

Three dimensional cold-plasma simulations on adaptive grids with internal boundaries

Robert Marskar^a

^a*SINTEF Energy Research*

Abstract

1. Introduction

A streamer is a non-thermal, self-propagating weakly ionized plasma filament with a bright (i.e. radiating) head which precedes electrical sparks. A streamer moves due to the motion of non-Maxwellian electrons in the head, and may move with or against the electron drift direction. A streamer which moves *with* the electron drift direction contains a net negative charge in its head, and is therefore called a negative streamer. Conversely, a streamer which moves *opposite* to the electron drift direction contains a net positive charge in the head and is called a positive streamer. Generally speaking, both modes propagate due to electron impact ionization (i.e. when ionization exceeds attachment) in the streamer head. However, a positive streamer additionally requires access to free electrons in front of the head, which are normally supplied through photoionization mechanisms or electron detachment from a background ionization. In air, for example, a positive streamer propagates primarily by electron excitation of molecular nitrogen which can radiatively relax down to the ground state. Since the energy difference between the excited and ground states of nitrogen exceed the ionization potential of molecular oxygen, photo-relaxation can produce seed electrons by photoionization of molecular oxygen.

Streamers are abundant in nature and are relevant in a number of industries. For example, streamer modeling is relevant for the high-voltage industry since its dynamics determine the electrical withstand. Furthermore, since a streamer may initiate in a super-critical field region (i.e. field regions where the electric field is above the breakdown field) and can then propagate through a subcritical region by field self-enhancement, streamer modeling can determine the minimum background field through which a streamer can propagate. In other types of industries, streamers are actually desired. For example, in high-intensity discharge (HID) lamps, a streamer is responsible for establishing the conductive path where the initial electric arc strikes. In such applications, a streamer crossing the gas gap is warranted, but its attachment of the streamer to the glass can damage the lamp and must be prevented.

The computational modelling of weakly ionized plasmas is inherently difficult due to the existence multiple length and time scales, as well as the presence of numerical stiffness. For atmospheric air, for example, a streamer channel is surrounded by a space charge layer with a thickness in the micrometer range, while the streamer itself can propagate tens of centimeters. Typical aspect ratios up to 1000 can therefore easily be envisioned, and adaptive mesh refinement (AMR) is almost universally required for numerically resolving such phenomena, especially in three dimensions. The inclusion of internal boundary conditions inside the computational domain leads to additional complexities, and it is essential that these can be resolved in a way that remain compatible with AMR. Other difficulties of a more fundamental nature are related to the treatment of the various species (neutral, ions, electrons). For the most part, neutrals and ions are close to equilibrium for weakly ionized plasmas, and the velocity distribution function (VDF) is therefore Maxwellian so that they can be treated in the continuum (i.e. fluid) approximation. This is not necessarily the case for electrons whose VDF may be far from Maxwellian and may therefore require a kinetic (i.e. particle based) treatment. Hybrid models that combine kinetic and fluid approaches are, strictly speaking, still in their infancy but are starting to gain traction in the plasma community. The stability of such methods have

been a challenge due to the inherent statistical noise that is present in e.g. Direct Monte Carlo Simulations (DMCS) or Particle-In-Cell (PIC) based approaches. Methods for direct solutions of the Boltzmann equation are also being developed. The Discrete Velocity Method, for example, reduces the Boltzmann equation to a system of hyperbolic equations for a set of velocities and is therefore a direct analogue of the Discrete Ordinates Method (DOM) for the radiative transfer equation (RTE). The DVM is, strictly speaking, different from Lattice-Boltzmann methods due to the number of velocities that are included in the discretization of phase space (usually 10^3 for DVM versus 10^1 for LBM). Nonetheless, simplified plasma models are still of great interest because of their simplicity and their straightforward compatibility with embedded boundary conditions.

In this paper we examine a minimal plasma model consisting of mass balance and field equations. We include embedded boundaries in the form of electrodes and dielectrics, and examine the interaction of streamers with such surfaces.

The structure of this paper is as follows: In Sec. 2 we review the minimal plasma model. We outline our numerical methods in Sec. 3 and provide numerical examples in Sec. 4. Some remarks regarding the minimal streamer model are given in Sec. 5.

2. Theory

The minimal plasma model consists of charge (or mass) conservation for the massive species (i.e. electrons, ions, and neutral molecules), the Poisson equation for the electric field, radiative transfer equations for photon transport, and a scalar conservation equation for the charge-balance on dielectric surfaces.

