

A hybrid Monte Carlo, discontinuous Galerkin method for linear kinetic transport equations

Johannes Krotz ^{a,*}, Cory D. Hauck ^b, Ryan G. McClarren ^c

^a Department of Mathematics, University of Tennessee Knoxville, Knoxville, 37996, TN, USA

^b Computational Mathematics Group, Computer Sciences and Mathematics Division, Oak Ridge National Laboratory, Oak Ridge, 37831, TN, USA

^c Department of Aerospace and Mechanical Engineering, University of Notre Dame, Notre Dame, 46545, IN, USA



ARTICLE INFO

Keywords:

Hybrid stochastic-deterministic method
Monte Carlo
Kinetic equations
Particle transport

ABSTRACT

We present a hybrid method for time-dependent particle transport problems that combines Monte Carlo (MC) estimation with deterministic solutions based on discrete ordinates. For spatial discretizations, the MC algorithm computes a piecewise constant solution and the discrete ordinates use bilinear discontinuous finite elements. From the hybridization of the problem, the resulting problem solved by Monte Carlo is scattering free, resulting in a simple, efficient solution procedure. Between time steps, we use a projection approach to “relabel” collided particles as uncollided particles. From a series of standard 2-D Cartesian test problems we observe that our hybrid method has improved accuracy and reduction in computational complexity of approximately an order of magnitude relative to standard discrete ordinates solutions.

1. Introduction

Numerical methods for kinetic transport equations are commonly divided into two classes: deterministic and Monte Carlo. Each of these approaches has strengths and weaknesses that complement the other.

Deterministic methods [24] directly discretize phase space (physical space, direction of flight, particle energy) as well as time, in the time-dependent setting. For this large seven-dimensional space (three for physical space, two for direction of flight, one for energy, and one for time), it is difficult to construct high resolution solutions for general problems. Indeed, the number of operations and the memory footprint required for deterministic solvers can be a challenge, even for leadership-class computers.

Monte Carlo methods [34], on the other hand, use sampling techniques to simulate particle transport processes. In its most basic form, the Monte-Carlo procedure is a computational analog of the actual physical processes being simulated: particles are sampled from sources and boundary conditions, then tracked as they stream through the domain, and along the way undergo scattering or absorption interactions with the material medium. As the particle traverses the physical domain, it contributes to integrated quantities of interest such as particle density or net fluence through a surface. For linear problems, the central limit theorem implies that the Monte Carlo solution is exact in the limit of an infinite number of samples [34]. Unlike deterministic methods, Monte Carlo methods are easy to extend to complicated 3-D geometries and can handle physical processes (such as particle interactions with the background material) in a continuous manner. Nevertheless, the uncertainty in Monte Carlo methods, as expressed in the standard deviation of an estimate, scales like $N^{-1/2}$, where N is the number of sample particles. Additionally, Monte Carlo methods are not well-suited

* Corresponding author.

E-mail address: jkrotz@nd.edu (J. Krotz).

for obtaining uniform spatial estimates due to the difficulty of getting sufficient samples in every region of the physical domain. Moreover, for nonlinear problems such as thermal radiative transfer, the Monte Carlo approach loses some of its attractive properties. For example the discretization of material temperature, to which the particles are coupled, means that an exact solution is not obtained in the limit of infinite samples [43,12,26]. Nevertheless, producing efficient, accurate Monte Carlo calculations is an active area of research [33,35].

Hybrid methods have been developed to harmonize the benefits of Monte Carlo and deterministic methods while minimizing their respective drawbacks. For steady-state nuclear reactor problems, methods such as COMET [27,44] use local Monte Carlo calculations to estimate properties of solutions in macroscopic regions of the problem and then use a deterministic procedure to couple these regions together. Other work has considered weight windows and other biasing techniques [37,6,7,38,28] wherein deterministic solutions are used to modify the flight of particles in Monte Carlo calculations so that computational effort is spent more efficiently. High-order low-order (HOLO) schemes [29,42,30] have been developed in which Monte Carlo is used to compute a closure term for a low-order, moment-based deterministic calculation.

This work presents a deterministic-Monte Carlo hybrid method for time-dependent problems based on the physics of particle transport. Previous work [21,9,10,22,40] has shown that treating particles from the beginning of a step to their first collision with a high-resolution discretization in angle and then treating the particles after they scatter with a low-resolution method can give efficient and accurate numerical calculations. Because the scattering process relaxes particles towards a weakly anisotropic angular distribution, one can combine methods that are appropriate for particle streaming for the uncollided particles during a time step with methods that are suitable for weakly anisotropic angular distributions. In previous work, deterministic methods with a large number of angular degrees of freedom were used for the uncollided particles while low-resolution deterministic methods were used for the collided particles. Moreover, it has been shown [40] that the benefits of this splitting approach extend to multigroup problems by applying a coarser discretization in both energy and angle to the collided particles.

Despite the benefits of deterministic hybrid methods, solutions still require a large number of degrees of freedom for problems with large streaming paths. A natural strategy to address this challenge, which for steady-state problems was first proposed in [3], is to use Monte Carlo for the uncollided particles. Indeed, in many respects this is the ideal situation for a Monte Carlo approach. During a time step, particles are tracked through the computational domain and a non-analog estimator of the solution known as implicit capture is employed, thereby avoiding the need to consider collisions at all. Thus the calculation of the contribution to the solution from uncollided particles is essentially a ray tracing algorithm, which has many efficient implementations on modern computing hardware [2].

The hybrid method considered here uses Monte Carlo to compute the contribution to the solution from uncollided particles and discrete-ordinates with a discontinuous Galerkin (DG) spatial discretization for the collided particles. A key advancement for extending the original steady-state formulation to time-dependent problems is a remapping step that resamples particles from the deterministic collided solution back into the uncollided component. This procedure is critical since, otherwise, the number of uncollided particles will decay exponentially and the hybrid solution will relax to a low-resolution, deterministic approximation of the collided solution [21]. We find that the hybrid approach leads to more accurate solutions obtained using lower computational complexity than pure deterministic calculations. An additional benefit of the hybrid is that, away from domain boundaries, it reduces asymptotically to the equation for the collided particles as the mean free path approaches zero [21]. Thus, unlike standard Monte-Carlo methods, the hybrid can capture the diffusion limit (see e.g. [23]) with an appropriate discretization of the collided equation. The DG method used in this work provides one such discretization.

The remainder of the paper is organized as follows. In Section 2, we introduce the hybrid method in the context of a single-group transport equation that is independent of particle energy. We also summarize the numerical methods used for the uncollided and collided components of the hybrid. In Section 3, we present numerical results for several standard test problems in a reduced two-dimensional geometry in physical space. In Section 4, we summarize the findings and present directions for future work. A short appendix describes the Monte Carlo implementation of a boundary for one of the test problems. A repository with the code used in the numerical experiments is available under <https://github.com/johannonymous123/MCDG-Hybrid.git>.

2. Basics of the hybrid method

2.1. Transport equation

Let $X \subset \mathbb{R}^3$ be a spatial domain with Lipschitz boundary and let \mathbb{S}^2 be the unit sphere in \mathbb{R}^3 . Let $\Psi = \Psi(\mathbf{x}, \Omega, t)$ be the angular flux depending on the position $\mathbf{x} = (x, y, z) \in X$, the direction of flight $\Omega \in \mathbb{S}^2$ and time $t > 0$. We assume that Ψ is governed by the linear transport equation

$$\frac{1}{c} \partial_t \Psi + \Omega \cdot \nabla_{\mathbf{x}} \Psi + \sigma_t \Psi = \frac{\sigma_s}{4\pi} \langle \Psi \rangle + Q, \quad \mathbf{x} \in X, \quad \Omega \in \mathbb{S}^2, \quad t > 0, \quad (1)$$

where $\sigma_t = \sigma_t(\mathbf{x})$, $\sigma_s = \sigma_s(\mathbf{x})$ and $\sigma_a = \sigma_t - \sigma_s$ are the total, scattering, and absorption cross-sections of the material, respectively; $Q = Q(\mathbf{x}, \Omega, t)$ is a known particle source; and angle brackets denote integration over the unit sphere:

$$\langle \Psi \rangle = \int_{\mathbb{S}^2} \Psi d\Omega. \quad (2)$$

The constant $c > 0$ is the particle speed; we assume that $c = 1$ for the remainder of this work. The transport equation (1) is equipped with initial conditions

$$\Psi(\mathbf{x}, \Omega, 0) = \Psi_0(\mathbf{x}, \Omega), \quad \mathbf{x} \in X, \quad \Omega \in \mathbb{S}^2, \quad (3)$$

and boundary condition

$$\Psi(\mathbf{x}, \Omega, t) = G(\mathbf{x}, \Omega, t) \quad \mathbf{x} \in \partial X, \quad \Omega \cdot \mathbf{n}(\mathbf{x}) < 0, \quad (4)$$

where Ψ_0 and G are known and $\mathbf{n}(\mathbf{x})$ is the unit outward normal at $\mathbf{x} \in \partial X$.

2.2. The hybrid method

The hybrid method is based on a first collision source splitting [3]. Let $\Psi = \Psi_u + \Psi_c$, where the *uncollided flux* Ψ_u and the *collided flux* Ψ_c satisfy the following system of equations

$$\partial_t \Psi_u + \Omega \cdot \nabla_{\mathbf{x}} \Psi_u + \sigma_t \Psi_u = Q, \quad (5a)$$

$$\partial_t \Psi_c + \Omega \cdot \nabla_{\mathbf{x}} \Psi_c + \sigma_t \Psi_c = \frac{\sigma_s}{4\pi} (\langle \Psi_u \rangle + \langle \Psi_c \rangle). \quad (5b)$$

Due to the linearity of (1), the splitting in (5) is exact; that is, if Ψ_u and Ψ_c solve (5a) and (5b), respectively, then $\Psi_u + \Psi_c$ solves (1). In practice, however, (5a) and (5b) are solved at different resolutions or even with different methods. Typically (5a) is solved with a method that has high resolution in angle, and because (5a) has no coupling in angle, it is easier to solve than (1) and also easy to solve in parallel. Meanwhile (5b) inherits the angular coupling in (1), but typically requires less angular resolution.

Since (5a) has no scattering source, particle densities will be transferred into the collided flux at an exponential rate, thus making the accuracy at which (5b) is solved the driving factor in the overall accuracy. This effect can be mitigated by abusing the autonomous nature of the equations and periodically relabeling the collided flux as uncollided at every time step.

To describe the implementation of the hybrid in more detail, let \mathcal{T} be an operator such that

$$u[t] = \mathcal{T}(t, t', s, v, b, \lambda_t, \lambda_s) \quad (6)$$

where $u[t](\mathbf{x}, \Omega) := u(\mathbf{x}, \Omega, t)$, satisfies

$$\begin{cases} \partial_t u + \Omega \cdot \nabla_{\mathbf{x}} u + \lambda_t u = \frac{\lambda_s}{4\pi} \langle u \rangle + s, & \mathbf{x} \in X, \quad \Omega \in \mathbb{S}^2, \quad t > t', \\ u(\mathbf{x}, \Omega, t') = v(\mathbf{x}, \Omega) & \mathbf{x} \in X, \quad \Omega \in \mathbb{S}^2, \\ u(\mathbf{x}, \Omega, t) = b(\mathbf{x}, \Omega, t) & \mathbf{x} \in \partial X, \quad \Omega \cdot \mathbf{n}(\mathbf{x}) < 0, \quad t > t'. \end{cases} \quad (7)$$

with source $s = s(\mathbf{x}, \Omega, t)$. Using the operator \mathcal{T} , we can write

$$\Psi[t_{n+1}] = \mathcal{T}(t_{n+1}, t_n, Q, \Psi[t_n], G, \sigma_t, \sigma_s), \quad (8)$$

$$\Psi_u[t_{n+1}] = \mathcal{T}(t_{n+1}, t_n, Q_u, \Psi[t_n], G, \sigma_t, 0), \quad Q_u = Q \quad (9)$$

$$\Psi_c[t_{n+1}] = \mathcal{T}(t_{n+1}, t_n, Q_c, 0, 0, \sigma_t, \sigma_s), \quad Q_c = \frac{\sigma_s}{4\pi} \langle \Psi_u \rangle. \quad (10)$$

