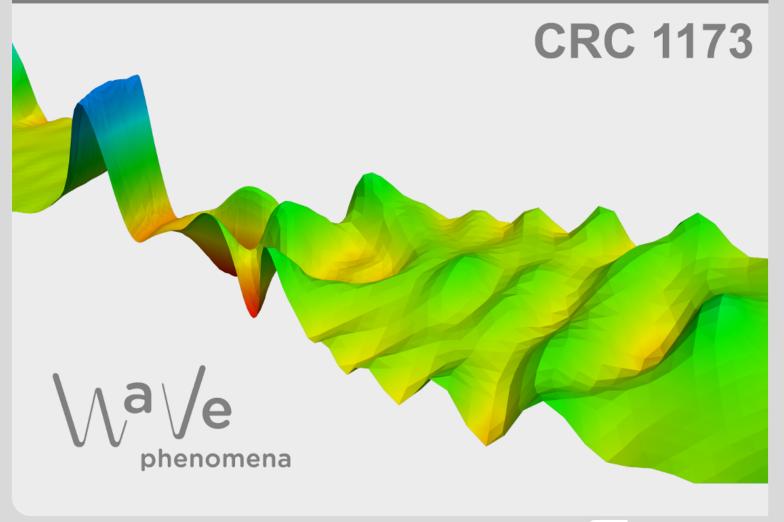# Dynamical low-rank approximation for Marshak waves

Gianluca Ceruti, Martin Frank, Jonas Kusch

KARLSRUHE INSTITUTE OF TECHNOLOGY

CRC 1173

Participating universities

Universität Stuttgart

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

TU
WIEN

Funded by

DFG

Technical report: Dynamical low-rank approximation for Marshak waves

Gianluca Ceruti[a], Martin Frank[b], Jonas Kusch[c]

[a]*École polytechnique fédérale de Lausanne, Lausanne, Switzerland, gianluca.ceruti@epfl.ch*
[b]*Karlsruhe Institute of Technology, Karlsruhe, Germany, martin.frank@kit.edu*
[c]*University of Innsbruck, Innsbruck, Austria, jonas.kusch1@gmail.com*

Abstract

Marshak waves are temperature waves which can arise from the background radiation in a material. A core limitation in the simulation of these temperature waves is the high-dimensional phase space of the radiation solution, which depends on time, the spatial position as well as the direction of flight. To obtain computationally efficient methods, we propose to use dynamical low-rank approximation (DLRA) which is a model order reduction method that dynamically determines and adapts dominant modes of the numerical solution. This is done by projecting the original dynamics onto the tangent space of the low-rank manifold. In this work, we investigate discontinuous Galerkin discretizations for two robust time integrators. By performing the derivation of the DLRA evolution equations on the continuous level, we are able to apply the needed slope limiter on the low-rank factors instead of the full solution. The efficiency of the method is presented through computational results for a Marshak wave originating from a heated wall.

Keywords: Dynamical low-rank approximation, kinetic equations, rank adaptivity, model order reduction

1. Introduction

Marshak waves are temperature waves which arise when radiation from a hot source enters a cold material. Though no material transport is present, a travelling heat wave arises in the material due to the interaction of radiation particles with the background medium: While radiation particles can heat up the medium through absorption, the heated material can in turn emit new particles. This interplay of particles and the medium results in a moving radiation and temperature front. Such phenomena arise in applications such as star formations, supernova explosions, the radiation emitted from a hohlraum striking a fusion target, or laser wakefield acceleration through pressure waves.

The dynamics of particles is described by a particle density (also called angular flux) $f(t, \mathbf{x}, \mathbf{\Omega})$ which depends on time t, position $\mathbf{x} \in \mathbb{R}^3$, and direction $\mathbf{\Omega} \in \mathbb{S}^2$. That is, the phase space that needs to be discretized is six-dimensional. Therefore, numerical simulations of Marshak waves are computationally costly and exhibit a high memory footprint. To enable efficient numerical simulations, coarse discretizations of the directional domain are chosen which results in unwanted numerical artifacts such as ray-effects [16, 22, 19] in nodal discretizations and Gibb's phenomenon in modal representations. To mitigate such artifacts, numerical methods are equipped with ray-effect mitigation techniques, see e.g. [17, 21, 1, 8] as well as filters [20, 9].

An alternative approach is to pick a finely resolved discretization in angle and space, and to resolve the issue of limited computational and memory resources by evolving the solution as a low-rank factorization. A method which is gaining interest in kinetic problems is dynamical low-rank approximation (DLRA) which has been proposed in [13]. The core idea of DLRA is to evolve the solution on the low-rank manifold $\mathcal{M}$ by projecting the right-hand side of the evolution equation onto the tangent plane of $\mathcal{M}$. The resulting evolution equations

can be solved robustly with the projector–splitting integrator [18] or the recently developed "unconventional" basis update & Galerking (BUG) integrator of [4]. The core advantage of the BUG integrator for applications in kinetic theory is the fact that unlike the projector–splitting integrator, it evolves the factorized solution solely forward in time, which facilitates the construction of stable spatial discretizations [14]. Moreover it allows for an extension to rank adaptivity [3]. The core idea of [3] is to augment the basis of the integrator before performing a Galerkin update. This adds conservation properties (up to a tolerance parameter) and can be used to preserve important moments in the Vlasov equations [7]. Further recent advances in dynamical low-rank approximation for radiative transfer include the construction of asymptotic–preserving schemes [5], sweeping methods [23] and collision source methods [15].

