Commun. Comput. Phys.
doi: 10.4208/cicp.OA-2016-0105

Vol. **22**, No. 1, pp. 157-181
July 2017

Implicit Asymptotic Preserving Method for Linear Transport Equations

Qin Li[1,3,*] and Li Wang[2]

[1] *Mathematics Department, University of Wisconsin-Madison, 480 Lincoln Dr., Madison, WI 53705, USA.*

[2] *Departments of Mathematics and Computational Data-Enabled Science and Engineering Program, State University of New York at Buffalo, 244 Mathematics Building, Buffalo, NY 14260, USA.*

[3] *The Optimization Group, The Wisconsin Institute of Discovery, Madison, WI 53715, USA.*

Received 12 July 2016; Accepted (in revised version) 5 October 2016

Abstract. The computation of the radiative transfer equation is expensive mainly due to two stiff terms: the transport term and the collision operator. The stiffness in the former comes from the fact that particles (such as photons) travel at the speed of light, while that in the latter is due to the strong scattering in the optically thick region. We study the fully implicit scheme for this equation to account for the stiffness. The main challenge in the implicit treatment is the coupling between the spacial and angular coordinates that requires the large size of the to-be-inverted matrix, which is also ill-conditioned and not necessarily symmetric. Our main idea is to utilize the spectral structure of the ill-conditioned matrix to construct a pre-conditioner, which, along with an exquisite split of the spatial and angular dependence, significantly improve the condition number and allows a matrix-free treatment. We also design a fast solver to compute this pre-conditioner explicitly in advance. Our method is shown to be efficient in both diffusive and free streaming limit, and the computational cost is comparable to the state-of-the-art method. Various examples including anisotropic scattering and two-dimensional problems are provided to validate the effectiveness of our method.

AMS subject classifications: 65M06, 65M30, 65F08, 65F35

PACS: 02.60.Dc, 02.60.Cb, 02.70.Bf

Key words: Implicit method, asymptotic preserving, pre-conditioner, diffusive regime, free streaming limit.

*Corresponding author. *Email addresses:* qinli@math.wisc.edu (Q. Li), lwang46@buffalo.edu (L. Wang)

 ©2017 Global-Science Press

https://doi.org/10.4208/cicp.OA-2016-0105 Published online by Cambridge University Press

1 Introduction

The linear transport equation describes the physical process of interaction of radiation with background material such as radiative transfer, neutron transport, and etc. It often contains a diffusive scaling that accounts for the strong scattering effect and leads to diffusion equations. On the contrary, when the scattering is weak, the propagation of radiation is almost a free transport with the speed of light, which is named as free streaming limit. In practice, the material usually contains both strong and weak scattering regimes, and thereby it is desirable to design a numerical method that is uniformly accurate in both cases without resolving the small scales.

Asymptotic preserving (AP) schemes arise to serve this purpose. As the name states, it preserves the asymptotic limit at the discrete level. More specifically, such method, when applied to certain equations with small parameters, should automatically become a stable solver for the corresponding limit equations without resolving the mesh size and time step. In the context of steady neutron transport, AP scheme was first studied by Larsen and Morel [21] and then Jin and Levermore [14], and Klar [17]. A rigorous convergence analysis was subsequently carried out by Golse, Jin and Levermore [11]. For time-dependent transport problem, a decomposition of the distribution function is often performed, either via a macro-micro decomposition [20] or an even-odd decomposition [15, 16]. Upon such decomposition, the stiff and non-stiff terms get separate off and an implicit-explicit scheme is applied to treat two terms respectively. The idea was later extended to a higher order implementation by Boscarino et al. [5]. Another related approach is termed the unified gas kinetic scheme framework, which was first proposed by Mieussens for linear transport equation [22] and recently modified by Sun et al. to deal with nonlinear problem [27].

Nevertheless, most AP schemes mentioned above, still suffer from a restrictive parabolic CFL condition that comes from the diffusion limit. Additionally, they are designed for diffusive scaling and thus may fail to capture the free streaming limit at which the radiation travels at the speed of light. Therefore, a fully implicit method is desired to remedy both problems. However, the main challenge that prevents researchers from directly applying the fully implicit method is the inversion of a large size of matrix due to the high dimension and the coupling between the spatial and angular coordinates. Even worse, this matrix is often ill-conditioned in some regimes and makes the inversion impractical.

Constant efforts have been made in the past few decades towards developing efficient implicit solver, both deterministic and stochastic. Here we only mention a few revolutionary works but an exhaustive bibliography is out of reach. For deterministic method, the current state-of-the-art is the Krylov iterative method for the discrete-ordinate system preconditioned by diffusion synthetic acceleration (DSA) [19, 30]. Besides the ray-effects [23] that generates by the discrete ordinate method, this preconditioned Krylov iteration has a drawback, mainly due to the complication of the method. Since each time iteration includes a sub-iterations of sweeps and diffusion solvers for DSA pre-

https://doi.org/10.4208/cicp.OA-2016-0105 Published online by Cambridge University Press

conditioner, it is very complicated to extend to nonlinear case. For stochastic method, the most widely used one is the implicit Monte Carlo (IMC) method [7]. But it is expensive because accurate solutions require an adequate sampling of points in phase space, and it also suffers from unavoidable fluctuations. Parallel to these two approaches is the development of moment method, which approximates the solution in terms of spherical harmonics. The foundation of this approximation was laid in [18, 26], and later an explosion of new developments emerges in two sub areas: one emphasis on the derivation of an approximation of the moment system such as [4, 9, 25, 28] and the other focuses primarily on designing applicable numerical methods such as [8, 12], to name just a few.

In this paper, we develop a highly efficient implicit solver for the linear transport equation that is uniformly accurate across different regimes: diffusive and free streaming. The main idea is to build a pre-conditioner based on the spectral structure of the collision, which is the main source of ill-conditioning. Then we reformulate the original system using an even-odd decomposition such that the resulting matrix to be inverted, upon discretization, is symmetric and positive definite, and thereby the conjugate gradient method can be used. This method is amenable to parallelization, and can be easily extended to anisotropic scattering.

In the following, we recall two state-of-the-art schemes that motivates our idea and will be compared to some extent with our newly designed scheme. To begin with, we summarize the typical equations we consider. In slab geometry, the radiation intensity $f(t,x,\mu)$ solves the following dimensionless equation:

$$\partial_t f+\frac{\mu}{\varepsilon}\partial_x f=\frac{\sigma}{\varepsilon^2}(\rho-f), \tag{1.1}$$

where $\sigma(x)$ is the scattering cross-section and ε is the Knudsen number. μ is the cosine of the angle between a particle's direction of flight and the x-axis. Here the external source and absorption effect are neglected for simplicity. $\rho=\langle f\rangle=\frac{1}{2}\int_{-1}^{1} f d\mu$ denotes the density. In the zero limit of ε, one has:

$$\partial_t\rho-\partial_x\left(\frac{1}{3\sigma}\partial_x\rho\right)=0, \tag{1.2}$$

which is termed the diffusion limit. In planar geometry, the equation rewrites as

$$\partial_t f+\frac{\Omega}{\varepsilon}\cdot\nabla_x f=\frac{\sigma}{\varepsilon^2}(\rho-f), \tag{1.3}$$

where Ω is the angular variable representing the two direction-of-flight; $\Omega=(\xi,\eta)$, $-1\leq\xi,\eta\leq 1$ and $\eta^2+\xi^2=1$. Here

$$\rho=\frac{1}{2\pi}\int_{|\Omega|=1} f d\Omega$$

and the corresponding diffusion limit is

$$\rho_t-\nabla\cdot\left(\frac{1}{2\sigma}\nabla\rho\right)=0. \tag{1.4}$$

https://doi.org/10.4208/cicp.OA-2016-0105 Published online by Cambridge University Press

1.1 Preconditioned Krylov method

The preconditioned Krylov method is similar to S_N source iteration (SI) with diffusion synthetic acceleration (DSA), but outperformed SI in computing diffusive problems with discontinuous material properties. We first recall SI with DSA to put it in the context [2]. Consider a semi-discrete version of (1.1)

$$f^{n+1}+\frac{\mu\Delta t}{\varepsilon}\partial_x f^{n+1}-\frac{\sigma\Delta t}{\varepsilon^2}(\rho^{n+1}-f^{n+1})=f^n, \tag{1.5}$$

where both the convection and scattering terms are treated implicitly, which results in an inversion of a large linear system. Due to the sparsity of the system, SI solves it iteratively that resembles the Richardson iteration. Denote

$$\mathcal{L}=1+\frac{\mu\Delta t}{\varepsilon}\partial_x+\frac{\sigma\Delta t}{\varepsilon^2}, \qquad \mathcal{P}=\langle\cdot\rangle, \tag{1.6}$$

as two operators. Given f^n and ρ^n the solution at time t^n, let $f^{(l)}$ be the approximation of f^{n+1} after l iterations, then it solves

$$\mathcal{L}f^{(l+1)}=\frac{\sigma\Delta t}{\varepsilon^2}\mathcal{P}f^{(l)}+f^n, \tag{1.7}$$

which is the main iteration in SI. When ε is small, the convergence of $f^{(l)}$ is rather slow; this is because the decreasing rate of the error $\frac{\delta f^{(l+1)}}{\delta f^{(l)}}$ (where $\delta f^{(l)}=f-f^{(l)}$ and f denotes the exact solution to (1.5)) is close to one. Then DSA helps to accelerate the convergence. Specifically, consider the equation for $\delta f^{(l)}$:

$$\mathcal{L}(\delta f^{(l+1)})=\frac{\sigma\Delta t}{\varepsilon^2}\mathcal{P}(\delta f^{(l+1)})+\frac{\sigma\Delta t}{\varepsilon^2}\left(\rho^{(l+1)}-\rho^{(l)}\right),$$

here $\rho^{(l)}=\mathcal{P}(f^{(l)})$ and $\delta\rho^{(l)}=\mathcal{P}(\delta f^{(l)})$. This equation is as hard to solve as (1.5) and the key idea is to use a diffusive approximation

$$\delta\rho^{(l+1)}-\frac{\Delta t}{3}\partial_x\left(\frac{1}{\sigma}\partial_x\delta\rho^{(l+1)}\right)=\frac{\sigma\Delta t}{\varepsilon^2}\left(\rho^{(l+1)}-\rho^{(l)}\right),$$

which is much easier to compute. Then SI with DSA can be summarized as follows.