We simulate the system (5) using a Monte Carlo method for the uncollided equation (5a) and a deterministic discretization of the collided equation (5b). Let

$$\mathcal{T}_{MC}(t, t', s, v_{MC}, b, \lambda_t, \lambda_s; N_p) \quad (11)$$

be the Monte Carlo approximation to (7) given a particle representation v_{MC} of v and using N_p pseudo-particles to represent the distribution of particles in phase space introduced by the source s over the interval (t', t) . Similarly,

$$\mathcal{T}_{SN}(t, t', s, v, b, \lambda_t, \lambda_s; N, N_x) \quad (12)$$

denote the S_N -DG approximation of (7) using a level N set of ordinates, N_x spatial cells per dimension with \mathbb{Q}_1 elements, and a backward Euler time discretization to get from t' to t . (The Monte Carlo method and S_N -DG method are presented in more detail below.) Then given N_p , N , and N_x , and a Monte Carlo approximation ψ^n of $\Psi(t_n)$, let

$$\psi_u^{n+1} = \mathcal{T}_{MC}(t_{n+1}, t_n, Q_u, \psi^n, G, \sigma_t, 0; N_p), \quad Q_u = Q \quad (13)$$

$$\psi_c^{n+1} = \mathcal{T}_{SN}(t_{n+1}, t_n, Q_c, 0, 0, \sigma_t, \sigma_s; N, N_x), \quad Q_c = \frac{\sigma_s}{4\pi} \langle \psi_u^{n+1} \rangle_{MC} \quad (14)$$

$$\mathcal{R}\psi_c^{n+1} = \mathcal{T}_{MC}(t_{n+1}, t_n, Q_r, 0, 0, \sigma_t, 0; N_p), \quad Q_r = Q_c + \frac{\sigma_s}{4\pi} \langle \psi_c^{n+1} \rangle_{SN} \quad (15)$$

$$\psi^{n+1} = \psi_u^{n+1} + \mathcal{R}\psi_c^{n+1} \quad (16)$$

where \mathcal{R} is the relabeling operator and $\langle \cdot \rangle_{\text{MC}}$ and $\langle \cdot \rangle_{\text{SN}}$ denote approximation of the angular integral over \mathbb{S}^2 with the respective method. When equations (13) through (15) are solved sequentially, ψ_u^{n+1} is a Monte Carlo approximation of (5a) with initial condition ψ^n , and $\mathcal{R}\psi_c^{n+1}$ is a Monte Carlo approximation of (5b) with zero initial condition. Thus ψ^{n+1} is a Monte Carlo approximation of (1) with initial condition ψ^n .¹

2.3. Discrete ordinate-discontinuous Galerkin

The discrete ordinates (S_N) method [5] approximates (7) by replacing the angular integral $\langle u \rangle$ by a discrete quadrature and then solving the resulting equation for the angles in the quadrature. This procedure yields a system of equations that depend only on space and time and can be further discretized by a variety of methods. Let

$$\{\Omega_q\}_{q=1}^{N_\Omega} \quad \text{and} \quad \{\omega_q\}_{q=1}^{N_\Omega} \quad (17)$$

be the angles and associated weights of the S_N quadrature, where $N_\Omega = N_\Omega(N)$ depends on the specific type of quadrature set being used. After discretizing in angle and applying an implicit Euler time discretization, the following semi-discrete system is obtained for each $q \in \{1, \dots, N_\Omega\}$ and $n \in \{0, 1, 2, \dots\}$,

$$\begin{cases} \frac{1}{\Delta t} (u_q^{n+1} - u_q^n) + \Omega_q \cdot \nabla_x u_q^{n+1} + \lambda_t u_q^{n+1} = \frac{\lambda_s}{4\pi} \sum_{r=1}^{N_\Omega} \omega_r u_r^{n+1} + s_q^{n+1}, & \mathbf{x} \in X, \\ u_q^{n+1}(\mathbf{x}) = b_q^{n+1}(\mathbf{x}), & \mathbf{x} \in \partial X_q^-, \end{cases} \quad (18)$$

where $\partial X_q^- = \{\mathbf{x} \in X : \Omega_q \cdot \mathbf{n}(\mathbf{x}) < 0\}$, $b_q^n(\mathbf{x}) = b(\mathbf{x}, \Omega_q, t_n)$, $s_q^n(\mathbf{x}) = s(\mathbf{x}, \Omega_q, t_n)$, and $u_q^n(\mathbf{x}) \approx u(\mathbf{x}, \Omega_q, t_n)$ is the approximation on the temporal and angular grid. After reassigning

$$u_q \leftarrow u_q^{n+1}, \quad s_q \leftarrow s_q^{n+1} + u_q^n, \quad \lambda_t \leftarrow \lambda_t + \frac{1}{\Delta t}, \quad \text{and} \quad b_q \leftarrow b_q^{n+1}, \quad (19)$$

the discretization in (18a) can be written in the equivalent steady-state form

$$\begin{cases} \Omega_q \cdot \nabla_x u_q + \lambda_t u_q = \lambda_s \bar{\mathbf{u}} + s_q, & \mathbf{x} \in X \\ u_q(\mathbf{x}) = b_q(\mathbf{x}), & \mathbf{x} \in \partial X_q^- \end{cases} \quad (20)$$

where $\mathbf{u} = (u_1, \dots, u_{N_\Omega})^\top$, $\bar{\mathbf{u}} := \frac{1}{4\pi} \sum_r \omega_r u_r$.

We discretize (20a) in physical space with a discontinuous Galerkin method and upwind numerical fluxes. The method by now is fairly standard (see for example [9,20]) and is often used because of its accuracy in scattering-dominated regimes relative to upwind finite-difference and finite-volume methods [1,23,32,19]. Because the DG method is well-known, we summarize it only briefly for the case of a two-dimensional Cartesian mesh with rectangular cells, which is sufficient for all of the numerical tests in Section 3. Let X be divided into open sets $C_{i,j}$ that are squares with side lengths h centered at (x_i, y_j) , and let $V_h = \{v \in L^2(X) : v|_{C_{i,j}} \in Q_1\}$. The goal will be to find an approximation of the weak solution of equation (20a); that is, find $\mathbf{u}^h = (u_1^h, \dots, u_{N_\Omega}^h)^\top \in [V_h]^{N_\Omega} := V_h \underbrace{\times \dots \times}_{N_\Omega \text{ times}} V_h$

such that

$$\mathcal{A}_q^{(i,j)}(u_q^h, v_q^h) + \mathcal{P}_q^{(i,j)}(u_q^h, v_q^h) = \mathcal{M}_q^{(i,j)}(u_q^h, v_q^h) + \mathcal{R}^{(i,j)}(\mathbf{u}^h, v_q^h) + \mathcal{S}_q^{(i,j)}(v_q^h) + \mathcal{B}_q^{(i,j)}(v_q^h) \quad (21)$$

for all $i, j \in \{1, \dots, N_x\}$, $q \in \{1, \dots, N\}$, and $\mathbf{v}^h \in V_h^{N_\Omega}$. Here

$$\mathcal{A}_q^{(i,j)}(u_q^h, v_q^h) = - \int_{C_{i,j}} (\Omega_q \cdot \nabla_x v_q^h) u_q^h d\mathbf{x} + \lambda_t \int_{C_{i,j}} v_q^h u_q^h d\mathbf{x}, \quad (22)$$

$$\mathcal{P}_q^{(i,j)}(u_q^h, v_q^h) = \int_{(\partial C_{i,j})_q^+} (\Omega_q \cdot \mathbf{n}) v_q^{h,-} v_q^{h,-} ds(\mathbf{x}), \quad \mathcal{M}_q^{(i,j)}(u_q^h, v_q^h) = \int_{(\partial C_{i,j})_q^-} (\Omega_q \cdot \mathbf{n}) v_q^{h,-} u_q^{h,+} ds(\mathbf{x}), \quad (23)$$

$$\mathcal{R}^{(i,j)}(\mathbf{u}^h, v_q^h) = \lambda_s \int_{C_{i,j}} \bar{\mathbf{u}}^h v_q^h d\mathbf{x}, \quad \mathcal{S}_q^{(i,j)}(v_q^h) = \int_{C_{i,j}} s_q v_q^h d\mathbf{x}, \quad \mathcal{B}_q^{(i,j)}(v_q^h) = \int_{C_{i,j} \cap \partial X_q^-} b_q v_q^h ds(\mathbf{x}), \quad (24)$$

$$v_q^{h,\pm}(\mathbf{x}) = \lim_{\theta \rightarrow 0^+} v_q^h(\mathbf{x} \pm \theta \mathbf{n}), \quad \text{and} \quad (\partial C_{i,j})_q^\pm = \{\mathbf{x} \in \partial C_{i,j} : \pm \Omega_q \cdot \mathbf{n}(\mathbf{x}) > 0\} \quad (25)$$

¹ As an alternative to solving equation (15), pseudo-particles could be sampled directly from ψ_c^{n+1} instead. However, sampling directly from an S_N -DG solution can be problematic, as it is only defined on discrete sets of angles and interpolation may lead to negative values locally in the phase space. Moreover, it is expected that the current approach will be more accurate for two reasons: First, (15) only uses the angular average of the S_N -DG approximation. By using Jensen's inequality, one can show that the error in the angular average of ψ_c is bounded above by the error in ψ_c itself. Second, the solution operator \mathcal{T} is dissipative, i.e., in the absence of sources and boundary data, the solution will decay from one time step to the next. This means that the error introduced via Q_r will decay, as long as the Monte Carlo operator \mathcal{T}_{MC} preserves this dissipative property. These benefits have also been discussed in [11].

We construct $\mathbf{u}^h = (u_1^h, \dots, u_N^h)^\top$ as follows: For each $i, j \in \{1, \dots, N_x\}$ and $q \in \{1, \dots, N_\Omega\}$, let

$$u_q^h(\mathbf{x}) = \sum_{|\mathbf{k}|_\infty \leq 1} \alpha_{q,\mathbf{k}}^{(i,j)} \phi_{\mathbf{k}}^{(i,j)}(x, y), \quad \mathbf{x} \in C_{i,j}, \quad (26)$$

where $\mathbf{k} = (k_1, k_2)$,

$$\phi_{\mathbf{k}}^{(i,j)} = P_{k_1} \left(\frac{x - x_i}{h/2} \right) P_{k_2} \left(\frac{y - y_j}{h/2} \right), \quad (27)$$

and P_k is the usual Legendre polynomial of degree k on $\xi \in [-1, 1]$ with normalization $\int_{-1}^1 P_{k_1}(\xi) P_{k_2}(\xi) d\xi = \frac{2}{2k_1+1} \delta_{k_1, k_2}$. Using this representation for \mathbf{u}^h , we derive the following linear system for the coefficients $\alpha_{q,\mathbf{k}}^{(i,j)}$ from equation (21): For each \mathbf{l} such that $|\mathbf{l}|_\infty \leq 1$,

$$\sum_{|\mathbf{k}|_\infty \leq 1} \left(\mathbf{A}_{q,\mathbf{l},\mathbf{k}}^{(i,j)} + \mathbf{P}_{q,\mathbf{l},\mathbf{k}}^{(i,j)} \right) \alpha_{q,\mathbf{k}}^{(i,j)} = \sum_{|\mathbf{k}|_\infty \leq 1} \left(\mathbf{M}_{q,\mathbf{l},\mathbf{k}}^{(i,j)} \alpha_{q,\mathbf{k}}^{(i^*,j^*,q)} + \mathbf{R}_{\mathbf{l},\mathbf{k}}^{(i,j)} \tilde{\alpha}_{\mathbf{k}}^{(i,j)} \right) + \mathbf{S}_{q,\mathbf{l}}^{(i,j)} + \mathbf{B}_{q,\mathbf{l}}^{(i,j)}, \quad (28)$$

where $\tilde{\alpha}_{\mathbf{k}}^{(i,j)} = \sum_{q=1}^{N_\Omega} w_q \alpha_{q,\mathbf{k}}^{(i,j)}$,

$$\mathbf{A}_{q,\mathbf{l},\mathbf{k}}^{(i,j)} = \mathcal{A}_q^{(i,j)}(\phi_{\mathbf{k}}^{(i,j)}, \phi_{\mathbf{l}}^{(i,j)}), \quad \mathbf{P}_{q,\mathbf{l},\mathbf{k}}^{(i,j)} = \mathcal{P}_q^{(i,j)}(\phi_{\mathbf{k}}^{(i,j)}, \phi_{\mathbf{l}}^{(i,j)}) \quad \mathbf{M}_{q,\mathbf{l},\mathbf{k}}^{(i,j)} = \mathcal{M}_q^{(i,j)}(\phi_{\mathbf{k}}^{(i^*,j^*)}, \phi_{\mathbf{l}}^{(i,j)}) \quad (29)$$

$$\mathbf{B}_{q,\mathbf{l}}^{(i,j)} = \mathcal{B}_q^{(i,j)}(\phi_{\mathbf{l}}^{(i,j)}), \quad \mathbf{S}_{q,\mathbf{l}}^{(i,j)} = \mathcal{S}_q^{(i,j)}(\phi_{\mathbf{l}}^{(i,j)}), \quad \mathbf{R}_{\mathbf{l},\mathbf{k}}^{(i,j)} = \mathcal{R}^{(i,j)}(\phi_{\mathbf{k}}^{(i,j)}, \phi_{\mathbf{l}}^{(i,j)}) \quad (30)$$

and, given the components $\mathbf{n} = (n_x, n_y)$ of the outward normal, the indices i^*, j^* are given by

$$i^*(\mathbf{x}) = i + n_x(\mathbf{x}) \quad \text{and} \quad j^*(\mathbf{x}) = j + n_y(\mathbf{x}). \quad (31)$$

To improve readability, we rewrite equation (28) as a matrix equation with respect to the indices \mathbf{k} and \mathbf{l} :

$$\left(\mathbf{A}_q^{(i,j)} + \mathbf{P}_q^{(i,j)} \right) \alpha_q^{(i,j)} = \mathbf{R}^{(i,j)} \tilde{\alpha}^{(i,j)} + \mathbf{M}_q^{(i,j)} \alpha_q^{(i^*,j^*)} + \mathbf{B}_q^{(i,j)} + \mathbf{S}_q^{(i,j)}. \quad (32)$$

The organization of the operators in (32) reflects a standard solution strategy combining source iteration and sweeping. In source iteration, $\tilde{\alpha}^{(i,j)}$ is lagged; that is, given an iteration index ℓ :

$$\alpha_q^{(i,j,\ell+1)} = \left(\mathbf{A}_q^{(i,j)} + \mathbf{P}_q^{(i,j)} \right)^{-1} \left(\mathbf{R}^{(i,j)} \tilde{\alpha}^{(i,j,\ell)} + \mathbf{M}_q^{(i,j)} \alpha_q^{(i^*,j^*,\ell+1)} + \mathbf{B}_q^{(i,j)} + \mathbf{S}_q^{(i,j)} \right), \quad (33a)$$

$$\tilde{\alpha}^{(i,j,\ell+1)} = \sum_{q=1}^{N_\Omega} w_q \alpha_q^{(i,j,\ell+1)}. \quad (33b)$$

Sweeping refers to process solving of (33) cell-by-cell: for each q , cells can be ordered such that $\alpha_q^{(i^*,j^*,\ell+1)}$ is known, prior to solving for $\alpha_q^{(i,j,\ell+1)}$. The details of this procedure are given in Algorithm 1.

