

# A low-rank method for two-dimensional time-dependent radiation transport calculations

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

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

<sup>b</sup>*Karlsruhe Institute of Technology, Steinbuch Centre for Computing, Karlsruhe, Germany*

---

## Abstract

The low-rank approximation is a complexity reduction technique to approximate a tensor or a matrix with a reduced rank, which has been applied to the simulation of high dimensional problems to reduce the memory required and computational cost. In this work, a dynamical low-rank approximation method is developed for the time-dependent radiation transport equation in 1-D and 2-D Cartesian geometries. Using a finite volume discretization in space and a spherical harmonics basis in angle, we construct a system that evolves on a low-rank manifold via an operator splitting approach. Numerical results on five test problems demonstrate that the low-rank solution requires less memory than solving the full rank equations with the same accuracy. It is furthermore shown that the low-rank algorithm can obtain high-fidelity results at a moderate extra cost by increasing the number of basis functions while keeping the rank fixed.

*Keywords:* Low-rank approximation, Radiation transport

---

## 1. Introduction

The numerical simulation of the radiation transport process is fundamental in a wide range of applications from supernovas to medical imaging, where accurate numerical solutions of a linear Boltzmann equation, also known as the

---

\*Corresponding author

*Email addresses:* `zpeng5@nd.edu` (Zhuogang Peng), `rmccclarr@nd.edu` (Ryan G. McClarren), `martin.frank@kit.edu` (Martin Frank)

radiative transfer equation (RTE), are required. In this equation, we are solving for the specific intensity, which is a seven-dimensional function that describes the movement of the flow of particles in terms of time, position, direction, and energy. Despite the rapid development of high-performance computing, the numerical solution of the RTE remains challenging due to the high computational costs and the large memory requirements caused by such a high-dimensional phase space. In this paper, we focus on reducing the memory required to solve the RTE to enable better performance on exascale-class computers [1].

Due to the rich phase space, computational methods for radiation transport require discretizations in energy and direction<sup>1</sup>, in addition to spatial and temporal discretizations. For the energy variable, the multigroup method is commonly used [2]. The direction, or angular variables, can be treated by solving the equations along particular directions via the discrete ordinates or  $S_N$  method [3]. An alternative technique employs a basis expansion using the natural basis for the sphere: spherical harmonics. The spherical harmonics or  $P_N$  method uses a truncated expansion to approximate the angular dependence [4–6]. Low-order approximations are also used such as flux-limited diffusion [7, 8], simplified  $P_N$  [9–11] along with hybrid methods such as quasi-diffusion [12]. Finally, stochastic methods based on the implicit Monte Carlo method [13–16] are an alternative approach, though often requiring significant computational cost.

Although the aforementioned methods require many degrees of freedom to describe the phase-space dependence of the solution, it is known that many transport problems require only a subspace of the full phase space (called a manifold in mathematical parlance) to describe the transport of particles. An example of this phenomenon is problems in the diffusion limit: these problems require only a linear dependence on angular variables. One can also formulate problems where this manifold over which the solution depends evolves over time: a beam entering a scattering medium would be described by a delta-function in

---

<sup>1</sup>The terminology direction and angle are typically used interchangeably to describe the angular component of the phase space.

space and angle at time zero, but eventually, relax to much smoother distribution that we could characterize using a simple basis expansion.

We desire to generalize this idea, and possibly automatically discover the manifold that describes the system evolution. We accomplish this task by expressing the solution to a transport problem as a basis expansion in space and angle and using techniques to determine what subspace of those bases are needed to describe the solution and how that subspace evolves. We use the dynamical low-rank approximation (DLRA) of Koch and Lubich to evolve time-dependent matrices by tangent-space projection [17]. DLRA has been extended to tensors [18], and further results can be found in [19]. DLRA has been used to reduce the computational complexity of quantum propagation [20] by restricting the evolution to lower-rank.

Our work builds on this and other previous work to demonstrate how low-rank methods can improve radiative calculations. An earlier paper applied the low-rank approximation to solve the Vlasov equation [21]. However, in that case, the rank is not fixed, so the singular value decomposition is required in each time step to truncate the solution to the designated rank. In this work, we apply the dynamical low-rank projector splitting technique developed by Lubich and Oseledets [22] to neutral particle transport. A similar algorithm has been applied to plasma dynamics [23][24][25] and computational fluid dynamics [26].

In terms of low-rank methods for radiative transfer calculations, we presented preliminary results using DLRA at a recent conference [27]. Additionally, the asymptotic analysis of this splitting method in a one-dimensional radiative transfer equation with the backward Euler and Crank-Nicolson scheme was made in [28]. Our work presents a detailed numerical scheme for 1-D and 2-D problems and we specifically study the required storage during the solution process.

### *1.1. Overview of Projector-Splitting Method*

Here we give a brief mathematical introduction to the robust and accurate projector-splitting method [22] to perform the DLRA for matrix differential

equations of the form

$$\frac{\partial}{\partial t} A(t) \equiv \dot{A}(t) = F(A(t)),$$

for  $A(t) \in \mathbb{R}^{m \times n}$ . DLRA seeks to find an approximating matrix  $Y(t) = U(t)S(t)V^T(t)$ , where  $U(t) \in \mathbb{R}^{m \times r}$ ,  $V(t) \in \mathbb{R}^{n \times r}$  are orthonormal matrices and  $S(t) \in \mathbb{R}^{r \times r}$  so that  $Y(t)$  has rank  $r$ . Note that rank  $r$  matrices are a manifold,  $\mathcal{M}_r$ , in the space  $\mathbb{R}^{m \times n}$ . We evolve the solution on  $\mathcal{M}_r$  directly by inserting the ansatz for  $Y(t)$  into the equation and projecting the right-hand side onto the tangent space of  $\mathcal{M}_r$ .

In this work, we develop a reduced system of the radiative transfer equation with this low-rank method, where the intensity only involves its low-rank orthogonal bases rather the full phase space. We design a numerical scheme for the 2-D RTE with a finite volume discretization in space and a spherical harmonic ( $P_N$ ) expansion in angle. We further demonstrate the memory saving features by comparing the low-rank results with full-rank solutions and benchmark results.

## 2. Derivation of Method

We consider an energy-independent radiation transport equation

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

Here the angular flux  $\psi(\mathbf{r}, \hat{\Omega}, t)$  with units of particles per area per steradian per time is a function of position  $\mathbf{r} \in D$  where  $D$  is the computational domain, direction  $\hat{\Omega}(\mu, \varphi)$  where  $\mu$  is the cosine of the polar angle and  $\varphi$  is the azimuthal angle, and time  $t$ . Note that  $\hat{\Omega}$  is a unit vector. The total and isotropic scattering macroscopic cross-sections with units of inverse length are denoted as  $\sigma_t(\mathbf{r})$  and  $\sigma_s(\mathbf{r})$ , respectively,  $c$  is the particle speed and  $Q(\mathbf{r}, t)$  is a prescribed source with units of particles per volume per time. We also write the scalar flux,  $\phi(\mathbf{r}, t)$ , as the integral of the angular flux

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

In this study we approximate the solution to Eq. (1) using the form

$$\psi(\mathbf{r}, \hat{\Omega}, t) \approx \sum_{i,j=1}^r X_i(\mathbf{r}, t) S_{ij}(t) W_j(\hat{\Omega}, t); \quad (3)$$

that is, we seek the best approximation with rank  $r$  to the solution of Eq. (1), where we have written  $X_i$  as an orthonormal basis for  $\mathbf{r}$  and  $W_j$  as an orthonormal basis for  $\hat{\Omega}$  using the inner products

$$\langle f, g \rangle_{\mathbf{r}} = \int_D f g d\mathbf{r}, \quad \langle f, g \rangle_{\hat{\Omega}} = \int_{4\pi} f g d\hat{\Omega}.$$

Due to orthonormality we also have  $\langle X_i, X_j \rangle_{\mathbf{r}} = \langle W_i, W_j \rangle_{\hat{\Omega}} = \delta_{ij}$ . We also use  $\bar{X} = \{X_1, X_2, \dots, X_r\}$  and  $\bar{W} = \{W_1, W_2, \dots, W_r\}$  as ansatz spaces. The expansion in Eq. (3) is not unique and we add orthogonal constraints  $\langle X_i, \dot{X}_j \rangle_{\mathbf{r}} = 0$  and  $\langle W_i, \dot{W}_j \rangle_{\hat{\Omega}} = 0$  as gauge conditions, which identify  $X$  and  $W$  in Grassmann manifolds [29]. We now define orthogonal projectors using the bases:

$$P_{\bar{X}} g = \sum_{i=1}^r X_i \langle X_i g \rangle_{\mathbf{r}}, \quad (4)$$

$$P_{\bar{W}} g = \sum_{j=1}^r W_j \langle W_j g \rangle_{\hat{\Omega}}. \quad (5)$$

The projector onto the tangent space of the low-rank manifold  $\mathcal{M}_r$  is given in [17] as

$$P_g = P_{\bar{W}} g - P_{\bar{X}} P_{\bar{W}} g + P_{\bar{X}} g. \quad (6)$$

We apply the projectors to define a Lie splitting of the original equations into three steps; each of these is solved consecutively for a time step :

$$\begin{aligned} \partial_t \psi(\mathbf{r}, \hat{\Omega}, t) = P_{\bar{W}} \Big( & -\hat{\Omega} \cdot \nabla \psi(\mathbf{r}, \hat{\Omega}, t) - \sigma_t(\mathbf{r}, t) \psi(\mathbf{r}, \hat{\Omega}, t) \\ & + \frac{1}{4\pi} \sigma_s(\mathbf{r}, t) \phi(\mathbf{r}, t) + Q(\mathbf{r}, t) \Big), \quad (7) \end{aligned}$$

$$\begin{aligned} \partial_t \psi(\mathbf{r}, \hat{\Omega}, t) = -P_{\bar{X}} P_{\bar{W}} \Big( & -\hat{\Omega} \cdot \nabla \psi(\mathbf{r}, \hat{\Omega}, t) - \sigma_t(\mathbf{r}, t) \psi(\mathbf{r}, \hat{\Omega}, t) \\ & + \frac{1}{4\pi} \sigma_s(\mathbf{r}, t) \phi(\mathbf{r}, t) + Q(\mathbf{r}, t) \Big), \quad (8) \end{aligned}$$

$$\begin{aligned} \partial_t \psi(\mathbf{r}, \hat{\Omega}, t) = P_{\hat{X}} \Bigg( & -\hat{\Omega} \cdot \nabla \psi(\mathbf{r}, \hat{\Omega}, t) - \sigma_t(\mathbf{r}, t) \psi(\mathbf{r}, \hat{\Omega}, t) \\ & + \frac{1}{4\pi} \sigma_s(\mathbf{r}, t) \phi(\mathbf{r}, t) + Q(\mathbf{r}, t) \Bigg). \end{aligned} \quad (9)$$

Thus, equation (8) uses the solution of equation (7) as an initial condition, and the equation (9) uses the solution of equation (8) as an initial condition. It can be shown that the above evolution is contained in the low-rank manifold  $\mathcal{M}_r$ , if the initial value is in  $\mathcal{M}_r$  [22] because the right-hand side of each step remains in the tangent space  $\mathcal{T}\mathcal{M}_r$ .