$$\begin{cases} f^{(l+\frac{1}{2})}=\dfrac{\sigma\Delta t}{\varepsilon^2}\mathcal{L}^{-1}\rho^{(l)}+\mathcal{L}^{-1}f^n, \\ \delta\rho^{(l+\frac{1}{2})}=\dfrac{\sigma\Delta t}{\varepsilon^2}(1+\Delta t\mathcal{D})^{-1}\left(\rho^{(l+\frac{1}{2})}-\rho^{(l)}\right), \\ \rho^{(l+1)}=\rho^{(l+\frac{1}{2})}+\delta\rho^{(l+\frac{1}{2})}, \end{cases} \tag{1.8}$$

where $\delta\rho^{(l+\frac{1}{2})}=\rho-\rho^{(l+\frac{1}{2})}$ and ρ is again the exact solution to (1.5) and $\mathcal{D}=-\partial_x(\frac{1}{3\sigma}\partial_x)$ is the diffusion operator.

https://doi.org/10.4208/cicp.OA-2016-0105 Published online by Cambridge University Press

As DSA requires the diffusion solver to be consistent with the transport solver, which can be very expensive and deficient in heterogeneous multi-dimensional media [3], a Krylov method is developed with DSA as a preconditioner. The idea is to use more advanced iterative technique than Richardson iteration to solve

$$\left(\mathcal{L}-\frac{\sigma\Delta t}{\varepsilon^2}\mathcal{P}\right)f^{n+1}=f^n. \tag{1.9}$$

As mentioned above, the condition number of the corresponding discretization matrix of $\mathcal{L}-\frac{\sigma\Delta t}{\varepsilon^2}\mathcal{P}$ can be very big when ε is small, and thus a preconditioner is needed. A good choice [30] would be to first multiply (1.9) by the operator $\mathcal{P}\mathcal{L}^{-1}$, which leads to

$$\left(\mathcal{I}-\frac{\sigma\Delta t}{\varepsilon^2}\mathcal{P}\mathcal{L}^{-1}\right)\rho^{n+1}=\mathcal{P}\mathcal{L}^{-1}f^n. \tag{1.10}$$

Then an operator including a diffusion is expected to further precondition the above system. Note from a manipulation of the equations in (1.8) and a comparison with (1.7), a reasonable pre-conditioner is $\mathcal{I}+\frac{\sigma\Delta t}{\varepsilon^2}\mathcal{D}^{-1}$. Multiply it with (1.10), one has

$$\left(\mathcal{I}+\frac{\sigma\Delta t}{\varepsilon^2}\mathcal{D}^{-1}\right)\left(1-\frac{\sigma\Delta t}{\varepsilon^2}\mathcal{P}\mathcal{L}^{-1}\right)\rho^{n+1}=\left(\mathcal{I}+\frac{\sigma\Delta t}{\varepsilon^2}\mathcal{D}^{-1}\right)\mathcal{P}\mathcal{L}^{-1}f^n, \tag{1.11}$$

whose discrete version can now be solved by Krylov method such as GMRES. Preconditioned Krylov method has been applied to linear transport equation in various contexts, please refer to [1,6,10] for an extensive study.

1.2 Asymptotic preserving scheme via even-odd decomposition

Another class of methods aim at capturing the diffusion limit as ε vanishes without resolving the mesh size. This is the spirit of so-called asymptotic preserving method. Various asymptotic preserving methods have been proposed for transport equation with diffusive scaling, and here we only recall the method proposed by Jin, Pareschi and Toscani [15] based on an even-odd decomposition as it motives our new method. Define an even and an odd part of f by:

$$f_{\mathrm{E}}=\frac{1}{2}(f(t,x,\mu)+f(t,x,-\mu)),\quad f_{\mathrm{O}}=\frac{1}{2\varepsilon}(f(t,x,\mu)-f(t,x,-\mu)), \tag{1.12}$$

then (1.1) splits into

$$\begin{cases}\partial_t f_{\mathrm{E}}+\mu\partial_x f_{\mathrm{O}}=\dfrac{\sigma}{\varepsilon^2}(\rho-f_{\mathrm{E}}),\\ \partial_t f_{\mathrm{O}}+\alpha\mu\partial_x f_{\mathrm{E}}=-\dfrac{\sigma}{\varepsilon^2}f_{\mathrm{O}}+\dfrac{1}{\varepsilon^2}(1-\alpha\varepsilon^2)\mu\partial_x f_{\mathrm{E}},\end{cases} \tag{1.13}$$

https://doi.org/10.4208/cicp.OA-2016-0105 Published online by Cambridge University Press

where $\alpha=\min\{1,\frac{1}{\varepsilon^2}\}$. In this setting, two scales–$\frac{1}{\varepsilon}$ and $\frac{1}{\varepsilon^2}$–in the original system has been unified to one scale in (1.13) such that the non-stiff terms can now be treated explicitly while the stiff terms are treated implicitly.

The rest of paper is organized as follows. In the next section, we explain at large the challenges in the implicit method and elucidate our idea—constructing a preconditioning matrix—by analyzing the spectral structure of the matrix that will be inverted. Detailed pseudo code is also provided. Section 3 is devoted to checking some properties of our scheme, especially the computational cost, asymptotic preservation and unconditionally stability. In Section 4 we present several numerical examples to illustrate the efficiency, accuracy, and AP properties of the schemes. Finally, the paper is concluded in Section 5.

2 Numerical scheme

Since our purpose is to capture both the diffusion and free streaming limit of (1.1) (or (1.3)) without the parabolic CFL constraint, we will take the fully implicit time discretization. However, as mentioned above, this amounts to invert a large scale matrix whose condition number is not necessarily acceptable. Below, we will first explain our main idea in

- utilizing the spectral structure of the collision operator to construct the preconditioner for the ill-conditioned matrix;
- reducing the size of the matrix by separating the spatial dimension from the angular dimension and building the sparsity structure into a matrix-free form.

Then a detailed algorithm is given in the next subsection. The idea is presented for the isotropic collision case and the anisotropic treatment will be demonstrated in Section 2.3.

2.1 Challenges

Recall the fully implicit semi-discretization of the equation in one dimension:

$$\frac{f^{n+1}-f^n}{\Delta t}+\frac{\mu}{\varepsilon}\partial_x f^{n+1}=\frac{\sigma}{\varepsilon^2}(\rho^{n+1}-f^{n+1}), \tag{2.1}$$

where $f^n=f(t^n,x,\mu)$. Here because of the convection, the stiff parts are asymmetric. To make it symmetric, we define an even-odd parity of the solution by:

$$f_{\rm E}=\frac{1}{2}(f(t,x,\mu)+f(t,x,-\mu)),\quad f_{\rm O}=\frac{1}{2}(f(t,x,\mu)-f(t,x,-\mu)), \tag{2.2}$$

then the Eq. (2.1) splits:

$$\begin{cases}\dfrac{f_{\rm E}^{n+1}-f_{\rm E}^n}{\Delta t}+\dfrac{1}{\varepsilon}\mu\partial_x f_{\rm O}^{n+1}=\dfrac{\sigma}{\varepsilon^2}(\rho^{n+1}-f_{\rm E}^{n+1}),\\ \dfrac{f_{\rm O}^{n+1}-f_{\rm O}^n}{\Delta t}+\dfrac{1}{\varepsilon}\mu\partial_x f_{\rm E}^{n+1}=-\dfrac{\sigma}{\varepsilon^2}f_{\rm O}^{n+1}.\end{cases} \tag{2.3}$$

https://doi.org/10.4208/cicp.OA-2016-0105 Published online by Cambridge University Press

Notice that the even-odd decomposition we used here is different from (1.12) as our purpose is to symmetrize the convection operator, not to separate the macro and micro parts of the solution. With easy calculation, one could show that the reformation (2.3) is equivalent to:

$$\begin{cases} f_{\mathrm{O}}^{n+1}=\dfrac{\varepsilon^2}{\varepsilon^2+\sigma\Delta t}\left(f_{\mathrm{O}}^{n}-\dfrac{\Delta t}{\varepsilon}\mu\partial_x f_{\mathrm{E}}^{n+1}\right), \\ -\partial_x\left(\dfrac{\varepsilon^2\mu^2\Delta t}{\varepsilon^2+\sigma\Delta t}\partial_x f_{\mathrm{E}}^{n+1}\right)+\left(\dfrac{\varepsilon^2}{\Delta t}+\sigma\right)f_{\mathrm{E}}^{n+1}-\sigma\rho^{n+1}=\dfrac{\varepsilon^2}{\Delta t}\left[f_{\mathrm{E}}^{n}-\partial_x\left(\dfrac{\varepsilon\mu\Delta t}{\varepsilon^2+\sigma\Delta t}f_{\mathrm{O}}^{n}\right)\right]. \end{cases} \tag{2.4}$$

Here we note that the f_{E} gets updated first in the second equation, and then we update f^{O} from the first equation.