2.4. Monte Carlo

In this section, we describe the Monte Carlo method used to compute the solution to (7) for the pure absorption problem when $\lambda_t = \lambda_a$ (i.e. no scattering):

$$\begin{cases} \partial_t u + \boldsymbol{\Omega} \cdot \nabla_{\mathbf{x}} u + \lambda_t u = s, & \mathbf{x} \in X, \quad \boldsymbol{\Omega} \in \mathbb{S}^2, \quad t > 0, \\ u(\mathbf{x}, \boldsymbol{\Omega}, 0) = v(\mathbf{x}, \boldsymbol{\Omega}) & \mathbf{x} \in X, \quad \boldsymbol{\Omega} \in \mathbb{S}^2, \\ u(\mathbf{x}, \boldsymbol{\Omega}, t) = b(\mathbf{x}, \boldsymbol{\Omega}, t) & \mathbf{x} \in \partial X, \quad \boldsymbol{\Omega} \cdot \mathbf{n}(\mathbf{x}) < 0, \quad t > 0. \end{cases} \quad (a) \quad (b) \quad (c) \quad (34)$$

We only consider the case of pure absorption, since, going forward, the Monte Carlo method will only be applied to equation (5a) for uncollided particles and during the relabeling process, both of which are scattering-free. The Monte Carlo method approximates the phase space distribution u using a finite set of pseudo-particles:

$$u(\mathbf{x}, \boldsymbol{\Omega}, t) \approx \sum_{\pi \in \Pi^t} w_\pi(t) \delta(\mathbf{x} - \mathbf{x}_\pi(t)) \delta(\boldsymbol{\Omega} - \boldsymbol{\Omega}_\pi), \quad (35)$$

where Π^t is a set of pseudo-particles π with position $\mathbf{x}_\pi(t)$, weight $w_\pi(t)$, and direction of flight $\boldsymbol{\Omega}_\pi$. Π^t will be defined more carefully below. A benefit of the hybrid is that scattering processes, which can slow down the method significantly [13], do not need to be modeled in (34).

The Monte Carlo implementation of (34) can be derived via a Green's function formulation for (34). Let $G(\mathbf{x}, \mathbf{y}, \boldsymbol{\Omega}, t, t_0)$ solve

$$\partial_t G + \boldsymbol{\Omega} \cdot \nabla_{\mathbf{x}} G + \lambda_t G = \delta(\mathbf{x} - \mathbf{y}) \delta(t - t_0), \quad (36)$$

with zero initial data and boundary conditions. Then

$$u(\mathbf{x}, \Omega, t) = \int_0^t \int_{\mathbb{R}^3} G(\mathbf{x}, \mathbf{y}, \Omega, t, t_0) s(\mathbf{y}, \Omega, t_0) d\mathbf{y} dt_0 \quad (37a)$$

$$+ \int_0^t \int_{\mathbb{R}^3} G(\mathbf{x}, \mathbf{y}, \Omega, t, t_0) s_v(\mathbf{y}, \Omega, t_0) d\mathbf{y} dt_0 \quad (37b)$$

$$+ \int_0^t \int_{\mathbb{R}^3} G(\mathbf{x}, \mathbf{y}, \Omega, t, t_0) s_b(\mathbf{y}, \Omega, t_0) d\mathbf{y} dt_0 \quad (37c)$$

solves (34), where s , s_v , and s_b are identically zero outside of the closure of X . The terms s_v and s_b are provisional source terms designed such that (37b) solves (34) when $s = 0$ and $b = 0$, while (37c) solves (34) when $s = 0$ and $v = 0$. The former is solved by setting $s_v(\mathbf{x}, \Omega, t) = v(\mathbf{x}, \Omega) \delta(t)$. However, determining s_b can be slightly more involved and problem dependent. An example that is used for numerical experiments in Section 3.3 is provided in the Appendix. Once s , s_v , and s_b are known, their sum can be treated as sole source term. Therefore any algorithm solving (37a) is sufficient to solve all three equations (37a), (37b) and (37c). Because of this and for simplicity, we restrict our attention to (37a) below.

For any $t > t_0$ and any fixed Ω , let $\mathbf{X}(t) = \mathbf{x}_0 + (t - t_0)\Omega$. Then $g(t) = G(\mathbf{X}(t), \mathbf{y}, \Omega, t, t_0)$ solves

$$\frac{dg(t)}{dt} = -\lambda(\mathbf{X}(t))g(t) + \delta(\mathbf{X}(t) - \mathbf{y})\delta(t - t_0) \quad (38)$$

or, equivalently,

$$g(t) = e^{-\int_{t_0}^t \lambda_t(\mathbf{X}(\xi)) d\xi} \delta(\mathbf{x}_0 - \mathbf{y}). \quad (39)$$

Setting $\mathbf{x} = \mathbf{X}(t)$ in (39) gives

$$G(\mathbf{x}, \Omega, t, t_0) = e^{-\int_{t_0}^t \lambda_t(\mathbf{x} - (t - \xi)\Omega) d\xi} \delta(\mathbf{x} - (t - t_0)\Omega - \mathbf{y}). \quad (40)$$

Plugging (40) back into (37a) yields, after some manipulation,

$$\begin{aligned} u(\mathbf{x}, \Omega, t) &= \int_0^t \int_{\mathbb{R}^3} e^{-\int_{t_0}^t \lambda_t(\mathbf{x} - (t - \xi)\Omega) d\xi} \delta(\mathbf{x} - (t - t_0)\Omega - \mathbf{y}) s(\mathbf{y}, \Omega, t_0) d\mathbf{y} dt_0 \\ &= \int_0^t e^{-\int_{t_0}^t \lambda_t(\mathbf{x} - (t - \xi)\Omega) d\xi} s(\mathbf{x} - (t - t_0)\Omega, \Omega, t_0) dt_0 \\ &= \int_0^t e^{-\int_{t-\tau}^t \lambda_t(\mathbf{x} - (\tau - \xi)\Omega) d\xi} s(\mathbf{x} - \tau\Omega, \Omega, t - \tau) d\tau \\ &= \int_0^t e^{-\int_0^\tau \lambda_t(\mathbf{x} - \xi\Omega) d\xi} s(\mathbf{x} - \tau\Omega, \Omega, t - \tau) d\tau. \end{aligned} \quad (41)$$

This representation of $u(\mathbf{x}, \Omega, t)$ can be interpreted as the density of particles that have reached the location \mathbf{x} at time t while moving in the direction Ω . These particles are emitted by the source s at time $t - \tau$ and location $\mathbf{x} - \tau\Omega$, and they carry a weight that decays exponentially due to absorption.

The Monte Carlo approach can be understood as an approximation of u based on sampling of pseudo-particles from the source s in (41). Let

$$\tilde{s}(\mathbf{x}, \Omega, t) = \sum_{\pi \in \Pi} w_\pi \delta(\mathbf{x} - \mathbf{x}_\pi) \delta(\Omega - \Omega_\pi) \delta(t - t_\pi) \quad (42)$$

where $\pi \in \Pi$ are pseudo-particles with weight $w_\pi > 0$, position $\mathbf{x}_\pi \in X$, and direction of flight $\Omega_\pi \in \mathbb{S}^2$ at $t_\pi > 0$, such that $\tilde{s}(\mathbf{x}, \Omega, t)$ approximates $s(\mathbf{x}, \Omega, t)$ for all $t > 0$. Then for any $C \subset X$, any $B \subset \mathbb{S}^2$, and any time interval (t_n, t_{n+1}) , the representation of u in (41), along with the approximation \tilde{s} gives

$$\begin{aligned} &\int_{t_n}^{t_{n+1}} \int_C \int_B u(\mathbf{x}, \Omega, t) d\Omega d\mathbf{x} dt \\ &\approx \int_{t_n}^{t_{n+1}} \int_C \int_B \int_0^t e^{-\int_0^\tau \lambda_t(\mathbf{x} - \xi\Omega) d\xi} \tilde{s}(\mathbf{x} - \tau\Omega, \Omega, t - \tau) d\tau d\Omega d\mathbf{x} dt \end{aligned}$$

$$\begin{aligned}
&= \int_{t_n}^{t_{n+1}} \int_C \int_B \int_0^t e^{-\int_0^\tau \lambda_t(x - \xi \Omega) d\xi} \sum_{\pi \in \Pi} w_\pi \delta(x - \tau \Omega - x_\pi) \delta(\Omega - \Omega_\pi) \delta(t - \tau - t_\pi) d\tau d\Omega d\mathbf{x} dt \\
&= \sum_{\pi \in \Pi} \int_{t_n}^{t_{n+1}} w_\pi e^{-\int_0^{t-t_\pi} \lambda_t(x_\pi + \xi \Omega_\pi) d\xi} \mathbb{1}_C(x_\pi + (t - t_\pi) \Omega_\pi) \mathbb{1}_{[0,t]}(t_\pi) \mathbb{1}_B(\Omega_\pi) dt \\
&= \sum_{\pi \in \Pi} \int_{t_n}^{t_{n+1}} w_\pi(t) \mathbb{1}_C(x_\pi + (t - t_\pi) \Omega_\pi) \mathbb{1}_{[0,t]}(t_\pi) \mathbb{1}_B(\Omega_\pi) dt
\end{aligned} \tag{43}$$

where $w_\pi(t) = w_\pi e^{-\int_0^{t-t_\pi} \lambda_t(x_\pi + \xi \Omega_\pi) d\xi}$ and $w_\pi(t_\pi) = w_\pi$.

Note that with the identification $x_\pi(t) = x_\pi + (t - t_\pi) \Omega_\pi$ we can also identify Π' in (35) as $\Pi' = \{\pi \in \Pi : t_\pi < t\}$.

We will denote (43) as $\mathcal{T}_{\text{MC}}(t_{n+1}, t_n, 0, \lambda_s, \lambda_s, s, 0; N_p)$ as the Monte Carlo solution for the case of the zero initial data and zero boundary data. A general Monte-Carlo solution can be obtained as

$$\begin{aligned}
\mathcal{T}_{\text{MC}}(t_{n+1}, t_n, s, v, b, \lambda_s, \lambda_s; N_p) &= \mathcal{T}_{\text{MC}}(t_{n+1}, t_n, s + s_b, v, 0, \lambda_s, \lambda_s; N_p) \\
&= \mathcal{T}_{\text{MC}}(t_{n+1}, t_n, s + s_v + s_b, 0, 0, \lambda_s, \lambda_s; N_p)
\end{aligned}$$

Numerically it is useful to realize that

$$\Pi^{n+1} := \Pi^{t_{n+1}} = \{\pi(t_n + \Delta t) : \pi \in \Pi^{t_n}\} \cup \{\pi \in \Pi : t_\pi \in (t_n, t_{n+1})\} \tag{44}$$

where in an abuse of notation $\pi(t_n + \Delta t)$ is a new particle with position $x_\pi(t + \Delta t)$, weight $w_\pi(t + \Delta t)$ and direction of flight Ω_π for some $\pi \in \Pi^{t_n} = \Pi^n$. Thus particles Π^{n+1} can be obtained by updating the weight and position of particles in Π^n and sampling new particles with birth times in (t_n, t_{n+1}) . From (43) we also obtain

$$\langle u \rangle_{\text{MC}}(\mathbf{x}, t) = \sum_{\pi \in \Pi} \int_{t_n}^{t_{n+1}} w_\pi(t) \mathbb{1}_C(x_\pi + (t - t_\pi) \Omega_\pi) \mathbb{1}_{[0,t]}(t_\pi) dt \tag{45}$$

With this formulation, particle weights $w_\pi(t)$ decrease exponentially at a rate proportional to the absorption, given by λ_t . This approach is known as the implicit capture method. It avoids the need to sample absorption times explicitly and, at the same time, reduces statistical noise and simplifies the implementation [14, Page 168][25, Chapter 22].

