

# A sweep-based low-rank method for the discrete ordinate transport equation

Zhuogang Peng<sup>a</sup>, Ryan G. McClarren<sup>a,\*</sup>

<sup>a</sup>*Department of Aerospace and Mechanical Engineering, University of Notre Dame, Notre Dame, IN, 46545, USA*

---

## Abstract

The dynamical low-rank (DLR) approximation is an efficient technique to approximate the solution to matrix differential equations. Recently, the DLR method was applied to radiation transport calculations to reduce memory requirements and computational costs. This work extends the low-rank scheme for the time-dependent radiation transport equation in 2-D and 3-D Cartesian geometries with discrete ordinates discretization in angle (SN method). The reduced system that evolves on a low-rank manifold is constructed via an “unconventional” basis update & Galerkin integrator to avoid a substep that is backward in time, which could be unstable for dissipative problems. The resulting system preserves the information on angular direction by applying separate low-rank decompositions in each octant where angular intensity has the same sign as the direction cosines. Then, transport sweeps and source iteration can efficiently solve this low-rank-SN system. The numerical results in 2-D and 3-D Cartesian geometries demonstrate that the low-rank solution requires less memory and computational time than solving the full rank equations using transport sweeps without losing accuracy.

**Keywords:** Low-rank approximation, Radiation transport, Discrete ordinates

---



---

\*Corresponding author

Email addresses: [zpeng5@nd.edu](mailto:zpeng5@nd.edu) (Zhuogang Peng), [rmccclarr@nd.edu](mailto:rmccclarr@nd.edu) (Ryan G. McClarren)

## 1. Introduction

The radiation transport equation (RTE) models the movement of particles through materials and their interactions with background media. The application of RTE among various science and engineering communities, such as optical imaging [1], neutron transport [2], rarefied gas dynamics [3], to name a few, require solving RTE accurately and efficiently. The accurate yet computationally efficient RTE solution has been an active research question for decades but remains open because of its rich dimensional spaces. The solution to the RTE is radiation intensity, which is determined by up to seven independent variables, including three in space, two in angle, one in time, and one in energy. Thus, the computation could consume a large amount of computer memory, challenging proposed exascale computing systems [4]. This work aims to design a fast and memory-efficient numerical scheme for solving the RTE.

The angular treatment is essential to solve the RTE. One commonly used numerical method to discretize the angular variable in RTE is the spherical harmonic ( $P_N$ ) method [5–7] that expresses the angular dependence of the radiation density using orthogonal bases on the unit sphere. This method has many desired properties, such as rotational invariance, but it suffers from oscillations that destroy the robustness of the method [8].

Another popular angular treatment is the discrete ordinates ( $S_N$ ) method [9] that solves the radiation intensity along with particular directions and uses quadrature to estimate moments of the intensity. Given that the direction of movement of radiation intensity is pre-selected, the resulting discrete ordinates system can be solved by a very efficient method called a transport sweep. This makes the  $S_N$  method popular in high-performance computing applications because it is computationally efficient. For this reason, we seek to develop novel methods based on  $S_N$  that is compatible with these transport sweeps.

This work follows the development of the dynamical low-rank (DLR) method applied in transport calculations to reduce the computer memory requirements and the computational cost. The DLR method aims to approximate large time-

dependent matrices determined by matrix differential equations [10]. The desired approximation has three components similar to factors in singular value decomposition (SVD). Each of them is solved by integrating the matrix differential equation projected onto the tangent space of the low-rank manifold. We refer to [11–13] for more background. In previous work, the authors applied the DLR method to transport calculations with a spherical harmonics expansion and an explicit time scheme [14]. Later, a high-order/low-order algorithm was developed in [15] to overcome the conservation loss in the low-rank evolution. For more analytical details, there is an error analysis for the backward Euler and Crank-Nicholson methods [16]. In [17] the asymptotic-preserving property is achieved by a macro-micro decomposition of the transport equation.

In this work, we develop a practical computation scheme with the discrete-ordinates model. By applying the time integrator proposed recently in [18], we avoid a substep that is backward in time, which could be unstable for dissipative problems as described in [16]. We then solve the low-rank equation with the iterative approach and transport sweeps.

The remainder of this paper begins with a brief review of the  $S_N$  formulation of the radiation transport equation. Then the low-rank representation of the  $S_N$  equations is derived. Section 4 will be the numerical implementation details, including the spatial discretization and the computational method for the resulting matrix equations. The efficiency and accuracy of our low-rank algorithms are demonstrated in section 5. Section 6 presents a discussion.

## 2. Discrete Ordinates Radiation Transport Equation

We begin with the time-dependent radiative transfer equation with one energy group given by

$$\frac{1}{c} \frac{\partial \psi(\mathbf{r}, \hat{\Omega}, t)}{\partial t} + \hat{\Omega} \cdot \nabla \psi(\mathbf{r}, \hat{\Omega}, t) + \sigma_t(\mathbf{r}) \psi(\mathbf{r}, \hat{\Omega}, t) = \frac{1}{4\pi} \sigma_s(\mathbf{r}) \phi(\mathbf{r}, t) + Q(\mathbf{r}, t). \quad (1)$$

In this equation,  $\psi(\mathbf{r}, \hat{\Omega}, t)$  is the radiation intensity with units of particles per area per steradian per time. We use the standard notation with  $\mathbf{r} = (x, y, z) \in$

$\mathbb{R}^3$  being the position, the unit angular vector  $\hat{\Omega}(\mu, \varphi) \in \mathbb{S}^2$  specified by the cosine of the polar angle  $\mu \in [-1, 1]$  and the azimuthal angle  $\varphi \in [0, 2\pi]$ , and  $t$  as the time. Additionally,  $\sigma_s(\mathbf{r})$  and  $\sigma_t(\mathbf{r})$  are isotropic scattering and total macroscopic cross-sections with units of inverse length;  $Q(\mathbf{r}, t)$  is a prescribed source. We integrate  $\psi(\mathbf{r}, \hat{\Omega}, t)$  over all angles to obtain the scalar intensity:

$$\phi(\mathbf{r}, t) = \int_{4\pi} \psi(\mathbf{r}, \hat{\Omega}, t) d\hat{\Omega}. \quad (2)$$

The scalar intensity is important because it can be used to compute reaction rate densities (e.g., absorption rate density) that determine the coupling with other physical operators in a given system.

We apply the  $S_N$  discretization to the angular variables with a finite quadrature set  $\{(\hat{\Omega}_n, w_n) | 1 \leq n \leq N\}$ , and solve Eq. (1) along these angular directions as

$$\frac{1}{c} \frac{\partial \psi_n(\mathbf{r}, t)}{\partial t} + \hat{\Omega}_n \cdot \nabla \psi_n(\mathbf{r}, t) + \sigma_t(\mathbf{r}) \psi_n(\mathbf{r}, t) = \frac{1}{4\pi} \sigma_s(\mathbf{r}) \phi(\mathbf{r}, t) + Q(\mathbf{r}, t). \quad (3)$$

where the discrete direction  $\hat{\Omega}_n$  is specified by the direction cosines  $\mu_n, \eta_n$  and  $\xi_n$ . The angular integral is approximated by quadrature rules, e.g., suppose there are  $N_\Omega$  directions, the scalar intensity can be written as

$$\phi(\mathbf{r}, t) = \sum_n^{N_\Omega} \omega_n \psi_n(\mathbf{r}, t), \quad (4)$$

Many kinds of quadrature sets can be implemented to the  $S_N$  method, such as level-symmetric [19], Legendre-Chebyshev, and Legendre-equal weight [20], to name a few. We refer to [21] for a comprehensive comparison. We use the Legendre-Chebyshev quadrature sets throughout this work because it enables a high quadrature order, which is not available with level-symmetric quadrature sets [22]. Then we have the formula for the number of discrete ordinates in each octant as  $N = \frac{SN^2}{4}$  and  $N_\Omega = Nn_o$ , where  $SN$  is the order of discrete ordinates,  $n_o$  is the number of octants, e.g.,  $n_o = 4$  for 2-D space and  $n_o = 8$  for 3-D space, and  $N$  is the number of angles per octant.

One most commonly used method to solve Eq. (3) is source iteration and transport sweep equipped with the implicit Euler method for time discretiza-

tions. To simplify the notation we define several operators that only used in this section [23]

$$L = \hat{\Omega} \cdot \nabla + \sigma_t(\mathbf{r}) \quad (5)$$

$$M\phi = \frac{1}{4\pi}\phi, \quad D\psi = \sum_{n=1}^{N_\Omega} w_n \psi_n d\hat{\Omega}, \quad S\phi = \sigma_s(\mathbf{r})\phi. \quad (6)$$

where  $L$  denotes the streaming and collision operator that operates on the angular intensity at the  $N_\Omega$  angles and outputs a vector of the same size,  $M$  is known as the moment-to-discrete operator that projects the scalar intensity to the angular intensity,  $S$  is the scattering operator, and  $D$  is the discrete-to-moment operator that represents the angular quadrature. Using these operators we obtain the abstract form of Eq. (3) as

$$\frac{1}{c\Delta t}\psi^{\ell+1} + L\psi^{\ell+1} = MSD\psi^{\ell+1} + Q + \frac{1}{c\Delta t}\psi^\ell, \quad (7)$$

where  $\psi^\ell$  is the solution for the radiation intensity at time step  $\ell$  at each angle  $\Omega_n$ . We can further simplify this equation into a quasi-steady form by defining

$$L^* = \hat{\Omega}_n \cdot \nabla + \left( \sigma_t(\mathbf{r}) + \frac{1}{c\Delta t} \right), \quad q = Q + \frac{1}{c\Delta t}\psi^\ell. \quad (8)$$

to get

$$L^*\psi^{\ell+1} = S\psi^{\ell+1} + q. \quad (9)$$

Discrete ordinates with implicit Euler time integration can be computed using an efficient algorithm called a “transport sweep,” which is highly desirable in computation. The idea is that we can solve for the unknowns by marching along the direction of the flow of particles from the given boundary conditions and iterating over a lagged scattering source. After spatial discretizations,  $(L^*)^{-1}$  is a triangular matrix for each angle so that

$$\phi^{\ell+1,c+1} = D(L^*)^{-1}MS\phi^{\ell+1,c} + Dq \quad (10)$$

can be computed by a simple lower or upper triangular solve in each angle. This iterative method is known as the source iteration (SI) method [9]. These iterations are repeated until the scalar intensity  $\phi$  converges, and we denote  $c$

as the iteration index. A further benefit of this method is that  $\psi^{\ell+1,c}$  does not need to be stored during the iteration process. After convergence, it can be computed using one more application of  $(L^*)^{-1}$ .