To compute the second equation in (2.4), however, is rather challenging. In a more compact form, it writes as:

$$(\mathcal{A}+\mathcal{B})f=b, \tag{2.5}$$

with the operators are defined:

$$\mathcal{A}=-\frac{\varepsilon^2\Delta t}{\varepsilon^2+\Delta t}\mu^2\partial_x^2,\quad \mathcal{B}=1+\frac{\varepsilon^2}{\Delta t}-\mathcal{P}\quad\text{and}\quad b=\frac{\varepsilon^2}{\Delta t}\left(f_{\mathrm{E}}^{n}-\frac{\varepsilon\Delta t}{\varepsilon^2+\Delta t}\mu\partial_x f_{O}^{n}\right). \tag{2.6}$$

We also used $\sigma\equiv 1$ and $\mathcal{P}f=\langle f\rangle$ for the conciseness of the notation. Under the finite difference framework, upon discretization, the two operators $\mathcal{A}$ and $\mathcal{B}$ will be represented using two matrices, denoted as A and B, and it is the following two properties of the matrices that make the problem difficult:

- The size of the system is huge. In the simplest 1D case, suppose we use an N_v-point quadrature for velocity and N_x grid points in space per direction, then f is a vector of length N_xN_v, and the two matrices A and B are both of size $(N_x\times N_v)^2$. Inverting them takes tremendous amount of computation. The situation is much worse for the 3D case, in which the matrices are of size $(N_x^3\times N_v^2)^2$ at the least. In a stereotype computation, $N_x\sim 10^9$. Not to mention the computation of the inverse, storing these matrices itself requires the memory that exceeds what is affordable.

- The matrices are ill-conditioned. In kinetic regime when $\varepsilon=\mathcal{O}(1)$ is relatively big, the matrix A+B is well-conditioned, but the condition number grows drastically to infinity in the diffusive regime as $\varepsilon\to 0$. Roughly speaking, the eigenvalues of A are very small–the $\varepsilon^2\Delta t$ term attenuate the effects of μ^2k^2 ($k=1,\cdots,N_x$), and thus the spectrum of the system is dominated by the eigenvalues of B. Realizing that $1-\mathcal{P}$ has a nontrivial null space with a big spectrum gap–the smallest eigenvalue of B is of ε^2 order while all others are of $\mathcal{O}(1)$, A+B produces a condition number of size $\mathcal{O}(1/\varepsilon^2)$ that grows to infinity with shrinking ε.

All efficient numerical algorithms, including the ones reviewed in the introduction, need to deal with the two difficulties listed above in one way or another. In the following subsection we explore the structure hidden inside the two matrices, and fully make use of these properties to tackle the difficulties listed above.

https://doi.org/10.4208/cicp.OA-2016-0105 Published online by Cambridge University Press

2.2 Ideas

In this section we explore the matrix structure, rely on which one could hope for better algorithms. Despite the size of the system and the ill-conditioning of the matrices, there are certain properties of A and B that might simplify the problem.

- The matrices are symmetric positive definite and block-wisely sparse.
  - Both A and B are symmetric.
  - Both A and B are positive definite. The eigenvalues of B is either $1+\frac{\varepsilon^2}{\Delta t}$ or $\frac{\varepsilon^2}{\Delta t}$ and since $\mathcal{A}$ is an elliptic operator, the positivity of A is out of question.
  - Both A and B are block-wisely sparse. In particular, A is sparse in both coordinates while B is dense in μ but sparse in x coordinate. We can actually write down its explicit expression:

$$\mathsf{A}=\mathsf{A}_x\otimes\mathrm{diag}\{\mu_k^2\},\quad \mathsf{B}=\mathsf{I}_{N_x}\otimes\mathsf{B}_\mu. \tag{2.7}$$

    Here $\otimes$ denotes the Kronecker product, and we have used the notation A_x for the discretization of $\frac{\varepsilon^2\Delta t}{\varepsilon^2+\Delta t}\partial_x^2$, $\mathsf{B}_\mu=(\mathsf{I}_{N_v}+\frac{\varepsilon^2}{\Delta t}-\mathsf{P})$ and the subindex for I indicate the size of this identity matrix. P corresponds to the averaging operator $\mathcal{P}$ and is a dense but low rank matrix.

- The matrices have a very clear pre-determined spectrum. Though A+B is ill-conditioned for small ε, the dominating part B has a very clear eigenspace structure. It reduces the constant vector by $\frac{\varepsilon^2}{\Delta t}<1$ in magnitude and magnifies all the vectors in the perpendicular space by $1+\frac{\varepsilon^2}{\Delta t}>1$. Specifically, we can write down the following eigendecomposition for B_μ:

$$\mathsf{B}_\mu=\frac{\varepsilon^2}{\Delta t}ee^t+\left(1+\frac{\varepsilon^2}{\Delta t}\right)\sum_{i\geq 2}v_iv_i^t. \tag{2.8}$$

  Here we have used the notation $e=\frac{1}{\sqrt{N_v}}[1,\cdots,1]^t$ and $\mathbb{R}^{N_v}=\mathrm{span}\{e,v_i,\text{ for } i\geq 2\}$, where e and all v_i are orthonormal, meaning that $\mathsf{X}=[e,v_2,\cdots,v_{N_v}]$ forms a unitary matrix.

These properties hold true regardless of the discretization of the operator $\mathcal{A}$ or the quadrature of $\mathcal{B}$. Only slight changes need to be made to account for different quadratures. If we denote $(\mu_1,\mu_2,\cdots\mu_{N_v})$ the quadrature points and $(w_1,w_2,\cdots w_{N_v})$ the corresponding weights, then B_μ simply write as:

$$\mathsf{B}_\mu=\frac{\varepsilon^2}{\Delta t}\sqrt{w}^{-1}v_1v_1^t\sqrt{w}+\left(1+\frac{\varepsilon^2}{\Delta t}\right)\sum_{i\geq 2}\sqrt{w}^{-1}v_iv_i^t\sqrt{w}, \tag{2.9}$$

https://doi.org/10.4208/cicp.OA-2016-0105 Published online by Cambridge University Press

where $\sqrt{w}=\text{diag}(\sqrt{w_1},\sqrt{w_2},\cdots,\sqrt{w_{N_v}})$ and $\sqrt{w}^{-1}$ is the inverse of $\sqrt{w}$. Here $\mathbb{R}^{N_v}=\text{span}\{v_j\}$ for $1\le j\le N_v$ with $v_1=[\sqrt{w_1},\cdots,\sqrt{w_{N_v}}]^t$, and v_j, $1\le j\le N_v$ are orthogonal with respect to matrix $w=\text{diag}(w_1,\cdots w_{N_v})$, i.e., $v_i^t w v_j=\delta_{ij}$. In fact, when we consider mid point rule on a uniform mesh, $w_i=\frac{1}{N_v}$ for all i and (2.9) boils down to (2.8).

It is easy to see that the first property suggests possible employment of fast algorithms such as conjugate gradient (CG) and the sparsity certainly should be utilized for matrix-free type algorithm designing, so that one could avoid storing all matrices inversion. The second property mainly explores the condition number of the system and should be used for preconditioning designing.

Remark 2.1. The previous two numerical methods in the introduction partially used these good properties. The unconditioned Krylov iteration method used the block-wisely sparse property of A and B but moves the P term to the right hand side, therefore suffering from slow convergence rate in the diffusion regime due to the canceling of I and P in the spectrum. The even-odd parity decomposition explicitly uses the eigenvalue structure of B and directly write the odd part as the $\mathcal{O}(\varepsilon)$ term, but since A term gets treated explicitly, the method could not overcome the parabolic scaling, meaning Δt is controlled by Δx^2.

2.3 Numerical method

Seen as above, we make use of the spectrum information to design pre-conditioner and use the positive-definite symmetric property as well as the sparsity of the matrices for computation speed-up. Unfortunately typically once a pre-conditioner is involved, the sparsity would be lost, making fast solvers unapproachable. We propose here to perform the computation in a matrix-free fashion and thus the number of flops are not increased, as will be explained in details below.

Since there is only one vector that gets mapped with a small eigenvalue, the strategy in designing the pre-conditioner is simply to pull this vector back by $\frac{\Delta t}{\varepsilon^2}$ in magnitude. For the isotropic term, simply put, the pre-conditioner we use is B^{-1}, and we leave the anisotropic cases for later. With it, the system (2.5) becomes:

$$(\mathsf{B}^{-1}\mathsf{A}+\mathsf{I})f=\mathsf{B}^{-1}b. \tag{2.10}$$

The benefits are immediate. As seen in (2.8) there are only two types of eigenvalues for B, they are either as small as $\frac{\varepsilon^2}{\Delta t}$ or order 1 as $1+\frac{\varepsilon^2}{\Delta t}$. Considering the fact that A has $\varepsilon^2\Delta t$ in its coefficients, we know roughly that the eigenvalues for $\mathsf{B}^{-1}\mathsf{A}+\mathsf{I}$ are all of $\mathcal{O}(1)$, ensuring a relatively small condition number.

Numerically it is a bad idea to invert $\mathsf{B}^{-1}\mathsf{A}+\mathsf{I}$ directly due to its large size, but since $\mathsf{B}^{-1}\mathsf{A}+\mathsf{I}$ is symmetric positive definite, it is natural to adopt the conjugate gradient method, which transforms inversion into multiplication till convergence, and the small condition number guarantees the speed of convergence. A fast matrix-vector multiplication is therefore called for.

https://doi.org/10.4208/cicp.OA-2016-0105 Published online by Cambridge University Press

This is, however, not straightforward since matrix B^{-1} is not as sparse as B in the original formulation, and if we do a naive multiplication, a large portion of computation effort will be needed, which degrades the advantage of using conjugate gradient method. To deal with this problem, we once again use the special structure of B_μ^{-1} and explore the low-rank decomposition. Specifically, let $g(x,\mu)$ be a function depends on both x and μ. Then upon discretization, $\mathsf{B}_\mu^{-1}g$ can be directly expressed by, as shown in (2.8):

$$\mathsf{B}_\mu^{-1}g=\frac{\Delta t}{\varepsilon^2}ee^t g+\frac{1}{1+\frac{\varepsilon^2}{\Delta t}}\left(g-ee^t g\right) \tag{2.11}$$

for a fixed x, and then $\mathsf{B}^{-1}g$ is to apply the equation above for every x. The multiplication with A is cheap due to sparsity.

Remark 2.2. As explained above, if a different set of quadrature is used, (2.11) generalizes to

$$\mathsf{B}_\mu^{-1}g=\frac{\Delta t}{\varepsilon^2}\sqrt{w}^{-1}v_1v_1^t\sqrt{w}g+\frac{1}{1+\frac{\varepsilon^2}{\Delta t}}\left(g-\sqrt{w}^{-1}v_1v_1^t\sqrt{w}g\right), \tag{2.12}$$

where v_1 and $\sqrt{w}$ are defined as in (2.9).