The main advantage of this scheme stems from the fact that only the low-rank components  $X(\mathbf{r}, t)$ ,  $S(t)$ , and  $W(\hat{\Omega}, t)$  need to be stored during the time evolution rather than the full size solution  $\psi(\mathbf{r}, \hat{\Omega}, t)$ . To demonstrate this we first formulate the projections (7 - 9) explicitly from time  $t_0$  to  $t_0 + h$  where  $h$  is the step size. The low-rank representation of  $\psi(\mathbf{r}, \hat{\Omega}, t_0)$  is given by the initial condition

$$\psi^{(0)}(\mathbf{r}, \hat{\Omega}, t_0) = \sum_{i,j=1}^r X_i^{(0)}(\mathbf{r}, t_0) S_{ij}^{(0)}(t_0) W_j^{(0)}(\hat{\Omega}, t_0). \quad (10)$$

In the first projection the basis  $W_j$  does not change with time and is evaluated at the initial value  $W_j^{(0)}(\hat{\Omega})$ . We simplify the notation by writing  $K_j(\mathbf{r}, t) = \sum_i X_i(\mathbf{r}, t) S_{ij}(t)$ , then (10) becomes

$$\psi^{(0)}(\mathbf{r}, \hat{\Omega}, t_0) = \sum_{j=1}^r K_j^{(0)}(\mathbf{r}, t_0) W_j^{(0)}(\hat{\Omega}). \quad (11)$$

We plug this solution into Eq. (7) and multiply by  $W_\ell^{(0)}(\hat{\Omega})$  and integrate over  $\mu$  and  $\varphi$  to get

$$\begin{aligned} \partial_t K_j = & - \sum_{l=1}^r \nabla K_l \langle \hat{\Omega} W_l W_j \rangle_{\hat{\Omega}} - \sigma_t K_j + \frac{1}{4\pi} \sigma_s \sum_{l=1}^r K_l \langle W_l \rangle_{\hat{\Omega}} \langle W_j \rangle_{\hat{\Omega}} \\ & + Q \langle W_j \rangle_{\hat{\Omega}}. \end{aligned} \quad (12)$$

Equation (12) resembles the standard  $P_N$  equations, a point we will return to later. It is a system of advection problems coupled through the streaming term.

We can then factor  $K_j^{(1)}(\mathbf{r}, t_0 + h)$ , which is the solution of Eq. (12) into  $X_i^{(1)}(\mathbf{r}, t_0 + h)$  and  $S_{ij}^{(1)}(t_0 + h)$  using a QR decomposition or singular value decomposition. In the second step both the  $X_i$  and  $W_j$  are preserved. The initial conditions to solve Eq. (8) are  $X_i^{(2)}(\mathbf{r}) = X_i^{(1)}(\mathbf{r}, t_0 + h)$ ,  $S_i^{(2)}(t_0) = S_i^{(1)}(t_0 + h)$ , and  $W_i^{(2)}(\hat{\Omega}) = W_i^{(0)}(\hat{\Omega})$ . Then, we can perform similar calculations on Eq. (8) to get

$$\begin{aligned} \frac{d}{dt} S_{ij} = & \sum_{kl}^r \langle \nabla X_k X_l \rangle_{\mathbf{r}} S_{kl} \langle \hat{\Omega} W_l W_j \rangle_{\hat{\Omega}} + \sum_k^r \langle \sigma_t X_k X_i \rangle_{\mathbf{r}} S_{kj} \\ & - \frac{1}{4\pi} \sum_{kl}^r \langle \sigma_s X_k X_i \rangle_{\mathbf{r}} S_{kl} \langle W_l \rangle_{\hat{\Omega}} \langle W_j \rangle_{\hat{\Omega}} - \langle X_i Q \rangle_{\mathbf{r}} \langle W_j \rangle_{\hat{\Omega}}. \end{aligned} \quad (13)$$

We call the solution  $S_{ij}^{(2)}$ . Equation (13) defines a set of  $r^2$  ordinary differential equations. The solution is used to create an initial condition for Eq. (9), where  $X_i^{(3)}(\mathbf{r}) = X_i^{(1)}(\mathbf{r}, t_0 + h)$ ,  $S_i^{(3)}(t_0) = S_i^{(2)}(t_0 + h)$ , and  $W_i^{(3)}(\hat{\Omega}, t_0) = W_i^{(0)}(\hat{\Omega})$ . Notice that  $X_i$  does not change with time in this step.

Writing  $L_i = \sum_j^r S_{ij}(t) W_j(\hat{\Omega}, t)$  we can multiply Eq. (9) by a spatial basis function and integrate over space to get

$$\begin{aligned} \partial_t L_i = & -\hat{\Omega} \sum_k^r \langle \nabla X_k X_i \rangle_{\mathbf{r}} L_k - \sum_k^r \langle \sigma_t X_k X_i \rangle_{\mathbf{r}} L_k + \frac{1}{4\pi} \sum_k^r \langle \sigma_s X_k X_i \rangle_{\mathbf{r}} \langle L_k \rangle_{\hat{\Omega}} \\ & + \langle Q X_i \rangle_{\mathbf{r}}, \end{aligned} \quad (14)$$

which evolves the solution in  $\mu$  and  $\varphi$  spaces. Upon factoring  $L_i = S_{ij}^{(3)}(t) W_j^{(3)}(\hat{\Omega}, t)$  using a QR decomposition we can write the low-rank solution as  $\psi(\mathbf{r}, \hat{\Omega}, t_0 + h) = \sum_{i,j=1}^r X_i^{(1)}(\mathbf{r}, t_0 + h) S_{ij}^{(3)}(t_0 + h) W_j^{(3)}(\hat{\Omega}, t_0 + h)$ . Notice that there is no need to calculate and store the full solution during the evolution.

### 3. Numerical Scheme

In this section the procedure outlined above of solving Eqs. (12), (13), and (14) is implemented in one and two spatial dimensions with a first-order explicit time integrator [22] and a finite volume discretization in space. For the angular basis, we use a spherical harmonics expansion. In this section, we describe the numerical method to solve the 2D problem in detail.

### 3.1. Discretization details

In the two dimensional system, we write the spatial variables as  $x$  and  $z$ . The transport equation in this reduced geometry is

$$\begin{aligned} \frac{1}{c} \frac{\partial \psi(x, z, \mu, \varphi, t)}{\partial t} + \mu \partial_z \psi(x, z, \mu, \varphi, t) + \sqrt{1 - \mu^2} \cos \varphi \partial_x \psi(x, z, \mu, \varphi, t) \\ + \sigma_t(x, z) \psi(x, z, \mu, \varphi, t) = \frac{1}{4\pi} \sigma_s(x, z) \phi(x, z, t) + Q(x, z, t). \end{aligned} \quad (15)$$

The projection system simplifies to

$$\begin{aligned} \partial_t K_j = - \sum_{l=1}^r \partial_z K_l \langle \mu W_j W_l \rangle_{\hat{\Omega}} - \sum_{l=1}^r \partial_x K_l \langle \sqrt{1 - \mu^2} \cos \varphi W_j W_l \rangle_{\hat{\Omega}} - \sigma_t K_j \\ + \frac{\sigma_s}{2} \sum_{l=1}^r K_l \langle W_l \rangle_{\mu} \langle W_j \rangle_{\mu} + \frac{\langle W_j \rangle_{\mu}}{2} Q, \end{aligned} \quad (16)$$

$$\begin{aligned} \partial_t S_{ij} = \sum_{kl}^r \langle \partial_z X_k X_l \rangle_{\mathbf{r}} S_{kl} \langle \mu W_l W_j \rangle_{\hat{\Omega}} + \sum_{kl}^r \langle \partial_x X_k X_l \rangle_{\mathbf{r}} S_{kl} \langle \sqrt{1 - \mu^2} \cos \varphi W_l W_j \rangle_{\hat{\Omega}} \\ + \sum_k^r \langle \sigma_t X_k X_i \rangle_{\mathbf{r}} S_{kj} - \frac{1}{2} \sum_{kl}^r \langle \sigma_s X_k X_i \rangle_{\mathbf{r}} S_{kl} \langle W_l \rangle_{\hat{\Omega}} \langle W_j \rangle_{\hat{\Omega}} - \frac{1}{2} \langle X_i Q \rangle_{\mathbf{r}} \langle W_j \rangle_{\hat{\Omega}}, \end{aligned} \quad (17)$$

$$\begin{aligned} \partial_t L_i = -\mu \sum_k^r \langle \partial_z X_k X_i \rangle_{\mathbf{r}} L_k - \sqrt{1 - \mu^2} \cos \varphi \sum_k^r \langle \partial_x X_k X_i \rangle_{\mathbf{r}} L_k - \sum_k^r \langle \sigma_t X_k X_i \rangle_{\mathbf{r}} L_k \\ + \frac{1}{2} \sum_k^r \langle \sigma_s X_k X_i \rangle_{\mathbf{r}} \langle L_k \rangle_{\hat{\Omega}} + \frac{1}{2} \langle Q X_i \rangle_{\mathbf{r}}. \end{aligned} \quad (18)$$