Despite recent advances in DLRA for radiative transfer, no results are available for Marshak wave problems. Moreover, to the best of the authors knowledge, the use of high-order spatial discretizations for radiation transport is limited to [10] and it is not clear how such spatial discretizations behave in combination with different integrators. In this report, we document a discontinuous Galerkin discretization for Marshak wave problems and present computational results for different integrators. Additionally, we present a strategy to efficiently incorporate non-linear slope limiters without requiring the computation of the full solution based on the continuous DLRA formulation [6].

This report is structured as follows: After the introduction in Section 1, we present a background on radiative transport and dynamical low-rank approximation in Section 2. Section 3 presents a derivation of the DLRA evolution equations for Marshak wave problems on a continuous level. In Section 4, we present a modal discretization in direction and space. Numerical results are presented in Section 5 and we conclude this report in Section 6.

2. Background

2.1. Recap: Radiative transfer
Marshak waves can be simulated by solving the thermal radiative transfer equation

$$\frac{1}{c}\partial_t f + \Omega \cdot \nabla_x f = \sigma_s(\phi - f) + \sigma_a(B - f) \tag{1a}$$

$$c_v \partial_t T = \int_{S^2} \sigma_a(f - B)\, d\Omega, \tag{1b}$$

where for the sake of compactness of notation, we have suppressed all arguments. We are interested in determining the angular flux $f(t, \mathbf{x}, \Omega)$ which depends on time t, position $\mathbf{x} \in \mathbb{R}^3$, and direction $\Omega \in \mathbb{S}^2$. The scalar flux is denoted by $\phi(t, \mathbf{x}) = \frac{1}{4\pi} \int_{4\pi} f(t, \mathbf{x}, \Omega') d\Omega'$, the speed of light is given by c, the material heat capacity is denoted as c_v, and h is Planck's constant. The material temperature is given by $T(t, \mathbf{x})$. Moreover, the Planck function for mono energetic particles reads $B(T) = acT^4$. Scattering and absorption cross sections are given by $\sigma_s(x)$ and $\sigma_a(x)$. The dynamics described by equation (1a) is the transport and isotropic scattering as well as absorption of photons. Due to absorption events, the temperature of the material can increase according to (1b). Moreover, the background medium emits blackbody radiation at its temperature.

2.2. Recap: Dynamical low-rank approximation
Let us provide a brief summary on dynamical low-rank approximation [13] in its continuous formulation [6]. To this end, we investigate a partial differential equation of the form

$$\partial_t f(t, x, \mu) = F(f(t, x, \mu)) \tag{2}$$

and define the continuous rank r ansatz for a general function $f(t, x, \mu)$ as

$$f_r(t, x, \mu) = \sum_{j,\ell=1}^{r} X_j(t, x) S_{j\ell}(t) W_\ell(t, \mu). \tag{3}$$

2

The domain of x, which will later be our spatial domain is denoted by $D_x \subset \mathbb{R}$ and the domain of μ, which will later serve as the directional domain is denoted as $D_\mu \subset \mathbb{R}$. For a given time $t \in [0, T]$, we wish to determine the orthonormal basis functions $X_j(t, \cdot) \in L^2(D_x)$, $W_\ell(t, \cdot) \in L^2(D_\mu)$ as well as the coefficients $S_{j\ell}(t) \in \mathbb{R}$. When deriving evolution equations for these factors, we need to ensure that the numerical solution f_r remains in the set of rank r functions $\mathcal{M}_r$. This is achieved by projecting the dynamics of the full problem onto the tangent space of $\mathcal{M}_r$ at every $f_r(t, x, \mu)$ denoted by $\mathcal{T}_{f_r(t,x,\mu)}\mathcal{M}_r$ according to

$$\partial_t f(t, x, \mu) \in \mathcal{T}_{f_r(t,x,\mu)}\mathcal{M}_r \qquad \text{such that} \qquad \|\partial f_r(t, x, \mu) - F(f_r(t, x, \mu))\|_{L^2(D_x \times D_\mu)} = \min. \qquad (4)$$

According to [13, Lemma 4.1], we can rewrite the above problem as

$$\partial_t f_r(t, x, \mu) = P(f_r(t, x, \mu))F(f_r(t, x, \mu)). \qquad (5)$$

To define the projector onto the tangent space P, we denote the integration over D_x and D_μ as $\langle \cdot \rangle_x$ and $\langle \cdot \rangle_\mu$ as well as projections onto the basis as

$$P_X g(t, x, \mu) = \sum_{j=1}^r X_j(t, x)\langle X_j(t, \cdot)g(t, \cdot, \mu)\rangle_x \quad \text{and} \quad P_W g(t, x, \mu) = \sum_{j=1}^r W_j(t, \mu)\langle W_j(t, \cdot)g(t, x, \cdot)\rangle_\mu.$$

Then P takes the form

$$P(g) = P_X(g) - P_X P_W(g) + P_W(g).$$

Using Einstein's sum convention (here and throughout the manuscript), the evolution equations to (5) can be determined as

$$\dot{S}_{ji} = \langle F(f_r(t, x, \mu))X_j W_i \rangle_{x\mu}, \qquad (6a)$$

$$\partial_t X_j = (1 - P_X)\langle F(f_r(t, x, \mu))W_i \rangle_\mu (\mathbf{S}^{-1})_{ji}, \qquad (6b)$$

$$\partial_t W_j = (1 - P_W)\langle F(f_r(t, x, \mu))X_i \rangle_x (\mathbf{S}^{-1})_{ij}. \qquad (6c)$$