We summarize the algorithm below in Algorithm 1.

Algorithm 1: Outline of the **main** algorithm.

Data: Initial data: $f(t=0,x,v)$, final time T
Result: $f(t_n,x,v)$ for all time steps t_n.
Discretization: Δt, Δv, Δx;
Initialization: $t=0$, $f_{\mathrm{E}}(0,x,v)=\frac{1}{2}(f(0,x,v)+f(0,x,-v))$, $f_{\mathrm{O}}=f-f_{\mathrm{E}}$;
Matrix Preparation: A_x, e;
while $t<T$ **do**
  Set b as in (2.6);
  Compute f_E using (2.5), call function **ABinverse**;
  Compute f_{O} using (2.4) directly;
  Set $t=t+\Delta t$;
end

Algorithm 2: Function **ABinverse**: solving $(\mathsf{A}+\mathsf{B})f=b$.

Data: Initial data: b.
Result: f.
Set $b_{\mathrm{mod}}=\mathsf{B}_v^{-1}b$ using (2.8);
Compute $\mathsf{B}^{-1}(\mathsf{A}+\mathsf{B})f=\mathsf{B}^{-1}b$ using CG, in which we call function **ABproduct**;

https://doi.org/10.4208/cicp.OA-2016-0105 Published online by Cambridge University Press

Algorithm 3: Function **ABproduct**: computing $f=\mathsf{B}^{-1}(\mathsf{A}+\mathsf{B})b$.

Data: Initial data: b.
Result: $f=\mathsf{B}^{-1}(\mathsf{A}+\mathsf{B})b$.
Compute $b_1=\mathsf{A}\cdot b$;
Compute $b_2=\mathsf{B}^{-1}\cdot b_1$ using (2.11);
Compute $f=b_2+b$;

Note that in Algorithm 2, CG is called which requires the matrix product of $\mathsf{B}^{-1}(\mathsf{A}+\mathsf{B})$. But it is not applied directly. Through the entire computation we do not store any of the matrices, and all the computation is done in the matrix-free fashion. It is also worth mentioning that construction of A can be done with great generality as neither the computation of our pre-conditioner nor the fast matrix-vector multiplication technique here depends on the specific form of spatial discretization. In the examples we will show later, we use center difference for spatial derivatives and mid-point rule on a uniform gird for the averaging operator in μ. But other quadrature rule such as Gaussian quadrature used in S_N approximations [2] and various spatial discretization such as discontinuous Galerkin method [24] or discontinuous finite-element method [29] can be easily adapted. Now, it is important to point out that our algorithm, majorly solving the preconditioned system (2.10), only requires a three simple matrix-vector multiplications, two with B^{-1} and one with A^{-1}, in each iteration, whereas in the previous method (1.11), quite a few operations are needed: a diffusion solver, a transport sweep and an average calculation, not to mentioned some of them are needed more than twice. Therefore the main benefit of our method is its simplicity.

Remark 2.3. We would like to point out that the even-odd decomposition is *not* an indispensable part of our scheme. Indeed, the above idea of constructing and computing the pre-conditioner can be applied directly to the original system (2.1) without symmetrization technique via the even-odd parity. Rewrite (2.1) in a compact form

$$(\mathcal{C}+\mathcal{B})f^{n+1}=d, \tag{2.13}$$

where $\mathcal{C}=\varepsilon\mu\partial_x$, $\mathcal{B}$ is the same as (2.6), and $d=\frac{\varepsilon^2}{\Delta t}f^n$. Here we again assume $\sigma\equiv 1$ for simplicity. Then it boils down to invert the discretization matrix $\mathsf{C}+\mathsf{B}$, which can be done similarly using the previous framework expect changing conjugate gradient method to Generalized Minimal Residual method (GMRES).

Remark 2.4. Our scheme is easily extended to second order. As a center difference is used for the spatial derivative, we only need to upgrade our time discretization to second order. This is done by replacing (2.1) with

$$\frac{3f^{n+1}-4f^n+f^{n-1}}{2\Delta t}+\frac{1}{\varepsilon}\mu\cdot\partial_x f^{n+1}=\frac{\sigma}{\varepsilon^2}(\rho^{n+1}-f^{n+1}). \tag{2.14}$$

https://doi.org/10.4208/cicp.OA-2016-0105 Published online by Cambridge University Press

Then the rest steps are the same as that described in Section 2.3 except varying a few constants.

2.4 Anisotropic scattering

In this subsection, we generalize the above framework to the anisotropic scattering. In one dimensional slab geometry, the transport equation takes the form

$$\partial_t f+\frac{\mu}{\varepsilon}\partial_x f=\frac{\sigma_0(x)}{2\varepsilon^2}\int_{-1}^{1}\sigma(\mu\cdot\mu')(f(\mu')-f(\mu))d\mu', \tag{2.15}$$

where

$$\frac{1}{2}\int_{-1}^{1}\sigma(\mu\cdot\mu')d\mu'=1.$$

In this case, the diffusion limit is

$$\rho_t+\frac{1}{2}\int_{-1}^{1}\mu\partial_x\left(\mathcal{Q}^{-1}\left(\frac{\mu\partial_x\rho}{\sigma_0(x)}\right)\right)d\mu=0, \tag{2.16}$$

where

$$\mathcal{Q}(f)=\frac{1}{2}\int_{-1}^{1}\sigma(\mu\cdot\mu')f(\mu')d\mu'-f(\mu)$$

is the linear collision operator. The main idea of the numerical scheme follows that in Section 2.2, and here we present a non-symmetric version as mentioned in Remark 2.3, a reformulation to symmetric form is similar to (2.4) and we leave it to the reader.

Denote

$$\mathcal{P}^\sigma f=\frac{1}{2}\int_{-1}^{1}\sigma(\mu\cdot\mu')f(\mu')d\mu', \tag{2.17}$$

and assume $\sigma_0(x)\equiv 1$ for simplicity, then the semi-discrete scheme reads

$$(\mathsf{B}^\sigma+\mathsf{C})f^{n+1}=b. \tag{2.18}$$

Here B^σ and C are the discretization matrices for operator $1+\frac{\varepsilon^2}{\Delta t}-\mathcal{P}^\sigma$ and $\varepsilon\mu\partial_x$, respectively, and $b=\frac{\varepsilon^2}{\Delta t}f^n$. Write

$$\mathsf{B}^\sigma=\mathsf{I}_{N_x}\otimes\mathsf{B}^\sigma_\mu \quad \text{and} \quad \mathsf{B}^\sigma_\mu=\left(1+\frac{\varepsilon^2}{\Delta t}\right)I_{N_v}-\mathsf{P}^\sigma_\mu,$$

where P^σ_μ is the corresponding matrix for $\mathcal{P}^\sigma$ with fixed x. Then the spectral structure of B^σ_μ is similar to B_μ in (2.8) and we summarize as follows:

- The matrix P^σ_μ is low rank as it depends on the inner product of μ and μ'. Denote its eigenvalues $\xi_1>\xi_2>\cdots>\xi_k$, $k\ll N_v$, and corresponding eigenvectors $v_1,\cdots,v_k$.

https://doi.org/10.4208/cicp.OA-2016-0105 Published online by Cambridge University Press

- The largest eigenvalue of P_μ^σ is $\xi_1=1$, and corresponding eigenvector is $e=\frac{1}{\sqrt{N_v}}[1,\cdots,1]^t$.
- Denote the eigenvalues of B_σ^μ as $\lambda_1\le\lambda_2\cdots\le\lambda_{N_v}$. Then $\lambda_1=\frac{\varepsilon^2}{\Delta t}$ and $\lambda_{k+1}=\lambda_{k+2}=\cdots\lambda_{N_v}=1+\frac{\varepsilon^2}{\Delta t}$. The eigenvector corresponding to λ_1 is e. Other eigenvalues depend on the form of $\sigma(\mu\cdot\mu')$.
- The eigenvectors v_j, $j=k+1,\cdots N_v$ corresponding to the eigenvalues $\lambda_j=1+\frac{\varepsilon^2}{\Delta t}$ all satisfy $\mathsf{P}_\sigma^\mu v_j=0$.

Therefore, for any vector g of size N_v, first find its projection to the first k eigenvectors v_1, $\cdots$, v_k. That is, write $g=\sum_{i=1}^{N_v}c_iv_i$, and find c_1, $\cdots$, c_k. Then, we have

$$\mathsf{B}_\mu^\sigma g=\sum_{i=1}^{k}c_k\lambda_kv_k+\left(1+\frac{\varepsilon^2}{\Delta t}\right)\left(g-\sum_{i=1}^{k}c_kv_k\right),\tag{2.19}$$

and

$$(\mathsf{B}_\mu^\sigma)^{-1}g=\sum_{i=1}^{k}c_k\lambda_k^{-1}v_k+\frac{1}{1+\frac{\varepsilon^2}{\Delta t}}\left(g-\sum_{i=1}^{k}c_kv_k\right).\tag{2.20}$$

Since $k\ll N_v$, the computation (2.19) and (2.20) are cheap. In fact, in most applications, the scattering $\sigma(\mu\cdot\mu')$ has special structures such that v_2, $\cdots$, v_k are easy to find out. Then what is left is to precondition (2.18) as

$$(\mathsf{B}^\sigma)^{-1}(\mathsf{B}^\sigma+\mathsf{C})f^{n+1}=b,\tag{2.21}$$

and solve the resulting linear system using GMRES.

3 Properties

We study the numerical properties of our scheme in this section. We will concentrate on the equation with isotropic collision operator and the extension to the anisotropic case is straightforward, which we will omit from here.