The bases we use arise from a finite volume discretization in space with a constant mesh area  $\Delta x \Delta z$  and  $N_x \times N_z$  zones, and the truncated at  $N_l$  spherical harmonics in angle. To make orthonormal bases we define

$$X_i(t, x, z) = \sum_{p=1}^{N_x} \sum_{q=1}^{N_z} Z_{pq}(x, z) u_{pqi}(t) \quad (19)$$

$$W_j(t, \mu, \varphi) = \sum_{l=0}^{N_l} \sum_{k=0}^l Y_l^k(\mu, \varphi) v_{lkj}(t) \quad (20)$$



The spatial basis functions are given by  $Z_{pq}(x, z) = \frac{1}{\sqrt{\Delta x \Delta z}}$  with  $x \in [x_{p-\frac{1}{2}}, x_{p+\frac{1}{2}}]$  and  $z \in [z_{q-\frac{1}{2}}, z_{q+\frac{1}{2}}]$ , where  $p$  and  $q$  are the cell index. Note that  $Z_{pq}(x, z) = 0$  outside the cell  $\{p, q\}$ . The spherical harmonics are defined as

$$Y_l^k(\mu, \varphi) = \sqrt{\frac{2l+1}{4\pi} \frac{(l-k)!}{(l+k)!}} P_l^k(\mu) e^{ik\varphi},$$

where  $P_l^k(\mu)$  is the associate Legendre polynomial. The negative  $k$  are not necessary here because of the recursion properties of the spherical harmonics [30]. Here  $u_{pqi}$  and  $v_{lkj}$  are components of the time dependent tensor  $U(t) \in \mathbb{R}^{N_x \times N_z \times r}$  and  $V(t) \in \mathbb{R}^{N_l \times (N_l+1) \times r}$ . Additionally, in 2-D Cartesian geometry, the imaginary part of the spherical harmonics can be decoupled and ignored [31].

We also notice that the computations in Eqs. (19), (20) can be performed by performing matrix multiplications instead of tensor operation. This is achieved by rearranging the matrices  $Z \in \mathbb{R}^{N_x \times N_z}$  and  $Y \in \mathbb{R}^{\frac{1}{2}(N_l+1)(N_l+2)}$  into vectors and tensors  $U$  and  $V$  to matrices.

To make the notation consistent we denote the number of degrees of freedom in space as  $m = N_x N_z$  and the angular degrees of freedom as  $n = \frac{1}{2}(N_l+1)(N_l+2)$ . In addition, the indices of  $U$  and  $V$  conform to Eqs. (19) and (20).

Consequently the new ansatz spaces  $X$  and  $W$  have the form

$$X = UZ = \begin{bmatrix} u_{1\ 1\ 1} & u_{1\ 1\ 2} & \dots & u_{1\ 1\ r} \\ \dots & \dots & \dots & \dots \\ u_{N_x\ 1\ 1} & u_{N_x\ 1\ 2} & \dots & u_{N_x\ 1\ r} \\ u_{1\ 2\ 1} & u_{1\ 2\ 2} & \dots & u_{1\ 2\ r} \\ \dots & \dots & \dots & \dots \\ u_{N_x\ N_z\ 1} & u_{N_x\ N_z\ 2} & \dots & u_{N_x\ N_z\ r} \end{bmatrix}^T \begin{bmatrix} Z_{1\ 1} \\ \dots \\ Z_{N_x\ 1} \\ Z_{1\ 2} \\ \dots \\ Z_{N_x\ N_z} \end{bmatrix}$$

$$W = VY = \begin{bmatrix} v_{0\ 0\ 1} & v_{0\ 0\ 2} & \dots & v_{0\ 0\ r} \\ v_{0\ 1\ 1} & v_{0\ 1\ 2} & \dots & v_{0\ 1\ r} \\ v_{1\ 1\ 1} & v_{1\ 1\ 2} & \dots & v_{1\ 1\ r} \\ v_{0\ 2\ 1} & v_{0\ 2\ 2} & \dots & v_{0\ 2\ r} \\ \dots & \dots & \dots & \dots \\ v_{N_l\ N_l\ 1} & v_{N_l\ N_l\ 2} & \dots & v_{N_l\ N_l\ r} \end{bmatrix}^T \begin{bmatrix} Y_0^0 \\ Y_1^0 \\ Y_1^1 \\ Y_2^0 \\ \dots \\ Y_{N_l}^{N_l} \end{bmatrix}$$

To solve Eqs. (17) and (18) we need to calculate spatial integration terms like  $\langle \partial_z X_k X_l \rangle_{\mathbf{r}}$ . Due to our use of finite volume method, the basis  $X_i$  is discontinuous and piecewise constant in space. We apply integration by parts and obtain

$$\begin{aligned} \langle \partial_z X_k X_l \rangle_{\mathbf{r}} &= \left\langle \partial_z \left( \sum_p^{N_x} \sum_q^{N_z} Z_{pq} u_{pqk} \right) \sum_{p'}^{N_x} \sum_{q'}^{N_z} Z_{p'q'} u_{p'q'l} \right\rangle_{\mathbf{r}} \\ &= \sum_{p'}^{N_x} \sum_{q'}^{N_z} u_{p'q'l} \left( \left\langle \partial_z (Z_{p'q'} \sum_p^{N_x} \sum_q^{N_z} Z_{pq} u_{pqk}) \right\rangle_{\mathbf{r}} - \left\langle \cancel{\partial_z Z_{p'q'}}^0 \sum_{p'}^{N_x} \sum_{q'}^{N_z} Z_{p'q'} u_{pqk} \right\rangle_{\mathbf{r}} \right) \\ &= \sum_p^{N_x} \sum_q^{N_z} u_{pql} \left( \frac{\{u\}_{p,q+\frac{1}{2},k} - \{u\}_{p,q-\frac{1}{2},k}}{2\Delta z} \right) \end{aligned} \quad (21)$$

Here,  $\{u\}_{p,q+\frac{1}{2},k}$  denotes the value at the cell boundary between the spatial cells  $q$  and  $q+1$ . We set the value at the cell boundary by the average of its left and right values, such as  $\{u\}_{p,q+\frac{1}{2},k} = \frac{1}{2}(u_{p,q+1,k} + u_{p,q,k})$ .

For the angular terms, e.g.,  $\langle \mu W_j W_{j'} \rangle_{\hat{\Omega}}$ , we can write the integral as

$$\begin{aligned} \langle \mu W_j W_{j'} \rangle_{\hat{\Omega}} &= \left\langle \sum_{l=1}^{N_l} \sum_{k=-N_l}^{N_l} Y_l^k(\mu, \varphi) v_{lkj}(t) \sum_{l'=1}^{N_l} \sum_{k'=-N_l}^{N_l} Y_{l'}^{k'}(\mu, \varphi) v_{l'k'j'}(t) \right\rangle_{\hat{\Omega}} \\ &= \left\langle \sum_{l=1}^{N_l} \sum_{k=-N_l}^{N_l} \sum_{l'=1}^{N_l} \sum_{k'=-N_l}^{N_l} v_{lkj}(t) \mu Y_l^k(\mu, \varphi) Y_{l'}^{k'}(\mu, \varphi) v_{l'k'j'}(t) \right\rangle_{\hat{\Omega}} \\ &= \sum_{l=1}^{N_l} \sum_{k=-N_l}^{N_l} \sum_{l'=1}^{N_l} \sum_{k'=-N_l}^{N_l} v_{lkj}(t) \left\langle \mu Y_l^k(\mu, \varphi) Y_{l'}^{k'}(\mu, \varphi) \right\rangle_{\hat{\Omega}} v_{l'k'j'}(t) \end{aligned} \quad (22)$$

There are two ways to calculate this term. One is to precompute the time-independent part  $\langle \mu Y_l^k(\mu, \varphi) Y_{l'}^{k'}(\mu, \varphi) \rangle_{\hat{\Omega}}$  which forms a  $n \times n$  matrix  $C$ . Thus

Eqs. (22) requires  $\mathcal{O}(n^2r)$  operations, which is affordable because usually  $n$  is not large and  $C$  is sparse. Alternatively, we could calculate  $\langle \mu W_j W_{j'} \rangle_\mu$  on-the-fly by choosing  $\mathcal{O}(n)$  quadrature points in angle, and it requires  $\mathcal{O}(nr^2)$  operations for all the  $r^2$  entries. In this work we chose the first method which is more straightforward to implement.

### 3.2. Upwind scheme with slope reconstruction

We apply the standard upwinding technique to solve the advection equation Eqs. (16). For the space derivative term we use upwinding to write

$$\begin{aligned} \partial_z K_j \langle \mu W_j W_l \rangle_{\hat{\Omega}} &= \frac{1}{\Delta z} (K_{p,q+\frac{1}{2},j} - K_{p,q-\frac{1}{2},j}) \langle \mu W_j W_l \rangle_{\hat{\Omega}} \\ &= \frac{K_{p,q+1,j} - K_{p,q-1,j}}{2\Delta z} (V^T C V)_{jl} - \frac{K_{p,q+1,j} - K_{p,q,j} + K_{p,q-1,j}}{2\Delta z} (V^T \Sigma V)_{jl} \\ &= \frac{1}{\sqrt{\Delta x \Delta z}} \sum_{i=1}^r \sum_{p=1}^{N_x} \sum_{q=1}^{N_z} \left( \left( \frac{u_{p,q+1,i} - u_{p,q-1,i}}{2\Delta z} \right) S_{ij} (V^T C V)_{jl} \right. \\ &\quad \left. - \left( \frac{u_{p,q+1,i} + u_{p,q-1,i} - 2u_{p,q,i}}{2\Delta z} \right) S_{ij} (V^T \Sigma V)_{jl} \right) \end{aligned} \quad (23)$$

where  $\Sigma$  is a stabilization matrix that we take to be a diagonal matrix with the singular values of  $C$ . Other stabilization terms could be used, including Lax-Friedrichs where  $V^T \Sigma V$  is replaced by a constant times an identity matrix. This method allows the spatial variables to be differentiated separately, so the same treatment can be applied to  $\partial_x K_l \langle \sqrt{1 - \mu^2} \cos \varphi W_j W_l \rangle_{\hat{\Omega}}$ .

