron T (for brevity, we will from now on suppress the cell index at the basis functions in equations related to a given cell K). An

example of the spatial integration of two rays through a single cell can be seen in Figure 6. It is easier to represent that process using 2D graphics. Here, a quadrilateral cell has been split into 4 triangles (the 2D analog of what happens in 3D using PWLD). Ray #1 (in red) crosses 3 of the 4 underlying triangles of the quadrilateral cell, while ray #2 only crosses 2 triangles. The entry/exit points in each triangular subcell for both rays are shown. The solid angle cones are represented using the dotted lines. The shaded zones associated with each ray represent the approximated volume used for the cell integration (in our 2D example, it is the approximated surface).

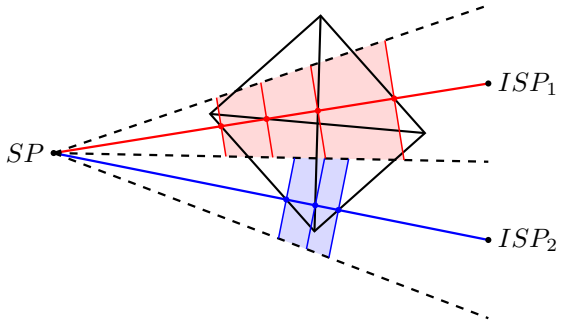


Figure 6: A 2D example of the spatial integration carried out for two different rays within a single cell.

In Eq. (18), we now only need to compute a 1D integral along the segment of the ray intersecting a given cell, from the entry to the exit points. For this, we simply use a 1D Gauss-Legendre quadrature to compute the integral in s :

$$\left[\mathbf{A}^K\right]_{i,j} = \sum_{m \in K} \Delta\omega_m \sum_{T \in K} \frac{|s_{out}^{T,m} - s_{in}^{T,m}|}{2} \sum_q w_q s_q^2 b_i(\boldsymbol{r}_q) b_j(\boldsymbol{r}_q) \ ,$$

where $|s_{out}^{T,m} - s_{in}^{T,m}|$ is the length of the m-th ray traveled inside in tetrahedron T, $\boldsymbol{r}_q$ are the 3D coordinates of the q-th quadrature point on the ray and w_q is the associated quadrature weight ($\sum_q w_q = 2$).

Finally, we need to evaluate the basis function values at any location along a given ray. Because we are employing piecewise linear basis functions over a polyhedral cell (i.e., linear basis function over each of its tetrahedral subcells),

we know that $b_i(\boldsymbol{r})$ varies linearly from its entry point at $\boldsymbol{r}_{in}$ to its exit point at $\boldsymbol{r}_{out}$. Hence, we only need to evaluate the basis functions at these two points and its value anywhere along the ray is simply given by linear interpolation

$$b_i(\boldsymbol{r}_q) = \frac{b_i(\boldsymbol{r}_{out}) + b_i(\boldsymbol{r}_{in})}{2} + \xi_q \frac{b_i(\boldsymbol{r}_{out}) - b_i(\boldsymbol{r}_{in})}{2}\,, \tag{19}$$

with $\xi_q \in (-1, 1)$ being the Gauss-Legendre quadrature abscissae.

4.3. Computing $\mathbf{y}$

We introduce the same change of variables for the spatial integration in the right-hand side term $\mathbf{y}$:

$$\left[\mathbf{y}_n^K\right]_i = \sum_{m\in K} \Delta\omega_m \sum_{T\in K} \int_{s_{in}^{T,m}}^{s_{out}^{T,m}} \mathrm{d}s\, s^2 b_i(\boldsymbol{r}(s)) \int_{4\pi} \mathrm{d}\Omega\, Y_n(\boldsymbol{\Omega}) \Psi^u(\boldsymbol{r}(s), \boldsymbol{\Omega}(s)) \tag{20}$$

The integrand in the s variable is the product of s^2 times a piecewise linear basis function $b_i(\boldsymbol{r}(s))$ times a solid-angle integral of $Y_n \times$ the uncollided flux.

The solid-angle integration range over which Ψ^u is non-zero is defined by the projected source-volume area, which can change significantly with s (but relatively smoothly) for s close to the source. For s far from the source the range decreases like $1/s^2$, which "cancels" the s^2 factor from the change of variables. In this far-from-source limit, the product of s^2 times the angular integral changes only because of within-cell attenuation, which is exponentially decreasing in s. We ensure that our quadrature integration becomes exact in that limit, as we describe next.

Closer to the source the product of s^2 times the angular integral changes in a more complicated way, but it is a smooth function of s except at values of s at which the number of visible surfaces of the source volume changes. For example, for small s, close to one surface of a source volume, only that surface will be visible, but as s increases along a ray it is possible that a second and even third surface may become visible. At such transition points, s^2 times the angular integral will have a discontinuous derivative.

4.3.1. Spatial Integration

The integral in s appearing in Eq. (20) is split into several smaller segments, centered around the abscissae of a 1D Gauss-Legendre quadrature, indexed by q :

$$\int_{s_{in}^{T,m}}^{s_{out}^{T,m}} \mathrm{d}s \left[s^2 b_i(\boldsymbol{r}(s)) \int_{4\pi} \mathrm{d}\Omega\, Y_n(\boldsymbol{\Omega}) \Psi^u(\boldsymbol{r}(s), \boldsymbol{\Omega}(s)) \right] = \sum_q \int_{s_{q-1/2}}^{s_{q+1/2}} \mathrm{d}s \left[s^2 b_i(\boldsymbol{r}(s)) \int_{4\pi} \mathrm{d}\Omega\, Y_n(\boldsymbol{\Omega}) \Psi^u(\boldsymbol{r}(s), \boldsymbol{\Omega}(s)) \right] .$$

The terms in brackets are identical on both sides. However, because of the decaying exponential behavior of the exact solution far from the source, we proceed as follows to make sure that limit is reproduced numerically :

$$\begin{aligned} I_{q,n}^K &= \int_{s_{q-1/2}}^{s_{q+1/2}} ds\, s^2 \int_{4\pi} d\Omega\, Y_n(\boldsymbol{\Omega}) \Psi^u(\boldsymbol{r}, \boldsymbol{\Omega}) \,, \\ &= s_q^2 \int_{4\pi} d\Omega\, Y_n(\boldsymbol{\Omega}) \Psi^u(\boldsymbol{r}_q, \boldsymbol{\Omega}) \int_{s_{q-1/2}}^{s_{q+1/2}} ds \frac{s^2 \int_{4\pi} d\Omega\, Y_n(\boldsymbol{\Omega}) \Psi^u(\boldsymbol{r}, \boldsymbol{\Omega})}{s_q^2 \int_{4\pi} d\Omega\, Y_n(\boldsymbol{\Omega}) \Psi^u(\boldsymbol{r}_q, \boldsymbol{\Omega})} \,. \end{aligned} \quad (21)$$

The above equality is exact. Next, we make the approximation, which is accurate in the large-s limit (as $s^2 \approx s_q^2$ in that situation), that the ratio in the integrand is an exponential :

$$\frac{s^2 \int_{4\pi} d\Omega\, Y_n(\boldsymbol{\Omega}) \Psi^u(\boldsymbol{r}, \boldsymbol{\Omega})}{s_q^2 \int_{4\pi} d\Omega\, Y_n(\boldsymbol{\Omega}) \Psi^u(\boldsymbol{r}_q, \boldsymbol{\Omega})} \simeq \exp\left(-\sigma_t^K (s - s_q)\right)$$

Finally,

$$I_{q,n}^K = \beta_q^K s_q^2 \int_{4\pi} d\Omega\, Y_n(\boldsymbol{\Omega}) \Psi^u(\boldsymbol{r}_q, \boldsymbol{\Omega}) = \beta_q^K s_q^2 J_{q,n}^K \,,$$

where

$$\begin{aligned} \beta_q^K &= \int_{s_{q-1/2}}^{s_{q+1/2}} ds \exp\left(-\sigma_t^K (s - s_q)\right) \\ &= \frac{\exp\left(-\sigma_t^K (s_{q-1/2} - s_q)\right) - \exp\left(-\sigma_t^K (s_{q+1/2} - s_q)\right)}{\sigma_t^K} \,. \end{aligned}$$

(note that, as $\sigma_t^K \to 0$, $\beta_q^K \to s_{q+1/2} - s_{q-1/2}$, as expected). The angular integral, $J_{q,n}^K$, is discussed in the next Section. So far, we have :

$$\left[\mathbf{y}_n^K\right]_i = \sum_{m \in K} \Delta\omega_m \sum_{T \in K} \sum_q b_i(\boldsymbol{r}_q) I_{q,n}^K \,.$$

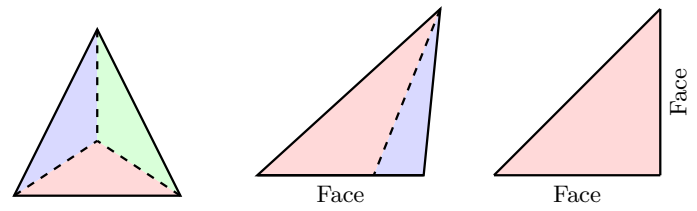
Note, there are no spatial quadrature weights (w_q) in this formula (they have been absorbed in the computation of β_q^K). See Eq. (19) to compute the basis function at quadrature point, $b_i(\boldsymbol{r}_q)$.

4.3.2. Angular Integration and Subcone Views

We must now approximate the angular integral :

$$J_{q,n}^K = \int_{4\pi} d\Omega \, Y_n(\boldsymbol{\Omega}) \Psi^u(\boldsymbol{r}_q, \boldsymbol{\Omega}) \ .$$

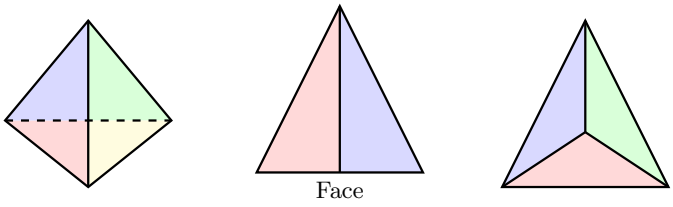
The solid angle over which we need to carry out this integration depends on the spatial position (spatial quadrature point along the segment). Indeed, one can think of it as follows: from quadrature point $\boldsymbol{r}_q$, one looks back at the source volume and the visible area depends on the position of the observer along the ray segment. Let us thus consider $\boldsymbol{r}_q$ as the viewpoint from which we view the source volume. From any such viewpoint, the view of the tetrahedral source volume must be one of the six shown in Figure 7, in which the colors represent separate solid angle subcones. Of the visible face(s) from the viewpoint, the faces are divided into subcones such that the thickness (depth) of source volume behind each subcone varies linearly. Each subcone can be further refined (e.g., multiple levels of refinement of the visible triangular surfaces into 4 smaller triangles) in order to converge the angular integration.



Case 1: One face visible and no faces aligned.

Case 2: One face visible and one face aligned.

Case 3: One face visible and two faces aligned.



Case 4: Two faces visible and no faces aligned.

Case 5: Two faces visible and one face aligned.

Case 6: Three faces visible.

Figure 7: The possible views of a tetrahedral source volume viewed from a point outside the volume. The colored planes represent the division of the faces (termed subcones) as seen by the viewpoint such that the thickness of source volume behind each subcone varies linearly.