3.1 Condition number of $\mathsf{B}^{-1}(\mathsf{A}+\mathsf{B})$

In the isotropic scattering case, it is easily seen that B^{-1} only has two eigenvalues, $\frac{\Delta t}{\varepsilon^2}$ and $\frac{\Delta t}{\Delta t+\varepsilon^2}$. Since A is the discretization of $-\frac{\varepsilon^2\Delta t}{\varepsilon^2+\Delta t}\mu^2\partial_x^2$, its spectrum is roughly given by (depending on the boundary too):

$$\frac{\varepsilon^2\Delta t}{\varepsilon^2+\Delta t}\mu^2k^2\quad\text{with}\quad k=0,1,\cdots,N_x-1.\tag{3.1}$$

Then the smallest possible and the biggest possible eigenvalues of $\mathsf{B}^{-1}\mathsf{A}$ are:

https://doi.org/10.4208/cicp.OA-2016-0105 Published online by Cambridge University Press

- biggest: $\frac{\Delta t}{\varepsilon^2}\cdot\frac{\varepsilon^2\Delta t^2}{\varepsilon^2+\Delta t^2}(N_x-1)^2$,
- smallest: 0.

Therefore, the condition number of $\mathsf{B}^{-1}(\mathsf{A}+\mathsf{B})$ is

$$\kappa_{\mathsf{B}^{-1}(\mathsf{A}+\mathsf{B})}\simeq\frac{1+\frac{\Delta t}{\varepsilon^2}\cdot\frac{\varepsilon^2\Delta t^2}{\varepsilon^2+\Delta t^2}(N_x-1)^2}{1}\sim 1+\frac{\Delta t}{\varepsilon^2+\Delta t^2}, \tag{3.2}$$

which is $\mathcal{O}(1)$ for both small and big ε. Compare this to the condition number of $\mathsf{A}+\mathsf{B}$

$$\kappa_{\mathsf{A}+\mathsf{B}}=\frac{1+\frac{\varepsilon^2}{\Delta t}+\frac{\varepsilon^2\Delta t}{\varepsilon^2+\Delta t}\mu^2(N_x-1)^2}{\frac{\varepsilon^2}{\Delta t}}\sim\mathcal{O}\left(\frac{\Delta t}{\varepsilon^2}\right), \tag{3.3}$$

we see clearly that the pre-conditioner plays an important role in the diffusive regime.

3.2 Computational cost

We first analyze the computational cost per each time step. Assume that we are given f_{E}^n and f_{O}^n, we calculate the flops it takes to update for f_{E}^{n+1} and f_{O}^{n+1}. As can be seen in Algorithm 1, b is computed using (2.6) and f_{O}^{n+1} is computed using (2.4), both of which require $\mathcal{O}(N_x)$ flops of computation per angular grid and thus $\mathcal{O}(N_xN_v)$ flops of computation in total. To compute f_{E}^{n+1}, the scheme calls Algorithm 2, in which b_{mod} requires N_xN_v flops, and conjugate gradient is used for f_{E}^{n+1}. Denote Tol the tolerance set for the method, the number of iteration needed for CG is:

$$N_{\mathrm{iter}}\geq\frac{\log\mathrm{Tol}}{\log\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)}, \tag{3.4}$$

where κ is the condition number for matrix $\mathsf{B}^{-1}(\mathsf{A}+\mathsf{B})$. With the results above, the matrix is well-conditioned with $\kappa\sim 1$, and thus the required iteration is small. For example, if we set $\mathrm{Tol}=1e-10$ and $\Delta t=1e-3$, then:

$$N_{\mathrm{iter}}\gtrsim\frac{\log\mathrm{Tol}}{\log\left(\frac{\frac{1}{2}\Delta t^3}{2+\frac{1}{2}\Delta t^3}\right)}\gtrsim\frac{\log\mathrm{Tol}}{3\log\Delta t}=\frac{10}{9}. \tag{3.5}$$

In each iteration in the conjugate gradient (CG) method, the matrix multiplication of $\mathsf{B}^{-1}(\mathsf{A}+\mathsf{B})$ needs to be applied three times. Notice that our scheme is matrix free (as shown in Algorithm 3), that is, we perform A product first, which requires $\mathcal{O}(N_xN_v)$ flops, and then perform B^{-1} product, which also requires the same amount of flops. Thus in total, each iteration in the CG method requires $\mathcal{O}(N_xN_v)$ flops. Multiplied by $\mathcal{O}(1)$ iterations required by the CG method, f_{E}^{n+1} is computed using $\mathcal{O}(N_xN_v)$ flops.

https://doi.org/10.4208/cicp.OA-2016-0105 Published online by Cambridge University Press

It is very important to compare the computational cost of our scheme with the state-of-the-art preconditioned Krylov method. The major computational burden in both schemes come from the iteration in solving the linear system. That is

$$\mathsf{B}^{-1}(\mathsf{A}+\mathsf{B})f=\mathsf{B}^{-1}b \tag{3.6}$$

in our scheme, and

$$\left(\mathsf{I}+\frac{\sigma\Delta t}{\varepsilon^2}\mathsf{D}^{-1}\right)\left(\mathsf{I}-\frac{\sigma\Delta t}{\varepsilon^2}\mathsf{P}\mathsf{L}^{-1}\right)\rho=\left(\mathsf{I}+\frac{\sigma\Delta t}{\varepsilon^2}\mathsf{D}^{-1}\right)\mathsf{P}\mathsf{L}^{-1}f \tag{3.7}$$

in the preconditioned Krylov method, where we recall

$$\mathsf{A}=\frac{\varepsilon^2\Delta t}{\varepsilon^2+\Delta t}\partial_x^2\otimes\mathrm{diag}\{\mu_k^2\},\quad \mathsf{B}=\mathsf{I}_{N_x}\otimes\left(\mathsf{I}_{N_v}+\frac{\varepsilon^2}{\Delta t}-\mathsf{P}\right),$$

and

$$\mathsf{D}=\partial_x\left(\frac{1}{3\sigma}\partial_x\right),\quad \mathsf{L}=\left(\mathsf{I}_{N_x}+\frac{\sigma\Delta t}{\varepsilon^2}\right)\otimes\mathsf{I}_{N_v}+\frac{\Delta t}{\varepsilon}\partial_x\otimes\mathrm{diag}\{\mu_k\}.$$

Although it seems from the expression that (3.6) is more expensive than (3.7) as (3.6) iterates for f, a N_xN_v vector and (3.7) iterates for ρ, a vector of size N_x, the truth is that the computational cost are comparable. Indeed, for any iterative method, the major cost comes from the matrix-vector multiplication, and the number of iterations needed. If we assume that the number of matrix-vector multiplications are the same for both methods per iteration, then from the above calculation, our method needs $\mathcal{O}(N_xN_v)$ flops and it is the same with (3.7). In fact, the matrix-vector multiplication in (3.7) is accomplished via a sequence of operations, including a sweep for each velocity $\mathsf{L}^{-1}\rho$ ($\mathcal{O}(N_xN_v)$ flops), averaging $\mathsf{P}\mathsf{L}^{-1}\rho$ ($\mathcal{O}(N_x)$ flops), and a diffusion solver ($\mathcal{O}(N_x)$ flops), and thus $\mathcal{O}(N_xN_v)$ flops in total. The number of iterations relies on the condition number of the matrix to be inverted (if conjugate gradient method is used) and then the calculation in Section 3.1 guarantees the fast convergence of our method. Therefore, within the loop of iteration in solving the linear system, (3.6) and (3.7) are at the same order of computational cost. And the preconditioned Krylov method needs an extra outer loop to iterate between ρ and f whereas ours does not.

3.3 Asymptotic preservation

Now we check the asymptotic preservation property for the scheme (2.1). First taking the average of (2.1), one has

$$\frac{\rho^{n+1}-\rho^n}{\Delta t}+\frac{1}{\varepsilon}\left\langle\mu\partial_x f^{n+1}\right\rangle=0. \tag{3.8}$$

https://doi.org/10.4208/cicp.OA-2016-0105 Published online by Cambridge University Press

Notice that

$$f^{n+1}=\rho^{n+1}-\frac{\varepsilon\mu}{\sigma}\partial_x f^{n+1}-\frac{\varepsilon^2}{\sigma\Delta t}(f^{n+1}-f^n).$$

Plugging it into (3.8) leads to

$$\frac{\rho^{n+1}-\rho^n}{\Delta t}+\partial_x\left(\frac{1}{3\sigma}\partial_x\rho^{n+1}\right)=\mathcal{O}(\varepsilon), \tag{3.9}$$

which, upon sending ε to zero, is a semi-discrete implicit solver for the diffusion limit.

3.4 Stability

Next, we prove the stability of our scheme (2.1). The result is summarized as follows.

Theorem 3.1. *The scheme* (2.1) *is unconditionally stable.*

Proof. Multiply the Eq. (2.1) by f^{n+1}, and integrate with respect to x and v, we have

$$\begin{aligned}&\frac{1}{2}\left[|||(f^{n+1})^2|||-|||(f^n)^2|||+|||(f^{n+1}-f^n)^2|||\right]\\=&\frac{\sigma\Delta t}{\varepsilon^2}\left(|||\rho^{n+1}f^{n+1}|||-|||(f^{n+1})^2|||\right).\end{aligned} \tag{3.10}$$

Here we use $|||\cdot|||$ to denote the integration with respect to both x and v and the equality $a(a-b)=\frac{1}{2}\left[a^2-b^2+(a-b)^2\right]$. Considering $|||\rho^{n+1}f^{n+1}|||=\|(\rho^{n+1})^2\|$, with $\|\cdot\|$ denoting integration only on x, and by Cauchy-Schwartz inequality, we have:

$$|||\rho^{n+1}f^{n+1}|||\leq|||(f^{n+1})^2|||, \tag{3.11}$$

making the right hand side negative. Therefore,

$$|||(f^{n+1})^2|||\leq|||(f^n)^2||| \tag{3.12}$$

regardless of the choice of Δt and thus unconditional stability is guaranteed. □

4 Numerical examples

In this section, we present several numerical examples using our AP scheme described in Algorithm 1-3. Here we use uniform mesh for both spatial and angular discretization. Periodic boundary condition is adopted in both directions.

https://doi.org/10.4208/cicp.OA-2016-0105 Published online by Cambridge University Press

4.1 Condition number comparison

We check the effectiveness of the preconditioner by comparing the condition number of $\mathsf{A}+\mathsf{B}$ and $\mathsf{B}^{-1}\mathsf{A}+\mathsf{I}$ with different ε. It can be seen from Table 1, for very small ε, the condition number is extremely big and the pre-conditioner B^{-1} drastically decreased the condition number. For moderate ε, the pre-conditioner is less effective. As seen in Table 2, the condition number of $\mathsf{A}+\mathsf{B}$ is of $\mathcal{O}(1)$ already and the pre-conditioner does not make much difference.