Additionally, the harmonic mean limiter is adopted to reconstruct this slope term  $K_j$  for better accuracy. We define the slope of  $z$  direction in the edge of the cell  $(p, q)$  as

$$m_{p,q}^+ = \frac{K_{p,q,j} - K_{p,q-1,j}}{\Delta z}, \quad m_{p,q}^- = \frac{K_{p,q+1,j} - K_{p,q,j}}{\Delta z}.$$

Then the slope of this cell is

$$m_{p,q} = \begin{cases} \frac{2m_{p,q}^+ m_{p,q}^-}{m_{p,q}^+ + m_{p,q}^-} & \text{if } m_{p,q}^+ m_{p,q}^- > 0 \\ 0 & \text{otherwise} \end{cases} \quad (24)$$

With the reconstructed edge value then

$$K_{p,q,j}^+ = K_{p,q,j} + \frac{\Delta z}{2} m_{p,q}, \quad K_{p,q,j}^- = K_{p,q,j} - \frac{\Delta z}{2} m_{p,q},$$

Eqs. (23) can be written as

$$\begin{aligned} \partial_z K_j \langle \mu W_j W_l \rangle_{\hat{\Omega}} &= \frac{1}{\Delta z} (K_{p,q+\frac{1}{2},j} - K_{p,q-\frac{1}{2},j}) \langle \mu W_j W_l \rangle_{\hat{\Omega}} \\ &= \frac{1}{2\Delta z} \left( (K_{p,q+1,j}^- + K_{p,q,j}^+ - K_{p,q-1,j}^+ - K_{p,q,j}^-) (V^T C V)_{jl} \right. \\ &\quad \left. - (K_{p,q+1,j}^- - K_{p,q,j}^+ - K_{p,q,j}^- + K_{p,q-1,j}^+) (V^T \Sigma V)_{jl} \right), \end{aligned} \quad (25)$$

Spherical harmonics expansions can yield oscillatory or negative solutions [4]. To address this issue we implemented angular filtering [5, 32, 33] which can significantly increase the performance of  $P_n$  method in solving radiative transfer equation by removing the oscillations. We implemented the Lanczos filter into our explicit solver by using an equivalent equation approach [33] and combined it with the low-rank approximation algorithm. The filtered equation has the form of

$$\begin{aligned} \frac{1}{c} \frac{\partial \psi(\mathbf{r}, \hat{\Omega}, t)}{\partial t} + \hat{\Omega} \cdot \nabla \psi(\mathbf{r}, \hat{\Omega}, t) + \sigma_t(\mathbf{r}) \psi(\mathbf{r}, \hat{\Omega}, t) + \beta \sigma_f(\mathbf{r}, \hat{\Omega}) \psi(\mathbf{r}, \hat{\Omega}, t) \\ = \frac{1}{4\pi} \sigma_s(\mathbf{r}) \phi(\mathbf{r}, t) + Q(\mathbf{r}, t). \end{aligned} \quad (26)$$

where the free parameter  $\beta$  is the filter strength,  $\sigma_f = \log \frac{\sin \eta}{\eta} \frac{l}{l+1}$  and  $\eta$  is the order of the  $P_N$  expansion and  $l$  is the index of  $P_N$  moments.

### 3.3. Memory Reduction and Computational Efficiency

The memory footprint required to compute the solution is based on storing the matrices  $U$ ,  $V$ , and  $S$ . Therefore, the memory required is  $2(mr + r^2 + nr)$  where the factor 2 assumes that we need to store the previous step solution as well as the new step. The full solution to this problem without splitting would require a memory footprint of  $2mn$ . Therefore, for  $r \ll m, n$ , there will be large memory savings.

Given that double-precision floating-point format occupying 8 bytes in computer memory and 1 megabyte (MB) =  $10^6$  megabytes, then the required mem-

ory for the low-rank algorithm measured in MB is

$$\text{memory} = 16 \times (mr + r^2 + nr) \times 10^{-6}, \quad (27)$$

and for the full solution the memory required is

$$\text{memory} = 16 \times mn \times 10^{-6}. \quad (28)$$

The low-rank scheme reduces computational cost significantly as well. It is straightforward that solving equation (16) requires  $\mathcal{O}(mr^2)$  operations. Note that the computation of the  $\langle \mu W_j W_{j'} \rangle_{\hat{\Omega}}$  term in Eq. (22) takes  $\mathcal{O}(nr^2)$  operations. Equation (17) also needs  $\mathcal{O}(mr^2)$  operations where the term (21) is the heaviest computational burden. Since  $X$  is preserved in equation (17) and (18), there is no need to compute term (21) again in the third equation. Therefore, equation (18) only requires  $\mathcal{O}(nr^2)$  operations. The two QR decompositions consume another  $\mathcal{O}(mr^2) + \mathcal{O}(nr^2)$ . Combining all of these costs, in total the low-rank scheme requires  $\mathcal{O}(mr^2) + \mathcal{O}(nr^2)$  operations whereas the full solution needs  $\mathcal{O}(mn^2)$ .

We note that we have not considered the applications of the above algorithm on distributed-memory parallel computation. In such a scenario the projection operators, and the procedures required to obtain them, may hurt scalability and performance when applied globally. We contend that applying the low-rank technique on individual processing elements (PEs) would alleviate this shortcoming. The full solution would only be required on domain boundaries, and each PE would be able to have its own low-rank structure, possibly improving the results. A study of such a implementation is outside the scope of this paper and will be the focus of future research.

### 3.4. Conservation

The low-rank algorithm we have described does not conserve the number of particles. This loss of conservation is a result of information lost in the algorithm when restricting the solution to low-rank descriptions. We have addressed this by globally scaling the solution after each time step to correct for

any particles lost. We can do this because we know the number of particles that are absorbed, travel across the boundary, and born from sources. This correction will not preserve higher moments of the solution, and it only preserves the zeroth-order moment in space and angle. This issue has also been addressed in [24], where a correction term calculated by the imposed conservation law is added to each splitting step to ensure global and/or local conservation of mass and momentum (the zeroth and first moments in our problem) by the solution of a minimization problem. The procedure outlined in [24] could be applied to the radiative transfer equations. We choose the simpler scaling because we are primarily interested global conservation of the zeroth moment as it is through this quantity where the primary coupling to other physics occurs.

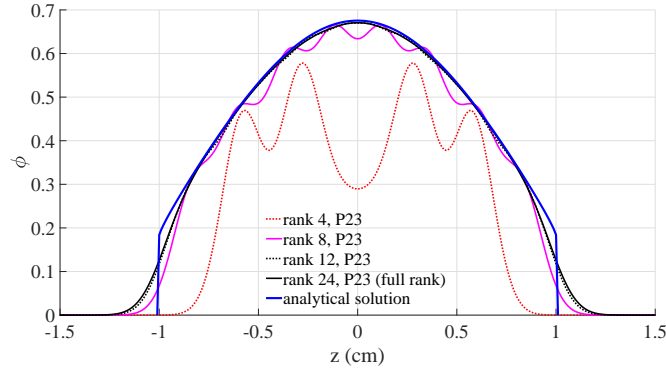
#### 4. Numerical Results

We run the transport simulations with five test problems, including the plane source problem and Reed’s problem in 1D slab geometry in addition to the line source problem, lattice problem and double chevron problem in 2D planar geometry. We note that it is common for radiation transport problems to have many more spatial degrees of freedom  $m$  than angular degrees of freedom  $n$ . To test our method under these common circumstances we fix the spatial resolution and vary  $n$  and the maximum rank  $r$  in the following simulations. The CFL condition with the formula  $CFL = \frac{\Delta t}{\Delta z}$  in 1D and  $CFL = \min(\frac{\Delta t}{\Delta x}, \frac{\Delta t}{\Delta z})$  in 2D is set to 0.2, except for the line source problem where  $CFL = 0.1$ . The conservation fix is applied to the plane and line source problems unless specified otherwise. Additionally, we used one simulation to demonstrate that the low-rank algorithm is compatible with the filter, other than that all the results are unfiltered. Our implementation is written in MATLAB, and our particle speeds are set to  $c = 1$  cm/s.

We use Eqs. (27) or (28) to calculate the theoretical value of the memory footprint for the computation. Additionally, we adopt a MATLAB built-in function “memory” to measure the actual memory usage for the 2D problems. It

is evaluated by recording the maximum running memory during the simulation and subtracting the memory occupied by MATLAB before the solution function was called. Along with the memory, we track the execution time with the “profile” function in MATLAB. We point out that the recorded memory and running time could vary a small percentage over different simulations. To account for this variation, we take the average values of the memory usage and the running time in 20 simulations for each test. The unit of memory in this section is MB.

(a) Low-rank solutions without scaling



(b) Low-rank solutions with scaling

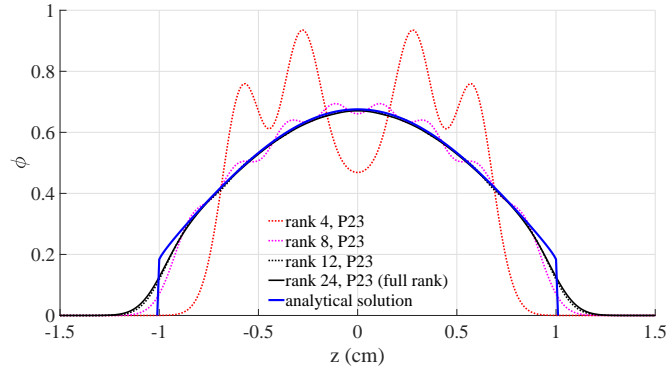


Figure 1: Solutions to the plane source problem at  $t = 1$  s using the low rank method compared to the analytic solution. Numerical smearing errors are observed in both edges due to numerical dissipation [34].

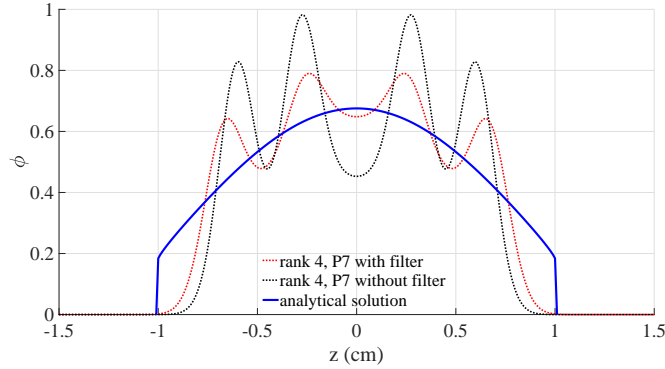


Figure 2: Comparison of  $P_7$  solutions of rank 4 with and without a filter to the analytic solution at  $t = 1$  s.