The angular integral $J^K_{q,n}$ is thus split into integrals over the subcones (and possibly smaller subcones after refinement):

$$J^K_{q,n} = \sum_{sc} \sum_{\text{ref. levels } \ell} \Delta\omega_{sc,\ell}\, Y_n(\mathbf{\Omega}_{CQ}) \Psi^u_{sc,\ell}(\boldsymbol{r}_q, \mathbf{\Omega}_{CQ}) \tag{22}$$

where $\Delta\omega_{sc,\ell}$ is the solid angle (computed using [23]) for subcone sc (with refinement level ℓ). $\mathbf{\Omega}_{CQ}$ is the unit direction vector from the centroid C of the sub-subcone (subcone sc, refinement level ℓ) to the view point (quadrature point) Q.

4.3.3. Angular Flux Exiting from a Subcone of the Source Volume

The exiting angular flux from a sub-subcone (subcone sc, refinement level ℓ) is to be attenuated by the optical thickness between the sub-subcone centroid C and the quadrature point Q in the target cell[3]:

$$\tau_{CQ} = \int_{s=0}^{s=\|\boldsymbol{CQ}\|} \sigma_t(s)ds.$$

However, we know only the optical thickness for ray m exiting the source volume at E to point Q (the τ that we accumulate in our ray tracing), τ_{EQ}, and the target volume cross section, σ_t^K.

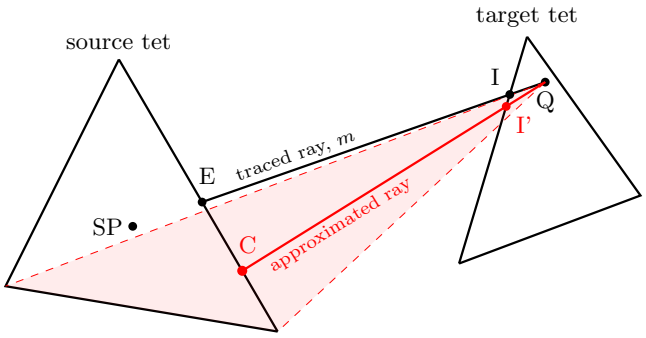


Figure 8: Illustration of the optical thickness approximation. The average cross section computed by tracing the primary ray m is used to approximate the optical thickness along the auxiliary ray (shown in red), in order to compute the angular integral contribution over the red-shaded sub-subcone associated with the auxiliary ray.

Consider the geometry configuration in Figure 8. We compute the average cross section, $\langle\sigma_t\rangle_{\text{outside}}$, in the region between the source volume at E and the target volume at I using τ_{EQ}, as

$$\langle\sigma_t\rangle_{\text{outside}} = \frac{\tau_{EI}}{\|\boldsymbol{EI}\|} = \frac{\tau_{EQ} - \sigma_t^K\|\boldsymbol{IQ}\|}{\|\boldsymbol{EI}\|}.$$

We then make the assumption that this average cross section is also valid between the points points C and I'. The optical thickness from the sub-subcone

[3] $\|\boldsymbol{XY}\|$ denotes the Euclidean distance between two points X and Y

centroid C and the quadrature point Q in the target cell is then approximated without the need for spawning an additional ray as

$$\tau_{CQ} = \langle \sigma_t \rangle_{\text{outside}} \|\boldsymbol{CI'}\| + \sigma_t^K \|\boldsymbol{I'Q}\|.$$

While this is exactly true in homogeneous zones or if the materials intersected by the two rays have the same optical thickness, it is only approximate in the general heterogeneous-medium case. The consequences of this approximation will be discussed in more detail in our numerical results (Section 5.2.3).

Now, we know the attenuation from sub-subcone to quadrature point:

$$\Psi^u_{sc,\ell}(\boldsymbol{r}_q, \boldsymbol{\Omega}_{CQ}) = \Psi^{exit}_{sc,\ell} \exp\left(-\tau_{CQ}\right), \tag{23}$$

and only need to determine the value of the uncollided flux exiting the sub-subcone, $\Psi^{exit}_{sc,\ell}$. This is given by

$$\Psi^{exit}_{sc,\ell} = S(\boldsymbol{\Omega}_{CQ}) \frac{1 - \exp\left(-\sigma_t^S L\right)}{\sigma_t^S}\ , \tag{24}$$

if the source strength (source rate density) S in the source volume and the total cross section σ_t^S in the source volume are spatially constant (otherwise, we employ a numerical quadrature). L is the distance traversed in the source volume from the point C (centroid of sub-subcone) in direction $-\boldsymbol{\Omega}_{CQ}$. Note that this expression remains valid in the (near-)void case, as

$$\lim_{\sigma_t^S \to 0} \Psi^{exit}_{sc,\ell} = S(\boldsymbol{\Omega}_{CQ}) L.$$

4.3.4. Source-Cell Contribution to the Uncollided Flux

So far, we have discussed the calculations of the uncollided flux (and its space-angle moments) in any target cell that is not the source cell emitting the radiation. In this section, we describe our method to compute the uncollided flux and its moments inside an arbitrarily shaped polyhedral source cell. As before, the polyhedral cell is decomposed into its underlying tetrahedral cells (using face midpoints and the cell centroid). Recall from Section 4.3.3 that for a given tetrahedron T containing a source of strength S, the angular flux due

to that source in direction $\mathbf{\Omega}$ at an exiting point $\boldsymbol{r}$ on an outgoing surface of the source volume is given by

$$\psi^T_{\text{exit}}(\boldsymbol{r},\mathbf{\Omega}) = \frac{S(\mathbf{\Omega})}{\sigma_t^S}(1-\exp(-\sigma_t^S L))\,,$$

where L is the chord length for that given direction $\mathbf{\Omega}$ and exiting point $\boldsymbol{r}$. Rather than computing the exiting angular flux for that chord, one can use the above expression to compute the chord-average angular flux:

$$\psi^T_{\text{chord}}(L,\mathbf{\Omega}) = \frac{1}{L}\int_0^L \mathrm{d}\ell \frac{S(\mathbf{\Omega})}{\sigma_t^S}\big(1-\exp(-\sigma_t^S\ell)\big) = \frac{S(\mathbf{\Omega})}{\sigma_t^S}\left(1-\frac{1-\exp(-\sigma_t^S L)}{\sigma_t^S L}\right).$$

Next, to obtain the cell-averaged angular flux in direction $\mathbf{\Omega}$, we need to integrate the previous expression over all outgoing surfaces. This is equivalent to integrating the chord-average expression over all possible chord lengths. For a tetrahedron, the distribution of chord lengths $p^T(L)$ is a linear distribution, increasing from a value of 0, reached at the largest chord length $L_{\max}$, to its maximum value, obtained for chord length values of 0:

$$p^T(L) = \frac{2}{L_{\max}}\left(1-\frac{L}{L_{\max}}\right) \qquad \text{for } 0 \le L \le L_{\max}\,.$$

Then, the tetrahedron-average angular flux is given by

$$\begin{aligned}\psi^T(\mathbf{\Omega}) &= \frac{\int_0^{L_{\max}} L\, p^T(L)\psi^T_{\text{chord}}(L,\mathbf{\Omega})\,\mathrm{d}L}{\int_0^{L_{\max}} L\, p^T(L)\,\mathrm{d}L}\,,\\ &= \frac{S(\mathbf{\Omega})}{\sigma_t^S}\left(\frac{6\tau_{\max}-3\tau_{\max}^2+\tau_{\max}^3-6+6\exp(-\tau_{\max})}{\tau_{\max}^3}\right), \end{aligned} \tag{25}$$

with $\tau_{\max} = \sigma_t^S L_{\max}$. For any direction $\mathbf{\Omega}$, the largest chord length in a tetrahedron is given by

$$L_{\max}(\mathbf{\Omega}) = 3\frac{V^T}{A_\perp(\mathbf{\Omega})}\,, \tag{26}$$

where $A_\perp(\mathbf{\Omega})$ is the projected (visible) area of the tetrahedron on a plane perpendicular to $\mathbf{\Omega}$. If we denote by $\boldsymbol{S}_i$ the surface vector of the i-th triangular face of T,

$$A_\perp(\mathbf{\Omega}) = \sum_{\substack{1\le i\le 4\\ \boldsymbol{S}_i\cdot\mathbf{\Omega}>0}} \boldsymbol{S}_i\cdot\mathbf{\Omega}\,.$$

Once the tetrahedron-average angular flux has been computed, it is straightforward to compute the angular moments ($\phi_n^{T,S}$) in given tetrahedron T with source S, using an angular quadrature formula; here, we re-use the angular discretization used in ray-tracing from the SP to the ISPs. Finally, to obtain the contribution from the source tetrahedron to the PWLD finite-element representation $\phi_n^{K,S}$ over the entire polyhedral cell, we use a simple least-square fitting, using the tetrahedron-averaged values.

4.4. Computing Point-wise Values of Uncollided Flux

In some applications (for instance when comparing with benchmark solutions), the values of the uncollided flux at specific points might be required. To compute these values, we trace the ray originating from each SP towards the specified target point using the same ray-tracing procedure described in Section 3). However, instead of accumulating the contributions to the finite element system $\mathbf{A}$, $\mathbf{y}$ cell by cell, we directly use Eqs. (22), (23) and (24), where $\boldsymbol{r}_q$ is now the coordinate of the target point, to compute the contribution of given source point to each angular moment of the uncollided flux at that point. Note that if the target point and the source point are in the same IV, there is no need to trace towards the ISPs.

If the specified target point is located in the source volume or on its boundary, we first define a set of points on each boundary face of the source volume that does not contain the target point. The same procedure used for refining the subcones for the computation of the angular integral $J_{q,n}^K$ is used here to obtain sufficient density of these boundary points (see Section 4.3.2). These points are thus the centroids of the non-overlapping sub-subcones that completely fill the portion of the solid angle from which the target point receives source particles. Finally, we use the distance from the target point to each of these source boundary points to determine the attenuated contribution to the uncollided flux at the target point. Denoting by $\boldsymbol{Q}$ the coordinates of the target point, we have

$$\phi_n(\boldsymbol{Q}) = \sum_{sc} \sum_{\text{ref. levels } \ell} \Delta\omega_{sc,\ell}\, Y_n(\boldsymbol{\Omega}_{CQ}) \frac{S(\boldsymbol{\Omega}_{CQ})}{\sigma_t^S} \big(1 - \exp(-\sigma_t^S \|\boldsymbol{CQ}\|)\big), \quad (27)$$

where $\Delta\omega_{sc,\ell}$ is the solid angle for subcone sc (refinement level ℓ) and $\mathbf{\Omega}_{CQ}$ is the unit direction vector from the centroid C of the sub-subcone (subcone sc, refinement level ℓ) to the target point Q.

5. Results

To assess the accuracy and effectiveness of our uncollided flux algorithm, we perform four tests. The first test case we present consists of two tetrahedral cells – source and target – placed in vacuum. This example demonstrates the accuracy of our method compared to traditional approaches in calculating the uncollided flux. The second test serves as numerical verification that the method is conservative. The third example illustrates one possible source of error introduced by our method when estimating the optical thickness required to compute the angular integrals and deal with the non-point source nature of the source volumes (see Section 4.3.3). The final test is one of the well-known Kobayashi benchmarks [9] developed to stress 3D transport production codes – the purely-absorbing case of a shield with nearly voided dog-leg duct and a localized source.