### 3. Low-rank Representations of the Solution

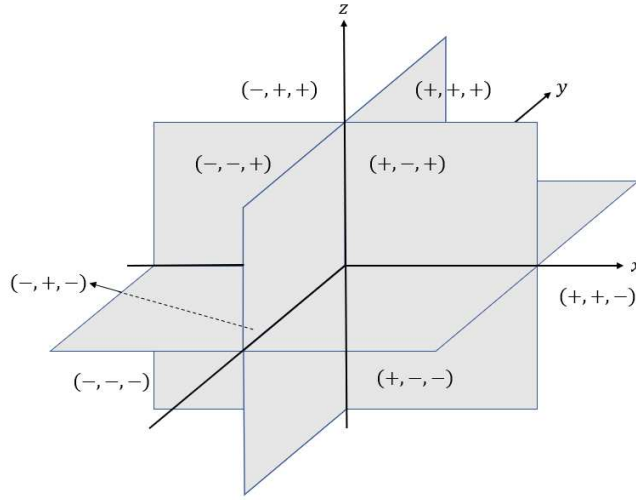


Figure 1: The sign of direction cosines in the form  $(x,y,z)$  in the 3D Cartesian geometry for each octant.

The preliminary requirement for transport sweeps is the known direction of particle advection. As mentioned above, the original  $S_N$  equations (3) can be updated with a transport sweep using a triangular solve along one discrete ordinate at a time. This property is not guaranteed to be preserved by low-rank equations if the low-rank procedure combines solutions along angles from different octants. As shown later, the low-rank formulation reduces the number of equations by coupling them with linear combinations, so different directions are mixed and solved together. The key idea of the low-rank method proposed in this work is to apply a separate low-rank decomposition to the unknowns with different sign combinations of the direction cosines, as shown in Figure 1. In other words, we group the equations in the same octant, so they have the

same propagation directions even after being coupled.

We begin the derivation by writing the low-rank approximation of rank  $r$  to the solution of (3) in the form of a linear combination of  $r$  basis functions similar as [14]

$$\boldsymbol{\psi}^k(\mathbf{r}, t) = \sum_{ij}^r X_i^k(\mathbf{r}, t) S_{ij}^k(t) V_j^k(t) \quad (11)$$

where  $\boldsymbol{\psi}^k = [\psi_1^k, \psi_2^k, \dots, \psi_N^k]^T$  collect all the intensities in octant  $k$ , and  $X_i^k(\mathbf{r}, t)$  and  $V_j^k(t)$  are orthonormal bases, e.g.,  $\langle X_i^k, X_j^k \rangle_{\mathbf{r}} = V_i^k V_j^k = \delta_{ij}$ . The inner product over space is defined as  $\langle f, g \rangle_{\mathbf{r}} = \int_{\mathbb{R}^3} f g d\mathbf{r}$ . We build the low-rank equation in each octant by using  $\bar{X}^k = \{X_1^k, X_2^k, \dots, X_r^k\}$  and  $\bar{V}^k = \{V_1^k, V_2^k, \dots, V_r^k\}$  as ansatz spaces. Then we define orthogonal projectors using the bases:

$$P(\bar{X}^k) g = \sum_{i=1}^r X_i^k \langle X_i^k g \rangle_{\mathbf{r}}, \quad (12)$$

$$P(\bar{V}^k) g = \sum_{j=1}^r V_j^k V_j^k g. \quad (13)$$

The projector onto the tangent space of the low-rank manifold  $\mathcal{M}_r$  is given by

$$P(\bar{X}, \bar{V}) g = P_{\bar{V}} g - P_{\bar{X}} P_{\bar{V}} g + P_{\bar{X}} g. \quad (14)$$

Using this projector, we can define the low-rank governing equations as

$$\partial_t \boldsymbol{\psi}^k = P(\bar{X}^k, \bar{V}^k) F(t, \boldsymbol{\psi}^k) \quad (15)$$

where the  $n_{th}$  component of  $F(t, \boldsymbol{\psi}^k)$  is

$$F(t, \boldsymbol{\psi}^k)_n = -\hat{\Omega}_n \cdot \nabla \psi_n^k - \sigma_t(\mathbf{r}) \psi_n^k + \frac{1}{4\pi} \sigma_s(\mathbf{r}) \phi + Q.$$

We apply the integrating method proposed in [18] to solve the Eq.(15) at  $t_0 + h$  from  $t_0$  implicitly where  $h$  is the step size. The initial condition is given by the low-rank formulation of  $\boldsymbol{\psi}^k(\mathbf{r}, t_0)$

$$\boldsymbol{\psi}^k(\mathbf{r}, t_0) = \sum_{i,j=1}^r X_i^k(\mathbf{r}, t_0) S_{ij}^k(t_0) V_j^k(t_0) = \sum_{i,j=1}^r X_i^{k,0} S_{ij}^{k,0} V_j^{k,0}. \quad (16)$$

and the intended solution at  $t_0 + h$  is denoted as

$$\begin{aligned}\boldsymbol{\psi}^k(\mathbf{r}, t_0 + h) &= \sum_{i,j=1}^r X_i^k(\mathbf{r}, t_0 + h) S_{ij}^k(t_0 + h) V_j^k(t_0 + h) \\ &= \sum_{i,j=1}^r X_i^{k,1} S_{ij}^{k,1} V_j^{k,1}.\end{aligned}\quad (17)$$

The first part of the solving procedure computes  $X_i^{k,1}$  and  $V_j^{k,1}$  by solving

$$\partial_t \boldsymbol{\psi}^k = P_{\bar{V}} F(t, \boldsymbol{\psi}^k) \quad (18)$$

and

$$\partial_t \boldsymbol{\psi}^k = P_X F(t, \boldsymbol{\psi}^k) \quad (19)$$

in parallel. In Eq. (18) the basis  $V_j^k$  does not change with time and is evaluated at the initial value  $V_j^{k,0}$ . We simplify the notation by writing  $K_j^k(\mathbf{r}, t) = \sum_i^r X_i^k(\mathbf{r}, t) S_{ij}^k(t)$ , then the initial condition (16) becomes

$$\boldsymbol{\psi}^{k,0} = \sum_{j=1}^r K_j^{k,0} V_j^{k,0}. \quad (20)$$

We plug Eq. (20) into Eq. (18) and multiply the both sides by  $V_\ell^{k,0}$  to get

$$\partial_t K_j^k = - \sum_{l=1}^r \nabla K_\ell^k V_\ell^k \hat{\boldsymbol{\Omega}}^k V_j^k - \sigma_t K_j^k + \frac{1}{4\pi} \sigma_s \phi V_j^k + Q V_j^k. \quad (21)$$

Note that we use  $\hat{\boldsymbol{\Omega}}^k$  to collect all the ordinates in octant  $k$ . This equation can be solved by the source iteration method and we will return to it with more details in the next subsection. The solution to Eq (21) is factored into  $K_j^k(\mathbf{r}, t_0 + h) = \sum_i^r X_i^{k,1} \tilde{T}_{ij}^k$  by a QR decomposition.

Similarly, by writing  $L_i^k(t) = \sum_j^r S_{ij}^k(t) V_j^k(t)$  we can expand the Eq.(19) as

$$\begin{aligned}\frac{d}{dt} L_i^k &= -\hat{\boldsymbol{\Omega}}^k \sum_l^r \langle \nabla X_\ell^k X_i^k \rangle_{\mathbf{r}} L_\ell^k - \sum_l^r \langle \sigma_t X_\ell^k X_i^k \rangle_{\mathbf{r}} L_l^k \\ &\quad + \frac{1}{4\pi} \sigma_s \langle X_i^k \phi \rangle_{\mathbf{r}} + \langle X_i^k Q \rangle_{\mathbf{r}}.\end{aligned}\quad (22)$$

With the initial condition  $L_i^{k,0} = \sum_j^r S_{ij}^{k,0} V_j^{k,0}$ , we obtain the basis  $V_j^{k,1}$  by factoring the solution  $L_i^{k,1}$  into  $\sum_{j=1}^r \tilde{R}_{ij}^k V_j^{k,1}$  using a QR decomposition. We mention that  $\tilde{R}_{ij}^k$  and  $\tilde{T}_{ij}^k$  are not kept in these two steps.