Note that the inverses of the coefficient matrix $\mathbf{S} = (S_{ij})_{i,j=1}^r \in \mathbb{R}^{r \times r}$ can have small singular values. I.e., time integration schemes for (6) will require small time step sizes Δt. A robust integrator which does not suffer from small singular values is the projector–splitting integrator [18]. The core idea of this integrator is to split the projected equation (5) into three substep and note that in each substep, two of the three factors remain constant. For an update from time t_0 to $t_1 = t_0 + \Delta t$, let us assume to have the factored solution at time t_0 as X^0, S^0, V^0. Then, the integrator reads as follows

1. **K-step**: Update the spatial basis from t_0 to t_1, update the coefficients to $\bar{\mathbf{S}}^1$ via

$$\partial_t K_i(t, x) = \langle F(K_\ell W_\ell^0) W_i^0 \rangle_\mu$$
$$K_i(t_0, x) = X_j^0 S_{ji}^0.$$

Determine the time updated X_j^1 and $\bar{S}_{ji}^1$ with $K_i(t_1, x) = X_j^1 \bar{S}_{ji}^1$ (by e.g. Gram-Schmidt).

2. **S-step**: Update the coefficient matrix $\bar{\mathbf{S}}^1 \to \widetilde{\mathbf{S}}^0$ according to

$$\dot{S}_{ij}(t) = -\langle F(X_\ell^1 \bar{S}_{\ell k}^1 W_k^0))X_i^1 W_j^0 \rangle_{x\mu}$$
$$S_{ij}(t_0) = \bar{S}_{ij}^1.$$

The updated coefficient matrix is then obtained as $\widetilde{\mathbf{S}}^0 = \mathbf{S}(t_0 + \Delta t)$.

3. **L-step**: Update the directional basis from t_0 to t_1, update the coefficients to $\mathbf{S}^1$ via

$$\partial_t L_i(t, \mu) = \langle F(X_\ell^1 L_\ell)X_i^1 \rangle_x$$
$$L_i(t_0, \mu) = \widetilde{S}_{ij}^0 W_j^0.$$

Determine the time updated W_j^1 and S_{ij}^1 with $L_i(t_1, x) = W_j^1 \bar{S}_{ij}^1$ (by e.g. Gram-Schmidt).

3

A main disadvantage of the projector–splitting integrator for kinetic problems is that the negative sign in the S-step can destroy stability when performing a spatial discretization [14]. A recently developed alternative is the BUG integrator [4]. It takes the following form:

1. **K-step**: Update the spatial basis from t_0 to t_1 via

$$\partial_t K_i(t, x) = \langle F\left(K_\ell W_\ell^0\right) W_i^0 \rangle_\mu$$
$$K_i(t_0, x) = X_j^0 S_{ji}^0.$$

Determine X_j^1 with $K_i(t_1, x) = X_j^1 R_{ji}$ (by e.g. Gram-Schmidt), store $\mathbf{M} = \left(\langle X_i^1 X_j^0 \rangle_x\right)_{i,j=1}^r$.

2. **L-step**: Update the directional basis from t_0 to t_1 via

$$\partial_t L_i(t, \mu) = \langle F(X_\ell^1 L_\ell) X_i^1 \rangle_x$$
$$L_i(t_0, \mu) = S_{ij}^0 W_j^0.$$

Determine W_j^1 with $L_i(t_1, x) = W_j^1 \overline{R}_{ij}$ (by e.g. Gram-Schmidt), store $\mathbf{N} = \left(\langle W_i^1 W_j^0 \rangle_\mu\right)_{i,j=1}^r$.

3. **S-step**: Update the coefficient matrix from t_0 to t_1 via

$$\dot{S}_{ij}(t) = \langle F(X_\ell^1 S_{\ell k}^1 W_k^0)) X_i^1 W_j^0 \rangle_{x\mu}$$
$$S_{ij}(t_0) = M_{i\ell} S_{\ell k}^0 N_{jk}.$$

The updated coefficient matrix is then obtained as $\mathbf{S}^1 = \mathbf{S}(t_0 + \Delta t)$.

Here, we can perform the K and L steps in parallel. Note that the BUG integrator can be extended to rank-adaptivity [3]. Besides moving only forward in time, the BUG integrator shares the exactness and the robust error bound of the projector-splitting integrator, see [4, 12].

3. Dynamical low-rank approximation for Marshak waves

In the following, we derive the evolution equations needed by both, the projector–splitting as well as the BUG integrator. To not repeat too many formulas, we only write down the evolution equations for the BUG integrator, since the projector–splitting counterpart only differs with respect to the minus sign in the S step and time indices. For sake of readability, we perform the derivation for the frequency-averaged (gray) equations in slab geometry. We start our derivation from

$$\partial_t f(t, x, \mu) = -\mu \partial_x f(t, x, \mu) + \sigma_s \left(\frac{1}{2}\int_{-1}^1 f d\mu' - f\right) + \sigma_a(B(T) - f) \tag{7a}$$

$$c_v \partial_t T = \sigma_a \int_{-1}^1 (f - B(T)) d\mu, \tag{7b}$$

where μ is the projected directional variable. First, let us write f as a low-rank factorization