5.1. Testing Methodology

In the various examples, several quantities of interest are reported; these are either the cell-average uncollided scalar flux, the PWLD finite element representation of the uncollided scalar flux, or point values (note that the cell average uncollided flux is not the same quantity as the point-wise uncollided flux computed at the cell centroid). The reference solutions were obtained by running a 1-group MCNP calculation with a significant number of particle histories to reach a low level of statistical fluctuations. For the Kobayashi benchmark, we compared against the point-wise analytical solution provided in [9] as well.

In all tables with numerical results, read 1.23456E±07 as $1.23456 \times 10^{\pm 7}$.

5.1.1. Direct Quadrature Approximation

As mentioned in Section 2.2, another method to calculate the uncollided flux on arbitrary finite-element grids would consist in ray-tracing from each quadrature point in the source volumes to all quadrature points in every single mesh of the computational domain; see Eq. (9). In this alternate approach, the source volumes are treated as a set of point sources. We refer to this approach as the *direct quadrature approximation*. It is employed in numerous transport codes (see references cited in Section 1).

To compare our method with the direct quadrature approximation, a small code was written to compute the uncollided flux solution via tracing rays from quadrature points in a source region to quadrature points in a target region. When the source/target regions are cuboids, the direct quadrature approximation uses a tensor-product of the standard 1D Gauss-Legendre quadrature; for tetrahedral cells, the Xiao-Gimbutas tetrahedral quadratures [25] are used.

5.1.2. Algorithm Parameters

The performance of our uncollided algorithm is determined by five significant parameters: IS refinement level, subcone refinement level, the number of source volume refinement levels, the number of nearest ISPs used for interpolation on incoming ISs, and the spatial quadrature order. In our experiments presented below, we found the first three parameters to have the largest impact on the accuracy of the solution – we typically vary those and keep the latter two fixed as described below.

The IS refinement level (ℓ_{IS}) specifies the number of uniform recursive refinements applied to each cell face shared with an IS. A general polygonal face with N_{fv} vertices is refined by connecting the edge midpoints to the face centroid, which at the first level produces N_{fv} quadrilateral patches and then 4 child quadrilateral patches for each parent patch at the subsequent levels. The

total number of ISPs per a cell face which is a part of an IS is then

$$N_{\text{ISP}} = \begin{cases} 1\,, & \text{for } \ell_{\text{IS}} = 0, \\ N_{fv} * 4^{\ell_{\text{IS}}-1}\,, & \text{for } \ell_{\text{IS}} = 1, 2, \ldots. \end{cases}$$

The uniform refinement of the ISs is a simple generic way to generate the ISPs. However, more efficient strategies may be utilized for specific problems. For example, in a beam problem created from a singular anisotropic point source (encountered, for instance, in external beam radiotherapy applications), one may choose to increase the density of the ISPs in the direction of the beam along with appropriate cell refinement in order to capture the discontinuity of the flux in the cells that are cut by the edges of the beam. In our future work, we also plan to explore the possibilities of adaptive placement of the ISPs that would ensure sufficient density of rays for an arbitrary mesh refinement.

The subcone refinement level (ℓ_{SC}) specifies the number of uniform recursive refinements applied to the subcones used for computing the angular integral on the right hand side of the cell finite element system as described in Section 4.3.2. Since the subcones are always triangular, we use a simple subdivision of a parent subcone into 4 triangular sub-subcones at each level, obtained by connecting the edge midpoints. This leads to the following number of sub-subcones considered for each quadrature point and subcone visible on the source surface from that point:

$$N_{\text{SSC}} = 4^{\ell_{\text{SC}}}, \quad \ell_{\text{SC}} = 0, 1, \ldots$$

In certain cases, refining the subcones for angular integration and/or increasing the number of ISPs is not enough to effectively reduce the discretization error – an example will be given in Section 5.2.3. We found that refining the source tetrahedron is needed in these cases for convergence and implemented a recursive splitting procedure: at each level, the parent tetrahedron can be split into 12 tetrahedral subcells by connecting two adjacent vertices of the tetrahedron, face centroid and the tetrahedron centroid (for each face), starting with the original source cell at level 0. Note that this is the same procedure as that used to define the subcells for a tetrahedral cell for the PWLD discretization, see

Section 2.3. The effect of source volume refinement is demonstrated in Section 5.2.3; all other results were obtained using the original source volume.

See Section 3.2.1 for more information about the number of nearest ISPs used for interpolation; we set this value to 4 in all our experiments.

We refer back to Sections 4.2 and 4.3 for more information about the spatial 1D quadrature along a given ray. Note that the integrand representing the elements of the matrix $\mathbf{A}$ is a quartic function in s and can be computed exactly with a three-point Gauss-Legendre quadrature; our modified quadrature integrates exactly the elements in $\mathbf{y}$ in the limit of large s, but for points closer to the source, we need to use a sufficient quadrature order. We use 4 quadrature points in all our experiments to evaluate the spatial integrals along a ray.

5.2. Test Results

5.2.1. Tetrahedral Source to Tetrahedral Target

Consider a problem with a single tetrahedral source and single tetrahedral target. Both source and target volumes have the same total cross-section value (we test two situations, with either $\Sigma_t = 0.075$ cm^{-1} or $\Sigma_t = 3.75$ cm^{-1}, corresponding to the cell diameters of about 0.1 and 5 mean free paths, respectively). The source rate density is such that the source volume emits 1 n/s and the material region between the source and the target is void. The placement of the source and target is given in Table 1 and shown in Figure 9. Coordinates of the source and target vertices are given in Table 1.

Table 1: Vertex coordinates (cm) for the tetrahedral-source-to-tetrahedral-target test case.

Vertex	Source	Target
1	$(\frac{5}{2}, \frac{5\sqrt{3}}{2}, -\frac{1}{3})$	$(1, 0, -\frac{1}{3})$
2	$(1, 2\sqrt{3}, -\frac{1}{3})$	$(-\frac{1}{2}, \frac{\sqrt{3}}{2}, -\frac{1}{3})$
3	$(\frac{5}{2}, \frac{3\sqrt{3}}{2}, -\frac{1}{3})$	$(-\frac{1}{2}, -\frac{\sqrt{3}}{2}, -\frac{1}{3})$
4	$(2, 2\sqrt{3}, 1)$	$(0, 0, 1)$

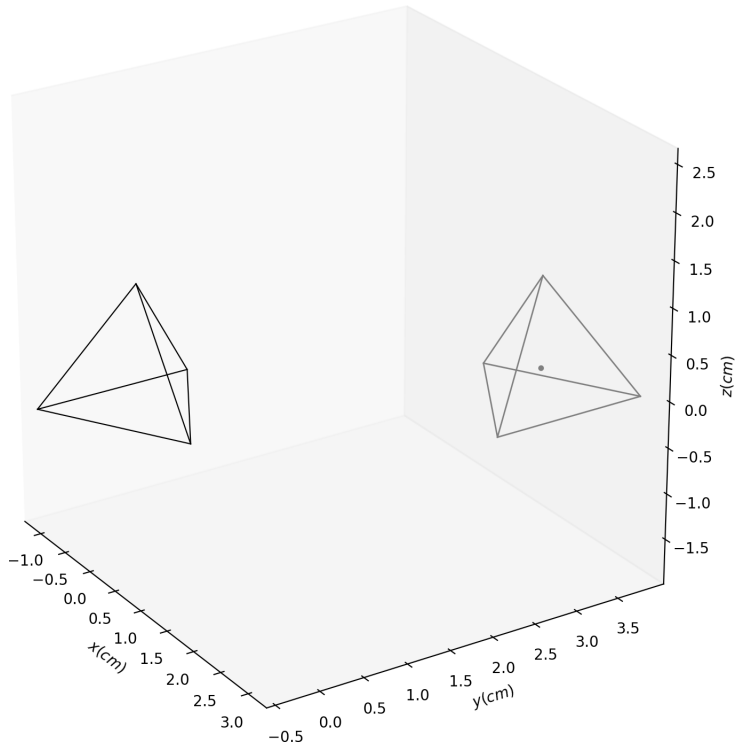


Figure 9: Visual rendition for the tetrahedral-source-to-tetrahedral-target case.

Table 2 provides the comparison of cell average values of uncollided flux for increasingly refined discretization parameters. For our method, we simultaneously increased the number of IS and subcone refinements, while for the direct quadrature method, we simultaneously increased the quadrature order in the target and the source. Note that in this simple geometry with a single target cell, the "interpolation surfaces" are the faces of the cell itself and no interpolation of average cross-sections is actually performed. In order to demonstrate the merits of our new algorithm, we compare the accuracy of the two methods for a given number of rays traced, corresponding to a given set of discretization parameters. For the direct quadrature method, the number of rays traced is the product of the number of the target quadrature points and the number of the source quadrature points.

Compared to the direct quadrature method, there is more work done per

ray with our method as we need to also perform the subcone refinement and accumulate the angular integral over the sub-subcones. The purposes of this test case was to assess the accuracy of our proposed method. Recall that, in full-scale calculations with multiple IVs, our method does not require the explicit ray tracing for a source volume to every other cell in the computational domain; this becomes an significant advantage for larger problems.

One of the shortcomings of the direct quadrature methods is that in optically thick source cells, most of the radiation leaving the source is emitted by the rim of the source cell and high-order quadrature is typically required to ensure some quadrature points are located in the rim. Our method handles this case naturally by employing the analytical solution for the uncollided flux exiting from the source volume. The results for the optically thick case ($\Sigma_t = 3.75$ cm^{-1}) shown in Table 2 demonstrate the superior accuracy of the new method, especially for low numbers of rays traced.

Table 2: Comparison of the average flux in the target cell for the two total cross-section values.

$\Sigma_t^\dagger$	MCNP		Our method			Direct quadrature		
(cm^{-1})	Flux (cm^{-2}s^{-1})	Error* (%)	IS/SC ref.	Rays	Rel. error (%)	Order	Rays	Rel. error (%)
0.075	4.76202E-03	0.01	0	3	8.49	1	1	−1.27
			1	9	2.62	2	16	−0.18
			2	36	0.97	3	36	0.00
			3	144	0.40	4	121	0.00
			4	576	0.17	5	196	0.00
3.750	8.28621E-04	0.02	0	3	26.44	1	1	−63.95
			1	9	4.03	2	16	−3.85
			2	36	1.39	3	36	−3.38
			3	144	0.98	4	121	−0.30
			4	576	0.59	5	196	−0.48

†Target and source cross section. Vacuum exists between the target and source.

*MCNP errors quoted at 2σ.

Finally, we used a modification of the two-tetrahedra configuration to perform a simple check of the calculation of higher order angular moments. The source tetrahedron was moved 1,000 cm away from the target and the cross sections in both cells were set to 0.0 cm^{-1} to simulate a point source in vacuum.

We used 1 level of IS refinement and varied the subcone refinement. For each subcone refinement level, we computed the relative errors of the 16 uncollided flux moments (P_3 anisotropy) with respect to an exact analytic solution. The maximum of these errors for 0, 1 and 2 levels of subcone refinement, respectively, was 1.3×10^{-3}, 3.3×10^{-4} and 1.8×10^{-5}, respectively, verifying the correctness of the higher angular moments calculation.