Last we update  $S_{ij}^k(t)$  by solving the matrix differential equation

$$\begin{aligned} \frac{d}{dt} S_{ij}^k = & - \sum_{ml}^r \langle \partial_r X_m^k X_i^k \rangle_r S_{ml}^k V_\ell^k \hat{\Omega}^k V_j^k - \sum_m^r \langle \sigma_t X_m^k X_i^k \rangle_r S_{mj}^k \\ & + \frac{1}{4\pi} \sigma_s \langle X_i^k \phi \rangle_r V_j^k + \langle X_i^k Q \rangle_r V_j^k \end{aligned} \quad (23)$$

with the initial condition  $S_{ij}^{k,0} = \sum_{m,l=1}^r \langle X_i^{k,1} X_m^{k,0} \rangle_r S_{ml}^{k,0} V_\ell^{k,0} V_j^{k,1}$ .

#### 4. Implementation Details

This section presents the numerical scheme for solving Eqs.(21)-(23). Specifically, we apply the finite volume method for the spatial discretization and the backward Euler for the implicit time integration. For simplicity we present the method in 1-D slab geometry. We begin with the governing equations discussed above

$$\partial_t K_j^k = - \sum_{l=1}^r \partial_x K_\ell^k V_\ell^k \mu^k V_j^k - \sigma_t K_j^k + \frac{1}{2} \sigma_s \phi V_j^k + \frac{1}{2} Q V_j^k, \quad (24)$$

$$\begin{aligned} \frac{d}{dt} L_i^k = & - \mu^k \sum_l^r \langle \partial_x X_\ell^k X_i^k \rangle_x L_\ell^k - \sum_l^r \langle \sigma_t X_\ell^k X_i^k \rangle_x L_l^k \\ & + \frac{1}{2} \sigma_s \langle X_i^k \phi \rangle_x + \langle X_i^k Q \rangle_x, \end{aligned} \quad (25)$$

and

$$\begin{aligned} \frac{d}{dt} S_{ij}^k = & - \sum_{ml}^r \langle \partial_x X_m^k X_i^k \rangle_x S_{ml}^k V_\ell^k \mu^k V_j^k - \sum_m^r \langle \sigma_t X_m^k X_i^k \rangle_x S_{mj}^k \\ & + \frac{1}{2} \sigma_s \langle X_i^k \phi \rangle_x V_j^k + \langle X_i^k Q \rangle_x V_j^k, \end{aligned} \quad (26)$$

where the superscript  $k = +$  or  $-$  denotes directions moving in positive or negative  $x$ -direction, and  $\mu^k$  collects the cosine of corresponding polar angles.

##### 4.1. Numerical Scheme

We define the orthonormal bases  $X_i$  and  $V_j$  as

$$X_i^k(t, x) = \sum_{m=1}^M Z_m(x) u_{mi}^k(t), \quad (27)$$

$$V_j^k(t) = v_{*,j}^k(t), \quad (28)$$

where  $Z_m(x) = \frac{1}{\sqrt{\Delta x}}$  with  $x \in [x_{m-\frac{1}{2}}, x_{m+\frac{1}{2}}]$  are based on a finite volume discretization in space with a constant mesh spacing  $\Delta x$  and  $M$  zones;  $m$  is the cell number. Here  $u_{mi}^k$  are components of the time dependent matrix  $U^k(t) \in \mathbb{R}^{M \times r}$ , and  $v_{*,j}^k$  refers to the  $j_{th}$  column of the  $V^k(t) \in \mathbb{R}^{N \times r}$ . Note that the  $N \times 1$  vector  $v_{*,j}^k$  is a group of linear factors of all positive or negative ordinates, so there are  $r$  combinations for each set of angles.

Next we describe procedures to solve Eq. (24) using source iteration and transport sweeps. By applying the standard upwinding technique, we write the spatial derivative term using upwind finite differences as

$$\begin{aligned} \partial_x K_\ell^+ V_\ell^+ \boldsymbol{\mu}^+ V_j^+ &\approx \frac{1}{\Delta x} (K_{i,\ell}^+ - K_{i-1,\ell}^+) V_\ell^+ \boldsymbol{\mu}^+ V_j^+, \\ \partial_x K_\ell^- V_\ell^- \boldsymbol{\mu}^- V_j^- &\approx \frac{1}{\Delta x} (K_{i+1,\ell}^- - K_{i,\ell}^-) V_\ell^- \boldsymbol{\mu}^- V_j^-. \end{aligned} \quad (29)$$

Note that  $V_\ell^k \boldsymbol{\mu}^k V_j^k$  forms a matrix  $R^k = (r_{\ell j}^k) \in \mathbb{R}^{r \times r}$  that is precomputed. Thus we can discretize Eq. (24) using backward Euler method for the time integration

$$\begin{aligned} \frac{1}{h} K_{i,j}^{+,1} &= -\frac{1}{\Delta x} \sum_{l=1}^r (K_{i,\ell}^{+,1} - K_{i-1,\ell}^{+,1}) r_{\ell j}^+ - \sigma_t K_{i,j}^{+,1} + s_{ij}^+ \quad \text{for } \mu > 0 \\ \frac{1}{h} K_{i,j}^{-,1} &= -\frac{1}{\Delta x} \sum_{l=1}^r (K_{i+1,\ell}^{-,1} - K_{i,\ell}^{-,1}) r_{l j}^- - \sigma_t K_{i,j}^{-,1} + s_{ij}^- \quad \text{for } \mu < 0 \end{aligned} \quad (30)$$

where  $(s_{ij}^k)^\pm = \frac{1}{2} \sigma_s \phi V^\pm + \frac{1}{2} Q V^\pm + \frac{1}{h} K^{\pm,0} \in \mathbb{R}^{M \times r}$  is the source term and the superscript 1 denotes the current time step and 0 denotes the previous time

step. For maximal clarity we write out Eq. (4.1) in full, matrix form:

$$\begin{aligned}
& \begin{bmatrix} P_1^+ & 0 & 0 & \dots & 0 \\ -\bar{R}^+ & P_2^+ & 0 & \dots & 0 \\ 0 & -\bar{R}^+ & P_3^+ & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & -\bar{R}^+ & P_M^+ \end{bmatrix} \begin{bmatrix} K_{1,*}^{+,1} \\ K_{2,*}^{+,1} \\ K_{3,*}^{+,1} \\ \dots \\ K_{M,*}^{+,1} \end{bmatrix} = \begin{bmatrix} s_{1,*}^+ + b_L^+ \\ s_{2,*}^+ \\ s_{3,*}^+ \\ \dots \\ s_{M,*}^+ + b_R^+ \end{bmatrix} \quad \text{for } \mu > 0, \\
& \begin{bmatrix} P_1^- & \bar{R}^- & 0 & \dots & 0 \\ 0 & P_2^- & \bar{R}^- & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & P_3^- & \bar{R}^- \\ 0 & 0 & \dots & 0 & P_M^- \end{bmatrix} \begin{bmatrix} K_{1,*}^{-,1} \\ K_{2,*}^{-,1} \\ K_{3,*}^{-,1} \\ \dots \\ K_{M,*}^{-,1} \end{bmatrix} = \begin{bmatrix} s_{1,*}^- + b_L^- \\ s_{2,*}^- \\ s_{3,*}^- \\ \dots \\ s_{M,*}^- + b_R^- \end{bmatrix} \quad \text{for } \mu < 0.
\end{aligned} \tag{31}$$

Here  $L_i^+ = \frac{1}{\Delta x} R^+ + (\sigma_t(i) + \frac{1}{h}) I_r$ ,  $L_i^- = -\frac{1}{\Delta x} R^- + (\sigma_t(i) + \frac{1}{h}) I_r$ ,  $I_r$  is the  $r \times r$  identity matrix,  $\bar{R}^k = \frac{1}{\Delta x} R^k$ ,  $\sigma_t(i)$  denotes the value of total cross-section in cell  $i$ ,  $b_L^k$  and  $b_R^k$  are the left and the right boundary values. We set  $b_L^k = b_R^k = 0$  to correspond to vacuum boundary conditions. A compact form of the marching scheme for Eq.(31) is

$$\begin{aligned}
K_{i,*}^{+,1} &= (P_i^+)^{-1} \left( \bar{R}^+ K_{i-1,*}^{+,1} + s_{i,*}^+ \right) \quad \text{for } \mu > 0 \\
K_{i,*}^{-,1} &= (P_i^-)^{-1} \left( -\bar{R}^- K_{i+1,*}^{-,1} + s_{i,*}^- \right) \quad \text{for } \mu < 0
\end{aligned} \tag{32}$$

Eqs. (25) and (26) are matrix ordinary differential equations and can be solved by an implicit Runge–Kutta method.

#### 4.2. Computational Cost

This work focuses on reducing the computer memory cost for radiative transfer simulations. The memory consumption for solving Eq. (3) using the classical iteration solution procedure is  $2 \times M \times (N_\Omega + 1)$  during each time step to store the angular and scalar intensities. Here we do not include the simulation parameters such as the scattering/absorption cross-sections and the prescribed

source based on the assumption that they can be stored efficiently during the computation.