$$f(t, x, \mu) \approx f_r(t, x, \mu) = X_i(t, x) S_{ij}(t) W_j(t, \mu)$$

and start by deriving the evolution equation of the K-step. Note that in the K-step, the directional basis is kept fixed at W_j^0. Moreover, we note that with our definition of $K_j(t, x) = X_i(t, x) S_{ij}(t)$ we can write the ansatz inside the K-step as $f_r^K(t, x, \mu) := K_j(t, x) W_j^0(\mu)$. We then plug this ansatz into (7a) and project with respect to $W_j^0(\mu)$. The resulting equation reads

$$\frac{1}{c}\partial_t K_j = -\langle \mu W_j^0 W_l^0 \rangle_\mu \partial_x K_l + \sigma_s \left(\frac{1}{2}\langle W_j^0 \rangle_\mu \langle W_l^0 \rangle_\mu K_l - K_j\right) + \sigma_a(B(T)\langle W_j^0 \rangle_\mu - K_j). \tag{8a}$$

4

Similarly, for the L-step equation, we fix the spatial basis at X_j^0. Using the definition of $L_i(t,\mu) = S_{ij}(t)W_j(t,\mu)$, we choose the ansatz inside the L-step as $f_r^L(t,x,\mu) := X_i(t,x)L_i(t,\mu)$. Again, we plug this ansatz into (7a) and project with respect to $X_j^0(x)$. The resulting equation reads

$$\frac{1}{c}\partial_t L_i = -\mu\langle X_i^0 \partial_x X_k^0\rangle_x L_k + \sigma_s\left(\frac{1}{2}\langle L_i\rangle_\mu - L_i\right) + \sigma_a(\langle B(T)X_i^0\rangle_x - L_i), \tag{8b}$$

Lastly, the S-step chooses fixed basis functions in space and direction, i.e., we pick the ansatz $f_r^S(t,x,\mu) := X_i^1(x)S_{ij}(t)W_j^1(\mu)$. Plugging this ansatz into (7a) and projecting with respect to $W_j^1(\mu)$ yields

$$\frac{1}{c}\dot{S}_{ij} = -\langle X_i^1 \partial_x X_k^1\rangle_x\langle \mu W_j^1 W_l^1\rangle_\mu S_{kl} + \sigma_s\left(\frac{1}{2}\langle W_j^1\rangle_\mu\langle W_l^1\rangle_\mu S_{il} - S_{ij}\right) + \sigma_a(B(T)\langle X_i^1\rangle_x\langle W_j^1\rangle_\mu - S_{ij}) \tag{8c}$$

Since the temperature T depends on space only and therefore allows for an efficient time evolution, we do not expand it in terms of basis functions. Note however that an expansion in terms of the spatial basis is possible.

4. A modal discretization for dynamical low-rank approximation

In the following, we perform a modal discretization of the derived evolution equations (8) for K, L and S. To this end, we define the normalized Legendre polynomials as $p_i(\mu)$ for $i \leq N$ and represent the directional basis as

$$W_j(t,\mu) \simeq \widehat{W}_{ji}(t)p_i(\mu), \quad L_j(t,\mu) \simeq \widehat{L}_{ji}(t)p_i(\mu). \tag{9}$$

Moreover, we write the spatial basis in terms of scaled Legendre polynomials $\varphi_i^{(k)}$ where $i \leq P$ as

$$X_j(t,x) \simeq \widehat{X}_{ji}^{(k)}(t)\varphi_i^{(k)}(x)\chi_{I_k}(x), \quad K_j(t,x) \simeq u_{ji}^{(k)}(t)\varphi_i^{(k)}(x)\chi_{I_k}(x). \tag{10}$$

Here, we use an equidistant spatial grid with spatial cells $I_k = [x_k, x_{k+1}]$ with $N_x + 1$ cell boundaries $x_1 \leq \cdots \leq x_{N_x+1}$ and grid spacing Δx. Moreover, χ denotes the indicator function. The modal basis in space is then defined as $\varphi_i^{(k)}(x) = \Phi_i\left(\frac{x-x_k}{\Delta x}\right)$ with $\Phi_i(x) := \sqrt{2}p_i(2x-1)\chi_{[0,1]}(x)$. Then, to derive the discretized L-step equation, we choose the ansatz (9) for L and test against $p_k(\mu)$. This gives

$$\frac{1}{c}\dot{\widehat{L}}_{i\ell}(t) = -\langle \mu p_\ell p_m\rangle_\mu\langle X_i^1 \partial_x X_k^1\rangle_x\widehat{L}_{ik} + \sigma_s\left(\frac{1}{2}\langle p_k\rangle_\mu\widehat{L}_{ik} - \widehat{L}_{i\ell}\right) + \sigma_a(\langle B(T)X_i^1\rangle_x - \widehat{L}_{i\ell}). \tag{11}$$

The terms $\langle X_j^1 \partial_x X_\ell^1\rangle_x$ can be written as

$$\langle X_i^1 \partial_x X_k^1\rangle_x = \sum_k \widehat{X}_{ji}^{(k),1}\widehat{X}_{\ell i'}^{(k),1}\left\langle \varphi_i^{(k)}\frac{d}{dx}\varphi_{i'}^{(k)}\right\rangle_{I_k}. \tag{12}$$