The Monte Carlo simulation of (34) from t_n to t_{n+1} is based on (43) and proceeds according to the following steps:

1. Let \mathcal{P} be a partition of X into disjoint cells C . For each $C \in \mathcal{P}$, calculate the total weight W^C of new particles generated in C by the source during the interval (t_n, t_{n+1}) :

$$W^C = \frac{1}{\Delta t} \frac{1}{4\pi} \int_{\mathbb{S}^2} \int_{t_n}^{t_{n+1}} \int_C s(\mathbf{x}, \Omega, t) d\Omega d\mathbf{x} dt, \tag{46}$$

where $\Delta t = t^{n+1} - t^n$, and let $W = \sum_{C \in \mathcal{P}} W^C$. Let N_p be the input for the total number of new particles to sample during the interval (t_n, t_{n+1}) . Then for each $C \in \mathcal{P}$, sample

$$N_p^C = \text{floor} \left[\frac{W^C}{W} \right] \tag{47}$$

particles and assign them each a weight $w = W/N_p^C$. The total number of particles in the system at this time is $\hat{N}_p^{n+1} = \hat{N}_p^{n+1} + \sum_{C \in \mathcal{P}} N_p^C$. Each new particle p is assigned a position \mathbf{x}_π sampled uniformly from C and a birth time $t^{n+1} - \tau_\pi$, where τ_π is sampled uniformly from $(0, \Delta t)$. Each particle p is also assigned an angle Ω_π . For isotropic sources, Ω_π is sampled uniformly from \mathbb{S}^2 . For non-isotropic sources, (such as the boundary source for the hohlraum problem in Section 3.3), the sampling distribution must be consistent with the angular dependence. The particles, including their space, angle, and time coordinates, are added to the current particle list.

2. Move each particle π in the current particle list from \mathbf{x}_π to $\mathbf{x}_\pi + \tau_\pi \Omega_\pi$ and update its weight to $w_\pi \leftarrow w_\pi(t_{n+1})$. The number of particles in the system will have to be adjusted accordingly. Reset its remaining time to $\tau_\pi \leftarrow \Delta t$. For each cell $C \in \mathcal{P}$, update the sum in (43) and (45) according the time spent in C during the interval (t_n, t_{n+1}) .
3. To reduce computational effort and memory, at the end of each time step any particle p with weight $w_\pi < w_{\text{kill}}$ will be dropped with a probability of $p_{\text{kill}} > 0$. Here $w_{\text{kill}} > 0$ is a user-defined parameter, called the ‘killing weight’, and $p_{\text{kill}} = (1 - w_\pi/w_{\text{kill}})$. To preserve the total mass in the system, any particle p with weight $w < w_{\text{kill}}$ that survives this ‘Russian roulette’ [25, Chapter 22] will have its weight readjusted to $w/(1 - p_{\text{kill}})$.

Table 1
Pseudocode parameters.

User defined parameters	
N_x	number of Cartesian cells along each dimension
N	order of discrete ordinates
N_p	number of new particles (up to rounding) generated from source
Δt	time step
δ	tolerance of iteration
ω_q	Gauss-Legendre weights
Material parameters	
$\lambda_t, \lambda_a, \lambda_s$	total, absorption and scattering crosssection
Additional notation	
$\mathbf{A}_q^{(i,j)}, \mathbf{P}_q^{(i,j)}, \mathbf{R}^{(i,j)}, \mathbf{M}_q^{(i,j)}, \mathbf{B}_q^{(i,j)}, \mathbf{S}_q^{(i,j)}$	matrices defined in Section 2.3
$\mathcal{U}(X)$	uniform distribution on set X .

2.5. Pseudocode

The algorithms that we use for our numerical results are detailed in Algorithms 1-3. The S_N method is given in Algorithm 1. The hybrid method is given in Algorithm 3. It requires Algorithm 1 for the collided component and Algorithm 2 for the Monte Carlo update. A listing of the notation used these algorithms is provided in Table 1.

Algorithm 1 S_N -algorithm: Propagate solution from t_k to t_{k+1} .

```

Input:  $u^n, s_q$                                 ▷ coefficients of solution  $u_q^k$  from previous step and source
Require:  $\lambda_t, \lambda_a, \lambda_s$                 ▷ Material properties
Require:  $\Delta t, \{C_{i,j}\}_{i,j=1}^{N_x}, \{\Omega_q, w_q\}_{q=1}^{N_\Omega}$     ▷ Discretization parameters
Require:  $\delta$                                   ▷ Convergence tolerance
1:  $\alpha_q^{(i,j)}(t_n)$  such that  $u_q^n(\mathbf{x}) = \sum_{|\mathbf{k}|_\infty \leq 1} a_{q,\mathbf{k}}^{(i,j)} \phi_{\mathbf{k}}^{(i,j)}(\mathbf{x})$  for  $\mathbf{x} \in C_{i,j}$ 
2:  $s_q \leftarrow s_q + \frac{1}{\Delta t} \alpha_q(t_n)$           ▷ Initialize source
3:  $\alpha_q \leftarrow \alpha_q(t_n)$                       ▷ Initialize coefficients
4:  $err = \delta + 1$ 
5: while  $err > \delta$  do
6:    $\beta_q \leftarrow \alpha_q$                             ▷ Store old coefficients
7:    $\bar{\alpha} \leftarrow \sum_{q=1}^{N_\Omega} w_q \alpha_q$ 
8:   for  $q \in \{1, \dots, N_\Omega\}$  do
9:     for  $(i,j) \in \{1, \dots, N_x\}^2$  do           ▷ Sweep through cells in direction  $\Omega_q$ 
10:     $\alpha_q^{(i,j)} \leftarrow \left( \mathbf{A}_q^{(i,j)} + \mathbf{P}_q^{(i,j)} \right)^{-1} \left( \mathbf{R}^{(i,j)} \bar{\beta}^{(i,j)} + \mathbf{M}_q^{(i,j)} \alpha_q^{(i,j)} + \mathbf{B}_q^{(i,j)} + \mathbf{S}_q^{(i,j)} \right)$       ▷ Update coefficients
11:   end for
12: end for
13:  $err = \max_q ||\alpha_q - \beta_q||_\infty$             ▷ Discrepancy between iterations
14: end while
15: return  $u_q^{n+1}(\mathbf{x}) = \sum_{|\mathbf{k}|_\infty \leq 1} a_{q,\mathbf{k}}^{(i,j)} \phi_{\mathbf{k}}^{(i,j)}(\mathbf{x}), \langle u_q^{n+1} \rangle_{SN}(\mathbf{x}) = \sum_{|\mathbf{k}|_\infty \leq 1} \tilde{a}_{\mathbf{k}}^{(i,j)} \phi_{\mathbf{k}}^{(i,j)}(\mathbf{x})$  for  $\mathbf{x} \in C_{i,j}$ 

```

3. Numerical results

In this section, we compare simulation results from the Monte Carlo- S_N hybrid method to those from a standard, monolithic S_N method. The goal is to demonstrate that the hybrid method provides a more efficient approach. The S_N computations for the monolithic method and for the collided component of the hybrid rely on product quadrature sets on the sphere [39,4].

We consider three well known test problems: the line source problem [17], the lattice problem [8], and the linearized hohlraum problem [9,8]. The specifications for each problem are provided in the following subsections. They are all formulated in a geometry for which $\partial_z \Psi = 0$. This means that they can be reduced to two dimensions in physical space and, by an abuse of notation, we write $\Psi(\mathbf{x}, \Omega, t) = \Psi(x, y, \Omega, t)$. A further consequence of the geometry is that product quadrature on the sphere can be reduced to just the upper hemisphere, in which case $N_\Omega = N^2$. The time step for each problem is tied to the grid resolution via the ratio $CFL = \Delta t / \Delta x$. For all calculations shown below, the iteration tolerance δ (see Algorithm 1) is set to 10^{-4} .²

² While not shown here, we also tested several runs using $\delta = 10^{-8}$, which leads to negligible improvements in accuracy when compared to the results shown below.

Algorithm 2 MC-algorithm: Propagate solution form t_k to t_{k+1} .

Input: Π^{n+1} with $u^n(\mathbf{x}, \Omega) = \sum_{\pi \in \Pi^n} w_\pi \delta(\mathbf{x} - \mathbf{x}_\pi) \delta(\Omega - \Omega_\pi)$

$s(\mathbf{x}, \Omega, t)$ ▷ previous particle distribution and source

Require: λ_t

Require: $\Delta t, \{C_{i,j}\}_{i,j}^{N_x}, N_p$ ▷ Discretization parameters

Require: w_{kill} ▷ killing weight

- 1: **for** $(i, j) \in \{1, \dots, N_x\}^2$ **do**
- 2: $W^{C_{i,j}} = \frac{1}{|C_{i,j}| \Delta t} \int_{t_n}^{t_{n+1}} \int_{C_{i,j}} s(\mathbf{x}, \Omega, t) d\Omega dx dt$
- 3: **end for**
- 4: $W \leftarrow \sum_{i,j} W^{C_{i,j}}$ ▷ Calculate total source
- 5: **for** $(i, j) \in \{1, \dots, N_x\}^2$ **do**
- 6: $N_p^{C_{i,j}} \leftarrow \left\lfloor \frac{W^{C_{i,j}} N_p}{W} \right\rfloor$ ▷ Number of particles on each cell
- 7: **end for**
- 8: $N_p \leftarrow \sum_{i,j} N_p^{C_{i,j}}$ ▷ number of new particles
- 9: $w \leftarrow \frac{W}{N_p}$
- 10: **for** (i, j) with $N_p^{C_{i,j}} > 0$ **do** ▷ sample new particles from source
- 11: **for** $k \in \{1, \dots, N_p^{C_{i,j}}\}$ **do**
- 12: Generate new particle π with
- 13: $\mathbf{x}_\pi \sim \mathcal{U}(C_{i,j})$ ▷ Draw particle's position
- 14: $(\Omega_x, \Omega_y, \Omega_z) \sim \frac{1}{W^{C_{i,j}} |C_{i,j}| \Delta t} \int_{t_n}^{t_{n+1}} \int_{C_{i,j}} s(\mathbf{x}, \Omega, t) dx dt$ ▷ Draw particle's direction of flight
- 15: $\Omega_\pi \leftarrow (\Omega_x, \Omega_y)$
- 16: $w_\pi \leftarrow w$ ▷ Assign particle weight
- 17: $\Pi^* \leftarrow \Pi^* \cup \{\pi\}$ ▷ Add new particle to existing
- 18: **end for**
- 19: **end for**
- 20: **for** $\pi \in \Pi^l \cup \Pi^*$ **do** ▷ Move particles
- 21: **if** $\pi \in \Pi^l$ **then**
- 22: $\tau_\pi \leftarrow \Delta t$ ▷ remaining time for particles from prev. step
- 23: **else**
- 24: randomly draw $\tau_\pi \sim \mathcal{U}([0, \Delta t])$ ▷ remaining time for particles generated this time step
- 25: **end if**
- 26: $\mathbf{x}_\pi \leftarrow \mathbf{x}_\pi + \tau_\pi \Omega_\pi$ ▷ Update particle's position
- 27: **for** (i, j) with $C_{i,j} \cap \{\mathbf{x}_\pi + t \Omega_\pi : t \in [0, \tau_\pi]\} \neq \emptyset$ **do** ▷ All cells intersected by particle's trajectory
- 28: $\Phi_{i,j} \leftarrow \Phi_{i,j} + w_\pi \int_0^{\tau_\pi} \exp\left(-\int_0^t \lambda_a(\mathbf{x}_\pi + t' \Omega_\pi) dt'\right) \mathbb{1}_{C_{i,j}}(\mathbf{x}_\pi + t \Omega_\pi) dt$ ▷ Update Φ
- 29: **end for**
- 30: $w_\pi \leftarrow w_\pi \exp\left(-\int_0^{\tau_\pi} \lambda_a(\mathbf{x}_\pi + t \Omega_\pi) dt\right)$ ▷ Update particle weight
- 31: **if** $\mathbf{x}_\pi \in X$ **then**
- 32: $\Pi^{n+1} \leftarrow \Pi^{n+1} \cup \{\pi\}$ ▷ Remove particles that left domain
- 33: **end if**
- 34: **end for**
- 35: **for** $\pi \in \Pi^{n+1}$ with $w_\pi < w_{\text{kill}}$ **do** ▷ Russian roulette
- 36: $r \sim \mathcal{U}([0, 1])$
- 37: **if** $r > \frac{w_\pi}{w_{\text{kill}}}$ **then** ▷ Determine survival of particle
- 38: $\Pi^{n+1} \leftarrow \Pi^{n+1} \setminus \{\pi\}$
- 39: **else**
- 40: $w_\pi \leftarrow w_{\text{kill}}$ ▷ Update surviving particle's weight to approx. preserve total mass
- 41: **end if**
- 42: **end for**
- 43: **return** $\Pi^{n+1}(\mathbf{x}, \Omega) = \sum_{\pi \in \Pi^{n+1}} w_\pi \delta(\mathbf{x} - \mathbf{x}_\pi) \delta(\Omega - \Omega_\pi)$ and $\langle u^{n+1} \rangle_{\text{MC}}(\mathbf{x}) = \Phi(\mathbf{x})$

For each problem, we assess the accuracy of the numerical solution and the efficiency with which it is obtained. To quantify the accuracy we compare our results to a reference solution. For the line source problem the reference is the semi-analytic solution from [18]; see also [17]. For the other two test problems, the reference is a high-resolution hybrid solution based solely on S_N discretization with a triangular-based quadrature referred to as T_N [36] for both collided and uncollided component, combined with a DG discretization in space and integral deferred correction in time [11].