5.2.2. Particle Conservation

No special provisions have been made in order to ensure the method is conservative. However, we performed two simple experiments to numerically assess the degree of particle conservation. The conservation statement is evaluated as follows:

$$\sum_S \left(\sum_q \delta J_q^S A_q + \sum_K \sigma_t^K \phi_0^{K,S} V_K \right) = \widehat{S} \tag{28}$$

where the sums on the left side are, from left to right, over all sources, over the subset of ISPs lying on outer boundary surfaces of the domain (which we call "global boundary surface points", or GBSP), and over all cells in the domain. V_K is the volume of cell K, A_q is the fractional area of the boundary surface associated with the GBSP with index q and $\widehat{S}$ is the total particle source rate (s^{-1}). The leakage term, δJ_q^S, is evaluated as in Eq. (22) (or (27) if the GBSP lies on a surface of a source volume but with $Y_n(\mathbf{\Omega}_{CQ})$ replaced by $\mathbf{\Omega}_{CQ}\cdot\boldsymbol{n}_Q$). These quantities are easily computable once the ray-tracing from the source to the GBSP has been completed, as described in Sec. 4.3.3.

The first test case consists of a cubic cell, 1 mean free path thick, with a single source point ($\widehat{S} = 1.0$ s^{-1}). The second experiment consists of two adjacent cubic cells, 1 and 1.5 mean free paths thick, respectively, with 4 source points in one of the cells ($\widehat{S}_i = 0.25$ s^{-1} so that the total source rate is again 1.0 s^{-1}); see Fig. 10. Two interpolation volumes have been used to incorporate the interpolation procedure in the evaluation.

Table 3 shows the convergence of the sum of the leakage and absorption rates (left-hand sides of Eq. (28)) as the number of GBSPs is increased. The subcone refinement level for angular integration was fixed to 2 for both cases.

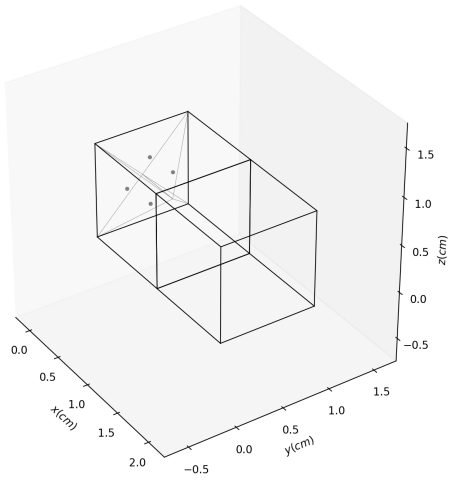


Figure 10: Geometry of the conservation tests. The source point centers are located at $[\frac{1}{8}, \frac{1}{4}, \frac{1}{2}]$, $[\frac{1}{8}, \frac{1}{2}, \frac{3}{4}]$, $[\frac{1}{8}, \frac{3}{4}, \frac{1}{2}]$, $[\frac{1}{8}, \frac{1}{2}, \frac{1}{4}]$.

In the second case, we note that the number of rays traced is slightly higher than if only one IV was used (as we explicitly trace rays to the ISPs lying at the interface between the two cells) – if only one IV is used, the sum of the balance terms becomes almost equal to the first case.

Table 3: Global particle conservation

	Case 1 (1 cell, 1 SP, 1 IV)			Case 2 (2 cells, 4 SPs, 2 IVs)		
IS ref.	Leakage	Absorption	Sum	Leakage	Absorption	Sum
0	0.8205	0.3119	1.1324	0.7892	0.3230	1.1123
1	0.7286	0.3142	1.0428	0.7071	0.3290	1.0361
2	0.6910	0.3213	1.0123	0.6712	0.3360	1.0072
3	0.6907	0.3146	1.0052	0.6713	0.3293	1.0005

5.2.3. Error due to Optical Thickness Approximation (the "τ-effect")

Recall from Section 4.3.3 that, in order to compute the angular flux moments at a spatial quadrature point Q in the target cell, we approximate (i.e., without launching a new ray) the optical thickness between Q and the source subcone using :

$$\tau_{CQ} = \tau_{EI} \frac{\|\boldsymbol{CI'}\|}{\|\boldsymbol{EI}\|} + \sigma_t^K \|\boldsymbol{I'Q}\| \tag{29}$$

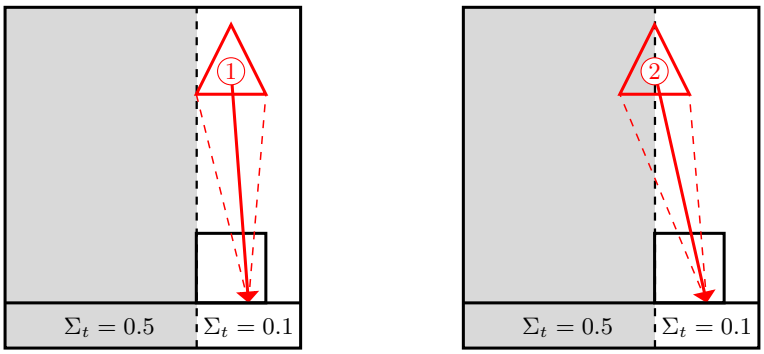
where C is the point through which the ray exits the source (centroid of a solid-angle subcone), E is the source exit point, I is the target cell entry point on the primary (traced) ray and I' is the target cell entry point on the approximated ray. The optical thickness (number of mean free paths) computed using the above formula is then used in Eqs. (23) and (22) to compute the angular integral $J^K_{q,n}$.

Consider now a problem with one cubical target cell with $\Sigma_t = 0.1$ cm^{-1}, one tetrahedral source cell emitting 1 n/s and having $\Sigma_t = 0.1$ cm^{-1} and two material regions between the cells ($\Sigma_t = 0.1$ cm^{-1} and $\Sigma_t = 0.5$ cm^{-1}, respectively). Figure 11 shows the xz-view of this problem and four different positions of the source (labeled 1-4 in the figure). The target cell is a cube of side 1 cm in length in which the front bottom left corner is at the origin. The centroid of the source tetrahedron in position 1 is at coordinate $(0.5, 0.375, 10.25)$ and its top vertex at coordinate $(0.5, 0.5, 11.0)$ (see Table 4 for the remaining source positions).

Table 4: Setup and reference solution for the "τ"-effect problem. Offset is the horizontal distance between the source centroid and the material interface.

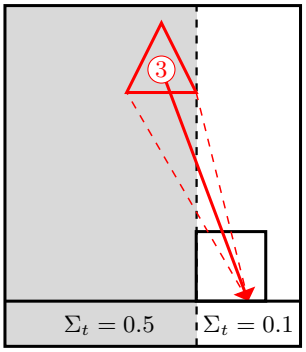
Source pos.	1	2	3	4
Offset (cm)	+0.5	0.0	-0.5	-9.5
MCNP Flux ($\mathrm{cm}^{-2}\mathrm{s}^{-1}$)	3.16705E-04	2.42363E-04	5.38449E-05	5.45520E-07
MCNP Error† (%)	0.01	0.01	0.01	0.04

†MCNP errors quoted at 2σ.

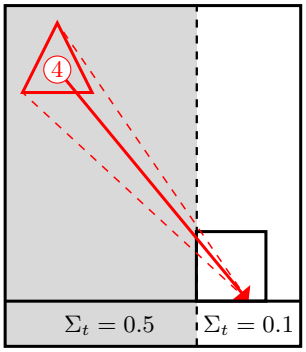


(a) Source position 1

(b) Source position 2



(c) Source position 3



(d) Source position 4

Figure 11: Illustration of the "τ"-effect. The regions with different cross-sections are distinguished by different shading; note that the source total cross-section is the same ($\Sigma_t = 0.1$ cm^{-1}) for all 4 source positions (the source region is shown as transparent without any shading) and that the figure is not to scale. The dashed lines represent the visible cone of solid angle of the source tetrahedron as seen by the point on the surface of the target cell.

For source position 1, the source particles that contribute to the uncollided flux in the target cell traverse only a homogeneous medium (the white region). Consider the single ray shown in Figure 11a that begins at the centroid of the source. The average cross section in the visible cone of solid angle of the source (within the dashed lines) is constant. Given this homogeneity, this single ray can approximate exactly using Eq. (29) the optical thickness between any two points in the cone. The uncollided solution for source position 1 converges to the reference solution obtained by MCNP as the subcone refinement level is

being increased (see Table 5); as expected, the refinement of the source cell into tetrahedral subcells, as described in Section 5.1.2, can be used to further improve the accuracy (see Table 6).

Three other source positions are chosen to investigate the error due to the approximation made in Eq. (29). For source positions 2-4, the source particles that contribute to the uncollided flux in the target cell traverse a heterogeneous medium. Consider now the single ray traced from the centroid of source position 2, shown in Figure 11b. This ray is traverses only the unshaded material region while, in reality, uncollided particles from that source position will sometimes traverse both material zones. In this situation, the approximation in Eq. (29) is prone to error in approximating the optical thickness across the material boundary, leading to significant error in computing the angular integral in Eq. (22). Moreover, splitting the cone into multiple subcones does not improve accuracy – while the contributions to J_q^n from the "right" subcones are added correctly, the contributions from the "left" subcones suffer from the same problem. Likewise, using more ISPs does not sufficiently reduce the error, as can be seen by comparing the results for 2 and 3 levels of IS refinement in Table 5. The same type of error is also manifested with source positions 3 and 4, but to a lesser degree as the approximation from Eq. (29) is more accurate.

In such configurations (such as source position 2), splitting of the source cell into subcells is the only effective way to obtain accurate results, as it effectively diminishes the differences between the primary and auxiliary rays. Table 6 shows the effect of using the original source cell without splitting (columns labeled "1SP"), with one level of splitting ("12SP") and of two levels of splitting ("144SP"), respectively (each refinement of a tetrahedral region leads to 12 smaller tetrahedra)– for any given level of subcone refinement, the relative error decreases adequately as the number of source subcells increases.

Finally, this example is simple enough to allow for an exact computation of the optical thickness along each auxiliary rays. The errors with respect to the MCNP solution when using this exact expression instead of Eq. (29) are presented in the column "1SPEx" in Table 6 and confirm that the failure of

Table 5: Comparison of the average flux in target cell for the 4 source positions shown in Figure 11, 4 different subcone refinement levels (SC ref.), IS refinement levels 2 (IS2) and 3 (IS3) and no source cell refinement.

Source pos.	Our method (1SP)					Direct quadrature		
	SC ref.	Rays		Rel. error (%)		Order	Rays	Rel. error (%)
		IS2	IS3	IS2	IS3			
1	0	80	320	−0.93	−0.93	1	1	−0.33
	1			−0.22	−0.22	2	32	0.01
	2			−0.04	−0.04	3	162	0.00
	3			0.00	0.00	4	704	0.00
2	0	64	256	28.92	28.91	1	1	29.74
	1			29.87	29.86	2	32	3.14
	2			30.10	30.09	3	162	−1.77
	3			30.16	30.15	4	704	−2.16
3	0	64	256	−10.58	−10.55	1	1	−10.13
	1			−9.83	−9.80	2	32	−3.98
	2			−9.65	−9.62	3	162	−1.14
	3			−9.60	−9.57	4	704	−0.27
4	0	64	256	−5.20	−5.21	1	1	5.78
	1			−1.21	−1.22	2	32	−0.32
	2			−0.21	−0.21	3	162	−0.66
	3			0.05	0.04	4	704	0.41

convergence of the algorithm as the subcone refinement level is increased is indeed caused by the optical thickness approximation in these "pathological" cases.

5.2.4. Kobayashi Benchmark

The final set of results demonstrates the accuracy of our algorithm on a standard benchmark used to evaluate the accuracy of 3D transport codes for problems with both void and highly absorbing regions [9]. To test our method for computing the uncollided flux, we selected the purely-absorbing case of Problem 3 from the reference. The problem consists of a shield region, nearly voided dog-leg duct and a localized source. The geometry is shown in Figure 12 and material properties are given in Table 7.