#### 4.1. Plane source problem

The plane source problem describes a plane of isotropically moving particles emitted at  $t = 0$  in a purely scattering medium with no source, i.e.,  $\sigma_t = \sigma_s = 1 \text{ cm}^{-1}$  and  $Q = 0$ . The only spatial variable in this slab geometry is  $z$ ; the initial condition is given by a delta function as  $\psi(z, \mu, t) = \delta(z)/2$ . For the simulation parameters, we set the spatial resolution to be  $\Delta z = 0.01 \text{ cm}$ , which corresponds to the number of mesh zones  $m = 300$  for the  $t = 1 \text{ s}$  solution and  $m = 1200$  for  $t = 5 \text{ s}$ .  $P_{23}$  solutions with different rank are compared. When used, the filter strength  $\beta$  is set to 50. The analytical benchmark solution that we compare to was given by Ganapol [35].

Figure 1 shows the solutions of varying rank and Legendre polynomial orders with and without the conservation fix where the zeroth moment is scaled. As can be seen in either case, the solution with rank 12, which is half of the full rank, matches the analytic solution to the scale of the graph in the middle part of the problem. It also agrees well with the full rank solution. The rank 8 solutions still capture the analytical solution well. However, the loss of conservation can be observed when the conservation fix is not applied. Rank 4 is not sufficient for accurate results and shows more affect from the loss of conservation. As shown in Figure 2, the low-rank solution can be improved by the filter:  $P_7$  solutions



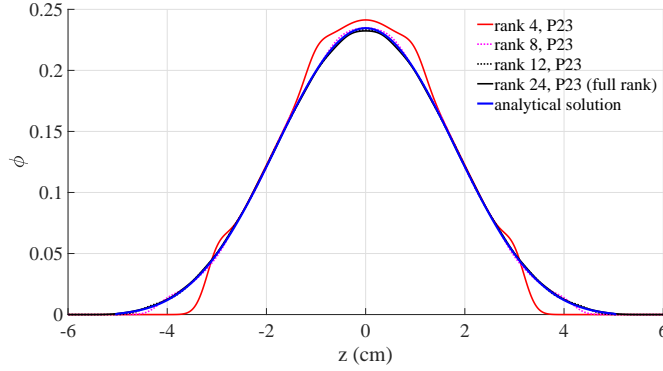
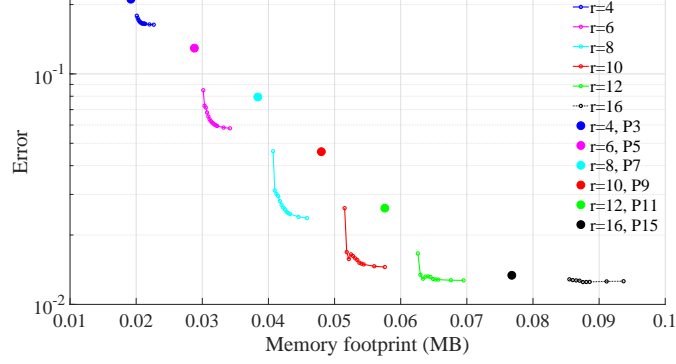


Figure 3: Solutions to the plane source problem using the low rank method compared to the analytic solution at  $t = 5$  s.

of reduced rank improve when a filter is used. Figure 3 presents the solution at  $t = 5$  s, at which even rank 8 appears to be sufficient. At this later time, there are few remaining uncollided particles from the initial condition. Therefore, fewer angular degrees of freedom is needed.

For a more quantitative comparison, the root mean square (RMS) error of the numerical results with different  $n$  and  $r$  is shown in Figure 4. In this figure, the colors for the symbols and lines correspond to the rank used in a calculation, and different values of the  $n$ , the number of angular basis functions, are corresponding dots. For each color the value of  $n$  ranges from  $r$  to 100. The large points are the value of the error using the standard full rank method with  $r = n$ . We can observe that the low-rank solution is more accurate than the full rank with the same memory usage. For example, the error of full rank solution  $n = 12$  at  $t = 1$  s using with a memory footprint of 0.057 MB is about 0.026. With the same memory, the error can be reduced to 0.013. We can also use 70% of the memory to achieve the same accuracy as seen in the  $r = 8$  error. Increasing the resolution and rank will contribute to the accuracy of solutions. Given the way we performed this study with a fixed spatial mesh and time step and the conservation fix we used, we can see some error stagnation in the low-rank solution at  $t = 5$  s. Other numerical experiments indicate that increasing

(a) The error of low rank solutions at  $t=1$



(b) The error of low rank solutions at  $t=5$

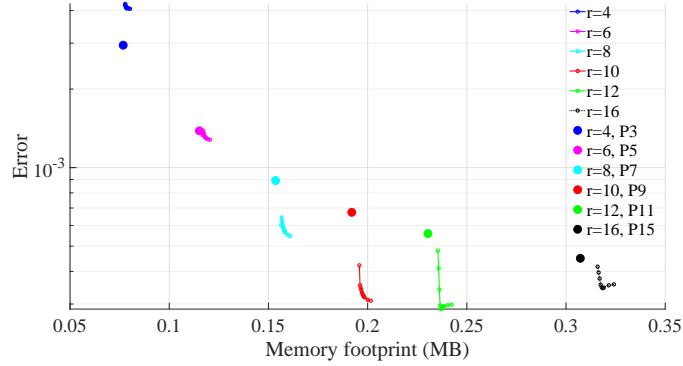


Figure 4: The comparison of errors on the plane source problem with  $m = 300$  and with different theoretical memory usage by varying  $r$  and  $n$  are shown. Each line represents the error with a fixed rank that varies the number of angular basis functions  $n$ . The bold dot denotes the full rank solution with  $n = r$ .

the number of spatial zones can further decrease the error.

#### 4.2. Reed's problem

The second test problem is Reed's problem, which is a multi-material problem, and its set-up is detailed in Figure 5. Because Reed's problem does not have an analytical solution, a numerical result with a high degree of angular basis and full rank, where  $\Delta z = 0.01$  cm and  $P_{99}$  (corresponding to  $m = 1600$

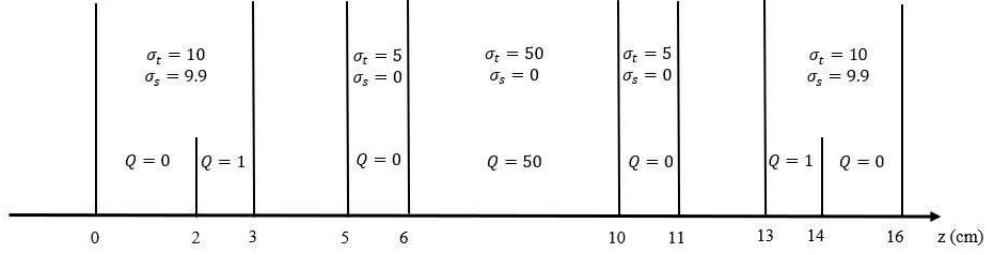


Figure 5: The material layout in Reed's problem where the blank zone means vacuum and the unit of cross sections is  $\text{cm}^{-1}$ .

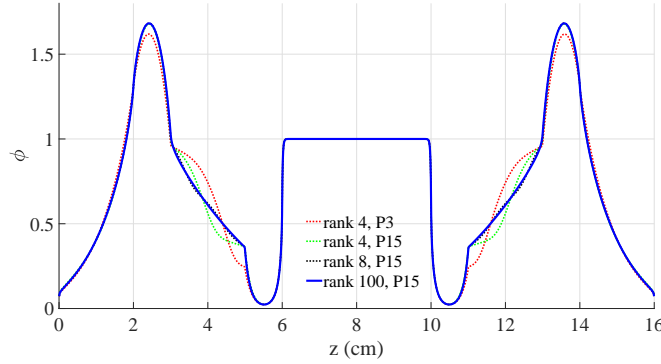


Figure 6: Solutions to the Reed's problem at  $t = 5$  s using the low rank method compared to the high-degree full rank solution. The rank 8  $P_{15}$  solutions are mostly obscured by the  $P_{99}$  solution in this plot.

and  $n = 100$ ) is set as a benchmark for memory analysis. Figure 6 shows that the rank 4 solutions differ in the vacuum regions ( $z \in (3, 5)$  and  $z \in (11, 13)$ ) and the scattering region with source ( $z \in (2, 3)$  and  $z \in (13, 14)$ ), but the rank 8 solution matches the  $P_{99}$  reference solution well.

It can be observed in Figure 7 that the low-rank solutions (solid lines with small dots) can give solutions with comparable errors to the full rank solutions (large dots) with much larger memory. For example, the rank 10 solutions obtain a solution error better than the full rank  $P_{19}$  solution with less memory.

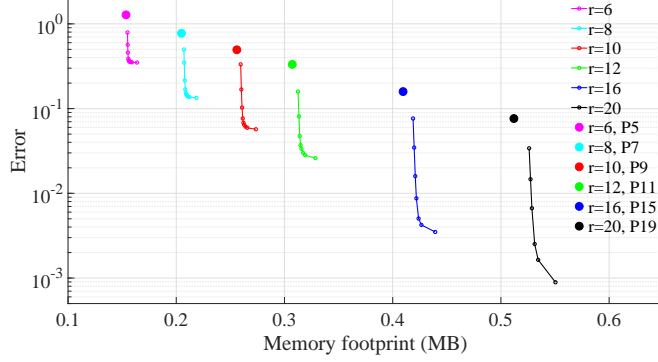


Figure 7: The comparison of errors for Reed's problem with  $m = 1600$  and with different values for  $r$  and  $n$ . Each line represents the error with a fixed rank  $r$  that varies the number of angular basis functions  $n$ . The bold dot denotes the full rank solution.

#### 4.3. Line source problem

The line source problem is a natural extension of the plane source problem to 2-D, where the plane source becomes a line. In this problem we have  $\sigma_t = \sigma_s = 1 \text{ cm}^{-1}$ ,  $Q = 0$  and the initial condition  $\psi(x, z, \mu, \varphi, t) = \delta(x)\delta(z)/4\pi$ . We use a computational domain of  $[-1.5, 1.5] \times [-1.5, 1.5]$  for the simulation time  $t = 1 \text{ s}$ , while the spatial grid is set to be  $150 \times 150$ .