In the low-rank algorithm the memory requirements are determined by the size of matrices  $U$ ,  $S$  and  $V$ . The initial conditions  $U^{k,0}$ ,  $V^{k,0}$ , and  $S^{k,0}$  take  $(M \times r + N \times r + r^2) \times n_o$  floating point numbers. Solving Eq. (24) requires  $M \times r \times n_o + M$  more memory to store the scattering source and the updated matrix  $U$  (factorized from the updated matrix  $K$ ). Eq.(25) is solved next and uses  $N \times r \times n_o$  memory for the updated  $V$  for a total of  $(M \times r + N \times r + r^2) \times n_o$  for the updated basis. Lastly, the  $n_o$  equations in Eq. (26) are  $r \times r$  Sylvester matrix equations which require  $n_o \times \mathcal{O}(r^3)$  of memory when using the algorithm proposed in [24]. In this step the memory usage is  $(M \times r + N \times r + r^2) \times n_o + n_o \times \mathcal{O}(r^3)$ . In practice, we choose the rank  $r \ll \min(M, N)$ , so we can assume  $\mathcal{O}(r^3) + r^2 < M \times r + N \times r$ . To summarize we use  $2 \times (M \times r \times n_o + N \times r \times n_o + M)$  to approximate the largest memory usage during the low-rank calculations. We can see that the low-rank solution requires much smaller memory than the full solution, when  $r \ll \min(M, N)$ .

In numerical experiments we measure the low-rank memory occupied by double precision floating point numbers in megabytes (MB) as

$$\text{memory} = 8 \times (M \times r \times n_o + N \times r \times n_o + M) \times 10^{-6}, \quad (33)$$

and for the full-rank solution the memory required is

$$\text{memory} = 8 \times 2 \times M \times (N_\Omega + 1) \times 10^{-6}. \quad (34)$$

#### 4.3. Initial Condition

Usually, the initial conditions  $U^{k,0}$ ,  $V^{k,0}$ , and  $S^{k,0}$  for the low-rank evolution are obtained by taking the SVD of the given initial intensity and then truncating the singular values smaller than the  $r$ th largest in matrix  $S$  and removing the corresponding rows and columns in  $U$  and  $V$ . Nevertheless, setting the initial condition is not always straightforward. If the intensities are zero or a constant initially (as in many common test problems), the initial condition will be of

rank 0 or 1. One way to deal with this issue is to add  $r$  randomly generated orthogonal basis to  $U$  and  $V$  and set  $S$  to a  $r \times r$  matrix of zeros [12]. We find that this approach does not work well for implicit schemes with large time steps. In this work, we choose to let the low-rank basis evolve for a very small time step, such as CFL= 0.01, to obtain initial conditions with a basis that is in the range of the low-rank operator. After this treatment, we can set time steps as intended.

## 5. Numerical Results

We present results for four 2-D and one 3-D benchmark problems. In all of the test results, we set the particle speed to 1 cm/s. The CFL condition is defined as  $\text{CFL} = \min(\frac{\Delta t}{\Delta x}, \frac{\Delta t}{\Delta y}, \frac{\Delta t}{\Delta z})$ . We implement the low-rank algorithm in MATLAB. For several problems, we measured the running memory in MATLAB with the built-in function “memory” and record the running time with the stopwatch timer functions.

### 5.1. Double Chevron problem

To demonstrate the accuracy of our low-rank algorithm, we consider a multi-material 2-D problem with an asymmetric layout originally presented in [14]. As detailed in Figure 2, this problem has purely scattering zones with  $\sigma_t = \sigma_s = 0.01 \text{ cm}^{-1}$ , absorbing zones  $\sigma_t = 100 \text{ cm}^{-1}, \sigma_s = 0.1 \text{ cm}^{-1}$ , and an isotropic source at the bottom with  $Q = 1$  of thickness 0.1 mm. We solve this problem using the spatial grid of size  $90 \times 90$  for the computational domain  $[0, 9 \text{ mm}] \times [0, 9 \text{ mm}]$ , and the simulation time  $t = 0.9 \text{ s}$ . The CFL number is set to be 3.

Figure 3 compares the low-rank solutions for the scalar intensity with  $S_{32}$  and varying rank to the full rank  $S_{32}$  solution and the  $S_{100}$  benchmark solution. As we can see  $S_{32}$  with rank 36 is close to the full rank  $S_{32}$  solution but it is not enough to resolve the particle distribution behind the second obstacle and the negative results are observed. The low-rank solution with rank 25 is nearly identical to the full rank  $S_{32}$  solution, which means rank 25 is sufficient to capture the important features in this problem.

Figure 4 presents a comparison for scalar intensities along  $x = 0.45$  cm and  $x = 0.6$  cm. The left plot emphasizes their different behavior behind the second chevron, and the right plot shows the particle distribution along the gap that particles can travel through. We find that the low-rank solution with rank 36 has a similar error to the full rank  $S_{32}$  solution when compared to the benchmark. We also notice that  $S_{32}$  is not enough to capture the particle stream along the gap and it requires high angular resolution to obtain.

The computational efficiency of our low-rank algorithm is presented in Figure 5. Our solutions' accuracy is measured by the root mean square (RMS) error to the benchmark solution. In Figure 5a we plot the error of our solutions versus the running memory in the sweeping process. As we can see, the third point in the red dotted line, which indicates the solution with rank 16, has already achieved the same error level as the full rank  $S_{32}$  solution, but with only 10% of the memory. It also can be seen that the low-rank and the full rank solution use a similar amount of memory for the same rank. As explained by our memory formula (33), the memory usage depends on the rank when  $M$  is fixed, and  $N$  is much smaller than  $M$ .

Figure 5b makes the similar comparison with the running time per one time step ( $\Delta t = 0.03$  s for this test). It appears that our rank 16 solution could save 95% of the computational time compared to the full rank solution. We also point out that our low-rank method will take more time than the full rank solution with the same rank. For example, the rightmost dot in the red line is the solution with rank 100, the same as the full rank  $S_{20}$  solution, but it runs twice as long. But when rank is properly chosen and relatively small, we can still save a large amount of memory and time.

The computer memory occupied in MATLAB during the calculation and the theoretical values are shown in Figure 6. The coefficients of determination ( $R^2$ ) for the linear fit are approximately one, which indicates a strong linear relationship between the theoretical and actual memory. We conclude that Eqs. (33) and (34) are valid estimations for the actual memory usage, and the low-rank algorithm reduces the memory requirement proportionally to the

reduction in rank.

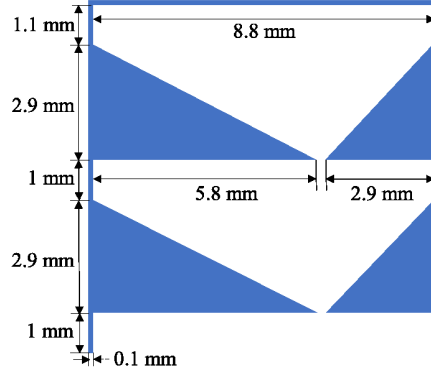


Figure 2: The layout for the double chevron problem is shown. The blue are dense materials and the blank are purely-scattering region. There is an isotropic source at the bottom and other three sides have vacuum boundary conditions.

### 5.2. Lattice problem

Next, we consider the lattice problem: a  $7 \times 7$  cm checkerboard consisting of pure absorbers, purely scattering regions, and a strong source turned on at  $t = 0$  with a zero initial condition. We use a computational domain of  $[0, 7] \times [0, 7]$  with a 210 spatial grid. We solve this problem with a single, large time step using  $\text{CFL} = 10^4$ . The layout and the reference solution calculated with  $S_{64}$  for this problem are shown in Figure 7.

This test aims to show the accuracy improvement by adding more angular directions while keeping the rank fixed. As shown in Figure 8, the full rank  $S_6$  solution has ray effects in the solution near the bottom, right and left sides. These can be alleviated by using more discrete ordinates as observed in solutions with  $S_{64}$  and rank 9. Similar phenomena are found in solutions with rank 16, where the full rank solution suffered from ray effects while the low-rank solution is closer to our reference. We also plot the solution along  $x = 3.5$  cm, where we can see that low-rank solutions are closer to the benchmark solution than the full rank solution.

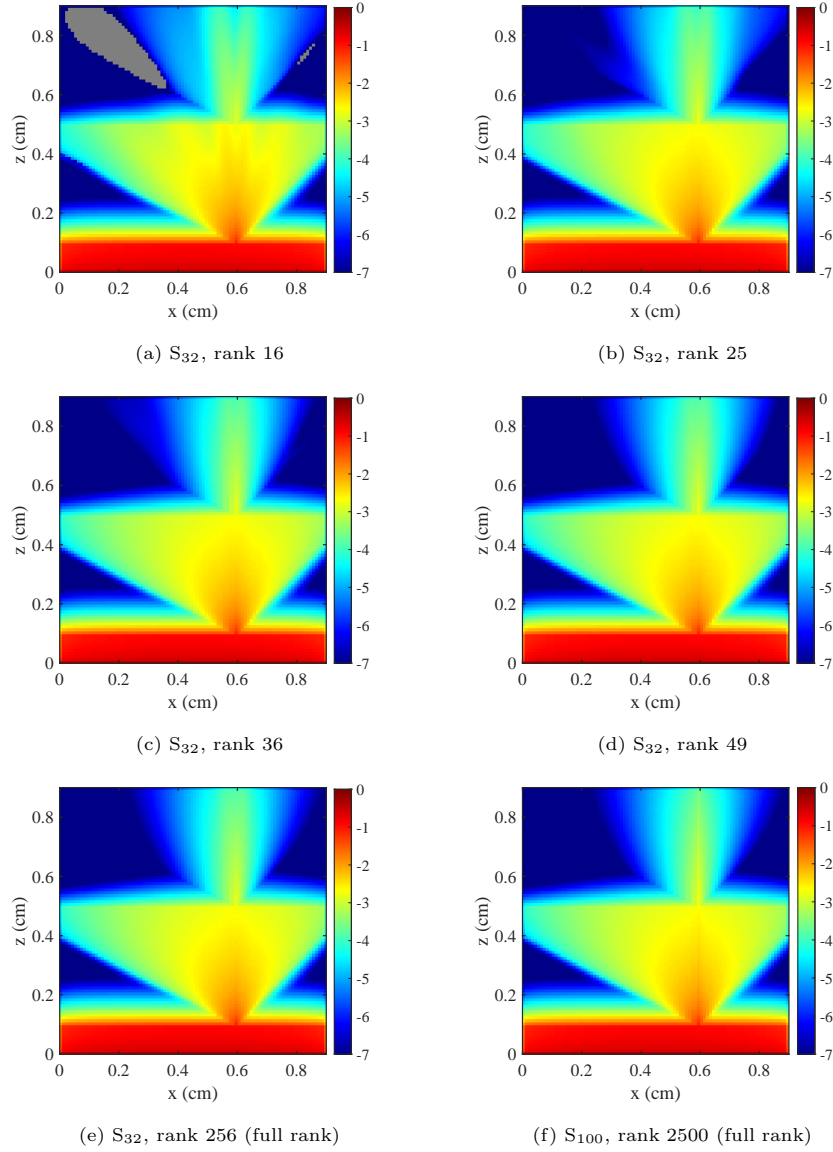


Figure 3: Solutions to the double chevron problem at  $t = 0.9$  s. The color scale is logarithmic and negative regions are shaded gray.