Table 6: Comparison of the average flux in target cell for the 4 source positions shown in Figure 11, 4 different subcone refinement levels (SC ref.) and IS refinement level 2. 1SP, 12SP and 144SP refer to 0, 1 and 2 levels of recursive splitting of the source cell and computing the optical thicknesses approximately by Eq. (29), while 1SPEx refers to the case with the non-refined source cell and computing the optical thicknesses exactly.

		Rays			Relative error (%)			
Source pos.	SC ref.	1SP	12SP	144SP	1SP	12SP	144SP	1SPEx
1	0	80	960	11520	−0.93	−0.37	−0.25	−0.93
	1				−0.22	−0.08	−0.06	−0.22
	2				−0.04	−0.01	0.00	−0.04
	3				0.00	0.01	−0.01	0.00
2	0	64	768	9216	28.92	2.23	−0.56	5.80
	1				29.87	2.52	−0.35	5.65
	2				30.10	2.60	−0.29	1.56
	3				30.16	2.61	−0.28	0.59
3	0	64	768	9216	−10.58	−5.56	−2.99	−11.33
	1				−9.83	−5.26	−2.74	−4.13
	2				−9.65	−5.18	−2.68	−1.26
	3				−9.60	−5.17	−2.66	−0.36
4	0	64	768	9216	−5.20	−2.22	−0.82	−5.42
	1				−1.21	−0.72	−0.33	−1.35
	2				−0.21	−0.33	−0.21	−0.33
	3				0.05	−0.23	−0.18	−0.07

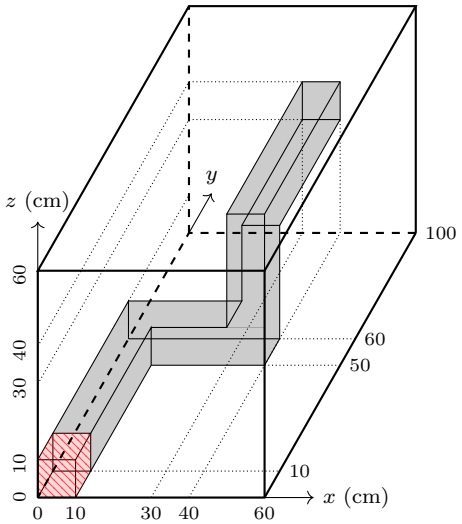


Figure 12: Kobayashi benchmark, problem 3 geometry. The hatched region is the source region, the filled region is the voided duct, and the non-filled region is the shield. Vacuum boundary conditions are applied on the right, top and back planes, remaining boundaries are reflective.

Table 7: Kobayashi benchmark, problem 3 material properties.

Region	Source strength ($\mathrm{cm}^{-3}\mathrm{s}^{-1}$)	Σ_t (cm^{-1})
Source (hatched)	1.0	0.1
Duct (filled)	0.0	10^{-4}
Shield	0.0	0.1

The problem is symmetric along the $x = 0$, $y = 0$ and $z = 0$ planes; since our code does not support reflective boundary conditions at this moment, we extended the domain by 10 cm across each plane of symmetry, so that the full source cell volume $[-10, 10]^3$ is taken into account. The interior planes passing through the explicitly marked locations on each axis in Figure 12 divided the domain into 64 interpolation volumes. We used cubical cells of side length 10 cm for spatial discretization, leading to a total of 539 cells. The 8 cubical cells comprising the source were split into 192 source tetrahedra, i.e., 192 SPs were considered in the computation (cf. Section 3).

Table 8 and Figure 13 show the ratios of the cell-volume average flux computed by our method to the reference solution obtained by MCNP runs tracking 10^{13} neutron histories (leading to statistical errors quoted at 2σ between 0.01% to 0.04% in the cells of interest). We ran our algorithm with three different refinement configurations (cf. Section 5.1.2) – as expected, closer to the source (first row of Figure 13), the subcone refinement level has more impact on the accuracy than the IS refinement, while far from the source (third row), the opposite is true.

Table 8: Cell-average flux comparison between our code and reference results for the Kobayashi problem 3i (no scattering). Reference: MCNP cell-average solution.

Case	Coord. (x,y,z) (cm^3)	Reference Flux ($cm^{-2}s^{-1}$)	Flux / Reference (IS ref., SC ref.) (1,1)	(2,1)	(2,2)
A	5,5,5	5.18720E-00	0.899	0.904	0.965
	5,15,5	1.43212E-00	0.977	0.972	1.006
	5,25,5	5.02923E-01	1.017	1.016	1.024
	5,35,5	2.52417E-01	1.025	1.025	1.027
	5,45,5	1.50064E-01	1.029	1.028	1.028
	5,55,5	9.89941E-02	1.030	1.029	1.028
	5,65,5	4.51500E-02	1.029	1.030	1.027
	5,75,5	1.21840E-02	1.024	1.024	1.022
	5,85,5	3.43559E-03	1.021	1.020	1.018
	5,95,5	1.00084E-03	1.018	1.018	1.016
B	5,55,5	9.89941E-02	1.030	1.029	1.028
	15,55,5	3.34409E-02	0.946	1.002	1.001
	25,55,5	4.93405E-03	1.025	1.010	1.012
	35,55,5	1.48845E-03	1.060	1.008	1.011
	45,55,5	3.55176E-04	1.037	1.001	1.005
	55,55,5	9.62717E-05	0.981	0.986	0.991
C	5,95,35	3.57931E-05	1.117	1.023	1.023
	15,95,35	2.93560E-05	0.957	0.980	0.980
	25,95,35	1.82746E-05	0.929	0.980	0.981
	35,95,35	2.51281E-05	0.948	0.976	0.976
	45,95,35	1.02846E-05	1.020	0.997	0.997
	55,95,35	3.63964E-06	0.994	1.001	1.002

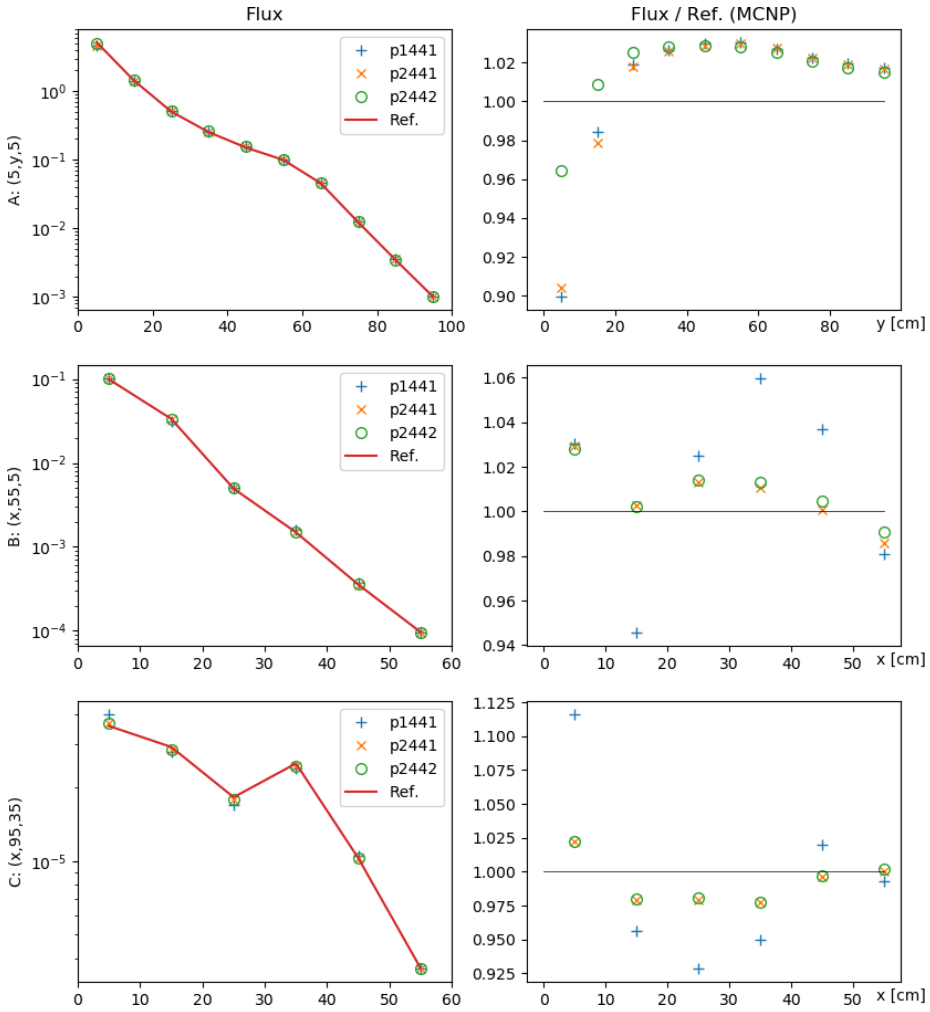


Figure 13: Cell-average flux comparison between our code and reference results for the Kobayashi problem 3i. Reference: MCNP solution. In the legends, "$XYZW$" means: IS refinement level X, spatial quadrature order Y, number of ISPs used for interpolation Z and SC refinement level W (cf. Sec. 5.1.2).

As can be seen in Table 8, with two levels of IS refinement and 2 levels of subcone refinement, the relative error is below 3% for all points outside of the source cell and 3.5% within the source cell. This corresponds to $\sim 10^6$ rays traced, one to four orders of magnitude less than needed by the methods based on the direct quadrature approximation. Table 9 provides a comparison

of the number of rays traced by our method and the estimated number of rays traced by two direct quadrature methods for which there is enough information available in the corresponding literature to compute that number (see Appendix B for details about the calculation in each case). The errors on the Kobayashi 3i benchmark reported for the AETIUS code range to 13.1%. While we realize that the reported errors are with respect to the point-wise analytical solution (as opposed to our cell-averaged results presented above) and that our method requires more work per ray trace (to perform the angular integration over the sub-subcones), we believe that the significant reduction in the number of rays to be traced demonstrates the important advantage over the traditional direct quadrature approach.

Table 9: Comparison of the number of rays traced.

Method		# Rays
Ours (IS ref., SC ref.)	(1,1)	779,520
	(2,1)	3,118,080
	(2,2)	3,118,080
FNSUNCL3 [10]		72,296,875
AETIUS [8]		19,883,997,000

Finally, if only point-wise values at a finite number of target points are needed by an application, we can easily use our algorithm as described in Section 4.4 without the need to trace to the ISPs; the good accuracy of this highly efficient approach is demonstrated for the Kobayashi benchmark in Table 10 and Figure 14.

Table 10: Point-wise flux comparison between our code and reference results for the Kobayashi problem 3i (no scattering). Reference: analytical solution from [9].