The analytical solution from Ganapol [35] and the rank 210 solutions with  $P_{19}$ ,  $P_{29}$  and  $P_{39}$  are shown in Figure 8, from which we can see that the  $P_{15}$  solution which is full rank, has large oscillations. The results obtained with the same rank but more angular basis functions are much better, as we can see in Figure 8e where fixing the rank and increasing the expansion order improves the solution dramatically. Figure 9 demonstrates that increasing the angular resolution while fixing the rank improves the solution considerably. In this figure, we note that the full rank  $P_{15}$  solution has negative values, which are due to the oscillatory property of the  $P_N$  method [4]. This nonphysical phenomenon can be alleviated by increasing the order of  $P_N$ ; using the low-rank approximation this requires only a modest increase in memory.

Figure 10 shows that memory saving is more significant in the 2D problem,

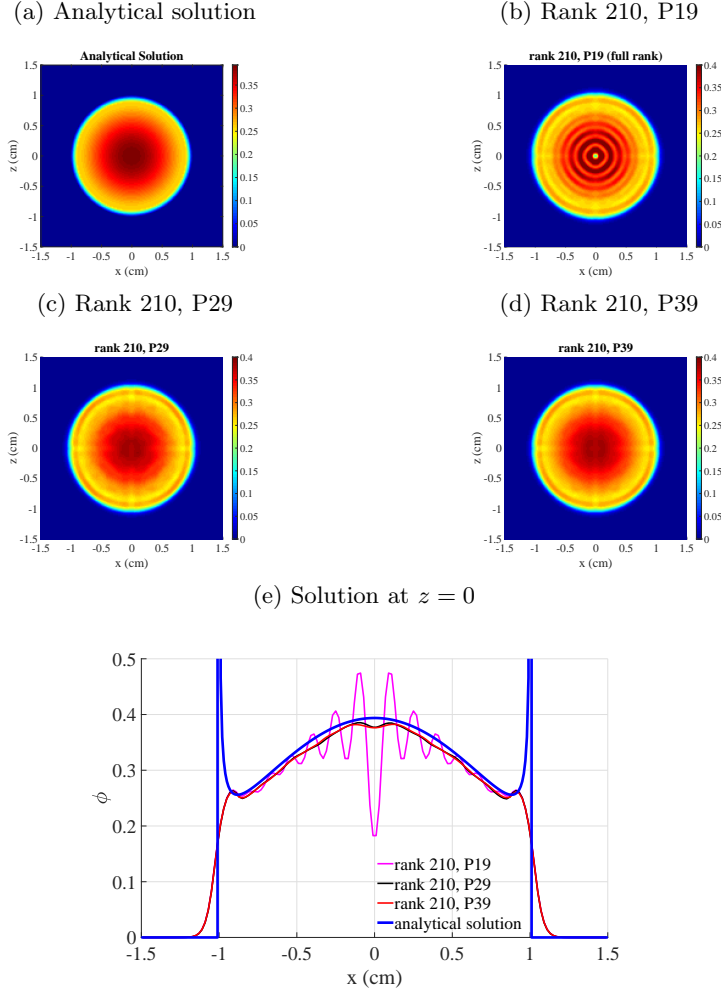


Figure 8: Solutions to the line source problem at  $t = 1$  s using rank 210 compared to the analytic solution.

where the number of spatial and angular degrees of freedom is large. As we can see, the theoretical memory footprint is a strong function of rank. It indicates that the accuracy can be improved as we increase the  $P_N$  order with modest additional cost. For example, the error of the solution at rank 78 with the low-rank method is four times smaller than the full rank solution.

For this problem we measured the memory usage while the program was

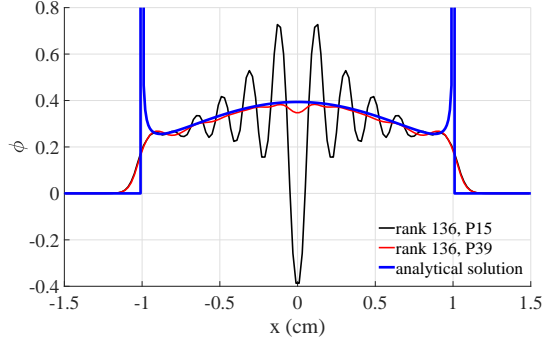


Figure 9: Solutions to the line source problem at  $t = 1$  s using rank 136 compared to the analytic solution at the cut  $z = 0$ .

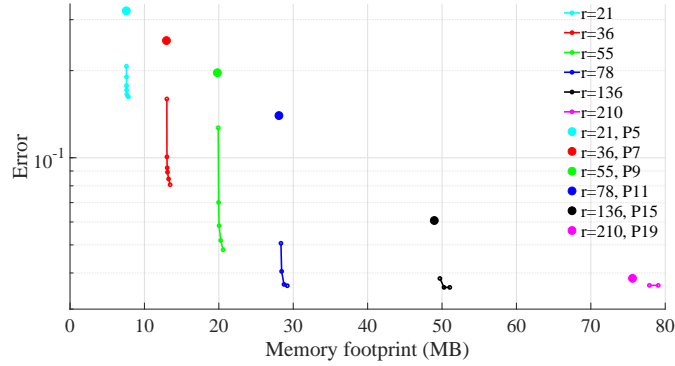


Figure 10: The comparison of errors for the line source problem with different memory usage are shown. Each dotted line represents the error with a fixed rank that varies the number of angular basis functions  $n$ . The bold dot denotes the full rank solution.

running to compare it to the theoretical values. We also measured the time required per time step. These results, as well as the theoretical memory usage and RMS error, are shown in Figure 11 and Table 1. The implementation used for the full-rank results is a standalone code that does not perform any of the projections or other special steps required in the low-rank algorithm. From this table we see that our MATLAB implementation is using memory very close to the theoretical values. Due to using sparse matrices where possible in both implementations, the actual memory used can be less than the theoretical

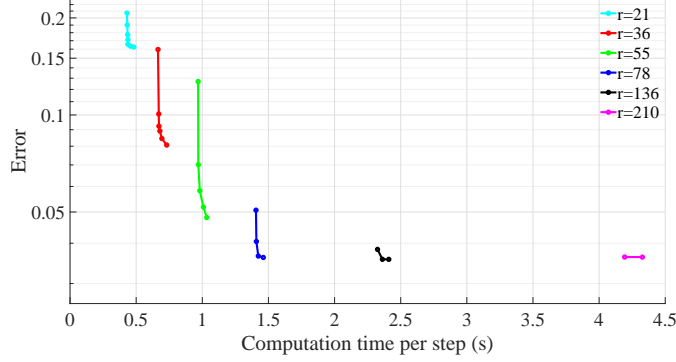


Figure 11: The comparison of errors for the low rank solutions of line source problem with different computation time are shown. Each dotted line represents the error with a fixed rank that varies the number of angular basis functions  $n$ .

values. Furthermore, as we shall discuss more later, the actual memory used is a linear function of the theoretical size, indicating that the theoretical value is a reasonable measure of the memory required.

As can be seen from Figure 11, the running time of the low-rank solutions depends strongly on rank as well. We point out that the shape of each line in this figure is similar to the line with same rank in Figure 10. This implies that increasing the  $P_N$  order in low-rank solutions with the rank fixed will only result in a slight growth in computational cost, when  $m \gg n, r$ . This is nearly ideal: we decrease memory and computation time by using our low-rank method.

From Table 1 we see that the memory and time difference between a full-rank calculation and a low-rank calculation of the same rank and larger  $n$  requires is relatively small. However, there is an increase in the running time when switching from the simpler full-rank method to the low-rank method. Nevertheless, when we look at the error per time we see that the low-rank results are significantly better in terms of memory and running time: The low-rank solution of  $P_{39}$  with  $r = 78$  requires an order of magnitude less memory (28.06 MB to 298.88 MB) and is over six times faster (1.46 to 9.79 s) to achieve a lower error (0.0361 to 0.0387).

Table 1: The quantitative comparison for the line source problem of memory footprints, running time and errors with the same rank but different angular approximations.

	rank 55, $P_9$ (full rank)	rank 55, $P_{39}$ (low rank)	rank 78, $P_{11}$ (full rank)	rank 78, $P_{39}$ (low rank)	rank 820, $P_{39}$ (full rank)
Theoretical memory (MB)	19.80	20.57	28.08	29.20	295.20
MATLAB memory (MB)	19.69	20.24	27.98	28.06	298.88
Running time per step (s)	0.68	1.08	0.92	1.46	9.79
Error	0.2071	0.0481	0.1473	0.0361	0.0387

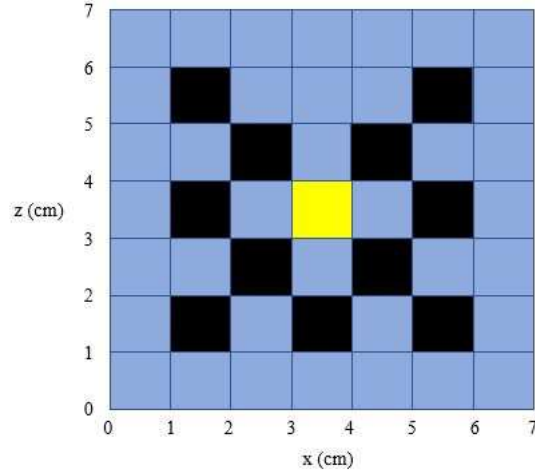


Figure 12: The material layout in the Lattice problem. The blue zones are purely scattering region, the black are absorbing region and the yellow is also the scattering region with an isotropic source which is turned on at  $t=0$ . The checkerboard is surrounded by vacuum.