Figure 9 gives a quantitative comparison between low-rank and full rank solutions in terms of their memory usage and error measured by RMS deviations to the reference. We notice that the low-rank solutions converge faster than the



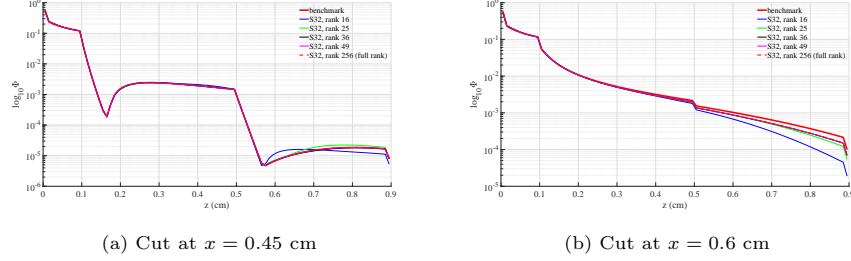


Figure 4: Logarithmic scalar insentities to the double chevron problem at  $t = 0.9$  s.

full rank solutions, where the low-rank solution with rank 49 achieves the same accuracy as the full rank  $S_{50}$  solution, but only requires 10% of the memory.

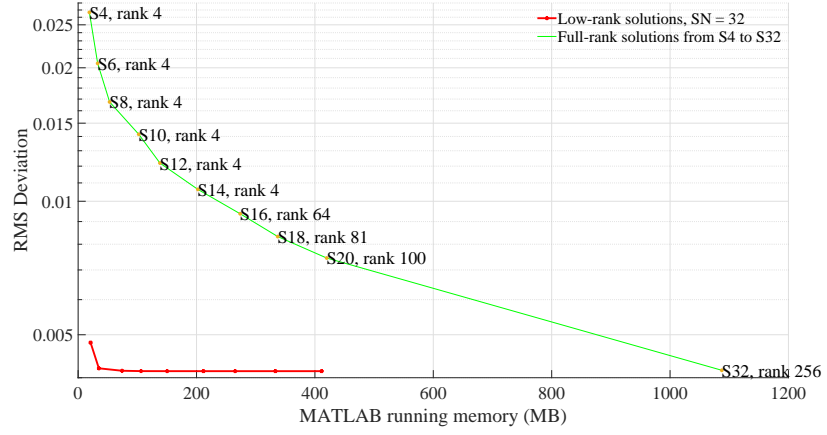
### 5.3. Line source problem

The line source problem describes a beam of radiation is spreading out in a purely scattering plane. In this problem we have the initial condition  $\psi(x, z, \mu, \varphi, t) = \delta(x)\delta(z)$ , and the purely scattering medium  $\sigma_s = \sigma_t = 1$  with no source  $Q = 0$ . We use a computational domain of  $[-1.5, 1.5] \times [-1.5, 1.5]$  for the simulation time  $t = 1$  s, while the spatial grid is set to be  $150 \times 150$ . Figure 10 shows the analytic solution and our benchmark solution.

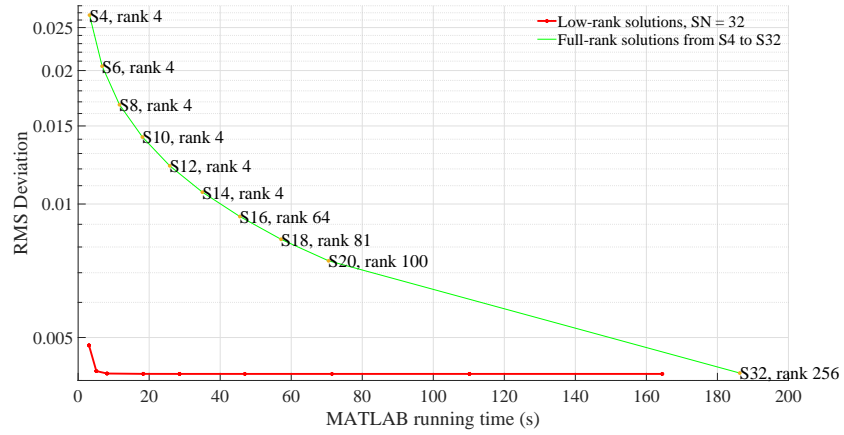
We use the line source test to demonstrate the benefits of the low-rank method with high angular resolutions. Figure 11 compares the full rank solution to low-rank solutions with the same rank. We notice that all the full-rank solutions have remarkable ray effects, which means the number of angular directions are insufficiently dense to move particles to all the regions. The low-rank method could significantly improve the full rank solution by using a small amount of extra memory. As we can see, the ray effects are alleviated in the solution with rank 16, and the solution with rank 36 is comparable to the benchmark solution.

### 5.4. Hohlraum problem

The last 2-D problem is a modified Hohlraum problem as shown in Figure 12. The hohlraum is surrounded by vacuums, except for the incoming source on



(a) Error versus running memory



(b) Error versus running time

Figure 5: The comparison of errors for the low-rank and full-rank solutions of double chevron problem with different memory usage and computational time is shown. The red dots represents low-rank solution with rank 4, 9, 16, 25, 49, 64, 81, and 100.

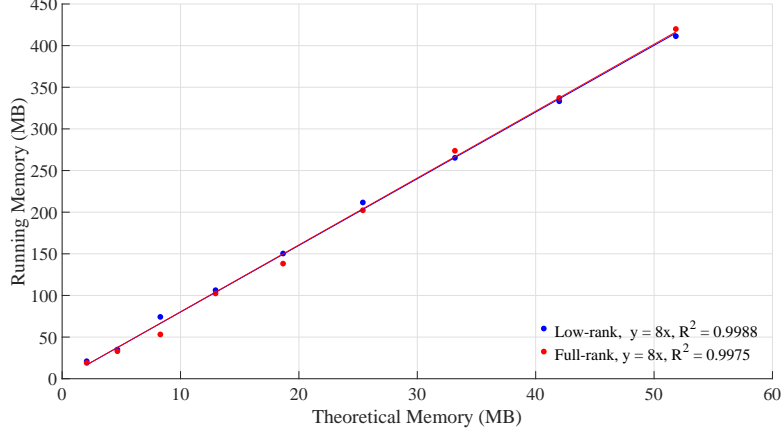


Figure 6: The relation between the memory calculate by formulas (33) and (34) and the running memory in MATLAB for the double chevron problem. The angular discretization for the low-rank solutions is  $S_{32}$  with ranks varying from 4 to 100, the full-rank solutions are with  $S_4$  to  $S_{20}$ . The spatial and time discretizations are the same as previous simulations.

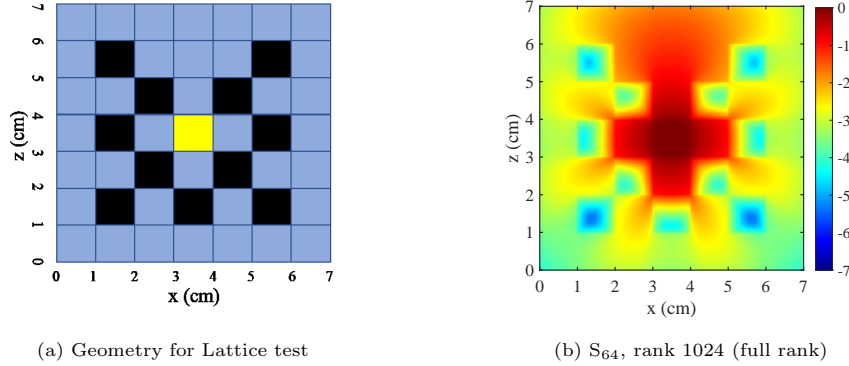


Figure 7: The left plot shows the layout of the lattice problem, where the blue zones are purely scattering region with  $\sigma_s = \sigma_t = 1 \text{ cm}^{-1}$ , the black are absorbing region with  $\sigma_s = 0$ ,  $\sigma_t = 10 \text{ cm}^{-1}$  and the yellow is the scattering region with an isotropic source  $Q = 1$ . The right plot is the benchmark solution with the logarithmic color scale.

the left. In this problem, we are interested in the particle distribution behind the block that particles cannot reach directly, which will require more angular direction samples to resolve. For this problem we use a computational domain of  $[0, 1.3] \times [0, 1.3]$  for the simulation time  $t = 0.9 \text{ s}$ , and set the spatial grid to