Case	Coord. (x,y,z) (cm^3)	Reference Flux ($cm^{-2}s^{-1}$)	Flux / Reference (SC refinement) 0	1	2
A	5,5,5	5.95659E+00	0.952	0.986	0.997
	5,15,5	1.37185E-00	0.857	0.969	0.996
	5,25,5	5.00871E-01	0.981	1.012	1.019
	5,35,5	2.52429E-01	1.010	1.020	1.023
	5,45,5	1.50260E-01	1.023	1.023	1.023
	5,55,5	9.91726E-02	1.032	1.025	1.023
	5,65,5	4.22623E-02	1.037	1.026	1.023
	5,75,5	1.14703E-02	1.030	1.021	1.018
	5,85,5	3.24662E-03	1.026	1.017	1.015
	5,95,5	9.48324E-04	1.022	1.015	1.013
B	5,55,5	9.91726E-02	1.032	1.025	1.023
	15,55,5	2.45041E-02	1.018	1.017	1.017
	25,55,5	4.54477E-03	0.988	0.995	0.997
	35,55,5	1.42960E-03	0.971	0.983	0.986
	45,55,5	2.64846E-04	0.965	0.981	0.985
	55,55,5	9.14210E-05	0.964	0.983	0.987
C	5,95,35	3.27058E-05	0.985	0.986	0.986
	15,95,35	2.68415E-05	1.000	1.001	1.001
	25,95,35	1.70019E-05	0.992	0.993	0.993
	35,95,35	3.37981E-05	0.972	0.970	0.969
	45,95,35	6.04893E-06	0.986	0.987	0.992
	55,95,35	3.36460E-06	0.992	0.993	1.001

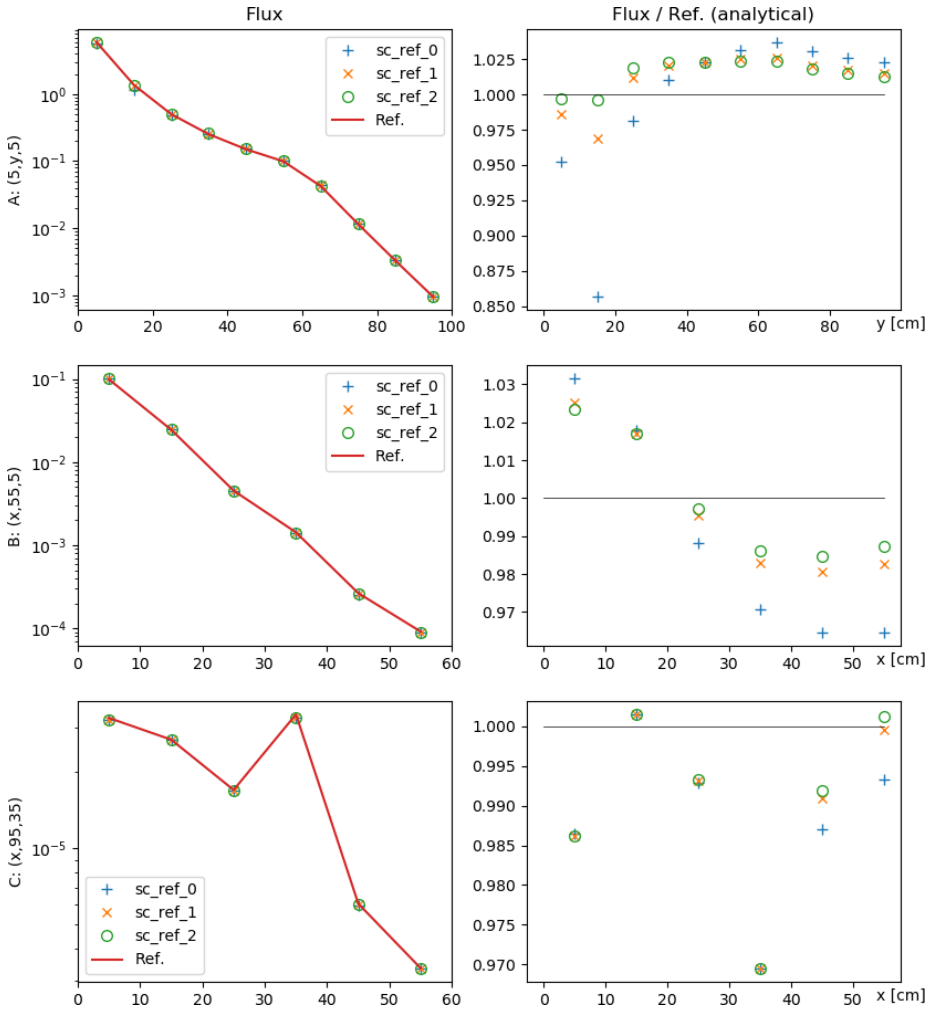


Figure 14: Point-wise flux comparison between our code and reference results for the Kobayashi problem 3i. Reference: analytical solution from [9].

6. Conclusions

In this paper, we have presented a novel algorithm for computing the spatial and angular moments of uncollided flux due to distributed sources. The algorithm is based on ray tracing in arbitrary polyhedral cells and uses a piecewise-linear discontinuous finite element representation of the flux in every cell. It provides accurate solution within, near, and far from the source regions, with

significantly lower number of rays than the traditional algorithms and exhibits a high potential of parallelism. The algorithm can be used to obtain the first/last-collision source in every cell in the mesh, as well as detailed exiting-flux information at specified points on the problem boundary.

This has been achieved by subdividing the computational domain into a set of subvolumes and tracing the rays from each source to the points on the surfaces enclosing such volumes, rather than to every cell in the domain. By a careful interpolation procedure on surfaces separating sub volumes, the rays can be spawned and traced independently in each subvolume (only satisfying the dependencies on the upwind subvolumes), leading to a more uniform ray density throughout the domain and better load balancing than achievable with current methods.

In the current paper, we described the overall algorithm and corresponding mathematical derivations, and performed several numerical experiments to evaluate the accuracy of the algorithm and validate its main features. In a subsequent publication, we will focus on the parallel performance. In our future work, we also plan to describe the treatment of reflective boundaries and address the question of selecting the density of the interpolation surface points in an adaptive manner in order to ensure sufficient amount of rays passing through every cell for an arbitrary mesh refinement.

Acknowledgments

This material is based upon work performed under the Predictive Science Academic Alliance Program (PSAAP) of the U.S. Department of Energy, National Nuclear Security Administration, contract reference DE-NA0002376.

References

[6] W. F. M. Jr., W. H. Reed, Ray-effect mitigation methods for two-dimensional neutron transport theory, Nuclear Science and Engineering 62 (1977) 391–411.

[17] R. Sanchez, On the singular structure of the uncollided and first-collided components of the green's function, Annals of Nuclear Energy 27 (2000) 1167 – 1186.

[1] R. Alcouffe, R. Baker, S. Dahl, J.A.and Turner, R. Ward, PARTISN: A time-dependent, parallel neutral particle transport code system, LA-UR-08-07258, 2008.

[2] T. M. Evans, A. S. Stafford, R. N. Slaybaugh, K. T. Clarno, Denovo: A new three-dimensional parallel discrete ordinates code in scale, Nuclear Technology 171 (2010) 171–200.

[12] R. Lillie, GRTUNCL3D: A Discontinuous Mesh Three-Dimensional First Collision Source Code, in: Proceedings of the 1998 ANS RP&S Div. Topical Conf.: Technologies for the New Century, Nashville, TN, ANS, LaGrange, IL, USA, 1998.

[10] C. Konno, TORT solutions with FNSUNCL3 for kobayashi's 3d benchmarks, Progress in Nuclear Energy 39 (2001) 167 – 179.

[11] K. Kosako, C. Konno, FNSUNCL3: First collision source code for TORT, Journal of Nuclear Science and Technology 37 (2000) 475–478.

[26] I. Zmijarevic, D. Sciannandrone, First collision source in the IDT discrete ordinates transport code, in: Proceedings of the 2017 International Conference on Mathematics & Computational Methods Applied to Nuclear Science & Engineering (M&C 2017), Jeju (KR), Korean Nuclear Society, 2017.

[24] T. Wareing, J. Morel, D. Parsons, A first collision source method for ATTILA, an unstructured tetrahedral mesh discrete ordinates code, in: Proceedings of the 1998 ANS RP&S Div. Topical Conf.: Technologies for the New Century, Nashville, TN, ANS, LaGrange, IL, USA, 1998.

[8] J. W. Kim, Y.-O. 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 (2018) 446 – 469.

[18] Sandia National Laboratories, The CUBIT Geometry and Mesh Generation Toolkit, 2015.

[14] Open CASCADE, S. A. S., OpenCASCADE technology, 3d modeling & numerical simulation, 2018. URL: `http://www.opencascade.org/`.

[20] H. Si, TetGen, a Delaunay-Based Quality Tetrahedral Mesh Generator, ACM Trans. Math. Softw. 41 (2015) 11:1–11:36.

[3] C. Geuzaine, J.-F. Remacle, Gmsh: A 3-d finite element mesh generator with built-in pre- and post-processing facilities, International Journal for Numerical Methods in Engineering 79 (2009) 1309–1331.

[21] H. G. Stone, M. L. Adams, A piecewise linear finite element basis with application to particle transport, in: Proc. ANS Topical Meeting Nuclear Mathematical and Computational Sciences Meeting, 2003.

[22] H. G. Stone, M. L. Adams, New spatial discretization methods for transport on unstructured grids, in: Proc. ANS Topical Meeting Mathematics and Computation, Supercomputing, Reactor Physics and Biological Applications, 2005.

[4] M. W. Hackemack, J. C. Ragusa, Quadratic serendipity discontinuous finite element discretization for sn transport on arbitrary polygonal grids, Journal of Computational Physics (2018).

[16] J. C. Ragusa, Discontinuous finite element solution of the radiation diffusion equation on arbitrary polygonal meshes and locally adapted quadrilateral grids, Journal of Computational Physics 280 (2015) 195 – 213.

[19] D. Shepard, A Two-dimensional Interpolation Function for Irregularly-spaced Data, in: Proceedings of the 1968 23rd ACM National Conference,

ACM '68, ACM, New York, NY, USA, 1968, pp. 517–524. URL: http://doi.acm.org/10.1145/800186.810616. doi:10.1145/800186.810616.

[23] A. van Oosterom, J. Strackee, The solid angle of a plane triangle, IEEE Transactions on Biomedical Engineering BME-30 (1983) 125–126.

[9] K. Kobayashi, N. Sugimura, Y. Nagaya, 3d radiation transport benchmark problems and results for simple geometries with void region, Progress in Nuclear Energy 39 (2001) 119 – 144.

[25] H. Xiao, Z. Gimbutas, A numerical algorithm for the construction of efficient quadrature rules in two and higher dimensions, Computers & mathematics with applications 59 (2010) 663–676.

[13] T. Möller, B. Trumbore, Fast, Minimum Storage Ray-Triangle Intersection, Journal of Graphics Tools 2 (1997) 21–28.

[15] N. Platis, T. Theoharis, Fast Ray-Tetrahedron Intersection Using Plucker Coordinates, Journal of Graphics Tools 8 (2003) 37–48.

[7] A. Kensler, P. Shirley, Optimizing Ray-Triangle Intersection via Automated Search 0 (2006) 33–38.

[5] J. J. Jimnez, R. J. Segura, F. R. Feito, A robust segment/triangle intersection algorithm for interference tests. Efficiency study, Computational Geometry 43 (2010) 474 – 492.

[1] R. Alcouffe, R. Baker, S. Dahl, J.A.and Turner, R. Ward, PARTISN: A time-dependent, parallel neutral particle transport code system, LA-UR-08-07258, 2008.

[2] T. M. Evans, A. S. Stafford, R. N. Slaybaugh, K. T. Clarno, Denovo: A new three-dimensional parallel discrete ordinates code in scale, Nuclear Technology 171 (2010) 171–200.