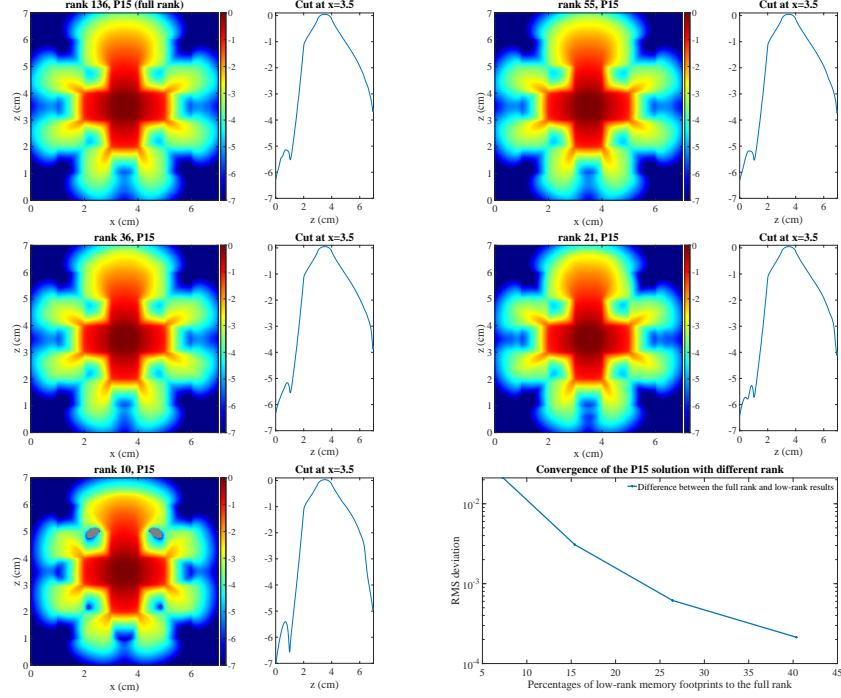


Figure 13: Solutions to Lattice problem at  $t = 3.2 s$  with  $P_{15}$  and different rank. The color scale is logarithmic.

#### 4.4. Lattice problem

As described in Figure 12, the lattice problem is a  $7 \times 7$  checkerboard with purely scattering zones  $\sigma_t = \sigma_s = 1 \text{ cm}^{-1}$ , purely absorbing zones  $\sigma_t = 10 \text{ cm}^{-1}$ ,  $\sigma_s = 0$ , and an isotropic source at the center where  $Q = 1$  [31]. In this problem, we use a spatial grid of size  $210 \times 210$  for the computational domain  $[0, 7] \times [0, 7]$ .

Figure 13 shows that the solution with different rank give similar results except in the absorbing zones, where solutions such as rank 55 and 36 contain more details than the rank 21 and 10 solutions. Additionally, low-rank solutions converge to the full rank as we increase the rank, which leads to more memory usage. As we can see that even with 26% of the full rank memory the difference is still small. Furthermore in this case the running time is 57% shorter than the full rank solution as suggested in Table 2.

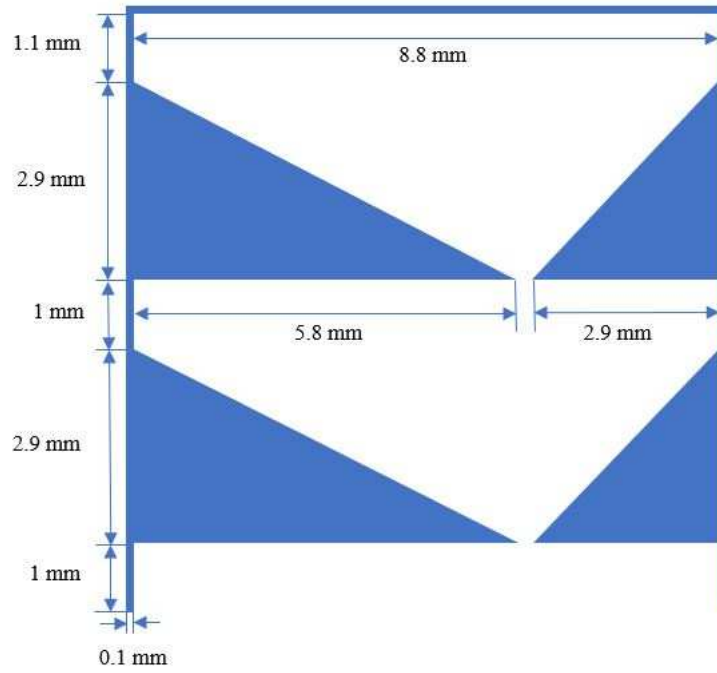


Figure 14: The layout for the double chevron problem. The blue areas are walls and obstacles made up by dense materials. The blank is the purely-scattering dilute medium. The bottom has an isotropic incoming source boundary condition and the other three sides have vacuum boundary conditions.

Table 2: The quantitative comparison for the Lattice problem of memory footprints, running time, and errors with the same angular discretizations but different rank.

	rank 136, $P_{15}$ (full rank)	rank 10, $P_{15}$ (low rank)	rank 21, $P_{15}$ (low rank)	rank 36, $P_{15}$ (low rank)	rank 55, $P_{15}$ (low rank)
Theoretical memory (MB)	95.96	7.08	14.87	25.50	38.98
MATLAB memory (MB)	95.59	6.95	14.65	25.24	39.94
Running time per step (s)	3.19	0.45	0.84	1.36	2.03

Table 3: The quantitative comparison for the double chevron problem of memory footprints, running time, and errors with the same rank but different angular discretizations.

	rank 36, $P_7$ (full rank)	rank 36, $P_{125}$ (low rank)	rank 136, $P_{15}$ (full rank)	rank 136, $P_{125}$ (low rank)	rank 8001, $P_{125}$ (full rank)
Theoretical memory (MB)	4.67	9.29	17.63	35.33	1036.93
MATLAB memory (MB)	4.50	9.09	17.48	34.76	1036.75
Running time per step (s)	0.26	1.16	0.64	2.31	34.48
RMS deviation to the full rank $P_{125}$ solution	$3 \times 10^{-3}$	$1.36 \times 10^{-5}$	$6.74 \times 10^{-4}$	$1.61 \times 10^{-8}$	—

#### 4.5. Double Chevron problem

In the last benchmark, we test our algorithm in an asymmetric geometry as shown in Figure 14. The  $9 \times 9$  spatial domain consists of 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  $Q = 1$ . In this problem, we use the spatial grid of size  $90 \times 90$  for the computational domain  $[0, 9 \text{ mm}] \times [0, 9 \text{ mm}]$ ,

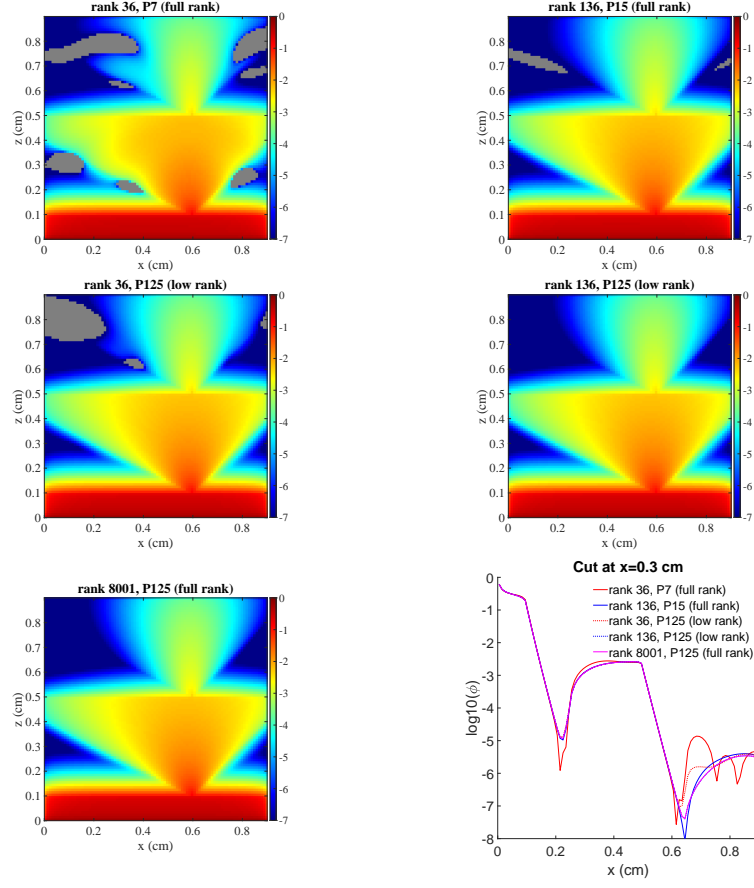


Figure 15: Solutions to the double chevron problem at  $t = 0.9$  s. The color scale is logarithmic and negative regions are shaded gray. Note the rank 136  $P_{125}$  solution is obscured by the full rank  $P_{125}$  solution.

and the simulation time  $t = 0.9$  s. This problem is designed to show the benefits of our low rank method when  $m \approx n$ . Note that  $m = 8100$  and  $n = 8001$  with  $P_{125}$ .

It can be seen from Figure 15 that the full rank  $P_{125}$  solution is identical in the scale of the figure to the rank 136 solution with  $P_{125}$ . Another observation we can make is that both the full rank  $P_7$  and  $P_{15}$  solutions are not able to capture the particle distribution behind the obstacles, leading to negative values, as indicated by the gray shading in the figure. Fortunately, these results can

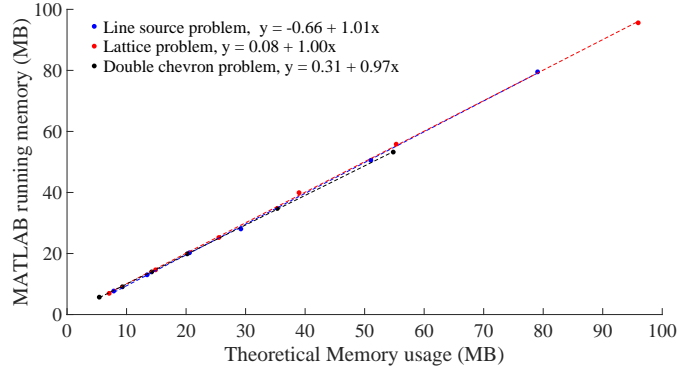


Figure 16: The relationship between the memory calculated by Eqs. (27) and the running memory in MATLAB for the line source problem, the lattice problem and the double chevron problem. The angular discretization for the line source problem is  $P_{39}$  with ranks varying from 21 to 210, the lattice problem is with  $P_{15}$  and  $r = 10, 21, 36, 55, 78, 136$  and the double chevron is with  $P_{125}$  and the ranks are the same as the line source problem. The spatial and time discretizations are the same as previous simulations. The coefficients of determination ( $R^2$ ) for the linear are approximately one, which indicates a strong linear relationship between the theoretical and actual memory for both problems.

be improved if we refine the angular discretizations by applying the low-rank algorithm. Low-rank solutions with  $P_{125}$  with the same rank as  $P_7$  or  $P_{15}$  are far better than the full-rank solutions.