4.2 Computational cost comparison

In this subsection we compare the computational cost for updating f_{E} by only one time step. We update f_{E} by both directly inverting the matrix and using CG method. In Table 4 we show the ratio of the time used. With $N_x=200$ and $N_v=20$, the time it takes by directly inverting the matrix is 130 more than utilizing the CG method. Note that here we only report the saving from the online computation, assuming the computer could store the matrix. In practice, one also needs to consider the memory cost for directly inverting the matrix. The new AP solver, on the other hand, is a matrix-free method, and there is no need to store matrices.

Table 1: Condition number for $\mathsf{A}+\mathsf{B}$ and $\mathsf{B}^{-1}\mathsf{A}+\mathsf{I}$ with $\varepsilon=10^{-5}$.

Condition number for $\mathsf{A}+\mathsf{B}$.				Condition number for $\mathsf{B}^{-1}\mathsf{A}+\mathsf{I}$.		
	$N_v=10$	$N_v=20$	$N_v=30$	$N_v=10$	$N_v=20$	$N_v=30$
$N_x=20$	1.59e8	1.59e8	1.59e8	15.88	16.14	16.19
$N_x=40$	8.12e7	8.12e7	8.12e7	31.50	32.04	32.14
$N_x=60$	5.46e7	5.46e7	5.46e7	47.12	47.94	48.09
$N_x=80$	4.11e7	4.11e7	4.11e7	62.74	63.84	64.05
$N_x=100$	3.30e7	3.30e7	3.30e7	78.37	79.75	80.00

Table 2: Condition number for $\mathsf{A}+\mathsf{B}$ and $\mathsf{B}^{-1}\mathsf{A}+\mathsf{I}$ with $\varepsilon=1$.

Condition number for $\mathsf{A}+\mathsf{B}$.				Condition number for $\mathsf{B}^{-1}\mathsf{A}+\mathsf{I}$.		
	$N_v=10$	$N_v=20$	$N_v=30$	$N_v=10$	$N_v=20$	$N_v=30$
$N_x=60$	1.36	1.40	1.42	1.36	1.40	1.41
$N_x=100$	1.36	1.40	1.42	1.36	1.40	1.41

Table 3: Condition number for $\mathsf{A}+\mathsf{B}$ and $\mathsf{B}^{-1}\mathsf{A}+\mathsf{I}$ with $\varepsilon=10^{-3}$.

Condition number for $\mathsf{A}+\mathsf{B}$.				Condition number for $\mathsf{B}^{-1}\mathsf{A}+\mathsf{I}$.		
	$N_v=10$	$N_v=20$	$N_v=30$	$N_v=10$	$N_v=20$	$N_v=30$
$N_x=60$	5.52e3	5.53e3	5.53e3	46.95	47.75	47.90
$N_x=100$	3.41e3	3.42e3	3.42e3	77.62	78.94	79.19

https://doi.org/10.4208/cicp.OA-2016-0105 Published online by Cambridge University Press

Table 4: Computational cost ratio between directly inverting the matrix, and utilizing the conjugate gradient method.

Computational Cost $t_{\text{inverse}}/t_{\text{cg}}$.			
	$\varepsilon=1$	$\varepsilon=0.1$	$\varepsilon=0.01$
$N_x=50$	1.5733	2.6637	3.1782
$N_x=100$	54.3670	3.0818	4.4672
$N_x=150$	38.2581	13.1954	6.2918
$N_x=200$	130.6267	12.1222	10.9596

4.3 One dimensional problem with isotropic scattering

In this section, we test the efficiency of our numerical schemes in slab geometry. The examples are similar to those in [13]. Here $x\in[0,2]$. In all examples, Δt is chosen to be $\Delta x/3$ when using our scheme.

4.3.1 Example I

We consider the following initial condition

$$f(0,x,\mu)=\begin{cases}2, & 0.8<x<1.2,\\ 0, & \text{otherwise},\end{cases} \tag{4.1}$$

and isotropic scattering with vanishing cross section (see the left figure of Fig. 1)

$$\sigma(x)=100(x-1)^4. \tag{4.2}$$

For $\varepsilon=1$, we compare the solution using our scheme with the solution from an explicit solver. And the profile of ρ at time $t_{\max}=1$ is displayed in the left figure of Fig. 2. Here we choose $N_x=200$ and $N_v=100$. In the diffusion scaling, we choose $\varepsilon=10^{-3}$ and compare our solution with the solution of the diffusion limit (1.2) at time $t_{\max}=0.1$, as shown in the right figure of Fig. 2. Good agreement are observed for both kinetic and diffusion regimes. Moreover, near the center of the computational domain where the scattering effect is extremely weak, our scheme successfully pick up the correct profile of density in the free streaming limit.

4.3.2 Example II

Initial condition is the same as in (4.1) and we consider striped cross section

$$\sigma(x)=\begin{cases}0.02, & x\in[0.35,0.65]\cup[1.35,1.65],\\ 1, & x\in[0,0.35)\cup(0.65,1.35)\cup(1.65,1],\end{cases} \tag{4.3}$$

so that the particles will transport from high scattering regime to low scattering regime and vice versa. Again we compute our solution for both $\varepsilon=1$ and $\varepsilon=10^{-3}$. In the former case, the solution is compared with the one using explicit solver, while in the later case, it

https://doi.org/10.4208/cicp.OA-2016-0105 Published online by Cambridge University Press

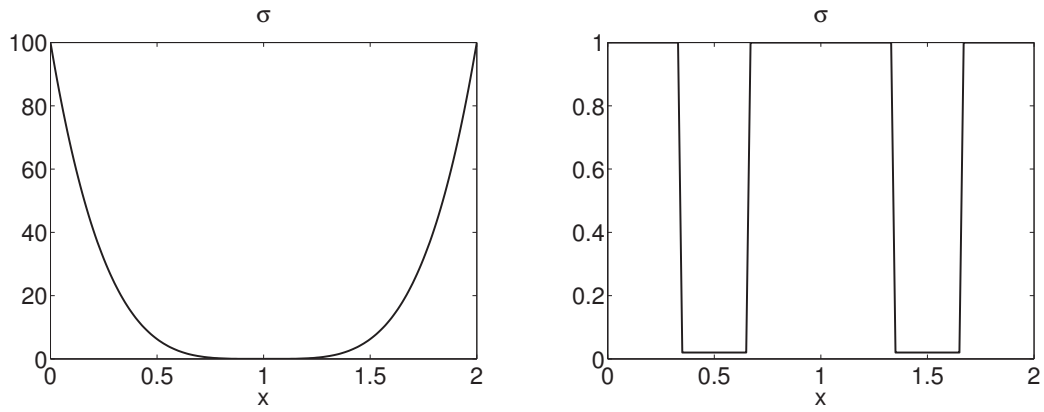


Figure 1: Scattering cross section σ for one dimensional examples. Left: vanishing cross-section (4.2). Right: striped cross-section (4.3).

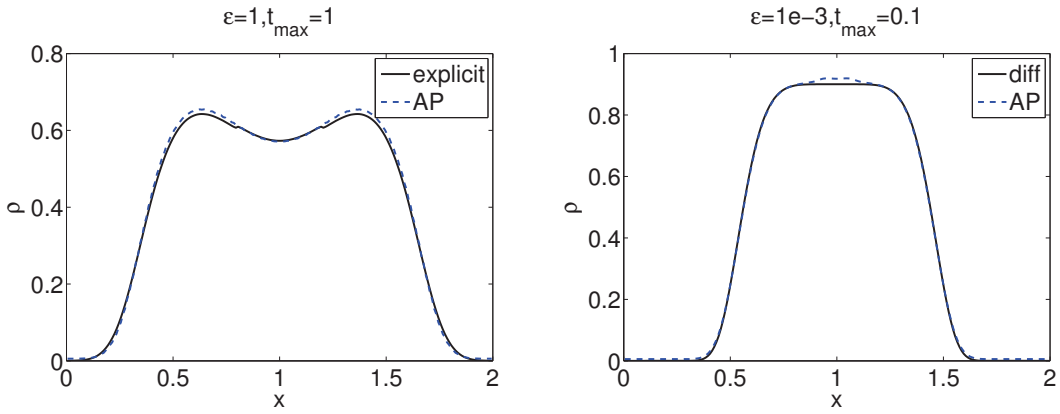


Figure 2: Example I. Here $N_x=200$, $N_v=100$. Left: $\varepsilon=1$, we compare the density ρ using our scheme (blue dashed curve) with the solution using explicit solver (black solid curve) at $t_{\max}=1$. Right: $\varepsilon=10^{-3}$, we compare the density ρ using our scheme (blue dashed curve) with the solution to the diffusion limit (black solid curve) at $t_{\max}=0.1$.

is compared with the one using diffusion solver. The results are gathered in Fig. 3. This example validates the efficiency of our scheme in computing the transport equation with discontinuous cross-section, which is often the case in many real materials.