Accuracy for the MC- S_N hybrid and the monolithic S_N method is measured in terms of the relative L^2 -difference in the scalar flux $\Phi = \langle \Psi \rangle$ at a given final time t_{final} . Given the numerical solution Φ_{num} and the reference Φ_{ref} :

$$\Delta = \frac{\|\Phi - \Phi_{\text{ref}}\|_{L_2}}{\|\Phi_{\text{ref}}\|_{L_2}}, \quad (48)$$

where L^2 -norm is approximated by $h^2 \sum_{C_{i,j}} \Phi_{i,j}^2$ and $\Phi_{i,j}$ is the average on the cell $C_{i,j}$. Because our implementation of the hybrid method and the S_N method are not run-time optimized, we use a complexity measure which counts the number of times a particle

Algorithm 3 H_{MC-S_N} -algorithm: Propagate solution from t_k to t_{k+1} .

Input: Π^n with $u^n(\mathbf{x}, \Omega) = \sum_p w_p \delta(\mathbf{x} - \mathbf{x}_p) \delta(\Omega - \Omega_p)$

▷ previous particle distribution and source

$s(\mathbf{x}, \Omega, t)$

Require: $\hat{\lambda}_t, \lambda_t, \lambda_a, \lambda_s$

Require: $\Delta t, \{C_{i,j}\}_{i,j=1}^{N_x}, \{\Omega_q, w_q\}_{q=1}^{N_\Omega}, N_p$

Require: δ, w_{kill}

- 1: $\Pi_u^{n+1}, \langle u_u^{n+1} \rangle_{MC} \leftarrow MC(\Pi^n, s)$
- 2: for $q \in \{1, \dots, N_\Omega\}$ do
- 3: $s_q \leftarrow \lambda_s \Phi_u$
- 4: end for
- 5: $u_c^{n+1}, \langle u_c^{n+1} \rangle_{SN} \leftarrow S_n(\mathbf{0}, s_q)$
- 6: $s \leftarrow \lambda_s (\langle u_c^{n+1} \rangle_{SN} + \langle u_u^{n+1} \rangle_{MC})$
- 7: $\Pi_R, \langle u_R^{n+1} \rangle_{MC} \leftarrow MC(0, s)$
- 8: $\Pi^{n+1} \leftarrow \Pi_u^{n+1} \cup \Pi_R$
- 9: $\langle u^{n+1} \rangle_{MC} \leftarrow \langle u_u^{n+1} \rangle_{MC} + \langle u_R^{n+1} \rangle_{MC}$
- 10: return $\Pi^{n+1}, \langle u^{n+1} \rangle_{MC}$

▷ Material properties

▷ Discretization parameters

▷ Convergence tolerance and killing weight

▷ Monte Carlo for uncollided

▷ turn $\langle u_u^{n+1} \rangle_{MC}$ into source for S_N

▷ S_N for collided

▷ sources for Relabelling

▷ Monte Carlo as relabeling

is moved or a *DG* unknown is updated in the course of a sweep. Let N^c be the level of the quadrature for the collided component of the hybrid. Then the complexity of the monolithic method is

$$\mathbb{C}_{S_N} = \frac{4}{\# \text{ of Legendre coefficients}} \times \frac{N_\Omega}{\# \text{ of ordinates}} \times \frac{N_x^2}{\# \text{ of cells}} \times \frac{N_i}{\# \text{ of source iterations}} \times \frac{T}{\Delta t} \quad (49a)$$

while the complexity of the hybrid is

$$\mathbb{C}_{\text{hybrid}} = \frac{(N_u + N_R)}{\substack{\text{avg. \# of particles moved} \\ \text{for uncollided}}} \times \frac{T}{\# \text{ of time steps}} + \mathbb{C}_{S_{N^c}} \quad (49b)$$

\uparrow \uparrow \uparrow \uparrow \uparrow
 avg. # of particles moved in
moved in
for relabeling S_N complexity
of collided equation

For convenience, we set $N_{MC}^{\text{tot}} = (N_u + N_R) \frac{T}{\Delta t}$ so that $\mathbb{C}_{\text{hybrid}} = N_{MC}^{\text{tot}} + \mathbb{C}_{S_{N^c}}$. N_u is the sum of particles added to the system N_p and the average number of particles still in flight from the previous time step N_{prev} , i.e. $N_u = N_p + N_{prev}$, while N_R is the number of particles added in the relabeling, which we set to be $N_R = N_p$.

3.1. The line source problem

In the line source problem, an initial pulse of uniformly distributed particles is emitted from the line $\ell = \{(x, y, z) : x = y = 0\}$ into the surrounding domain $X = \mathbb{R}^3$, which contains a purely scattering material with $\sigma_s = \sigma_t = 1$. Because the geometry of the domain and initial condition are invariant in z , the spatial domain can be reduced to \mathbb{R}^2 . In this two-dimensional setting, the initial condition can be represented by an isotropic delta function $\frac{1}{4\pi} \delta(x, y)$, but to reduce numerical artifacts, we use a mollified version of the initial condition:

$$\Psi_0(x, y, \Omega) = \frac{1}{4\pi} \frac{1}{2\pi\zeta} \exp\left(-\frac{x^2 + y^2}{2\zeta}\right), \quad \zeta = 0.03. \quad (50)$$

Meanwhile, the computational domain is restricted to the square $[-1.5, 1.5]^2$ and equipped with zero inflow boundary conditions.

We perform monolithic S_N and MC- S_N hybrid simulations at various spatial and angular resolutions. The spatial domain is subdivided into equal $N_x \times N_x$ square cells with $N_x \in \{51, 101, 201\}$. For the S_N -runs we let $N \in \{4, 8, 16, 32\}$, resulting in $N_\Omega = N^2$ ordinates on the northern hemisphere of \mathbb{S}^2 . The collided part of the hybrid algorithm employs an S_N method with $N \in \{4, 8\}$. In the hybrid method, the number of particles was also changed between runs. At time $t = 0$ an initial pulse of N_p particles distributed according to (50) is added to the system. Since the source term is zero, new particles are only added due to relabeling. Thus the number of particles newly inserted into the system is roughly N_p per time step where $N_p = 10^k$ and $k \in \{2, 3, 4, 5, 6\}$.

Due to rounding and particles being dropped via Russian roulette, the exact number of particles inserted into the system varies slightly. The killing weight is fixed at $w_{kill} = 10^{-15}$. The CFL is also fixed at 0.5 across all runs. The reference solution is the semi-analytic solution from [18]; see also [17].

Fig. 1 depicts several approximations of the scalar flux Φ at $t_{\text{final}} = 1$, calculated using the S_N method and the hybrid method. The solutions calculated using the S_N method clearly show ray-effects that only get resolved after a significant increase in the angular resolution. No such effects are seen in the hybrid solutions. The hybrid solutions do contain some noise, as the particle count is relatively low, but they preserve the symmetry of the problem up to a reasonable error. Unlike the S_N -method, the hybrid is able to capture the wavefront of unscattered particles traveling away from the center with speed 1.

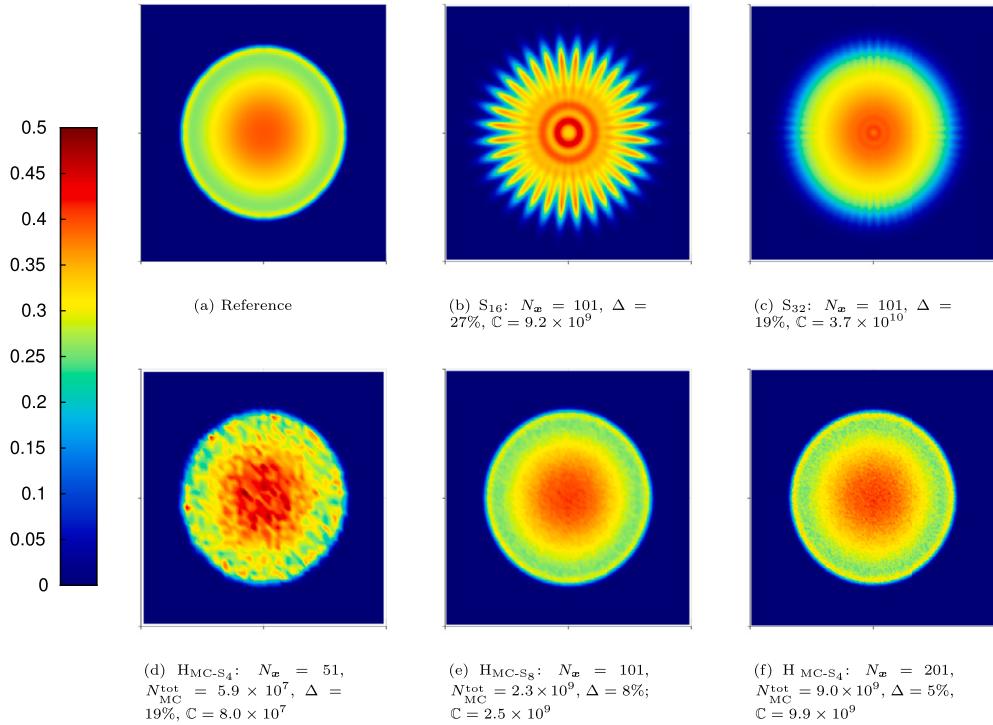


Fig. 1. Line Source problem: Numerical approximation of the scalar flux Φ at $t_{\text{final}} = 1$ with CFL 0.5. Each numerical solution is characterized by a relative L^2 difference Δ with respect to the reference, defined in (48), and a complexity \mathbb{C} , defined in (49a) for the monolithic S_N method and (49b) for the hybrid.

Fig. 2 shows L_2 -errors of the numerical solutions in log scale; the same trends are apparent. While the hybrid solutions mainly suffer from noise due to the stochastic nature of the Monte Carlo method, the S_N method has strong ray-effects and struggles to capture the analytic solution at the wavefront.

A more systematic analysis of the numerical results is presented in Fig. 3. This plot shows the relative L_2 -error Δ of various runs versus their respective computational complexity \mathbb{C} . For the hybrid method, increasing the angular resolution N in the collided component yields a marginal improvement at best. However, changes in the particle number N_p for the uncollided component have a significant impact. For the S_N method, on the other hand, increasing the angular resolution yields a significant improvement in the accuracy. For smaller values of N , increasing the spatial resolution may actually increase the error. This is especially apparent in Fig. 3 for the S_4 and S_8 results. For a fixed angular resolution, additional spatial accuracy will begin to resolve the ray effect anomalies in the solution. Conversely, S_N results with lower spatial resolution benefit from error cancellation due to the numerical diffusion smoothing ray effects. The hybrid may also have larger errors if the particles per cell are too low.

Overall the hybrid method outperforms the monolithic S_N method. For example, the hybrid error can match the most refined S_N calculation ($N = 32$) with a complexity that is roughly 2-3 orders of magnitude smaller; compare Figs. 2 (c) and (d). In fact the hybrid method can obtain an error with half the size with a complexity that is an order of magnitude less. In general hybrid runs tend to be 3-4 times more accurate than their S_N counterparts of similar complexity.

3.2. The lattice problem

In the lattice problem, a checkerboard of highly absorbing material is embedded in a scattering material with a central source. The layout of this problem along with its material parameters can be found in Fig. 4. The computational domain is a 7×7 rectangle with zero inflow data at the boundaries. The center square (red) contains an isotropic particle source, while the blue squares are pure absorbers. The red and white squares are purely scattering with $\sigma_s = \sigma_t = 1$. The initial condition is identically zero everywhere in the domain.

We perform S_N and hybrid runs with varying spatial and angular resolution. The spatial domain is subdivided into equal $N_x \times N_x$ square cells with $N_x \in \{56, 112, 224\}$. For the S_N -runs we use $N \in \{4, 8, 16, 32\}$, resulting in $N_\Omega = N^2$ ordinates on the northern hemisphere of \mathbb{S}^2 . The collided part of the hybrid algorithm employs an S_N method with $N \in \{4, 8\}$. In the hybrid method the number of particles is also changed between runs. The number of particles newly inserted into the system every time step is roughly $2 \times N_p$ where $N_p = 10^k$ for $k \in \{2, 3, 4, 5, 6\}$.³

³ The factor of 2 is because N_p particles are used for the uncollided equations (13) and N_p particles are used for the relabeling (15).