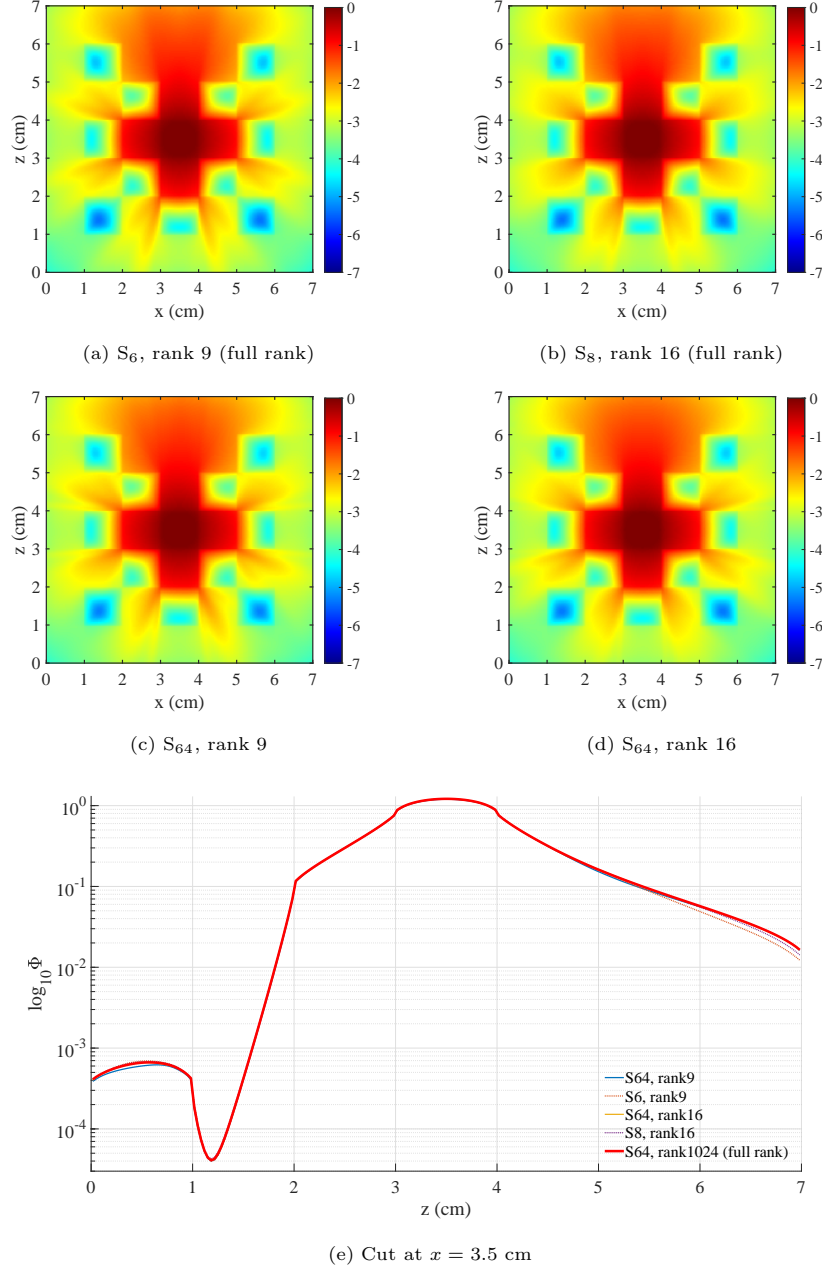


Figure 8: The logarithmic scalar intensity to the lattice problem calculated by the low-rank method with  $S_{64}$  are compared to the full rank solutions with the same rank.

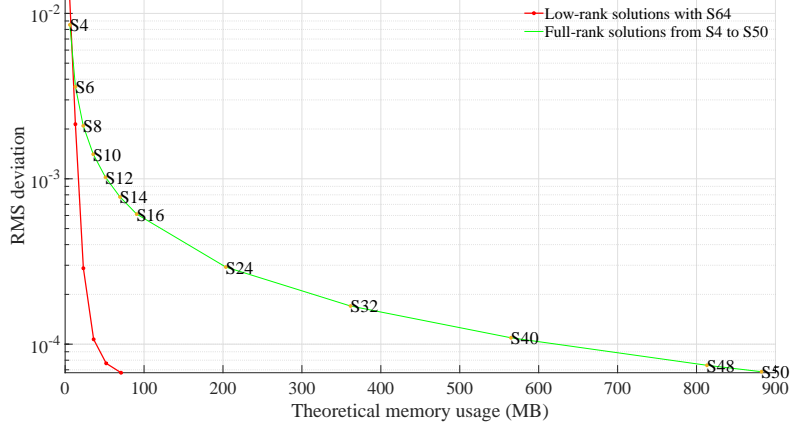


Figure 9: The comparison of errors for the lattice problem with different memory usage. The red dot line represents the error of the full rank solution that varies the number of discrete ordinates. The solid line represents the error of the low-rank solutions with  $S_{64}$  that varies the rank (rank = [4, 9, 16, 25, 36, 49]).

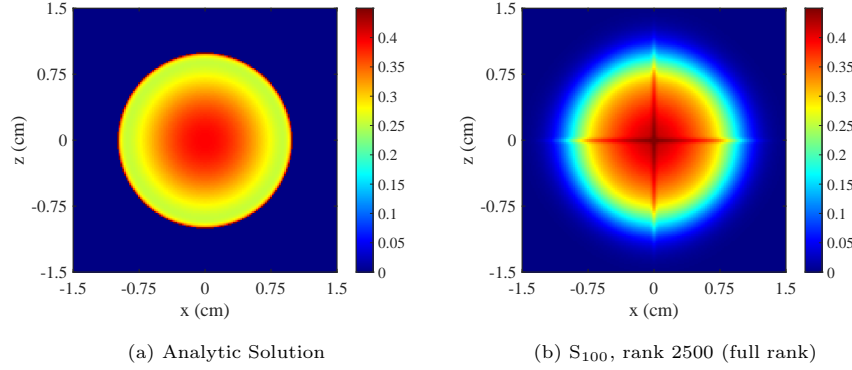


Figure 10: The scalar intensity  $\phi$  of the line source problem at  $t = 1$  s calculated by full-rank  $S_{100}$  is compared to the benchmark solution.

150 × 150.

We make a similar comparison between the low-rank solutions to the full rank solutions with the same rank in Figure 13. We observe beam-shaped shadows in the left part of the full rank  $S_8$  solution, but not in the low-rank solution with  $S_{100}$  and rank 16. There are negative regions in the low-rank solutions brought

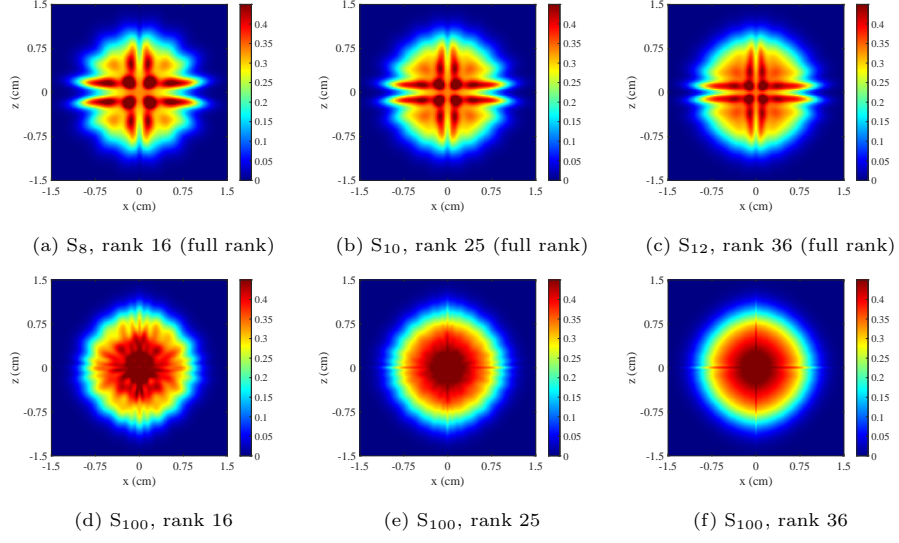


Figure 11: The scalar intensity to the line source problem calculated by the low-rank method are compared to the full rank solution with the same rank.

by the truncation from low-rank algorithms.

Figure 14 presents the cut along  $z = 0.65$  cm to compare the particle distribution behind the obstacle. The left plot shows that the solution with rank 36 is nearly identical to the full rank solution. The right plot shows that the low-rank solutions are closer to the benchmark than the full rank solution with the same rank.

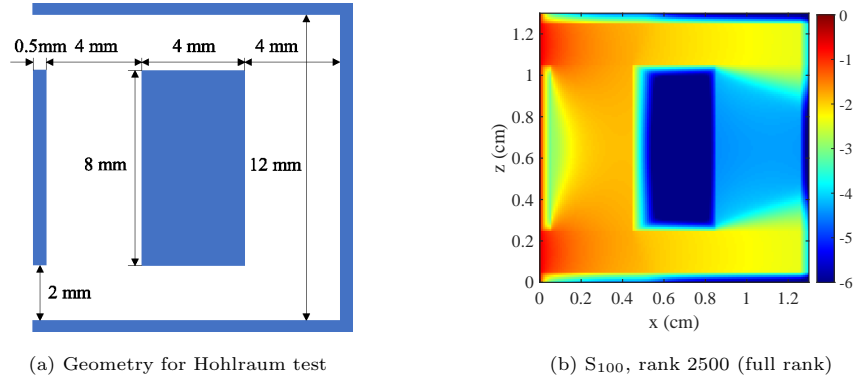


Figure 12: The layout of the Hohlraum problem and the high-order benchmark solution.