[3] C. Geuzaine, J.-F. Remacle, Gmsh: A 3-d finite element mesh generator with built-in pre- and post-processing facilities, International Journal for Numerical Methods in Engineering 79 (2009) 1309–1331.

[4] M. W. Hackemack, J. C. Ragusa, Quadratic serendipity discontinuous finite element discretization for sn transport on arbitrary polygonal grids, Journal of Computational Physics (2018).

[5] J. J. Jimnez, R. J. Segura, F. R. Feito, A robust segment/triangle intersection algorithm for interference tests. Efficiency study, Computational Geometry 43 (2010) 474 – 492.

[6] W. F. M. Jr., W. H. Reed, Ray-effect mitigation methods for two-dimensional neutron transport theory, Nuclear Science and Engineering 62 (1977) 391–411.

[7] A. Kensler, P. Shirley, Optimizing Ray-Triangle Intersection via Automated Search 0 (2006) 33–38.

[8] J. W. Kim, Y.-O. 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 (2018) 446 – 469.

[9] K. Kobayashi, N. Sugimura, Y. Nagaya, 3d radiation transport benchmark problems and results for simple geometries with void region, Progress in Nuclear Energy 39 (2001) 119 – 144.

[10] C. Konno, TORT solutions with FNSUNCL3 for kobayashi's 3d benchmarks, Progress in Nuclear Energy 39 (2001) 167 – 179.

[11] K. Kosako, C. Konno, FNSUNCL3: First collision source code for TORT, Journal of Nuclear Science and Technology 37 (2000) 475–478.

[12] R. Lillie, GRTUNCL3D: A Discontinuous Mesh Three-Dimensional First Collision Source Code, in: Proceedings of the 1998 ANS RP&S Div. Topical

Conf.: Technologies for the New Century, Nashville, TN, ANS, LaGrange, IL, USA, 1998.

[13] T. Möller, B. Trumbore, Fast, Minimum Storage Ray-Triangle Intersection, Journal of Graphics Tools 2 (1997) 21–28.

[14] Open CASCADE, S. A. S., OpenCASCADE technology, 3d modeling & numerical simulation, 2018. URL: http://www.opencascade.org/.

[15] N. Platis, T. Theoharis, Fast Ray-Tetrahedron Intersection Using Plucker Coordinates, Journal of Graphics Tools 8 (2003) 37–48.

[16] J. C. Ragusa, Discontinuous finite element solution of the radiation diffusion equation on arbitrary polygonal meshes and locally adapted quadrilateral grids, Journal of Computational Physics 280 (2015) 195 – 213.

[17] R. Sanchez, On the singular structure of the uncollided and first-collided components of the green's function, Annals of Nuclear Energy 27 (2000) 1167 – 1186.

[18] Sandia National Laboratories, The CUBIT Geometry and Mesh Generation Toolkit, 2015.

[19] D. Shepard, A Two-dimensional Interpolation Function for Irregularly-spaced Data, in: Proceedings of the 1968 23rd ACM National Conference, ACM '68, ACM, New York, NY, USA, 1968, pp. 517–524. URL: http://doi.acm.org/10.1145/800186.810616. doi:10.1145/800186.810616.

[20] H. Si, TetGen, a Delaunay-Based Quality Tetrahedral Mesh Generator, ACM Trans. Math. Softw. 41 (2015) 11:1–11:36.

[21] H. G. Stone, M. L. Adams, A piecewise linear finite element basis with application to particle transport, in: Proc. ANS Topical Meeting Nuclear Mathematical and Computational Sciences Meeting, 2003.

[22] H. G. Stone, M. L. Adams, New spatial discretization methods for transport on unstructured grids, in: Proc. ANS Topical Meeting Mathematics

and Computation, Supercomputing, Reactor Physics and Biological Applications, 2005.

[23] A. van Oosterom, J. Strackee, The solid angle of a plane triangle, IEEE Transactions on Biomedical Engineering BME-30 (1983) 125–126.

[24] T. Wareing, J. Morel, D. Parsons, A first collision source method for ATTILA, an unstructured tetrahedral mesh discrete ordinates code, in: Proceedings of the 1998 ANS RP&S Div. Topical Conf.: Technologies for the New Century, Nashville, TN, ANS, LaGrange, IL, USA, 1998.

[25] H. Xiao, Z. Gimbutas, A numerical algorithm for the construction of efficient quadrature rules in two and higher dimensions, Computers & mathematics with applications 59 (2010) 663–676.

[26] I. Zmijarevic, D. Sciannandrone, First collision source in the IDT discrete ordinates transport code, in: Proceedings of the 2017 International Conference on Mathematics & Computational Methods Applied to Nuclear Science & Engineering (M&C 2017), Jeju (KR), Korean Nuclear Society, 2017.

[1] R. Alcouffe, R. Baker, S. Dahl, J.A.and Turner, R. Ward, PARTISN: A time-dependent, parallel neutral particle transport code system, LA-UR-08-07258, 2008.

[2] T. M. Evans, A. S. Stafford, R. N. Slaybaugh, K. T. Clarno, Denovo: A new three-dimensional parallel discrete ordinates code in scale, Nuclear Technology 171 (2010) 171–200.

[3] C. Geuzaine, J.-F. Remacle, Gmsh: A 3-d finite element mesh generator with built-in pre- and post-processing facilities, International Journal for Numerical Methods in Engineering 79 (2009) 1309–1331.

[4] M. W. Hackemack, J. C. Ragusa, Quadratic serendipity discontinuous finite element discretization for sn transport on arbitrary polygonal grids, Journal of Computational Physics (2018).

[5] J. J. Jimnez, R. J. Segura, F. R. Feito, A robust segment/triangle intersection algorithm for interference tests. Efficiency study, Computational Geometry 43 (2010) 474 – 492.

[6] W. F. M. Jr., W. H. Reed, Ray-effect mitigation methods for two-dimensional neutron transport theory, Nuclear Science and Engineering 62 (1977) 391–411.

[7] A. Kensler, P. Shirley, Optimizing Ray-Triangle Intersection via Automated Search 0 (2006) 33–38.

[8] J. W. Kim, Y.-O. 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 (2018) 446 – 469.

[9] K. Kobayashi, N. Sugimura, Y. Nagaya, 3d radiation transport benchmark problems and results for simple geometries with void region, Progress in Nuclear Energy 39 (2001) 119 – 144.

[10] C. Konno, TORT solutions with FNSUNCL3 for kobayashi's 3d benchmarks, Progress in Nuclear Energy 39 (2001) 167 – 179.

[11] K. Kosako, C. Konno, FNSUNCL3: First collision source code for TORT, Journal of Nuclear Science and Technology 37 (2000) 475–478.

[12] R. Lillie, GRTUNCL3D: A Discontinuous Mesh Three-Dimensional First Collision Source Code, in: Proceedings of the 1998 ANS RP&S Div. Topical Conf.: Technologies for the New Century, Nashville, TN, ANS, LaGrange, IL, USA, 1998.

[13] T. Möller, B. Trumbore, Fast, Minimum Storage Ray-Triangle Intersection, Journal of Graphics Tools 2 (1997) 21–28.

[14] Open CASCADE, S. A. S., OpenCASCADE technology, 3d modeling & numerical simulation, 2018. URL: `http://www.opencascade.org/`.

[15] N. Platis, T. Theoharis, Fast Ray-Tetrahedron Intersection Using Plucker Coordinates, Journal of Graphics Tools 8 (2003) 37–48.

[16] J. C. Ragusa, Discontinuous finite element solution of the radiation diffusion equation on arbitrary polygonal meshes and locally adapted quadrilateral grids, Journal of Computational Physics 280 (2015) 195 – 213.

[17] R. Sanchez, On the singular structure of the uncollided and first-collided components of the green's function, Annals of Nuclear Energy 27 (2000) 1167 – 1186.

[18] Sandia National Laboratories, The CUBIT Geometry and Mesh Generation Toolkit, 2015.

[19] D. Shepard, A Two-dimensional Interpolation Function for Irregularly-spaced Data, in: Proceedings of the 1968 23rd ACM National Conference, ACM '68, ACM, New York, NY, USA, 1968, pp. 517–524. URL: `http://doi.acm.org/10.1145/800186.810616`. doi:`10.1145/800186.810616`.

[20] H. Si, TetGen, a Delaunay-Based Quality Tetrahedral Mesh Generator, ACM Trans. Math. Softw. 41 (2015) 11:1–11:36.

[21] H. G. Stone, M. L. Adams, A piecewise linear finite element basis with application to particle transport, in: Proc. ANS Topical Meeting Nuclear Mathematical and Computational Sciences Meeting, 2003.

[22] H. G. Stone, M. L. Adams, New spatial discretization methods for transport on unstructured grids, in: Proc. ANS Topical Meeting Mathematics and Computation, Supercomputing, Reactor Physics and Biological Applications, 2005.

[23] A. van Oosterom, J. Strackee, The solid angle of a plane triangle, IEEE Transactions on Biomedical Engineering BME-30 (1983) 125–126.

[24] T. Wareing, J. Morel, D. Parsons, A first collision source method for ATTILA, an unstructured tetrahedral mesh discrete ordinates code, in:

Proceedings of the 1998 ANS RP&S Div. Topical Conf.: Technologies for the New Century, Nashville, TN, ANS, LaGrange, IL, USA, 1998.

[25] H. Xiao, Z. Gimbutas, A numerical algorithm for the construction of efficient quadrature rules in two and higher dimensions, Computers & mathematics with applications 59 (2010) 663–676.

[26] I. Zmijarevic, D. Sciannandrone, First collision source in the IDT discrete ordinates transport code, in: Proceedings of the 2017 International Conference on Mathematics & Computational Methods Applied to Nuclear Science & Engineering (M&C 2017), Jeju (KR), Korean Nuclear Society, 2017.

[1] R. Alcouffe, R. Baker, S. Dahl, J.A.and Turner, R. Ward, PARTISN: A time-dependent, parallel neutral particle transport code system, LA-UR-08-07258, 2008.

[2] T. M. Evans, A. S. Stafford, R. N. Slaybaugh, K. T. Clarno, Denovo: A new three-dimensional parallel discrete ordinates code in scale, Nuclear Technology 171 (2010) 171–200.

[3] C. Geuzaine, J.-F. Remacle, Gmsh: A 3-d finite element mesh generator with built-in pre- and post-processing facilities, International Journal for Numerical Methods in Engineering 79 (2009) 1309–1331.

[4] M. W. Hackemack, J. C. Ragusa, Quadratic serendipity discontinuous finite element discretization for sn transport on arbitrary polygonal grids, Journal of Computational Physics (2018).

[5] J. J. Jimnez, R. J. Segura, F. R. Feito, A robust segment/triangle intersection algorithm for interference tests. Efficiency study, Computational Geometry 43 (2010) 474 – 492.

[6] W. F. M. Jr., W. H. Reed, Ray-effect mitigation methods for two-dimensional neutron transport theory, Nuclear Science and Engineering 62 (1977) 391–411.

[7] A. Kensler, P. Shirley, Optimizing Ray-Triangle Intersection via Automated Search 0 (2006) 33–38.

[8] J. W. Kim, Y.-O. 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 (2018) 446 – 469.

[9] K. Kobayashi, N. Sugimura, Y. Nagaya, 3d radiation transport benchmark problems and results for simple geometries with void region, Progress in Nuclear Energy 39 (2001) 119 – 144.

[10] C. Konno, TORT solutions with FNSUNCL3 for kobayashi's 3d benchmarks, Progress in Nuclear Energy 39 (2001) 167 – 179.