4.3.3 Example III

Here we consider anisotropic scattering cross section

$$\sigma(x,\mu,\mu')=\sigma_0(x)\left(1+\mu\cdot\mu'\right) \tag{4.4}$$

https://doi.org/10.4208/cicp.OA-2016-0105 Published online by Cambridge University Press

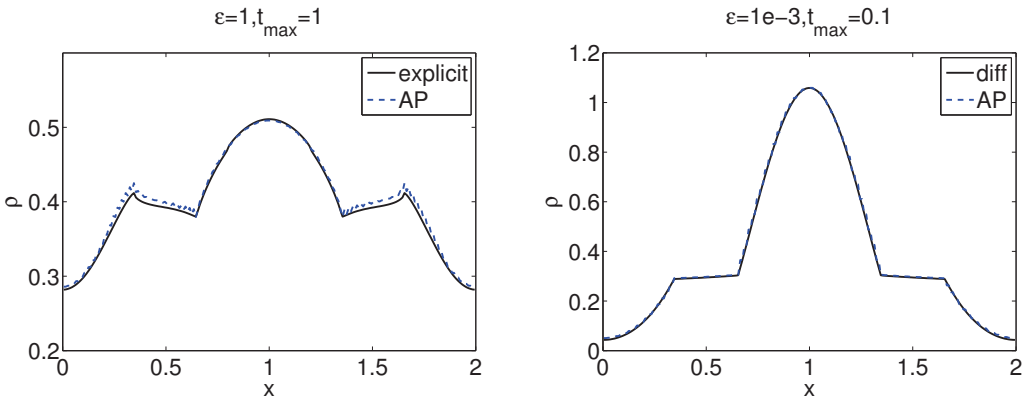


Figure 3: Example II. Here $N_x=200$, $N_v=100$. Left: $\varepsilon=1$, we compare the density ρ using our scheme (blue dashed curve) with the solution using explicit solver (black solid curve) at $t_{\max}=1$. Right: $\varepsilon=10^{-3}$, we compare the density ρ using our scheme (blue dashed curve) with the solution to the diffusion limit (black solid curve) at $t_{\max}=0.1$.

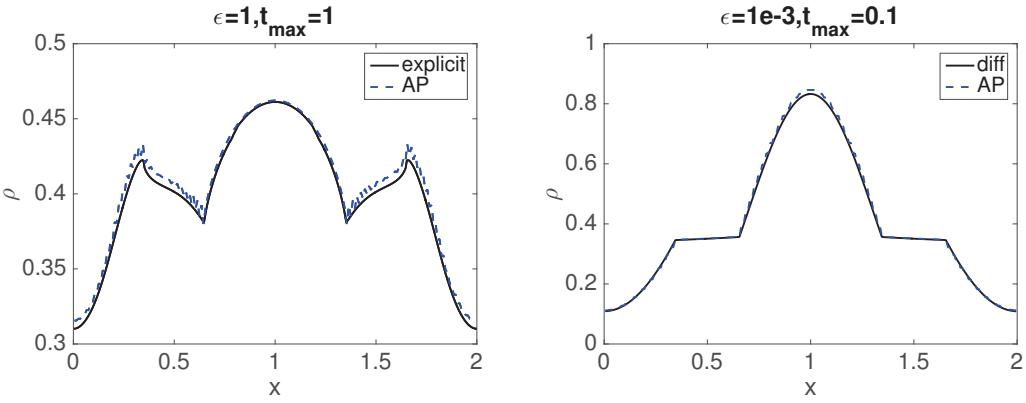


Figure 4: Example III: anisotropic case. Here $N_x=200$, $N_v=100$. Left: $\varepsilon=1$, we compare the density ρ using our scheme (blue dashed curve) with the solution using explicit solver (black solid curve) at $t_{\max}=1$. Right: $\varepsilon=10^{-3}$, we compare the density ρ using our scheme (blue dashed curve) with the solution to the diffusion limit (black solid curve) at $t_{\max}=0.1$.

with $\sigma_0(x)$ taking the form of (4.3). For this special choice of σ, the diffusion limit (2.16) reduces to

$$\rho_t+\frac{1}{2}\partial_x\left(\frac{1}{\sigma_0}\partial_x\rho\right)=0. \tag{4.5}$$

Also, the eigenvalues of B_μ^σ are $\lambda_1=\frac{\varepsilon^2}{\Delta t}$, $\lambda_2=1+\frac{\varepsilon^2}{\Delta t}-\frac{1}{3}$, and corresponding normalized eigenvectors are $v_1=e$ and $v_2=\frac{\mu}{\sqrt{\mu^t\mu}}$. And the rest eigenvalues are all equal to $1+\frac{\varepsilon^2}{\Delta t}$. Then for any vector $g=\sum_{i=1}^{N_v}c_iv_i$, notice that $\mathsf{P}_\mu^\sigma g=c_1v_1+\frac{1}{3}c_2v_2$, and v_1 is orthogonal to v_2, we have, $c_1=v_1^t\mathsf{P}_\mu^\sigma g$ and $c_2=3v_2^T\mathsf{P}_\mu^\sigma g$. Since P_μ^σ is low rank, the matrix-vector multiplication

https://doi.org/10.4208/cicp.OA-2016-0105 Published online by Cambridge University Press

$\mathsf{P}^{\sigma}_{\mu}g$ is cheap. Once c_1 and c_2 are computed, we can use the formula (2.20) with $k=2$ to compute the inverse of $\mathsf{B}^{\sigma}_{\mu}$ and the rest steps are the same as isotropic ones.

To illustrate, Fig. 4 on the left compares our solution with the solution using explicit scheme, and on the right compares with the solution to the diffusion limit. Here $\Delta x=0.01$ and $\Delta v=0.02$.

4.4 Two-dimensional problems

Finally we test our scheme in planar geometry.

4.4.1 Example V

Consider smooth initial condition

$$f(0,x,y,\xi,\eta)=1+e^{-40(x-0.5)^2-40(y-0.5)^2},\quad 0\le x,y\le 1, \tag{4.6}$$

and uniform cross section $\sigma(x,y)\equiv 1$. Here we check the asymptotic property of our scheme by computing the l^2 distance between f and ρ, namely

$$|f-\rho|_2=\sqrt{\sum_i\sum_j|f(x_i,v_j)-\rho(x_i)|^2\Delta x\Delta\mu} \tag{4.7}$$

with various ε along time, and the results are collected in Fig. 5 left. As expected, this error decreases with ε. And Fig. 5 on the right further confirms the asymptotic property by comparing the density using our kinetic solver with that using a diffusion solver

$$|\rho_{\mathrm{k}}-\rho_{\mathrm{d}}|_2=\sqrt{\sum_i|\rho_{\mathrm{k}}(x_i)-\rho_{\mathrm{d}}(x_i)|^2\Delta x}. \tag{4.8}$$

4.4.2 Example VI

Initial data is the same as in (4.6) and scattering cross-section takes the following form

$$\sigma(x,y)=\begin{cases}0.02, & (x,y)\in[0.25,0.35]\times[0.25,0.35]\cup[0.65,0.75]\times[0.65,0.75],\\ 1, & \text{elsewhere},\end{cases} \tag{4.9}$$

which again contains both strong and weak scattering regimes (see Fig. 6) on the upper left. For $\varepsilon=1e-4$, we compare the solution using our scheme with the solution to the diffusion limit at time $t_{\max}=0.1$, which is displayed in Fig. 6 with good agreement. And we also compare our solution with the solution using explicit solver the transport equation for $\varepsilon=1$ at time $t_{\max}=0.1$, and collect the result in Fig. 7, where again good agreement is observed.

https://doi.org/10.4208/cicp.OA-2016-0105 Published online by Cambridge University Press

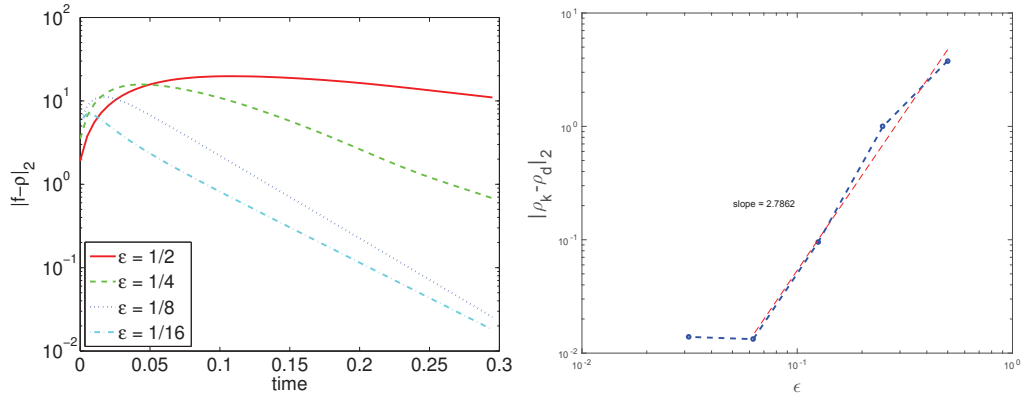


Figure 5: Example V. Left: l^2 distance between f and ρ (4.7) versus time using our kinetic solver. Right: l^2 distance between ρ_k, the density of the kinetic and ρ_d, the density to the diffusion equation. The error saturates at the last point of computation.