The memory and running time for this problem is given in Table 3. We observe that the low-rank solution is far more accurate than the full rank solution with reasonable extra cost. For example, the error of the full rank  $P_{15}$  solution can be reduced by a factor several thousand by employing the low rank method with  $r = 136$ , which comes with a cost of only twice the memory and three times the computation time. Comparing to the full rank  $P_{125}$  solution, this low rank solution only uses 3.4% of the memory and 6.7% of the computation time, but the overall error is small.

#### 4.6. Theoretical versus Actual Memory Used

Figure 16 plots the actual memory occupied during the calculation with the theoretical values. Linear regression demonstrates that the memory used in our

implementation is a linear function of the theoretical values. This verifies that the low-rank algorithm does reduce the memory requirement proportionally to the reduction in rank. Therefore, we are justified in using the theoretical memory usage estimates in our discussion and figures above.

## 5. Conclusion and Future Work

We have developed a practical algorithm to find the low-rank solution of the slab and planar geometry transport equation using explicit time integration. The method is based on projecting the equation to low-rank manifolds and numerically integrating with three steps. The numerical simulations show that on several test problems, a memory savings of about an order of magnitude can be achieved with the low-rank algorithm without sacrificing accuracy. This study establishes a projection-based framework for the direct model decomposition of the radiation transport problems.

In future work will attempt to solve the conservation issues present in the low-rank method using a high-order low-order approach (such as quasi-diffusion), where the low-rank method is used only to create a closure. In this approach, the low-rank method is estimating a ratio of moments that should be insensitive to conservation issues. Furthermore, there are several open research topics to explore, including asymptotic preservation, implicit discretization, the extension to energy-dependent problems, and other angular treatments, such as discrete ordinates. A recent preprint by Einkemmer, Hu, and Wang has presented a low-rank method based on an IMEX scheme that is asymptotic preserving [36], and we believe this is a first of several new developments in low-rank algorithms for transport problems.

## References

- [1] J. Shalf, S. Dosanjh, J. Morrison, Exascale computing technology challenges, in: International Conference on High Performance Computing for Computational Science, 2010, pp. 1–25.

- [2] G. C. Pomraning, The Equations of Radiation Hydrodynamics, Dover Publications, 2005.
- [3] E. Lewis, W. Miller, Computational Methods of Neutron Transport, John Wiley and Sons, 1984.
- [4] R. G. McClarren, T. A. Brunner, J. P. Holloway, On solutions to the Pn equations for thermal radiative transfer, Journal of Computational Physics 227 (5) (2008) 2864 – 2885.
- [5] V. M. Laboure, R. G. McClarren, C. D. Hauck, Implicit filtered PN for high-energy density thermal radiation transport using discontinuous Galerkin finite elements, Journal of Computational Physics 321 (2016) 624–643.
- [6] S. Dargaville, A. G. Buchan, R. P. Smedley-Stevenson, P. N. Smith, C. C. Pain, Angular adaptivity with spherical harmonics for Boltzmann transport, Journal of Computational Physics 397 (2019) 108846.
- [7] J. E. Morel, Diffusion-limit asymptotics of the transport equation, the  $P_{1/3}$  equations, and two flux-limited diffusion theories, Journal of Quantitative Spectroscopy and Radiative Transfer 65 (2000) 769–778.
- [8] G. Olson, L. Auer, M. Hall, Diffusion,  $P_1$ , and other approximate forms of radiation transport, Journal of Quantitative Spectroscopy and Radiative Transfer 64 (6) (2000) 619–634.
- [9] R. G. McClarren, Theoretical Aspects of the Simplified Pn Equations, Transport Theory and Statistical Physics 39 (2-4) (2011) 73–109.
- [10] M. Frank, A. Klar, E. W. Larsen, S. Yasuda, Time-dependent simplified PN approximation to the equations of radiative transfer, Journal of Computational Physics 226 (2) (2007) 2289–2305.
- [11] M. F. Modest, S. Lei, The Simplified Spherical Harmonics Method For Radiative Heat Transfer, Journal of Physics: Conference Series 369 (2012) 012019.

- [12] D. Anistratov, V. Y. Goldin, Solution of the multigroup transport equation by the quasidiffusion method, Preprint of the Keldysh Institute for Applied Mathematics 128 (1986) 19.
- [13] J. A. Fleck, J. D. Cummings, An implicit Monte Carlo scheme for calculating time and frequency dependent nonlinear radiation transport, *Journal of Computational Physics* 8 (3) (1971) 313–342.
- [14] R. G. McClarren, T. J. Urbatsch, A modified implicit Monte Carlo method for time-dependent radiative transfer with adaptive material coupling, *Journal of Computational Physics* 228 (16) (2009) 5669–5686.
- [15] A. B. Wollaber, Four Decades of Implicit Monte Carlo, *Journal of Computational and Theoretical Transport* 45 (1–2) (2016) 1–70.
- [16] R. P. Smedley-Stevenson, R. G. McClarren, Asymptotic diffusion limit of cell temperature discretisation schemes for thermal radiation transport, *Journal of Computational Physics* 286 (2015) 214–235.
- [17] O. Koch, C. Lubich, Dynamical low-rank approximation, *SIAM Journal on Matrix Analysis and Applications* 29 (2) (2007) 434–454.
- [18] O. Koch, C. Lubich, Dynamical Tensor Approximation, *SIAM Journal on Matrix Analysis and Applications* 31 (5) (2010) 2360–2375.
- [19] A. Nonnenmacher, C. Lubich, Dynamical low-rank approximation: applications and numerical experiments, *Mathematics and Computers in Simulation* 79 (4) (2008) 1346–1357.
- [20] B. Kloss, I. Burghardt, C. Lubich, Implementation of a novel projector-splitting integrator for the multi-configurational time-dependent Hartree approach, *Journal of Chemical Physics* 146 (17).
- [21] Katharina Kormann, A semi-Lagrangian Vlasov solver in tensor train format, *SIAM Journal on Scientific Computing* 37 (4) (2015) B613–B632. [arXiv:1408.7006](https://arxiv.org/abs/1408.7006), [doi:10.1137/140971270](https://doi.org/10.1137/140971270).



- [22] C. Lubich, I. V. Oseledets, A projector-splitting integrator for dynamical low-rank approximation, *BIT Numerical Mathematics* 54 (1) (2014) 171–188.
- [23] L. Einkemmer, C. Lubich, A low-rank projector-splitting integrator for the Vlasov–Poisson equation, *SIAM Journal on Scientific Computing* 40 (5) (2018) B1330–B1360.
- [24] L. Einkemmer, C. Lubich, A quasi-conservative dynamical low-rank algorithm for the vlasov equation, *SIAM Journal on Scientific Computing* 41 (5) (2019) B1061–B1081.   
[arXiv:https://doi.org/10.1137/18M1218686](https://doi.org/10.1137/18M1218686),  
[doi:10.1137/18M1218686](https://doi.org/10.1137/18M1218686).  
[URL https://doi.org/10.1137/18M1218686](https://doi.org/10.1137/18M1218686)
- [25] L. Einkemmer, A. Ostermann, C. Piazzola, A low-rank projector-splitting integrator for the Vlasov–Maxwell equations with divergence correction, *Journal of Computational Physics* 403 (2019) 109063.   
[arXiv:1902.00424](https://arxiv.org/abs/1902.00424),  
[doi:10.1016/j.jcp.2019.109063](https://doi.org/10.1016/j.jcp.2019.109063).  
[URL http://arxiv.org/abs/1902.00424](http://arxiv.org/abs/1902.00424)
- [26] L. Einkemmer, A low-rank algorithm for weakly compressible flow, *SIAM Journal on Scientific Computing* 41 (5) (2019) A2795–A2814.
- [27] Z. Peng, R. G. McClarren, M. Frank, A low-rank method for time-dependent transport calculations, in: *International Conference on Mathematics and Computational Methods applied to Nuclear Science and Engineering (M&C 2019)*, 2019.   
[arXiv:1906.09940](https://arxiv.org/abs/1906.09940).
- [28] Z. Ding, L. Einkemmer, Q. Li, Error analysis of an asymptotic preserving dynamical low-rank integrator for the multi-scale radiative transfer equation (2019).   
[arXiv:1907.04247](https://arxiv.org/abs/1907.04247).
- [29] A. Edelman, T. A. Arias, S. T. Smith, The Geometry of Algorithms with Orthogonality Constraints, *SIAM Journal on Matrix Analysis and Applications* 20 (2) (1998) 303–353.

- [30] P. N. Brown, D. E. Shumaker, C. S. Woodward, Fully implicit solution of large-scale non-equilibrium radiation diffusion with high order time integration, *Journal of Computational Physics* 204 (2) (2005) 760–783.
- [31] T. A. Brunner, J. P. Holloway, Two-dimensional time dependent Riemann solvers for neutron transport, *Journal of Computational Physics* 210 (1) (2005) 386–399.
- [32] R. G. McClarren, C. D. Hauck, Robust and accurate filtered spherical harmonics expansions for radiative transfer, *Journal of Computational Physics* 229 (16) (2010) 5597–5614.
- [33] D. Radice, E. Abdikamalov, L. Rezzolla, C. D. Ott, A new spherical harmonics scheme for multi-dimensional radiation transport I. Static matter configurations, *Journal of Computational Physics* 242 (SEPTEMBER 2012) (2013) 648–669.
- [34] G. D. Raithby, Evaluation of discretization errors in finite-volume radiant heat transfer predictions, *Numerical Heat Transfer, Part B: Fundamentals* 36 (3) (1999) 241–264.   
arXiv:<https://doi.org/10.1080/104077999275631>,  
doi:10.1080/104077999275631.  
URL <https://doi.org/10.1080/104077999275631>
- [35] B. Ganapol, Analytical Benchmarks for Nuclear Engineering Applications, Organisation for Economic Co-Operation and Development, 2008.
- [36] L. Einkemmer, J. Hu, Y. Wang, An asymptotic-preserving dynamical low-rank method for the multi-scale multi-dimensional linear transport equation, arXiv preprint arXiv:2005.06571.