2.1. Electrodynamic equations

For a weakly ionized plasma such as a streamer, the streamer current is sufficiently small so that one may safely disregard magnetic field effects. The remaining equation for the evolution of the electromagnetic field is Poisson's equation:

$$\nabla \cdot (\epsilon \nabla \Phi) = -\frac{\rho}{\epsilon_0}, \quad (1)$$

where Φ is the electric potential, ϵ the relative permittivity, and ρ the charge density. The remaining Maxwell equation for the evolution of $\mathbf{E} = -\nabla \Phi$ is $\mathbf{J} + \epsilon_0 \frac{\partial \mathbf{E}}{\partial t} = 0$, but we only use this for estimating a time step size which ensures numerical stability during our simulations.

The boundary conditions that are available the Poisson equation are of the Dirichlet, Neumann, or Robin type. For the most part we use Dirichlet boundary conditions on electrodes and Neumann boundary conditions on domain faces (edges in 2D).

2.2. Plasma-fluid equations

The spatiotemporal evolution of electrons, ions, and neutrals is solved in the plasma-fluid approximation. We use the minimal plasma model which solves only for mass conservation (thus closing the system by computing drift velocities) for each species μ :

$$\frac{\partial n_\mu}{\partial t} + \nabla \cdot (\mathbf{v}_\mu n_\mu - D_\mu \nabla n_\mu) = S_\mu, \quad (2)$$

where n_μ is the volumetric density of species μ ; \mathbf{v}_μ its drift velocity, D_μ its diffusion coefficient, and S_μ a source term (describing the interplay between attachment, impact ionization, photoionization etc.). Equation (2) is a convection-diffusion-reaction (CDR) equation which describes the evolution of individual species densities.

Equation (2) is supplemented by boundary conditions describing the mass flux into or out of the computational domain. Our approach is to use outflow conditions on domain walls, and to expose the boundary conditions on embedded electrodes and dielectrics in an abstract plasma-kinetic framework which allows us to modify the surface kinetics of the plasma without affecting the underlying solver code.

2.3. Radiative transfer

Radiative transfer is handled by solving the radiative transfer equation (RTE) in the diffusive approximation. The RTE is

$$\frac{\partial f_\nu(\mathbf{x}, \boldsymbol{\Omega}, t)}{\partial t} + c\boldsymbol{\Omega} \cdot \nabla f_\nu(\mathbf{x}, \boldsymbol{\Omega}, t) = -c\kappa_\nu(\mathbf{x})f_\nu(\mathbf{x}, \boldsymbol{\Omega}, t) + \frac{1}{4\pi}\eta_\nu(\mathbf{x}), \quad (3)$$

where f_ν is the photon distribution function (i.e. the number of photon with frequency ν at (\mathbf{x}, t) traveling in direction $\boldsymbol{\Omega}$), κ_ν is the Beer's length for photons ν , and η_ν is an isotropic source term. Thus, ν_ν is the total number of photons produced at (\mathbf{x}, t) (thus the factor of 4π).

2.3.1. The multigroup approximation

In the RTE, the frequency ν is generally speaking a continuous variable. For most applications it becomes necessary to reduce the computational load by invoking the monochromatic multigroup approximation. That is, one assumes that f_ν consists of a number of frequency bands γ where each frequency band is sufficiently sharp-line in order to individually invoke a monochromatic approximation. That is, $f_\nu = \sum_\gamma f_\gamma \delta(\gamma)$ which essentially replaces Eq. (3) with a finite set of equations for each frequency band γ :

$$\frac{\partial f_\gamma(\mathbf{x}, \boldsymbol{\Omega}, t)}{\partial t} + c\boldsymbol{\Omega} \cdot \nabla f_\gamma(\mathbf{x}, \boldsymbol{\Omega}, t) = -c\kappa_\gamma(\mathbf{x})f_\gamma(\mathbf{x}, \boldsymbol{\Omega}, t) + \frac{1}{4\pi}\eta_\gamma(\mathbf{x}), \quad (4)$$

The multigroup RTE is solved in the diffusive SP₁ approximation (i.e. the *Eddington* approximation) by closing the first order moment equation. That is, taking the first and second moments of Eq. (3) with respect to $\boldsymbol{\Omega}$ yields

$$\frac{\partial E_\gamma}{\partial t} + c\nabla \cdot \mathbf{F}_\gamma = -c\kappa_\gamma E_\gamma + \eta_\gamma, \quad (5a)$$

$$\frac{\partial \mathbf{F}_\gamma}{\partial t} + c\nabla \cdot \boldsymbol{\Pi}_\gamma = -c\kappa_\gamma \mathbf{F}_\gamma, \quad (5b)$$