In the same manner, we obtain evolution equation equations for S. For the K-step, we need to take special care of the stabilization. To shorten notation, let us denote the flux matrix of the K-step as $\mathbf{A} \in \mathbb{R}^{r\times r}$ with entries $a_{jl} = \langle \mu W_j^0 W_l^0\rangle_\mu$. Moreover, let us omit scattering and absorption terms which are trivial to discretize and do not require a stabilization. Then, with the test functions $\widetilde{\varphi}_i^{(k)}(x) := \frac{1}{\Delta x}\varphi_i^{(k)}(x)$, we directly obtain at cell I_k the system

$$\dot{\boldsymbol{u}}_i(t)\langle \varphi_i\widetilde{\varphi}_j\rangle_x + \langle \mathbf{A}\partial_x\mathbf{K}(t,\cdot)\widetilde{\varphi}_j\rangle_x = \mathbf{0},$$

where we drop the index k for simplicity of notation and collect $\mathbf{K}(t,x) = (K_i(t,x)) \in \mathbb{R}^r$. Using integration by parts and orthogonality of basis and test functions, we arrive at

$$\partial_t\boldsymbol{u}_j(t) + \mathbf{A}\mathbf{K}(t,x)\widetilde{\varphi}_j(x)\big|_{x_k}^{x_{k+1}} - \langle \mathbf{A}\mathbf{K}(t,\cdot)\widetilde{\varphi}_j'\rangle_x = \mathbf{0}.$$

The flux term is approximated with an upwind method

$$\mathbf{A}\mathbf{K}(t,x)\widetilde{\boldsymbol{\varphi}}(x)^T\big|_{x_k}^{x_{k+1}} \approx \frac{1}{2}\left[\mathbf{A}(\mathbf{u}_{k+1}^T\boldsymbol{\varphi}(x_{k+1}^+) + \mathbf{u}_k^T\boldsymbol{\varphi}(x_{k+1}^-)) - |\mathbf{A}|(\mathbf{u}_{k+1}^T\boldsymbol{\varphi}(x_{k+1}^+) - \mathbf{u}_k^T\boldsymbol{\varphi}(x_{k+1}^-))\right]\widetilde{\boldsymbol{\varphi}}(x_{k+1}^-)^T$$

$$- \frac{1}{2}\left[\mathbf{A}(\mathbf{u}_k^T\boldsymbol{\varphi}(x_k^+) + \mathbf{u}_{k-1}^T\boldsymbol{\varphi}(x_k^-)) - |\mathbf{A}|(\mathbf{u}_k^T\boldsymbol{\varphi}(x_k^+) - \mathbf{u}_{k-1}^T\boldsymbol{\varphi}(x_k^-))\right]\widetilde{\boldsymbol{\varphi}}(x_k^-)^T,$$

where we use the notation $\boldsymbol{\varphi}(x_k^-) := \boldsymbol{\varphi}_{k-1}(x_k)$ and $\boldsymbol{\varphi}(x_k^+) := \boldsymbol{\varphi}_k(x_k)$. This can be reformulated as an advection and a diffusion part

$$\mathbf{A}\mathbf{K}(t,x)\widetilde{\varphi}_j(x)\big|_{x_k}^{x_{k+1}} \approx \frac{1}{2}\mathbf{A}\left[(\mathbf{u}_{k+1}^T\boldsymbol{\varphi}(x_{k+1}^+) + \mathbf{u}_k^T\boldsymbol{\varphi}(x_{k+1}^-))\widetilde{\boldsymbol{\varphi}}(x_{k+1}^-)^T - (\mathbf{u}_k^T\boldsymbol{\varphi}(x_k^+) + \mathbf{u}_{k-1}^T\boldsymbol{\varphi}(x_k^-))\widetilde{\boldsymbol{\varphi}}(x_k^+)^T\right]$$

$$+ \frac{1}{2}|\mathbf{A}|\left[(\mathbf{u}_k^T\boldsymbol{\varphi}(x_k^+) - \mathbf{u}_{k-1}^T\boldsymbol{\varphi}(x_k^-))\widetilde{\boldsymbol{\varphi}}(x_k^-)^T - (\mathbf{u}_{k+1}^T\boldsymbol{\varphi}(x_{k+1}^+) - \mathbf{u}_k^T\boldsymbol{\varphi}(x_{k+1}^-))\widetilde{\boldsymbol{\varphi}}(x_{k+1}^-)^T\right],$$

where $|\mathbf{A}|$ is the absolute value of $\mathbf{A} = \mathbf{T}\mathbf{D}\mathbf{T}^\top$ defined as $|\mathbf{A}| = \mathbf{T}|\mathbf{D}|\mathbf{T}^\top$. Hence, the full semi-discrete update equation reads

$$\partial_t \boldsymbol{u}(t) = -\frac{1}{2}\mathbf{A}\left[(\mathbf{u}_{k+1}^T\boldsymbol{\varphi}(x_{k+1}^+) + \mathbf{u}_k^T\boldsymbol{\varphi}(x_{k+1}^-))\widetilde{\boldsymbol{\varphi}}(x_{k+1}^-)^T - (\mathbf{u}_k^T\boldsymbol{\varphi}(x_k^+) + \mathbf{u}_{k-1}^T\boldsymbol{\varphi}(x_k^-))\widetilde{\boldsymbol{\varphi}}(x_k^+)^T - \left\langle \mathbf{K}\frac{d}{dx}\widetilde{\boldsymbol{\varphi}}^T\right\rangle_x\right]$$

$$- \frac{1}{2}|\mathbf{A}|\left[(\mathbf{u}_k^T\boldsymbol{\varphi}(x_k^+) - \mathbf{u}_{k-1}^T\boldsymbol{\varphi}(x_k^-))\widetilde{\boldsymbol{\varphi}}(x_k^-)^T - (\mathbf{u}_{k+1}^T\boldsymbol{\varphi}(x_{k+1}^+) - \mathbf{u}_k^T\boldsymbol{\varphi}(x_{k+1}^-))\widetilde{\boldsymbol{\varphi}}(x_{k+1}^-)^T\right].$$