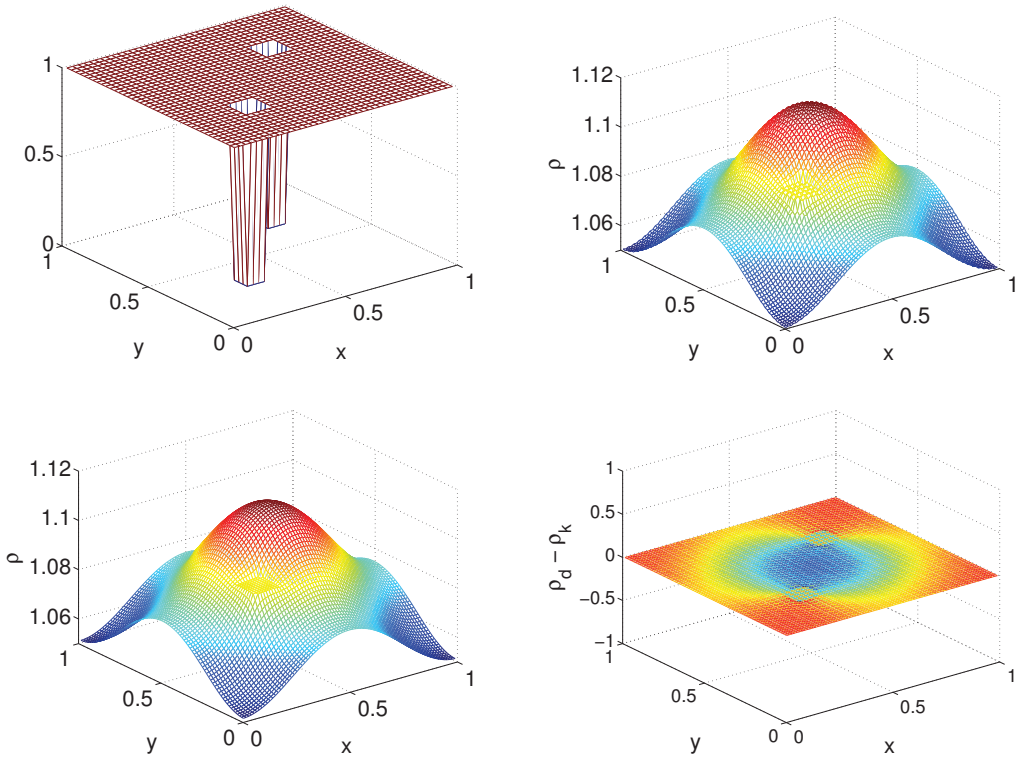


Figure 6: Example VI. Upper Left: Cross section (4.9). Upper right: solution with our kinetic solver. Lower left : solution to the diffusion limit (1.4). Lower right: difference between the two. Here $N_x = N_y = 80$, $N_v = 10$ and $\varepsilon = 10^{-4}$.

https://doi.org/10.4208/cicp.OA-2016-0105 Published online by Cambridge University Press

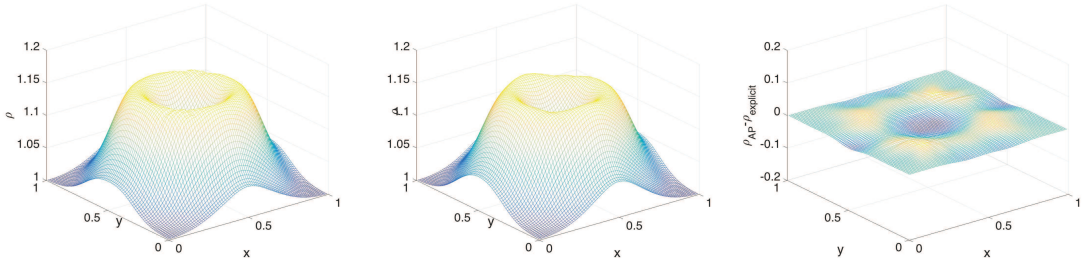


Figure 7: Example VI. Left: solution with our AP solver. Middle: solution with the explicit kinetic solver. Right: difference between the two. Here $N_x = N_y = 80$, $N_v = 10$ and $\varepsilon = 1$.

5 Conclusions

In this paper, we designed a fast solver for the fully implicit treatment of the linear transport equation. When the scattering effect is strong, this equation exhibits diffusive scaling such that both the convection and collision become stiff. On the other hand, when the scattering is very weak, the photon dynamics will be dominated by a free transport at the speed of light. In either case, numerically solving the equation requires a special care to deal with the stiffness. The fully implicit time discretization we considered here effectively treat the stiffness without resolving the mesh size, but at the cost of generating a large algebraic system that needs to invert, which is also ill-conditioned and not necessarily symmetric. We propose an efficient pre-conditioner which significantly improve and condition number and allows matrix-free treatment. The key ingredient is to use the spectral structure for the collision operator, which is also the source of ill-conditioning, to compute the pre-conditioner. We also reformulate the system via an even-odd parity so that the resulting linear system is symmetric and positive definite that can be inverted using conjugate gradient method with ease. An asymmetric version is also available and can be inverted through Krylov method such as GMRES. A major benefit of our new method is that it does not depend on the specific form of spatial or angular discretization, therefore it can be used with great generality. In the near future, we will generalize this method to nonlinear transport equation.

Acknowledgments

The work of Q. Li is supported in part by a start-up fund from UW-Madison and National Science Foundation under the grant DMS-1619778. The work of L. Wang is supported in part by the National Science Foundation under the grant DMS-1620135. Both authors would like to express gratitude to the support from the NSF research network grant RNMS11-07444 (KI-Net). We also thank Professors Shi Jin, Jim Morel and Cory Hauck for fruitful discussions.

https://doi.org/10.4208/cicp.OA-2016-0105 Published online by Cambridge University Press

References

[1] S. Ashby, P. Brown, M. Dorr and A. Hindmarsh, A linear algebraic analysis of diffusion synthetic acceleration for the Boltzmann transport equation, SIAM J. Numer. Anal., 32 (1995), 128–178.

[2] M. Adams and E. Larsen, Fast iterative methods for discrete-ordinate particle transport calculations, Prog. Nucl. Eng., 40 (2002), 3–159.

[3] Y. Azmy, Unconditionally stable and robust adjacent-cell diffusive preconditioning of weighted-difference particle transport methods is impossible, J. Comput. Phys., 182 (2002), 213.

[4] P. Brantley and E. Larsen, The simplified P3 approximation, Nucl. Sci. Eng., 134 (2001), 1–21.

[5] S. Boscarino, L. Pareschi and G. Russo, Implicit-explicit Runge-Kutta scheme for hyperbolic systems and kinetic equations in the diffusion limit, SIAM J. Sci. Comput., 35 (2013), 22–51.

[6] P. Brown, A linear algebraic development of diffusion synthetic acceleration for three-dimensional transport equations, SIAM J. Numer. Anal., 32 (1995), 179–214.

[7] J. A. Fleck and J. D. Cummings, An implicit Monte Carlo scheme for calculating time and frequency dependent nonlinear radiation transpor, J. Comput. Phys., 8 (1971), 313–342.

[8] M. Frank, A. Klar, E. Larsen and S. Yasuda, Time-dependent simplified pn approximation to the equations of radiative transfer, J. Comput. Phys., 226 (2007), 2289–2305.

[9] M. Frank and B. Seibold, Optimal prediction for radiative transfer: a new perspective on moment closure, Kinet. Relat. Models, 31 (2011), 717–733.

[10] B. Guthrie, J. Holloway and B. Patton, GMRES as a multi-step transport sweep accelerator, Trans. Theory Stat. Phys., 289(1) (1999), 83–102.

[11] F. Golse, S. Jin and D. Levermore, The convergence of numerical transfer schemes in diffusive regimes i: The discrete-ordinate method, SIAM J. Numer. Anal., 36 (1999), 1333–1369.

[12] C. Hauck, High-order entropy-based closures for linear transport in slab geometries, Commun. Math. Sci., 9 (2011), 187–205.

[13] C. Hauck and R. Lowrie, Temporal regularization of the p_n equations, Multiscale Model. Simul., 7 (2009), 1497–1524.

[14] S. Jin and D. Levermore, Fully discrete numerical transfer in diffusive regimes, Transp. Theory Stat. Phys., 22 (1993), 739–791.

[15] S. Jin, L. Pareschi and G. Toscani, Uniformly accurate diffusive relaxation schemes for multiscale transport equations, SIAM J. Numer. Anal., 38 (2000), 913–936.

[16] K. Kupper, M. Frank and S. Jin, An asymptotic preserving 2D staggered grid method for multiscale transport equations, submitted, 2015.

[17] A. Klar, An asymptotic-induced scheme for nonstationary transport equations in the diffusive limit, SIAM J. Numer. Anal., 35(6) (1998), 1097–1094.

[18] E. Larsen, Diffusion theory as an asymptotic limit of transport theory for nearly critical systems with small mean free paths, Ann. Nucl. Energy, 7 (1980), 249–255.

[19] E. Lewis Jr., and W. Miller, Computational Methods of Neutron Transport John Wiley and Sons, 1983.

[20] M. Lemou and L. Mieussens, New asymptotic preserving scheme based on micro-macro formulation for linear kinetic equations in the diffusion limit, SIAM J. Sci. Comput., 31 (2008), 334–368.

[21] E. Larsen and J. Morel, Asymptotic solutions of numerical transport problems in optically thick, diffusive regimes ii, J. Comput. Phys., 83 (1989), 212–236.

[22] L. Mieussens, On the asymptotic preserving property o fate unified gas kinetic scheme for

https://doi.org/10.4208/cicp.OA-2016-0105 Published online by Cambridge University Press

the diffusion limit of linear kinetic modle, J. Comput. Phys., 253 (2013), 138–156.
[23] J. Morel, T. Wareing, R. Lowrie and D. Parsons, Analysis of ray-effect mitigation techniques, Nucl. Sci. Eng., 144(1) (2003), 1–22.
[24] E. Olbrant, C. Hauck and M. Frank, A realizability-preserving discontinuous Galerkin method for the m1 model of radiative transfer, J. Comput. Phys., 231 (2012), 5612–5639.
[25] E. Olbrant, E. Larsen, M. Frank and B. Seibold, Asymptotic derivation and numerical investigation of time-dependent simplified P_N equations, J. Comput. Phys., 238(1) (2012), 315–336.
[26] G. Pomraning, Asymptotic and variational derivations of the simplified PN equations, Ann. Nucl. Energy, 20 (1993), 623–637.
[27] W. Sun, S. Jiang and K. Xu, An asymptotic preserving unified gas kinetic scheme for gray radiative transfer equations, J. Comput. Phys., 285(5) (2015), 265–279.
[28] D. Tomasevic and E. Larsen, The simplified $P2$ approximation, Nucl. Sci. Eng., 122 (1996), 309–325.
[29] T. A. Wareing, J. McGhee, J. Morel and S. Pautz, Discontinuous finite element s_n methods on three-dimensional unstructured grids, Nucl. Sci. Eng., 138(3) (2001), 256–268.
[30] J. Warsa, T. Wareing and J. Morel, Krylov iterative methods and the degraded effectiveness of diffusion synthetic acceleration for multidimensional Sn calculations in problems with material discontinuities, Nuclear Math. Comput. Sci., 147 (2004), 218–248.

https://doi.org/10.4208/cicp.OA-2016-0105 Published online by Cambridge University Press