where $E_\gamma = \int_{4\pi} f_\gamma d\Omega$ is the radiative density, $\mathbf{F}_\gamma = \int_{4\pi} f_\gamma \boldsymbol{\Omega} d\Omega$ is the radiative flux, and $\Pi_\gamma^{ij} = \int_{4\pi} f_\gamma \Omega_i \Omega_j d\Omega$ is the Eddington tensor. This system is closed by assuming

$$\Pi_\gamma^{ij} = \frac{1}{3}\delta_{ij}E_\gamma, \quad (6)$$

which is equivalent to the Eddington approximation. Insertion of Eq. (6) into Eq. (5b) in the stationary approximation $\partial_t f_\gamma = 0$ yields the radiative flux

$$\mathbf{F}_\gamma = -\frac{1}{3\kappa_\gamma} \nabla E_\gamma, \quad (7)$$

which is what one expects from the diffusive approximation. Insertion of Eq. (7) into Eq. (5a) yields

$$\nabla \cdot \left(\frac{1}{3\kappa_\gamma} \nabla E_\gamma \right) - \kappa_\gamma E_\gamma = \frac{\eta_\gamma}{c}. \quad (8)$$

This equation must be supplemented by appropriate boundary conditions for radiative transport. In the Eddington approximation, the appropriate outflow boundary is of the Robin type:

$$\partial_n E_\gamma + \frac{3\kappa_\gamma}{2} \frac{1+3r_2}{1-2r_1} E_\gamma = \frac{g}{\kappa_\gamma}, \quad (9)$$

where r_1 , and r_2 are reflection coefficients and g is a surface source. For most of our applications we do not consider reflection of photons from boundaries, nor injection of photons into the domain, so we normally take $r_1 = r_2 = g = 0$ (although our implementation is not restricted to this). The outflow boundary condition on photons therefore simplifies to

$$\frac{\partial E_\gamma}{\partial n} + \frac{3\kappa_\gamma}{2} E_\gamma = 0. \quad (10)$$

2.4. Surface charge conservation

Our final equation of motion is local conservation of charge on *dielectric* surfaces. We do not solve an equivalent problem on electrodes because we assume that complete neutralization occurs on both anodes and cathodes, and that the voltage on the live electrodes are maintained by an external circuit. Charge conservation on dielectric surfaces is given by:

$$\frac{\partial \sigma}{\partial t} = F_\sigma, \quad (11)$$

where F_σ is the charge flux out of the surface and into the gas phase. F_σ is, generally speaking, coupled to the boundary conditions for the species densities n_μ since charge must be conserved at the surface. However, additional physical phenomena, e.g. Schottky emission, can easily be added to Eq. (11).

2.5. Minimal plasma model

To summarize, our plasma model consists of the following equations:

$$\nabla \cdot (\epsilon_r \nabla) \phi = -\frac{\rho}{\epsilon_0}, \quad (12a)$$

$$\frac{\partial \sigma}{\partial t} = F_\sigma, \quad (12b)$$

$$\nabla \cdot \left(\frac{1}{3\kappa_\gamma} \nabla E_\gamma \right) - \gamma \nabla E_\gamma = -\frac{\eta_\gamma}{c}, \quad (12c)$$

$$\frac{\partial n_\mu}{\partial t} + \nabla \cdot (\mathbf{v}_\mu n_\mu) - \nabla \cdot (D_\mu \nabla n_\mu) = S_\mu, \quad (12d)$$

where the final two equations denote *sets* of equations for μ and γ .

For streamer discharges Eq. (12) is nonlinearly coupled through source terms and species velocities. Our approach has been to develop a code framework that exposes only the *coupling* rather than the actual solvers, so that the user can specify his own bulk and surface plasma kinetics within the minimal model.

3. Numerical method

Extensive experimental and computational efforts have long since established that a streamer channel is a very narrow, slightly conductive channel with fine spatial features that evolves on multiple spatial scales. Based on these experiences, our efforts have been guided by the need for adaptive resolution, efficient remeshing, and well-tested scalability of the underlying data structures. We have therefore chosen to implement Eq. (12) by using the Chombo library, which supplies the parallelized AMR infrastructure on cut-cell Cartesian grids. The use of cut-cells introduces some additional complexity into the discretization methods since the equations must be specially treated in the cut-cells, but nonetheless support the inclusion of internal boundaries inside the computational domain.

3.1. Spatial discretization

Here, we give a brief summary to the spatial discretization. In general, we discretize the physical domain by a set of Cartesian control volumes centered on points $\mathbf{i} = (i_0, \dots, i_{D-1})^\top \in \mathbb{Z}^D$ where D is the dimension (typically, $D = 2$ or $D = 3$). Each control volume occupies the region $[\mathbf{i} - \frac{1}{2}\mathbf{u}\Delta x, \mathbf{i} + \frac{1}{2}\mathbf{u}\Delta x]$ where Δx is a resolution and \mathbf{u} is a vector whose components are all one (i.e. $\mathbf{u} = (1, 1)^\top$ in 2D). Thus, a mesh $\Gamma \subset \mathbb{Z}^D$ simply consists of a set of connected cells $\mathbf{i} \in \Gamma$.