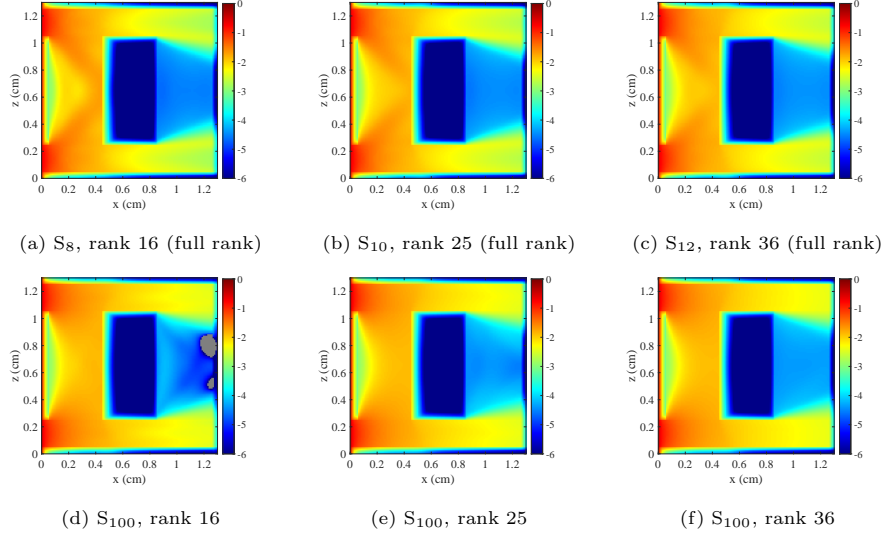


Figure 13: The scalar intensity to the Hohlraum problem calculated by the low-rank method with  $S_{100}$  are compared to the same rank solutions without rank reduction. The color scale is logarithmic and negative regions are shaded gray.

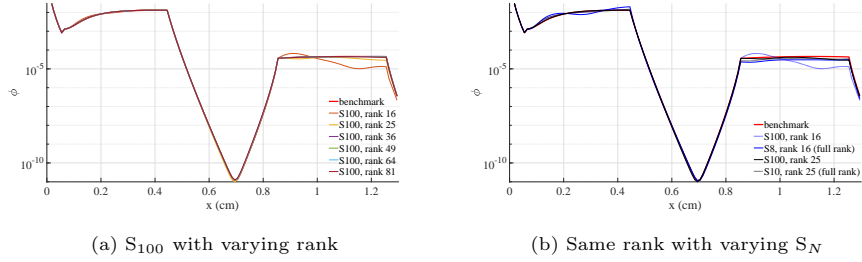


Figure 14: The logarithm of the scalar intensity along  $z = 0.65$  cm.

### 5.5. Wires Problem

As shown in Figure 15, we introduce the 3-D wires problem, inspired by the 2-D problem in [8], has an emitting dense material embedded vacuum. Particles are emitted from the central wire with  $Q = 1$ . All the five wires are made by the same material with  $\sigma_s = 0.1$  and  $\sigma_t = 5$ . The streaming regions has  $\sigma_s = \sigma_t = 0$  and we use vacuum boundary conditions. The computational domain for this

problem is set to  $[-0.01, 0.01] \times [-0.01, 0.01] \times [0, 0.02]$  with a  $40 \times 40 \times 40$  mesh grid. We simulate this problem to  $t = 0.6\text{s}$  using a time step  $\Delta t = 0.2$  (CFL= 1200).

The following results compare the low-rank solution and the full rank solution with the same rank. Figure 16 shows the solution in the plane of  $x - y$  along  $z = 0.1$ . First, we notice strong ray effects in the full rank  $S_4$  solution, especially along  $y = 0$ . The low-rank solution with  $S_{64}$  and rank 4 looks better in this aspect. The full rank solution with  $S_8$  achieves better distributions than  $S_4$  but is still unable to avoid the ray effects along  $y = 0$ . But we can see that the low-rank solution with rank 16 is close to the full rank solution. If we further increase the rank to 36, the low-rank solution is identical to the benchmark solution.

We also plot the solution in the y-z plane along  $x = 0$  shown in Figure 17. We can see that the solution with rank 4 is has large, qualitative errors, especially in the center source region. The low-rank solution significantly improves the situation with the errors greatly reduced.

The running memory to carry out these simulations in MATLAB is shown in Figure 18. The rank 64 solution with  $S_{64}$ , represented by the rightmost dot in the solid red line, uses 25% of the full rank  $S_{32}$  solution but achieves the same accuracy. It also reaches five times lower error than the full rank  $S_{16}$  solution with slightly more memory.

## 6. Conclusion

In this work, we have presented a low-rank- $S_N$  scheme to simulate discrete ordinate ( $S_N$ ) radiative transfer equations using reduced computational costs. By applying the unconventional low-rank integrator, our scheme can be solved implicitly by transport sweeps, enabling more efficient radiation transport simulations on computing platforms with reduced memory per core. With the carefully chosen rank based on the intrinsic property of problems, this low-rank method can save up to 90% of the computer memory and 95% of the compu-



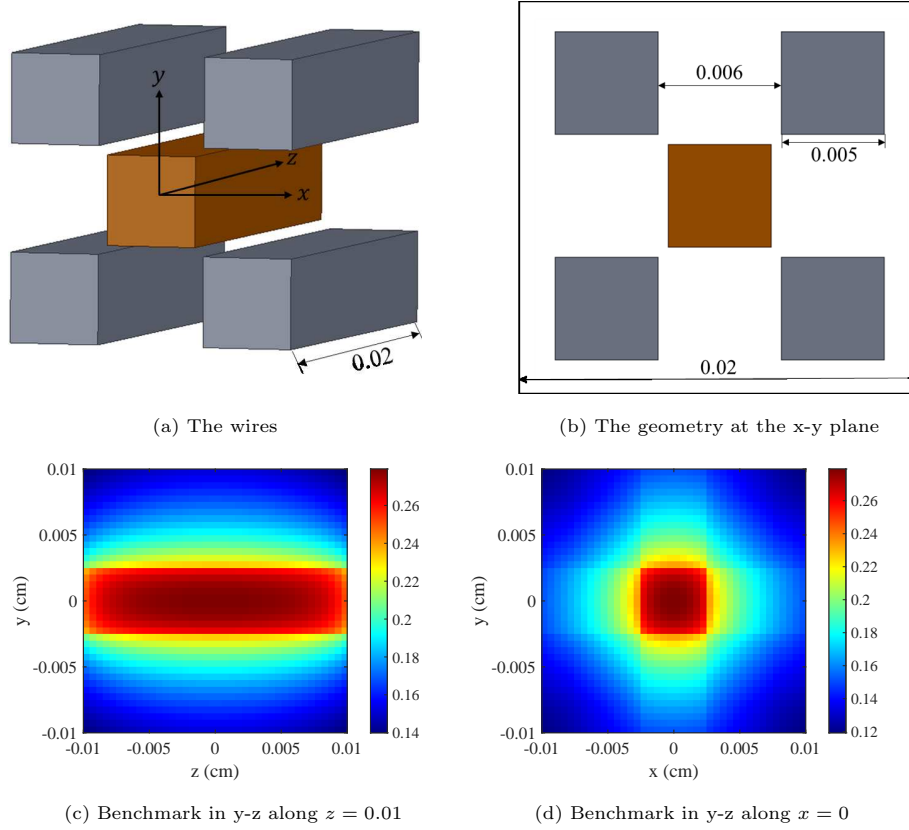


Figure 15: The layout of the wires problem and the high-order benchmark solution.

tational time. We also observe that if the rank is chosen to be too large, the solution does not suffer, but the calculation is not maximally efficient.

We point out that the effect of the computational cost saving on a specific problem relies on the discovery of its “true” rank. We are working on various radiative transfer problems simulations to find a practical strategy for selecting the rank. Another open question in low-rank methods is conserving the intrinsic properties of the underlying problem such as conservation of particles or the appropriate asymptotic limits. As in our previous work [15], the high-order/low-order (HOLO) algorithm is one way to perform the fix. The implementation is straightforward since we can also calculate the Eddington tensor with the low-rank  $S_N$  solution. Furthermore, we would be interested in applying the low-rank

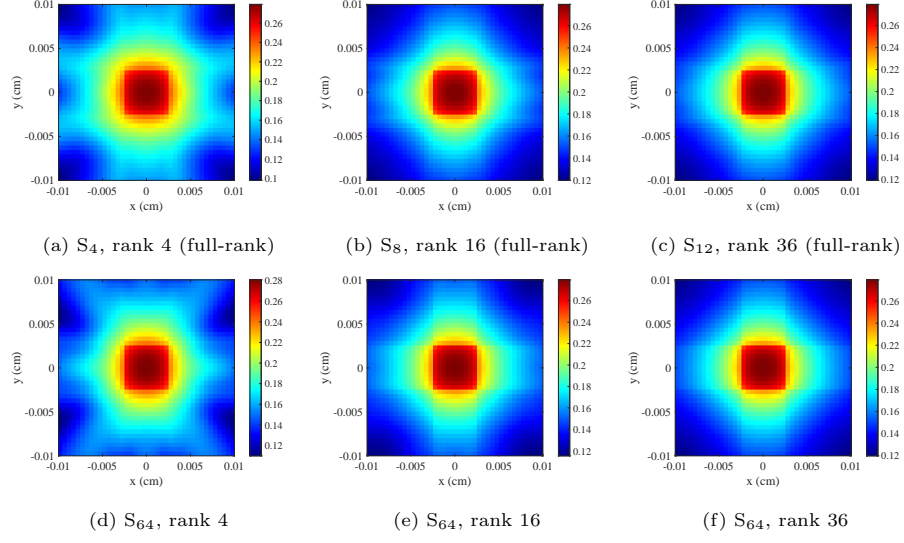


Figure 16: The logarithmic scalar intensity solution the wires problem calculated by the low-rank method with  $S_{64}$  are compared to the same rank solutions without rank reduction.