To ensure stability, we must apply a limiter on the higher-order corrections to the mean value in cell k, which is $\mathbf{u}_{k0}$. We define

$$\mathbf{u}_{k+1}^T\boldsymbol{\varphi}(x_{k+1}^-) := \mathbf{u}_{k0} + \bar{\boldsymbol{u}}_k^-, \qquad \mathbf{u}_k^T\boldsymbol{\varphi}(x_k^+) := \mathbf{u}_{k0} - \bar{\boldsymbol{u}}_k^+.$$

Using the limiter function $\widetilde{m} : \mathbb{R} \times \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ for the correction terms $\bar{\boldsymbol{u}}_k^{-,+}$, we get (element-wise)

$$\widetilde{\bar{\boldsymbol{u}}}_{k,\ell}^- := \widetilde{m}(\bar{\boldsymbol{u}}_k^-, \boldsymbol{u}_{k+1,0} - \boldsymbol{u}_{k,0}, \boldsymbol{u}_{k,0} - \boldsymbol{u}_{k-1,0}), \tag{13a}$$

$$\widetilde{\bar{\boldsymbol{u}}}_{k,\ell}^+ := \widetilde{m}(\bar{\boldsymbol{u}}_k^+, \boldsymbol{u}_{k+1,0} - \boldsymbol{u}_{k,0}, \boldsymbol{u}_{k,0} - \boldsymbol{u}_{k-1,0}). \tag{13b}$$

The limiter is constructed as

$$\widetilde{m}(a,b,c) := \begin{cases} a & \text{if } |a| \leq M\bar{H}_k^2 \\ m(a,b,c) & \text{else} \end{cases}.$$

where M and $\bar{H}_k$ are user-determined parameters. Recall that the minmod function m takes the form

$$m(a,b,c) := \begin{cases} s \cdot \min(a,b,c) & \text{if } s = \text{sign}(a) = \text{sign}(b) = \text{sign}(c) \\ 0 & \text{else} \end{cases}.$$

Then, the limited cell edge corrections are

$$\widetilde{\boldsymbol{u}}_{k+1}^T\boldsymbol{\varphi}(x_{k+1}^-) := \mathbf{u}_{k0} + \widetilde{\bar{\boldsymbol{u}}}_k^-, \qquad \widetilde{\boldsymbol{u}}_k^T\boldsymbol{\varphi}(x_k^+) := \mathbf{u}_{k0} - \widetilde{\bar{\boldsymbol{u}}}_k^+. \tag{14}$$

In order to directly define a limiter for the expansion coefficients $\widetilde{\boldsymbol{u}}_{k+1}$ instead of the cell edge corrections alone, we need to solve the system of equations. That is, we compute the correction via (13) in cell k and then solve (14) for $\widetilde{\boldsymbol{u}}_{k+1}$ and $\widetilde{\boldsymbol{u}}_{k+1}$. This system is uniquely solvable for orders two and three. At higher order, one needs to pick additional degrees of freedom. Let us denote the action of the limiter as

$\mathcal{L} : \mathbb{R}^{P \times N} \to \mathbb{R}^{P \times N}$. Then, when for example using a forward Euler method, we get

$$\boldsymbol{u}_k^{n+1} = \widetilde{\boldsymbol{u}}_k^n - \frac{\Delta t}{2} \mathbf{A} \left[(\widetilde{\boldsymbol{u}}_{k+1}^{n,T} \boldsymbol{\varphi}(x_{k+1}^+) + \widetilde{\boldsymbol{u}}_k^{n,T} \boldsymbol{\varphi}(x_{k+1}^-)) \widetilde{\boldsymbol{\varphi}}(x_{k+1}^-)^T - (\widetilde{\boldsymbol{u}}_k^{n,T} \boldsymbol{\varphi}(x_k^+) + \widetilde{\boldsymbol{u}}_{k-1}^{n,T} \boldsymbol{\varphi}(x_k^-)) \widetilde{\boldsymbol{\varphi}}(x_k^+)^T \right]$$

$$- \frac{\Delta t}{2} |\mathbf{A}| \left[(\widetilde{\boldsymbol{u}}_k^{n,T} \boldsymbol{\varphi}(x_k^+) - \widetilde{\boldsymbol{u}}_{k-1}^{n,T} \boldsymbol{\varphi}(x_k^-)) \widetilde{\boldsymbol{\varphi}}(x_k^-)^T - (\widetilde{\boldsymbol{u}}_{k+1}^{n,T} \boldsymbol{\varphi}(x_{k+1}^+) - \widetilde{\boldsymbol{u}}_k^{n,T} \boldsymbol{\varphi}(x_{k+1}^-)) \widetilde{\boldsymbol{\varphi}}(x_{k+1}^-)^T \right]$$

$$+ \Delta t \left\langle \widetilde{\boldsymbol{u}}^{n,T} \boldsymbol{\varphi} \frac{d}{dx} \widetilde{\boldsymbol{\varphi}}^T \right\rangle_x ,$$

$$\widetilde{\boldsymbol{u}}_k^{n+1} = \mathcal{L}(\boldsymbol{u}_k^{n+1}).$$

To bring the above scheme in a more compact form, let us write it as

$$\boldsymbol{u}^{n+1} = \widetilde{\boldsymbol{u}}^n + \Delta t \boldsymbol{D}_x \widetilde{\boldsymbol{u}}^n \mathbf{A}^T + \Delta t \boldsymbol{D}_{xx} \widetilde{\boldsymbol{u}}^n |\mathbf{A}|^T, \tag{15a}$$

$$\widetilde{\boldsymbol{u}}_k^{n+1} = \mathcal{L}(\boldsymbol{u}_k^{n+1}). \tag{15b}$$