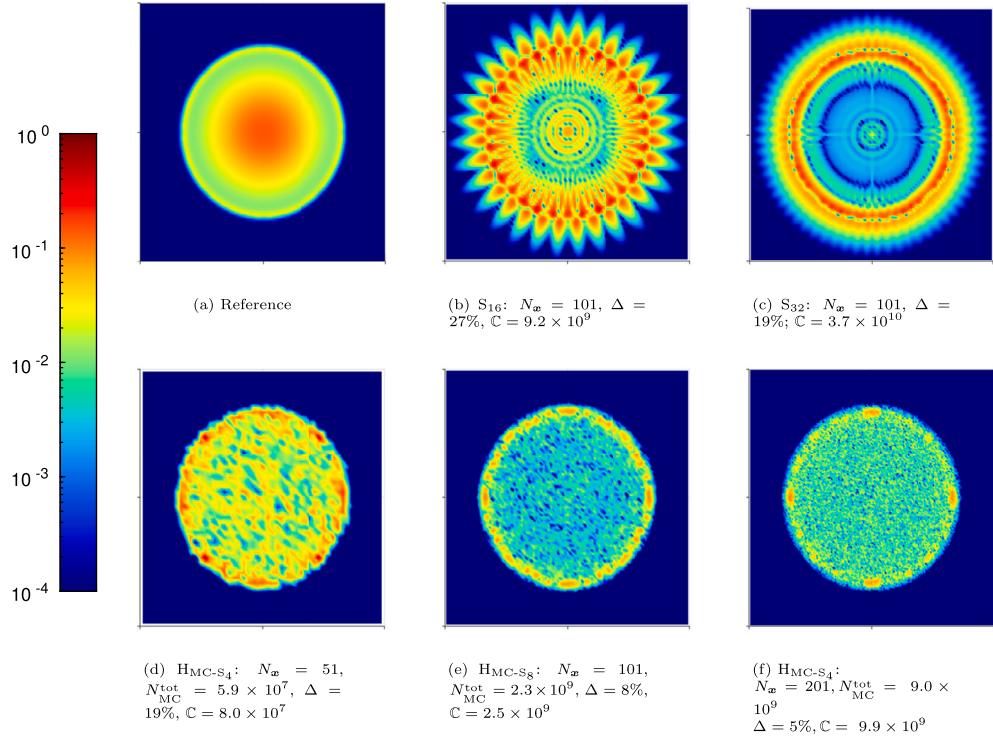


Fig. 2. Line Source problem: Absolute difference between the analytical solution and various numerical solutions to the at $t = 1$ with CFL 0.5. Each numerical solution is characterized by a relative L^2 difference Δ with respect to the reference, defined in (48), and a complexity C , defined in (49a) for the monolithic S_N method and (49b) for the hybrid.

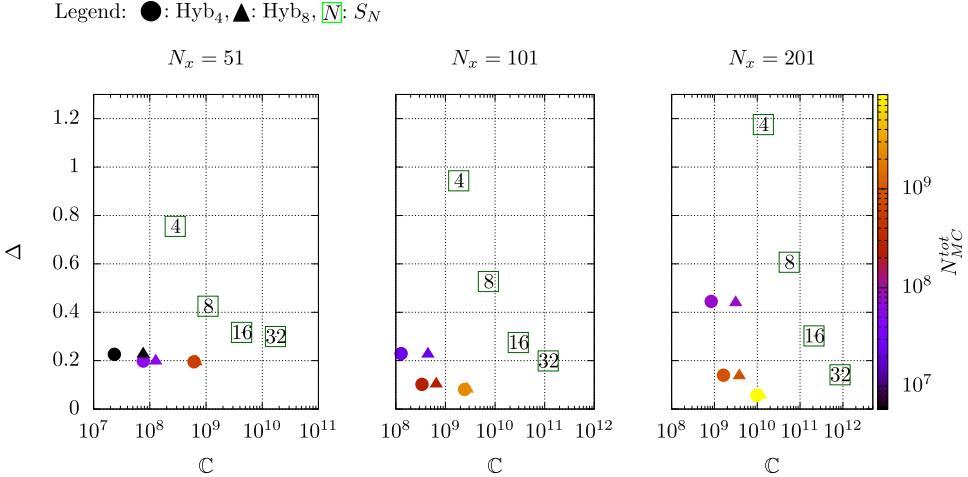


Fig. 3. Line source problem: Relative L^2 -difference Δ vs. complexity C for the scalar flux Φ , using the hybrid method with S_4 (filled circle markers), the hybrid with S_8 (filled triangle markers) and the monolithic S_N method (empty, green markers). Points that are down and to the left are more efficient. All methods were run for three different spatial resolutions $N_x = 51$ (left), $N_x = 101$ (middle), $N_x = 201$ (right). Coloring of the hybrid data points corresponds to the total number of particles N_{MC}^{tot} according to the colorbar. The S_N method was run for $N \in \{4, 8, 16, 32\}$, and each S_N data point is assigned a numerical label according to the value of N . The formula for Δ is given in (48) which the complexity C is given by (49a) for the monolithic S_N method and by (49b) for the hybrid. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

The killing weight is fixed at $w_{kill} = 10^{-15}$. All runs are performed to a final time of $t_{final} = 3.2$ and the CFL is kept fixed at 25.6. The reference solution is a S_{96} - S_{16} hybrid (meaning a S_{96} for the uncollided component and S_{16} for the collided component) using a T_N quadrature in angle, a third-order DG method in space on a 448 by 448 grid, and a defect correction time integrator [10].

Selected results for the scalar flux Φ are depicted in Fig. 5, and the relative L_2 -error for these same solutions is depicted in Fig. 6. While the hybrid solutions are not completely free of ray-effects, these effects are much more pronounced in the S_N runs. A more rigorous analysis of the performance of the two algorithm in dependence of their respective parameters can be seen in Fig. 7. This

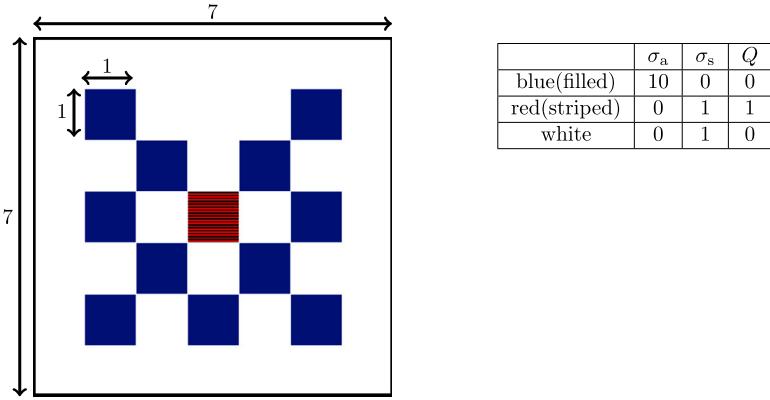
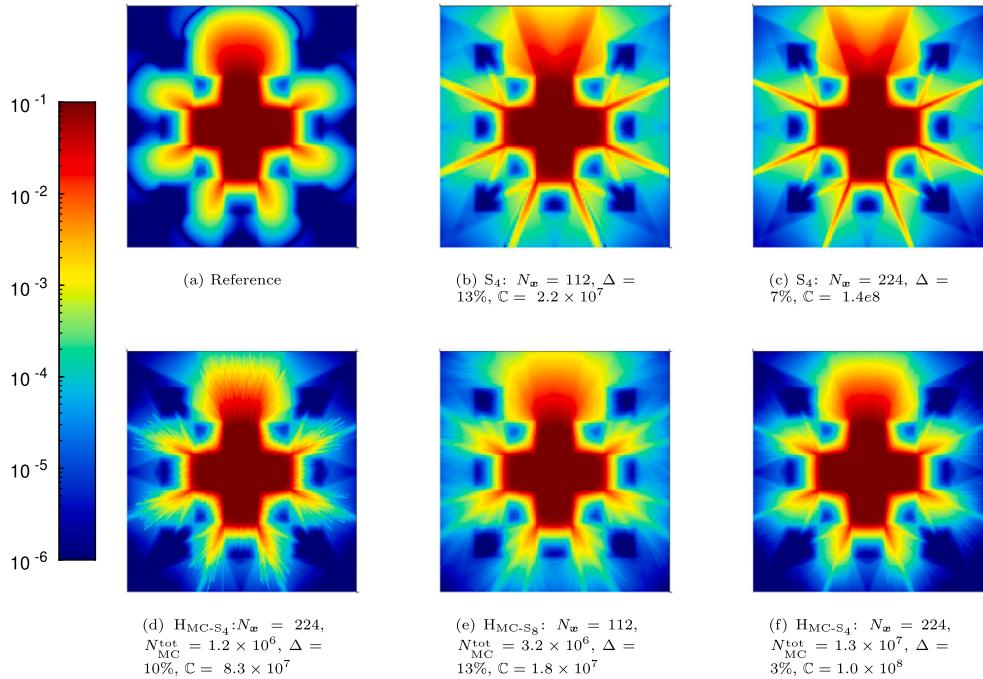


Fig. 4. Geometric layout and table of material properties for the Lattice problem.

Fig. 5. Lattice problem: Numerical solutions at $t = 3.2$ using a CFL of 25.6. Each numerical solution is characterized by a relative L^2 difference Δ with respect to the reference, defined in (48), and a complexity C , defined in (49a) for the monolithic S_N method and (49b) for the hybrid.

plot shows the relative error Δ of various runs in dependence of their complexity C . It is noteworthy that for this test problem the accuracy is mostly independent of the angular resolution, but depends significantly on the spatial resolution. Increasing the overall particle count in the hybrid method is most effective at higher spatial resolution; compare for example the vertical separation in colored triangles vs. colored circles vs. colored squares in Fig. 7.

It turns out that increasing the number of particles in the hybrid algorithm does not necessarily increase the algorithm complexity. This is because with increased particle count the iterative solver for the collided components often needs fewer iterations. In cases where the complexity is dominated by these iterations, an increase in particles can even cause a decrease in computational complexity. Overall, Fig. 7 shows that the hybrid algorithm produces results with comparable or slightly better accuracy than the standard S_N solver, while being close to an order of magnitude of lower complexity.

3.3. The Linearized Hohlraum problem

In the Linearized Hohlraum problem [9], nonlinear coupling between particles and the material medium is approximated in a linear way by adjusting the absorption and scattering cross-sections according to the expected material temperature profile of the nonlinear problem [8]. The geometry of the setup along with the material parameters can be found in Fig. 8. The domain is $X = [0, 1.3] \times [0, 1.3]$, and the initial condition is identically zero everywhere. For boundary conditions, we assume a constant influx

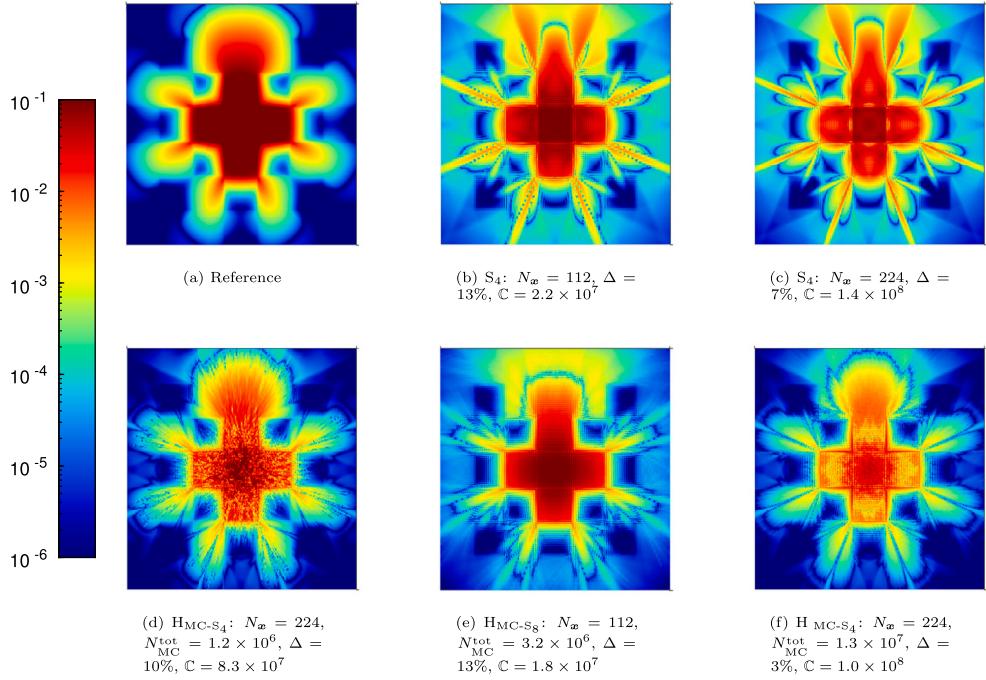


Fig. 6. Lattice problem: Absolute difference between the reference solution and various numerical solutions at $t = 3.2$ with CFL 25.6. Each numerical solution is characterized by a relative L^2 difference Δ with respect to the reference, defined in (48), and a complexity C , defined in (49a) for the monolithic S_N method and (49b) for the hybrid.

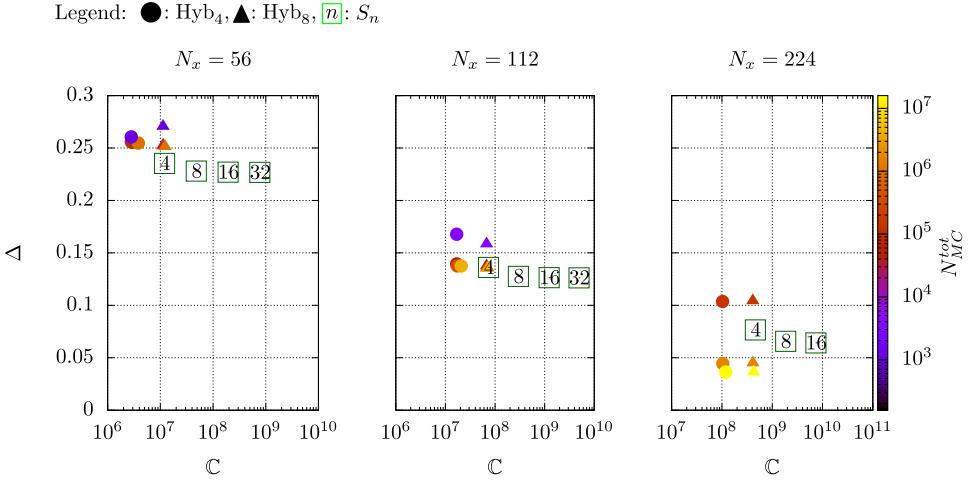


Fig. 7. Lattice problem: Relative L^2 -difference Δ vs. complexity C for the scalar flux Φ , using the hybrid method with S_4 (filled circle markers), the hybrid with S_8 (filled triangle markers) and the monolithic S_N method (empty, green markers). Points that are down and to the left are more efficient. All methods were run for three different spatial resolutions: $N_x = 56$ (left), $N_x = 112$ (middle), $N_x = 224$ (right). Coloring of the hybrid data points corresponds to the total number of particles N_{MC}^{tot} , according to the colorbar. The S_N method was run for $N \in \{4, 8, 16, 32\}$, and each S_N data point is assigned a numerical label according to the value of N . The formula for Δ is given in (48) which the complexity C is given by (49a) for the monolithic S_N method and by (49b) for the hybrid.

from the left side of the domain, i.e. $\Psi(x = 0, y, \Omega, t) = 1$ for $\Omega_x > 0$. As discussed in the appendix, this boundary condition can be treated as a surface source, modeled by setting $\Omega_x = \sqrt{\xi}$, where $\xi \sim U([0, 1])$ is sampled uniformly on $[0, 1]$. The spatial distribution along the boundary is sampled uniformly.