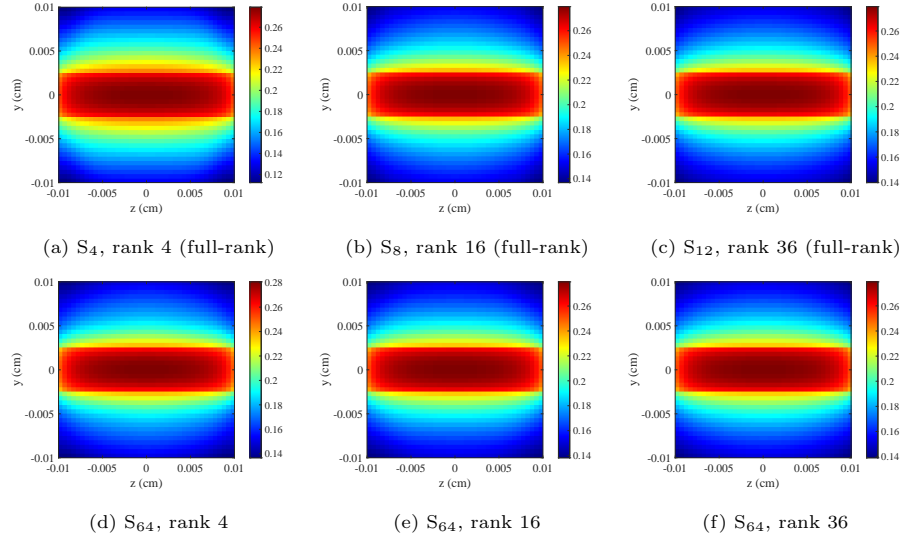


Figure 17: The logarithmic scalar intensity solution to the wires problem calculated by the low-rank method with  $S_{64}$  are compared to the same rank solutions without rank reduction.

method to the multi-group transport problems for future study. This would give another dimension to potentially compress and potentially give even greater

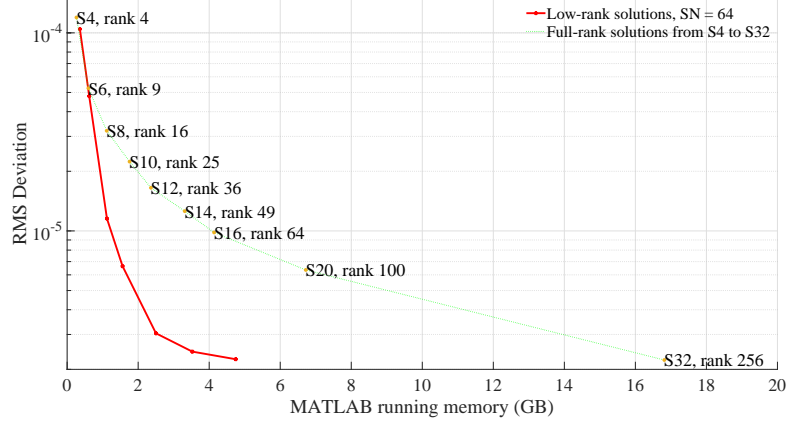


Figure 18: The comparison of errors for the wires problem with different memory usage. The green dots represent the error of the full rank solutions that varies the number of discrete ordinates. The red solid line represents the error of the low-rank solutions with  $S_{64}$  that varies the rank (rank = [4, 9, 16, 25, 36, 49, 64]).

memory savings. Recent work on diffusion-based particle transport problems with multigroup indicates that this could be a fruitful investigation [25].

## References

- [1] A. D. Kim, M. Moscoso, Radiative transfer computations for optical beams, *Journal of Computational Physics* 185 (1) (2003) 50 – 60.
- [2] Y. Azmy, E. Sartori, *Nuclear computational science: A century in review*, Springer Netherlands, 2010.
- [3] C. Cercignani, *Rarefied gas dynamics: from basic concepts to actual calculations*, Vol. 21, Cambridge university press, 2000.
- [4] J. Shalf, The future of computing beyond moore’s law, *Philosophical transactions of the Royal Society of London. Series A: Mathematical, physical, and engineering sciences* 378 (2166) (2020) 20190061–20190061.
- [5] G. C. Pomraning, *The equations of radiation hydrodynamics*, Courier Corporation, 2005.
- [6] S. I. Heizler, Asymptotic telegrapher’s equation ( $P_1$ ) approximation for the transport equation, *Nuclear science and engineering* 166 (1) (2010) 17–35.
- [7] R. G. McClarren, J. P. Holloway, T. A. Brunner, T. A. Mehlhorn, A quasi-linear implicit riemann solver for the time-dependent pn equations, *Nuclear science and engineering* 155 (2) (2007) 290–299.
- [8] R. G. McClarren, J. P. Holloway, T. A. Brunner, On solutions to the  $P_n$  equations for thermal radiative transfer, *Journal of Computational Physics* 227 (5) (2008) 2864–2885.
- [9] E. Lewis, W. Miller, *Computational Methods of Neutron Transport*, John Wiley and Sons, 1984.
- [10] O. Koch, C. Lubich, Dynamical Low-Rank Approximation, *SIAM Journal on Matrix Analysis and Applications* 29 (2) (2007) 434–454.
- [11] A. Nonnenmacher, C. Lubich, Dynamical low-rank approximation: applications and numerical experiments, *Mathematics and Computers in Simulation* 79 (4) (2008) 1346–1357. doi:10.1016/j.matcom.2008.03.007.

- [12] C. Lubich, I. V. Oseledets, A projector-splitting integrator for dynamical low-rank approximation, *BIT Numerical Mathematics* 54 (1) (2014) 171–188. doi:[10.1007/s10543-013-0454-0](https://doi.org/10.1007/s10543-013-0454-0).
- [13] E. Kieri, C. Lubich, H. Walach, Discretized dynamical low-rank approximation in the presence of small singular values, *SIAM Journal on Numerical Analysis* 54 (2) (2016) 1020–1038.
- [14] Z. Peng, R. G. McClarren, M. Frank, A low-rank method for two-dimensional time-dependent radiation transport calculations, *Journal of Computational Physics* 421 (2020) 109735. doi:<https://doi.org/10.1016/j.jcp.2020.109735>.
- [15] Z. Peng, R. G. McClarren, A high-order/low-order (holo) algorithm for preserving conservation in time-dependent low-rank transport calculations, *Journal of Computational Physics* 447 (2021) 110672. doi:<https://doi.org/10.1016/j.jcp.2021.110672>.
- [16] L. E. Zhiyan Ding, Q. Li, Dynamical low-rank integrator for the linear boltzmann equation: Error analysis in the diffusion limit, *SIAM Journal on Numerical Analysis* 59 (4) (2021) 2254–2285.
- [17] Y. W. Lukas Einkemmer, Jingwei Hu, An asymptotic-preserving dynamical low-rank method for the multi-scale multi-dimensional linear transport equation, *Journal of Computational Physics* 439 (2021) 110353.
- [18] G. Ceruti, C. Lubich, An unconventional robust integrator for dynamical low-rank approximation (2020). [arXiv:2010.02022](https://arxiv.org/abs/2010.02022).
- [19] J. Jenal, P. Erickson, W. Rhoades, D. Simpson, M. Williams, The Generation of a Computer Library for Discrete Ordinates Quadrature Sets, Tech. rep., Oak Ridge National Laboratory (1977).
- [20] G. Longoni, Advanced quadrature sets, acceleration and preconditioning techniques for the discrete ordinates, Ph.D. thesis, University of Florida (2004).  
URL <http://purl.fcla.edu/fcla/etd/UFE0007560>

- [21] B. Hunter, Z. Guo, Comparison of quadrature schemes in DOM for anisotropic scattering radiative transfer analysis, *Numerical Heat Transfer, Part B: Fundamentals* 63 (6) (2013) 485–507. doi:10.1080/10407790.2013.777644.
- [22] T. Endo, A. Yamamoto, Development of new solid angle quadrature sets to satisfy even- and odd-moment conditions, *Journal of Nuclear Science and Technology* 44 (10) (2007) 1249–1258. doi:10.1080/18811248.2007.9711368.
- [23] J. S. Warsa, T. A. Wareing, J. E. Morel, Krylov Iterative Methods and the Degraded Effectiveness of Diffusion Synthetic Acceleration for Multidimensional SN Calculations in Problems with Material Discontinuities, *Nuclear Science and Engineering* 147 (3) (2017) 218–248. doi:10.13182/NSE02-14.
- [24] A. Bouhamidi, K. Jbilou, A note on the numerical approximate solutions for generalized sylvester matrix equations with applications, *Applied Mathematics and Computation* 206 (2008) 687–694.
- [25] J. Kusch, B. Whewell, R. G. McClarren, M. Frank, A low-rank power iteration scheme for neutron transport criticality problems (2022). doi:10.48550/ARXIV.2201.12340.