[11] K. Kosako, C. Konno, FNSUNCL3: First collision source code for TORT, Journal of Nuclear Science and Technology 37 (2000) 475–478.

[12] R. Lillie, GRTUNCL3D: A Discontinuous Mesh Three-Dimensional First Collision Source Code, in: Proceedings of the 1998 ANS RP&S Div. Topical Conf.: Technologies for the New Century, Nashville, TN, ANS, LaGrange, IL, USA, 1998.

[13] T. Möller, B. Trumbore, Fast, Minimum Storage Ray-Triangle Intersection, Journal of Graphics Tools 2 (1997) 21–28.

[14] Open CASCADE, S. A. S., OpenCASCADE technology, 3d modeling & numerical simulation, 2018. URL: `http://www.opencascade.org/`.

[15] N. Platis, T. Theoharis, Fast Ray-Tetrahedron Intersection Using Plucker Coordinates, Journal of Graphics Tools 8 (2003) 37–48.

[16] J. C. Ragusa, Discontinuous finite element solution of the radiation diffusion equation on arbitrary polygonal meshes and locally adapted quadrilateral grids, Journal of Computational Physics 280 (2015) 195 – 213.

[17] R. Sanchez, On the singular structure of the uncollided and first-collided components of the green's function, Annals of Nuclear Energy 27 (2000) 1167 – 1186.

[18] Sandia National Laboratories, The CUBIT Geometry and Mesh Generation Toolkit, 2015.

[19] D. Shepard, A Two-dimensional Interpolation Function for Irregularly-spaced Data, in: Proceedings of the 1968 23rd ACM National Conference, ACM '68, ACM, New York, NY, USA, 1968, pp. 517–524. URL: http://doi.acm.org/10.1145/800186.810616. doi:10.1145/800186.810616.

[20] H. Si, TetGen, a Delaunay-Based Quality Tetrahedral Mesh Generator, ACM Trans. Math. Softw. 41 (2015) 11:1–11:36.

[21] H. G. Stone, M. L. Adams, A piecewise linear finite element basis with application to particle transport, in: Proc. ANS Topical Meeting Nuclear Mathematical and Computational Sciences Meeting, 2003.

[22] H. G. Stone, M. L. Adams, New spatial discretization methods for transport on unstructured grids, in: Proc. ANS Topical Meeting Mathematics and Computation, Supercomputing, Reactor Physics and Biological Applications, 2005.

[23] A. van Oosterom, J. Strackee, The solid angle of a plane triangle, IEEE Transactions on Biomedical Engineering BME-30 (1983) 125–126.

[24] T. Wareing, J. Morel, D. Parsons, A first collision source method for ATTILA, an unstructured tetrahedral mesh discrete ordinates code, in: Proceedings of the 1998 ANS RP&S Div. Topical Conf.: Technologies for the New Century, Nashville, TN, ANS, LaGrange, IL, USA, 1998.

[25] H. Xiao, Z. Gimbutas, A numerical algorithm for the construction of efficient quadrature rules in two and higher dimensions, Computers & mathematics with applications 59 (2010) 663–676.

[26] I. Zmijarevic, D. Sciannandrone, First collision source in the IDT discrete ordinates transport code, in: Proceedings of the 2017 International Conference on Mathematics & Computational Methods Applied to Nuclear Science & Engineering (M&C 2017), Jeju (KR), Korean Nuclear Society, 2017.

Appendix A. Determining Ray-Tetrahedron Intersection

Recall that in order to compute the uncollided flux at any point in the computational domain, we need to accumulate the mean-free-path along a ray from the source volume exit to that point. Since the elementary entity in our geometrical representation of the domain is a tetrahedral "subcell" (constituting the decomposition of each arbitrary polyhedral cell), an efficient and robust algorithm is needed to compute the intersection of a ray and a tetrahedron.

For a ray in given direction $\mathbf{\Omega}$ that starts at a given source point, the algorithm is first used to determine the exiting point and face of the source tetrahedron. This is done by testing each of the four faces of the tetrahedron for the intersection with the ray. The exiting face is then used to determine the adjacent tetrahedron and becomes the entering face of that tetrahedron. From now on, to determine the next exit point and face, the algorithm needs to test the ray intersection with only three faces of each new tetrahedron visited along the ray, excluding the entering face.

Ray-Triangle Intersection

As can be seen, computing the ray-triangle intersection is fundamental to the ray-tetrahedron intersection algorithm. Many algorithms have been proposed to perform such calculations [13, 15, 7, 5]. The algorithm that we implemented is based on the sign comparisons of the signed volumes of the tetrahedra formed by the pairs of vertices of the triangle and the endpoints of the ray and has the advantage of directly providing the barycentric coordinates of the intersection point with respect to the vertices of the triangle, as well as identifying degenerate cases. It is also very efficient as the intersection tests for all faces can be implemented with worst-case complexity of 3 cross-products and 6 dot-products (compared to, e.g., the Plücker coordinate approach presented in [15], which has a worst-case complexity of 6 cross-products and 10 dot-products).

Consider the situation shown in Figure A.15. The triangle ABC represents one of the faces of one of the tetrahedral subcells into which we decompose given

polyhedral cell; the vertices A, B, C are two of the vertices of the cell and either the cell-volume or cell-face centroid, as described in Section 2.3. The triangle ABC is oriented clock-wise relative to the ray origin P (the vectors **PA**, **PB**, **PQ** form a right-handed system) and hence the sign of the scalar triple product:

$$\gamma = [\mathbf{PA}, \mathbf{PB}, \mathbf{PQ}] = (\mathbf{PA} \times \mathbf{PB}) \cdot \mathbf{PQ}$$

is positive. The ray **PQ** passes clock-wise around the oriented edge **AB**. If the ray passes around all three edges in the same clock-wise manner, i.e

$$\begin{gathered} \alpha > 0 \text{ and } \beta > 0 \text{ and } \gamma > 0, \text{ where} \\ \alpha = [\mathbf{PB}, \mathbf{PC}, \mathbf{PQ}], \quad \beta = [\mathbf{PC}, \mathbf{PA}, \mathbf{PQ}], \quad \gamma = [\mathbf{PA}, \mathbf{PB}, \mathbf{PQ}], \end{gathered} \tag{A.1}$$

then the ray intersects the triangle ABC.

The absolute value of γ is one-sixth of the volume of the tetrahedron $ABPQ$ and is proportional to the shaded area in Figure A.15. The barycentric coordinates of the intersection point are thus obtained as

$$u = \frac{\alpha}{\alpha + \beta + \gamma}, \; v = \frac{\beta}{\alpha + \beta + \gamma}, \; w = \frac{\gamma}{\alpha + \beta + \gamma}$$
$$\mathbf{X} = u\mathbf{A} + v\mathbf{B} + w\mathbf{C}.$$

The barycentric coordinates can be used to easily evaluate the PWLD basis functions at the exit point from the subcell (and hence at any point within the subcell via Eq. (19)) using their nodal values at the cell vertices.

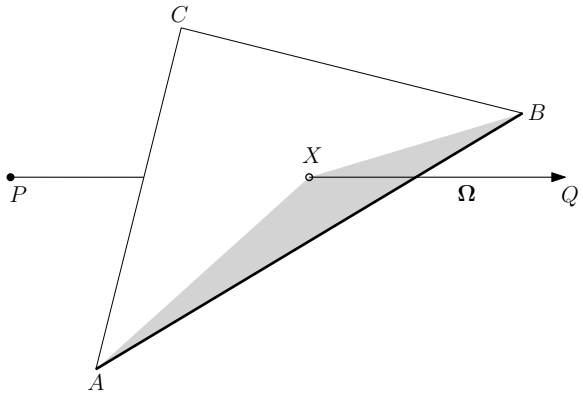


Figure A.15: Ray-triangle intersection test. The ray **PQ** passes clock-wise around the oriented edge AB.

Entry Point Shifting

If one of the triple products in Eq. (A.1) is zero (within a small ε-tolerance to guard against floating point rounding errors), the ray exits the tetrahedral subcell through a point lying on its edge. Likewise, two vanishing triple products signal an exit through a vertex. In both cases, the exiting face cannot be uniquely determined. This becomes an issue when we need to determine the neighboring subcell in order to propagate the ray throughout the domain. To overcome this problem, we shift the entry point towards another point in the subcell and repeat the calculation[4].

Let $\mathbf{P}_{\text{enter}}$ be the entry point, $\mathbf{P}_{\text{tgt}}$ the target point to shift the entry point towards and δ a small shift amount. Then the entry point is shifted as

$$\mathbf{P}_{\text{enter}}^{\text{shift}} = \mathbf{P}_{\text{enter}} + \delta \frac{\mathbf{P}_{\text{tgt}} - \mathbf{P}_{\text{enter}}}{\|\mathbf{P}_{\text{tgt}} - \mathbf{P}_{\text{enter}}\|}. \tag{A.2}$$

As target points, the vertices of the subcell and the midpoints of the subcell edges are used (the latter are needed for the case when the ray coincides with the edge of the subcell, i.e. when both the entry and exit point are the subcell vertices). These target points are used one by one as $\mathbf{P}_{\text{tgt}}$ in Eq. (A.2) until an exiting point lying on a face is found.

Appendix B. Computation of the Number of Rays Traced for the Kobayashi Benchmark

FNSUNCL3

In the reference [10], the authors appear to have considered the whole problem domain without the reflective boundaries. When restricted to the same domain as in our algorithm: $[-10, 60] \times [-10, 100] \times [-10, 60]$ cm^3 and using the discretization parameters reported in the reference (mesh spacing of 2 cm outside of the source, 0.25 cm in the source region), we arrive at:

[4]Note that the ray-triangle intersection algorithm is used in other parts of the calculation as well, e.g. to compute the chord length L in the source volume in Eq. (24); in this case, we only need to determine the exit point and the entry point shifting is not needed.

- Out-of-source regions: $70 \times 110 \times 70 \times -20 \times 20 \times 20 = 531,000$ cm^3, divided into 66,375 cells of volume 8 cm^3.

- Source region: $10 \times 10 \times 10$ cm^3, divided into $80 \times 80 \times 80$ cells $\Rightarrow$ 512,000 cells.

That is, the FNSUNCL3-equivalent of the computation domain used by our algorithm would consist of a total of 578,375 cells. Multiplying by the 125 point sources used to simulate the distributed source by FNSUNCL3, we arrive at a total of 72,296,875 rays traced.

AETIUS

As described in reference [8], the AETIUS calculation of the Kobayashi 3i benchmark was also performed on the full-scale domain. The authors discretized the domain using 918,086 tetrahedral elements:

- Source region: 36,351 elements; in our case, we model the full source region as well.

- Void region: A volume of 120,000 cm^3 was modeled by AETIUS using 46,000 elements, while our calculation was run over the volume of 33,000 cm^3. This corresponds to roughly 13,000 elements in the AETIUS model.

- Shield region: A volume of 2,752,000 cm^3 was modeled in AETIUS using 834,835 elements. We modeled the domain of volume

$$70 \times 70 \times 110 - 33,000 - 20 \times 20 \times 20 = 498,300 \text{ cm}^3,$$

  corresponding to roughly 150,000 elements in the AETIUS model.

Therefore, the AETIUS-equivalent of the computation domain used by our algorithm would consist of roughly 36,000 + 13,000 + 498,000 = 547,000 elements. Multiplying by the 36,351 source cells leads to the total of 19,883,997,000 rays traced.