Here, we define

$$(D_{xx})_{\text{idx}(k,i),\text{idx}(k+1,j)} = \frac{1}{2} \varphi_j(x_{k+1}^+) \widetilde{\varphi}_i(x_{k+1}^-),$$

$$(D_{xx})_{\text{idx}(k,i),\text{idx}(k-1,j)} = \frac{1}{2} \varphi_j(x_k^-) \widetilde{\varphi}_i(x_k^+),$$

$$(D_{xx})_{\text{idx}(k,i),\text{idx}(k,j)} = - \frac{1}{2} \varphi_j(x_k^+) \widetilde{\varphi}_i(x_k^+) - \frac{1}{2} \varphi_j(x_{k+1}^-) \widetilde{\varphi}_i(x_{k+1}^-).$$

and

$$(D_x)_{\text{idx}(k,i),\text{idx}(k+1,j)} = - \frac{1}{2} \varphi_j(x_{k+1}^+) \widetilde{\varphi}_i(x_{k+1}^-),$$

$$(D_x)_{\text{idx}(k,i),\text{idx}(k-1,j)} = \frac{1}{2} \varphi_j(x_k^-) \widetilde{\varphi}_i(x_k^+),$$

$$(D_x)_{\text{idx}(k,i),\text{idx}(k,j)} = \frac{1}{2} \varphi_j(x_k^+) \widetilde{\varphi}_i(x_k^+) - \frac{1}{2} \varphi_j(x_{k+1}^-) \widetilde{\varphi}_i(x_{k+1}^-) + \langle \varphi_j \widetilde{\varphi}_i' \rangle_x.$$

Remark 1. *It is important to point out that for our derivation it has been essential to formulate the DLRA evolution equations on the continuous level and then discretize in space and direction in a subsequent step. Otherwise, to apply the limiter, the full solution needs to be computed, thereby destroying the efficiency of the method.*

5. Numerical Results

The code used to generate the presented results in openly available [2]. In the following, we compute the propagation of a Marshak wave from a heated wall of a cold domain, following [11]. In this setting, even though the temperature equation does not include any convection terms, a temperature wave which travels into the cold domain will form. The advection is driven by the interation of radiation particles and the background medium. Here, the wave velocity is significantly smaller than the velocity of particles. Let us assume a wall temperature is $T_{\text{wall}} = 80$ eV, whereas the domain has a temperature of 0.02 eV. Additional parameters are

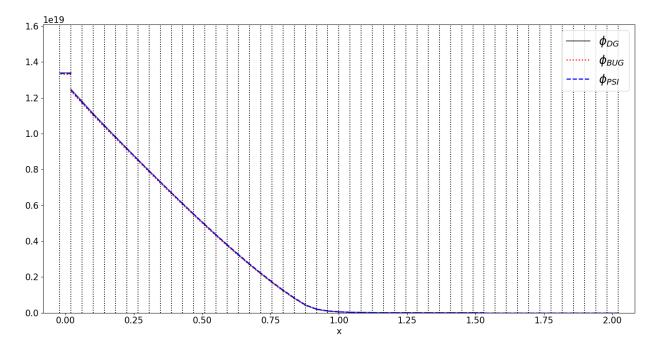|	
$D = [0, 0.002]$	spatial domain in cm
$N_x = 51$	number spatial cells
$P = 2$	expansion coefficients in space
$N = 100$	expansion coefficients in angle
$t_{\text{end}} = 5 \cdot 10^{-12}$	end time in seconds
$\lambda_{tr} = 92.6 \cdot 10^{-2}$	transport path length in μm
$\rho = 0.1$	material density in g/cm^3
$c_v = 0.831 \cdot 10^5$	specific heat in $J/(g \cdot K)$
$\sigma_s = 0$	scattering cross section in $1/\mu m$

Figure 1: Scalar flux profile for the full method (DG) as well as the PSI and BUG integrator. The spatial domain has been rescaled.

The heat capacity is given by $C_V = \rho \cdot c_V$ and the opacity is $\sigma_a = 1/\lambda_{tr}$.

In the following, we investigate the presented setting with a conventional full-rank discontinuous Galerkin (DG) method compared to the projector–splitting integrator and the BUG integrator. Both, the BUG and projector–splitting integrator (PSI) use a fixed rank of $r = 5$. The resulting scalar flux can be found in Figure 1 and the corresponding temperature profile can be found in Figure 2. It is observed that both integrators yield an accurate representation of the Marshak wave that shows satisfactory agreement with the full rank DG solution. The computational costs are reduced heavily from 85 seconds for the full rank method to 5 seconds for the dynamical low-rank methods

6. Conclusion

In this report, we have derived evolution equations for dynamical low-rank formulation in Marshak wave problems. We have presented an efficient fully modal discretization in space and angle. To be able to derive an efficient discontinuous Galerkin discretization of the DLRA evolution equations, we first perform the DLRA projection on the continuous level and then discretize in space and angle. We can show that the DLRA method is able to accurately compute a Marshak wave at reduced computational costs.