We again perform S_N and hybrid runs with varying spatial and angular resolution. The spatial domain is subdivided into equal $N_x \times N_x$ square cells with $N_x \in \{52, 104, 208\}$. For the S_N -runs we use $N \in \{4, 8, 16, 32\}$, resulting in $N_\Omega = N^2$ ordinates on the northern hemisphere of \mathbb{S}^2 . The collided part of the hybrid algorithm employs an S_N method with $N \in \{4, 8\}$. In the hybrid method the number of particles is also changed between runs, but the killing weight remains fixed at $w_{kill} = 10^{-15}$. The number of particles newly inserted into the system every time step is 2×10^k for $k \in \{2, 3, 4, 5, 6\}$. All runs are performed to a final time of $t_{final} = 2.6$ and

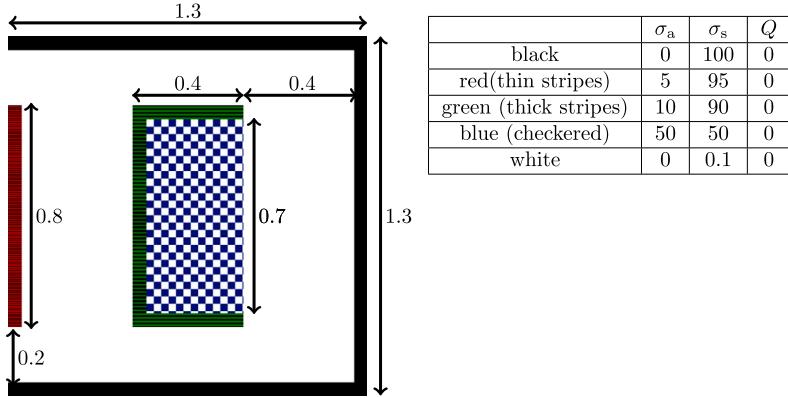


Fig. 8. Geometric layout and table of material parameters for the Linearized Hohlraum problem. All walls have a thickness of 0.05.

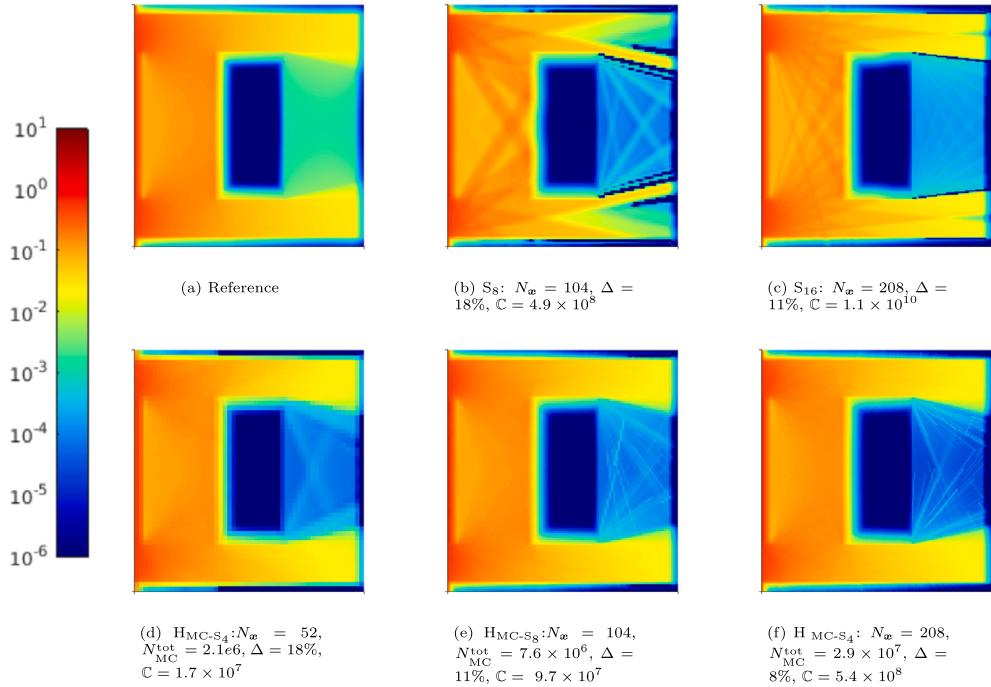


Fig. 9. Linearized Hohlraum problem: Numerical solutions at $t = 2.6$ with CFL of 52. Each numerical solution is characterized by a relative L^2 difference Δ with respect to the reference, defined in (48), and a complexity \mathbb{C} , defined in (49a) for the monolithic S_N method and (49b) for the hybrid.

the CFL is kept fixed at 52. The reference solution is a S_{96} - S_{16} hybrid using a T_N quadrature in angle, a third-order DG method in space on a 448×448 grid, and a defect correction time integrator [10].

In Fig. 9, we show densities of a few select runs, calculated using S_N and hybrid methods. The logarithms of the corresponding relative L_2 -errors are depicted in Fig. 10. As before, the S_N solutions suffer from ray-effects that are marginally reduced as the number of angle increases. Meanwhile, most of the disparities between the reference and hybrid solutions can be attributed to stochastic noise. The hybrid has a mix of ray effects from the collided equation solve and particle tracks from the uncollided equation solve on the backside of the hohlraum. However, these errors here are on the order of 10^{-2} - 10^{-3} , which is much smaller than the errors in other parts of the domain.

Detailed comparisons between the relative error against the computational complexity are depicted in Fig. 11. As before, increasing the angular resolution in the collided equation does not benefit the accuracy of the hybrid method. Increasing particles also has less effect than in the previous problems. The S_N solutions benefit most from finer spatial resolution, while the angular resolution does not matter as much. Spatial resolution also plays the biggest role for the hybrid. We do observe that for small particle counts (the purple points in the figure) increasing resolution can actually increase the error. This is explained by the fact that an under-sampled MC calculation does not benefit from more spatial resolution. Nevertheless, for a fixed spatial resolution, we do observe an improvement in the error when N_p is increased.

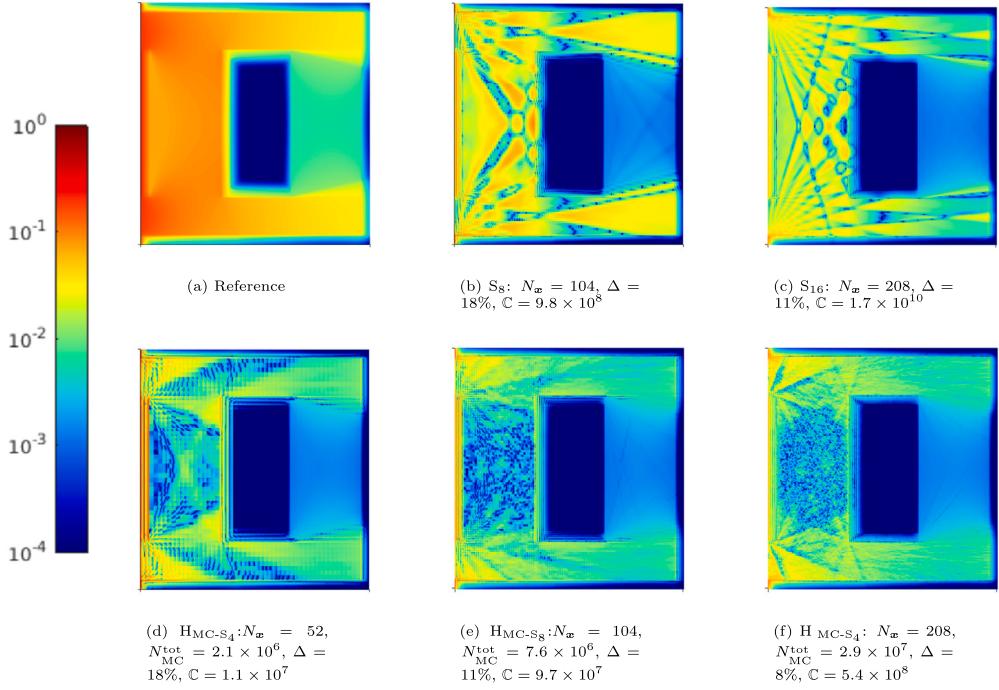


Fig. 10. Linearized Hohlraum problem: Absolute difference between reference solution and various numerical solutions at $t = 2.6$ with CFL of 52. Each numerical solution is characterized by a relative L^2 difference Δ with respect to the reference, defined in (48), and a complexity C , defined in (49a) for the monolithic S_N method and (49b) for the hybrid.

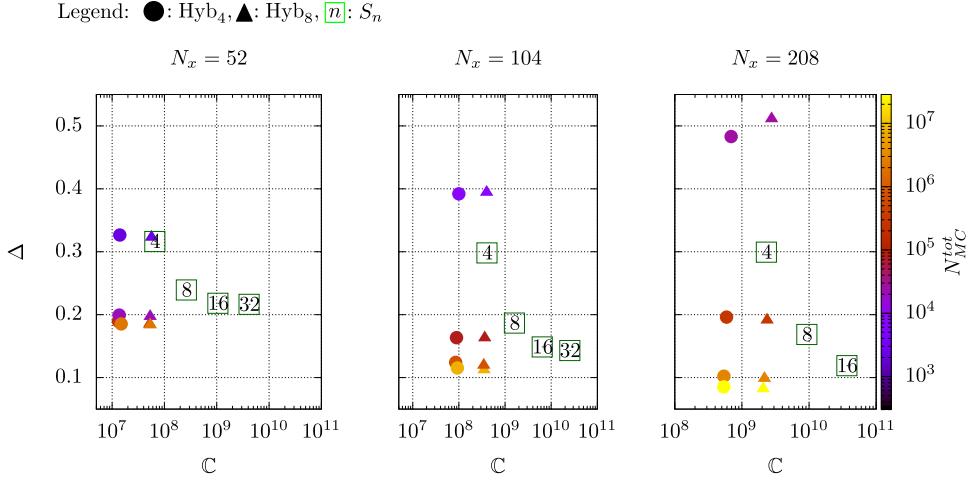


Fig. 11. Linearized Hohlraum problem: The relative L^2 -difference Δ vs. complexity C for the scalar flux Φ in the hohlraum problem, using the hybrid method with S_4 (filled circle markers), the hybrid with S_8 (filled triangle markers) and the monolithic S_N method (empty, green markers). Points that are down and to the left are more efficient. All methods were run for three different spatial resolutions $N_x = 52$ (left), $N_x = 104$ (middle), $N_x = 208$ (right). Coloring of the hybrid data points corresponds to the total number of particles N_{MC}^{tot} according to the colorbar. The S_N method was run for $N \in \{4, 8, 16, 32\}$, and each S_N data point is assigned a numerical label according to the value of N . The formula for Δ is given in (48) which the complexity C is given by (49a) for the monolithic S_N method and by (49b) for the hybrid.

Overall the hybrid runs produce solutions with comparable or better accuracy than monolithic S_N runs with the same spatial resolution. Hybrid runs using S_8 for the collided equation achieve improved accuracy at nearly the same complexity while runs using S_4 for the collided equation achieved improved accuracy with even less complexity.

4. Conclusion and discussion

In this work, we have presented a collision-based hybrid method that uses a Monte Carlo method for the uncollided solution and a discrete ordinate discretization for the collided solution. This combination of methods was originally proposed for the first collision

source strategy used in [3] in the steady-state setting. Thus this work can be considered as an extension to the time-dependent setting that requires a remap procedure after every time step.

Experimental simulations have been performed on three standard benchmarks. For each benchmark, the results demonstrate that the hybrid method is more efficient, in the sense that it achieves greater accuracy with the comparable or less complexity or is less complexity with comparable or greater accuracy. Here complexity is a measure of how many unknowns are updated during particle moves for Monte Carlo or sweeping iterations for discrete ordinates.

This work has concentrated on single-energy particle transport problems. However recent work has shown that when considering energy-dependent problems, more opportunities for hybridization arise when considering fully deterministic hybrids [41]. In those results it was shown that low-resolution in energy can be used for the collided solution as well as low-resolution in angle. With the introduction of Monte Carlo, new opportunities arise. For example, continuous energy cross-sections could be used in the uncollided portion. This could be important to treat resonances in neutron transport problems, but investigation is needed to quantify any benefits from this approach. The methodology here may also be extended to nonlinear radiative transfer using standard linearization strategies, although corrections may be needed to handle opacities that depend on radiation energy and material temperature. In addition future investigations should be made regarding the use of Monte Carlo techniques inside the high-order time accuracy methods developed for hybrid problems in [9,10].

Finally, a clear strategy for choosing of discretization parameters does not exist at this point. The selection of spatial and temporal discretization parameters is much like any other method. However, the appropriate choice for the relative degrees of freedom in angle between the uncollided and collided equation is not clear; nor is the time interval to wait until the relabeling is performed. While some work exists to understand errors introduced by the hybrid [16], the analysis is quite involved, even for a very simple case. Thus, the best approach is most likely an adaptive strategy based on a-posteriori estimates. This will be the topic of future work.

CRediT authorship contribution statement

Johannes Krotz: Writing – original draft, Visualization, Software, Conceptualization. **Cory D. Hauck:** Writing – review & editing, Writing – original draft, Supervision. **Ryan G. McClarren:** Writing – original draft, Supervision, Conceptualization.

Declaration of competing interest

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

Data availability

No data was used for the research described in the article.

Acknowledgements

J.K. gratefully acknowledges support from the 2022 National Science Foundation Mathematical Sciences Graduate Internship to conduct this research at Oak Ridge National Laboratory.

Appendix A. Boundary conditions for the hohlraum problem

Unlike the line source and lattice problems, the linearized hohlraum problem involves non-zero boundary conditions. A Monte Carlo implementation of this boundary condition is stated in [15]; here we present a derivation of the approach that is used.

In the hohlraum problem, it is assumed that $\Psi(x, y, t, \Omega) = 1$ for $x = 0$ and $\Omega_x > 0$, i.e., a constant flux of 1 along the left boundary is assumed for each incoming direction. To model this with Monte Carlo, we assume that this flux is due to a source s_b (see (37c)) located on an infinitesimal slab just left of the boundary.

Consider first a finite slab $S_a = \{(x, y) \in [-a, 0] \times [0, 1.3]\}$, where $a > 0$. We assume that $\sigma_a = \sigma_s = 0$ on S , that the source $s_b(x, y, \Omega; a) = s_b(x, \Omega; a)$ is independent of y and t , and that

$$\hat{s}_b(\Omega) := \int_{-a}^0 s_b(x, \Omega; a) dx \quad (\text{A.1})$$

is independent of a . Thus, $s_b(x, \Omega, t) \rightarrow \hat{s}_b(\Omega)\delta(x)$ as $a \rightarrow 0$.

To determine \hat{s}_b , we assume that Ψ is independent of y and t on S and satisfies the steady-state equation

$$\Omega_x \Psi_x(x, y, \Omega) = s_b, \quad (x, y) \in S, \quad \Omega \in \mathbb{S}^2, \quad (\text{A.2a})$$

$$\Psi(-a, y, \Omega) = 0, \quad y \in [0, 1.3], \quad \Omega_x > 0. \quad (\text{A.2b})$$

This formulation is consistent with (34), given the assumptions made on Ψ , s_b , and the material cross-sections. Integrating (A.2b) with respect to x and applying the boundary condition in (A.2b) gives

$$\hat{s}_b(\Omega) = \Omega_x, \quad \Omega_x > 0. \quad (\text{A.3})$$

Hence $s_b = \Omega_x \delta(x)$.

Finally, to sample particles according to the probability density $\rho(\Omega_x) \propto \hat{s}_b$ for $\Omega_x > 0$, we compute the cumulative distribution function (CDF):

$$F(\Omega_x) = \int_0^{\Omega_x} 2\mu d\mu = \Omega_x^2. \quad (\text{A.4})$$

According to the fundamental theorem of simulation [31, pp. 19–22], the correct angular distribution can be sampled by generating uniform variables $u \in [0, 1]$ and setting $\Omega_x = F^{-1}(u) = \sqrt{u}$.

References

- [1] Marvin L. Adams, Discontinuous finite element transport solutions in thick diffusive problems, *Nucl. Sci. Eng.* 137 (3) (2001) 298–333.
- [2] Timo Aila, Samuli Laine, Understanding the efficiency of ray traversal on GPUs, in: Proceedings of the Conference on High Performance Graphics 2009, 2009, pp. 145–149.
- [3] Raymond E. Alcouffe, A first collision source method for coupling Monte Carlo and discrete ordinates for localized source problems, in: Monte-Carlo Methods and Applications in Neutronics, Photonics and Statistical Physics: Proceedings of the Joint Los Alamos National Laboratory-Commissariat à l'Energie Atomique Meeting Held at Cadarache, Castle, Provence, France, April 1985, pp. 22–26, Springer, 2006, pp. 352–366.
- [4] Kendall Atkinson, Numerical integration on the sphere, *ANZIAM J.* 23 (3) (1982) 332–347.
- [5] Yousry Azmy, Enrico Sartori, Edward W. Larsen, Jim E. Morel, Advances in discrete-ordinates methodology, in: Nuclear Computational Science: A Century in Review, 2010, pp. 1–84.
- [6] Guillaume Bal, Anthony B. Davis, Ian Langmore, A hybrid (Monte Carlo/deterministic) approach for multi-dimensional radiation transport, *J. Comput. Phys.* 230 (20) (2011) 7723–7735.
- [7] Troy L. Becker, Allan B. Wollaber, Edward W. Larsen, A hybrid Monte Carlo-deterministic method for global particle transport calculations, *Nucl. Sci. Eng.* 155 (2) (2007) 155–167.
- [8] Thomas A. Brunner, Forms of approximate radiation transport, Technical Report SAND2002-1778, Sandia National Laboratories, Albuquerque, NM, United States, 2002.
- [9] Michael M. Crockatt, Andrew J. Christlieb, C. Kristopher Garrett, Cory D. Hauck, An arbitrary-order, fully implicit, hybrid kinetic solver for linear radiative transport using integral deferred correction, *J. Comput. Phys.* 346 (2017) 212–241.
- [10] Michael M. Crockatt, Andrew J. Christlieb, C. Kristopher Garrett, Cory D. Hauck, Hybrid methods for radiation transport using diagonally implicit Runge–Kutta and space-time discontinuous Galerkin time integration, *J. Comput. Phys.* 376 (2019) 455–477.
- [11] Michael M. Crockatt, Andrew J. Christlieb, Cory D. Hauck, Improvements to a class of hybrid methods for radiation transport: Nyström reconstruction and defect correction methods, *J. Comput. Phys.* 422 (2020) 109765.
- [12] Jeffery D. Densmore, Edward W. Larsen, Asymptotic equilibrium diffusion analysis of time-dependent Monte Carlo methods for grey radiative transfer, *J. Comput. Phys.* 199 (1) (2004) 175–204.
- [13] Jeffery D. Densmore, Todd J. Urbatsch, Thomas M. Evans, Michael W. Buksas, A hybrid transport-diffusion method for Monte Carlo radiative-transfer simulations, *J. Comput. Phys.* 222 (2) (2007) 485–503.
- [14] Stephen A. Dupree, Stanley K. Fraley, A Monte Carlo Primer: A Practical Approach to Radiation Transport, vol. 1, Springer Science & Business Media, 2002.
- [15] Joseph A. Fleck Jr., The calculation of nonlinear radiation transport by a Monte Carlo method, Technical report, Lawrence Livermore National Lab. (LLNL), Livermore, CA, United States, 1961.
- [16] Andrés Galindo-Olarde, Victor P. DeCaria, Cory D. Hauck, Numerical analysis of a hybrid method for radiation transport, arXiv preprint, arXiv:2306.04714, 2023.
- [17] B.D. Ganapol, Homogeneous infinite media time-dependent analytic benchmarks for x-tm transport methods development, Los Alamos National Laboratory, 1999.
- [18] C. Kristopher Garrett, Cory D. Hauck, A comparison of moment closures for linear kinetic transport equations: the line source benchmark, *Transp. Theory Stat. Phys.* 42 (6–7) (2013) 203–235.
- [19] Jean-Luc Guermond, Guido Kanschat, Asymptotic analysis of upwind discontinuous Galerkin approximation of the radiative transport equation in the diffusive limit, *SIAM J. Numer. Anal.* 48 (1) (2010) 53–78.
- [20] Weimin Han, Jianguo Huang, Joseph A. Eichholz, Discrete-ordinate discontinuous Galerkin methods for solving the radiative transfer equation, *SIAM J. Sci. Comput.* 32 (2) (2010) 477–497.
- [21] Cory D. Hauck, Ryan G. McClarren, A collision-based hybrid method for time-dependent, linear, kinetic transport equations, *Multiscale Model. Simul.* 11 (4) (2013) 1197–1227.
- [22] Vincent Heningburg, Cory D. Hauck, Hybrid solver for the radiative transport equation using finite volume and discontinuous Galerkin, arXiv preprint, arXiv: 2002.02517, 2020.
- [23] Edward W. Larsen, Jim E. Morel, Warren F. Miller Jr., Asymptotic solutions of numerical transport problems in optically thick, diffusive regimes, *J. Comput. Phys.* 69 (2) (1987) 283–324.
- [24] Elmer Eugene Lewis, Warren F. Miller, Computational Methods of Neutron Transport, 1984.
- [25] Ryan McClarren, Computational Nuclear Engineering and Radiological Science Using Python, Academic Press, 2017.
- [26] Ryan G. McClarren, Todd J. Urbatsch, A modified implicit Monte Carlo method for time-dependent radiative transfer with adaptive material coupling, *J. Comput. Phys.* 228 (16) (2009) 5669–5686.
- [27] Scott W. Mosher, Farzad Rahnema, The incident flux response expansion method for heterogeneous coarse mesh transport problems, *Transp. Theory Stat. Phys.* 35 (1–2) (2006) 55–86.
- [28] Madicken Munk, Rachel N. Slaybaugh, Review of hybrid methods for deep-penetration neutron transport, *Nucl. Sci. Eng.* (2019).
- [29] H. Park, D.A. Knoll, R.M. Rauenzahn, A.B. Wollaber, R.B. Lowrie, Moment-based acceleration of Monte Carlo solution for multifrequency thermal radiative transfer problems, *J. Comput. Theor. Transp.* 43 (1–7) (2014) 314–335.
- [30] Zhuogang Peng, Ryan G. McClarren, A high-order/low-order (holo) algorithm for preserving conservation in time-dependent low-rank transport calculations, *J. Comput. Phys.* 447 (2021) 110672.
- [31] Steve PRESSÉ, Ioannis Sgouralis, Data Modeling for the Sciences: Applications, Basics, Computations, Cambridge University Press, 2023.
- [32] Qiwei Sheng, Cory Hauck, Uniform convergence of an upwind discontinuous Galerkin method for solving scaled discrete-ordinate radiative transfer equations with isotropic scattering, *Math. Comput.* 90 (332) (2021) 2645–2669.

- [33] Yi Shi, Xiaole Han, Wenjun Sun, Peng Song, A continuous source tilting scheme for radiative transfer equations in implicit Monte Carlo, *J. Comput. Theor. Transp.* 50 (1) (2020) 1–26.
- [34] Jerome Spanier, Ely M. Gelbard, *Monte Carlo Principles and Neutron Transport Problems*, Courier Corporation, 2008.
- [35] Elad Steinberg, Shay I. Heizler, Multi-frequency implicit semi-analog Monte-Carlo (ISMC) radiative transfer solver in two-dimensions (without teleportation), *J. Comput. Phys.* 450 (2022) 110806.
- [36] C.P. Thurgood, A. Pollard, H.A. Becker, *The TN Quadrature Set for the Discrete Ordinates Method*, 1995.
- [37] John C. Wagner, Alireza Haghishat, Automated variance reduction of Monte Carlo shielding calculations using the discrete ordinates adjoint function, *Nucl. Sci. Eng.* 128 (2) (1998) 186–208.
- [38] John C. Wagner, Douglas E. Peplow, Scott W. Mosher, Fw-cadis method for global and regional variance reduction of Monte Carlo radiation transport calculations, *Nucl. Sci. Eng.* 176 (1) (2014) 37–57.
- [39] Wallace F. Walters, Use of the Chebyshev-Legendre quadrature set in discrete-ordinate codes, in: *Proc. Conf. Numerical Methods in High Temperature Physics, 1987*, pp. 2–6.
- [40] Ben Whewell, Ryan G. McClarren, Cory D. Hauck, Minwoo Shin, Multigroup neutron transport using a collision-based hybrid method, *Nucl. Sci. Eng.* (2023) 1–20.
- [41] Ben Whewell, Ryan G. McClarren, Cory D. Hauck, Minwoo Shin, Multigroup neutron transport using a collision-based hybrid method, *Nucl. Sci. Eng.* (2023).
- [42] Jeffrey Alan Willert, *Hybrid Deterministic/Monte Carlo Methods for Solving the Neutron Transport Equation and K-Eigenvalue Problem*, North Carolina State University, 2013.
- [43] Allan B. Wollaber, Four decades of implicit Monte Carlo, *J. Comput. Theor. Transp.* 45 (1–2) (2016) 1–70.
- [44] Dingkang Zhang, Farzad Rahnema, *Comet Solutions to a Stylized Bwr Benchmark Problem*, Technical report, American Nuclear Society, Inc., 2012.