Acknowledgment

References

[1] T. Camminady, M. Frank, K. Küpper, and J. Kusch. Ray effect mitigation for the discrete ordinates method through quadrature rotation. *Journal of Computational Physics*, 382:105–123, 2019.
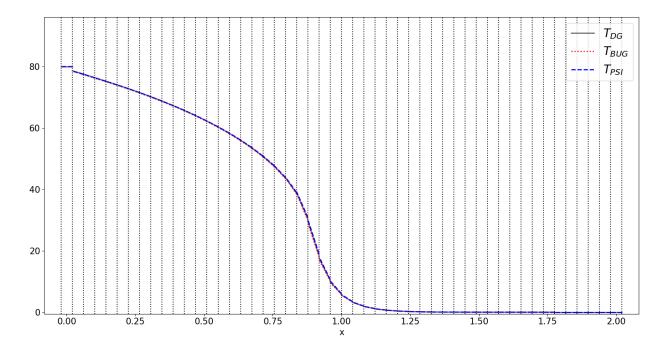
Figure 2: Temperature profile for the full method (DG) as well as the PSI and BUG integrator. The spatial domain has been rescaled.

[2] G. Ceruti, M. Frank, and J. Kusch. *Numerical testcases for "Technical report: Dynamical low-rank approximation for Marshak waves"*, 2022. `https://github.com/JonasKu/Testcases-for-Technical-Report---Dynamical-low-rank-approximation-for-Marshak-waves.git`.

[3] G. Ceruti, J. Kusch, and C. Lubich. A rank-adaptive robust integrator for dynamical low-rank approximation. *BIT Numerical Mathematics*, pages 1–26, 2022.

[4] G. Ceruti and C. Lubich. An unconventional robust integrator for dynamical low-rank approximation. *BIT Numerical Mathematics*, 62(1):23–44, 2022.

[5] L. Einkemmer, J. Hu, and Y. Wang. An asymptotic-preserving dynamical low-rank method for the multi-scale multi-dimensional linear transport equation. *Journal of Computational Physics*, 439:110353, 2021.

[6] L. Einkemmer and C. Lubich. A low-rank projector-splitting integrator for the vlasov–poisson equation. *SIAM Journal on Scientific Computing*, 40(5):B1330–B1360, 2018.

[7] L. Einkemmer, A. Ostermann, and C. Scalone. A robust and conservative dynamical low-rank algorithm. *arXiv preprint arXiv:2206.09374*, 2022.

[8] M. Frank, J. Kusch, T. Camminady, and C. D. Hauck. Ray effect mitigation for the discrete ordinates method using artificial scattering. *Nuclear Science and Engineering*, 194(11):971–988, 2020.

[9] C. Hauck and V. Heningburg. Filtered discrete ordinates equations for radiative transport. *Journal of Scientific Computing*, 80(1):614–648, 2019.

[10] C. Hauck and S. Schnake. A predictor-corrector strategy for adaptivity in dynamical low-rank approximations. *arXiv preprint arXiv:2209.00550*, 2022.

[11] W. Höbel, B. Goel, A. Ni, and H. Marten. Energy transport in matter with rapidly changing states. *Nuclear science and engineering*, 137(3):334–351, 2001.

[12] E. Kieri, C. Lubich, and H. Walach. Discretized dynamical low-rank approximation in the presence of small singular values. *SIAM Journal on Numerical Analysis*, 54(2):1020–1038, 2016.

[13] O. Koch and C. Lubich. Dynamical low-rank approximation. *SIAM Journal on Matrix Analysis and Applications*, 29(2):434–454, 2007.

[14] J. Kusch, L. Einkemmer, and G. Ceruti. On the stability of robust dynamical low-rank approximations for hyperbolic problems. *arXiv preprint arXiv:2107.07282*, 2021.

[15] J. Kusch and P. Stammer. A robust collision source method for rank adaptive dynamical low-rank approximation in radiation therapy. *arXiv preprint arXiv:2111.07160*, 2021.

[16] K. D. Lathrop. Ray effects in discrete ordinates equations. *Nuclear Science and Engineering*, 32(3):357–369, 1968.

[17] K. D. Lathrop. Remedies for ray effects. *Nuclear Science and Engineering*, 45(3):255–268, 1971.

[18] C. Lubich and I. V. Oseledets. A projector-splitting integrator for dynamical low-rank approximation. *BIT Numerical Mathematics*, 54(1):171–188, 2014.

[19] K. A. Mathews. On the propagation of rays in discrete ordinates. *Nuclear science and engineering*, 132(2):155–180, 1999.

[20] R. G. McClarren and C. D. Hauck. Robust and accurate filtered spherical harmonics expansions for radiative transfer. *Journal of Computational Physics*, 229(16):5597–5614, 2010.

[21] W. Miller Jr and W. H. Reed. Ray-effect mitigation methods for two-dimensional neutron transport theory. *Nuclear Science and Engineering*, 62(3):391–411, 1977.

[22] J. E. Morel, T. A. Wareing, R. B. Lowrie, and D. K. Parsons. Analysis of ray-effect mitigation techniques. *Nuclear science and engineering*, 144(1):1–22, 2003.

[23] Z. Peng and R. G. McClarren. A sweep-based low-rank method for the discrete ordinate transport equation. *Journal of Computational Physics*, page 111748, 2022.