Adaptive Mesh Refinement (AMR) grids are used by allowing overlapping grids with varying resolutions. We work with a set of grids Γ^l that are nested hierarchically from the coarsest grid to the finest grid. Thus, we always require that a grid Γ^l covers the overlapping region of a coarser grid Γ^{l-1} and a finer grid Γ^{l+1} . In general, the use of AMR grids for streamer simulations is advantageous due to need for frequent regridding the solution; the use of a static mesh requires one to fully resolve the entire spatial region at the finest length

scale, which leads to excessive resource requirements. Moreover, the computational overhead for remeshing a spatial region using Cartesian AMR is usually negligible (unlike Delaunay triangulation and least-squares reconstruction, for example).

Internal boundary conditions are resolved by cutting some of the cells with a level-set function $s(\mathbf{x}) = 0$ which describes the boundary interface. Inside each cut-cell, the boundary is linearized so that it intersects through the cell as a line in 2D, and a plane in 3D. Control volume faces that are cut by this surface are referred to as cut-faces, and the boundary itself is referred to as an embedded boundary (EB).

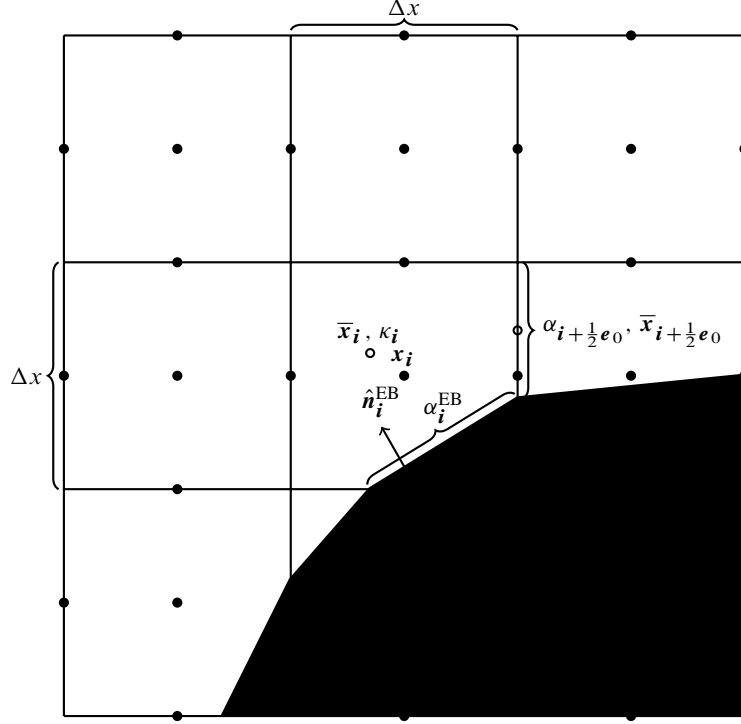


Figure 1: Cut-cell discretization.

Figure 1 shows the grid hierarchy and the cut cells in greater details. Here, a cell is identified by its index \mathbf{i} . We will take $\mathbf{x}_{\mathbf{i}}$ to be the *cell center*, and $\bar{\mathbf{x}}_{\mathbf{i}}$ to be the *cell centroid*. The volume fraction is $0 \leq \kappa_{\mathbf{i}} \leq 1$: Cells with $\kappa_{\mathbf{i}} = 0$ are denoted as *covered* cells and lie completely outside the computational domain; cells with volume fractions $0 < \kappa_{\mathbf{i}} < 1$ are called *irregular* cells and lie partially inside the computational domain (see e.g. the middle cell in Fig. 1). Furthermore, cell faces are denoted by $f_d^{\pm}(\mathbf{i})$ where \pm indicates the high (+) or low (−) face of the cell \mathbf{i} in the coordinate direction d . A face center for face $f_d^{\pm}(\mathbf{i})$ is therefore located at $\mathbf{x}_{\mathbf{i} \pm \frac{1}{2} \mathbf{e}_d}$ where \mathbf{e}_d is a unit vector in the d -direction. Face centroids are likewise denoted $\bar{\mathbf{x}}_{\mathbf{i} \pm \frac{1}{2} \mathbf{e}_d}$ and the corresponding area fractions are denoted by $\alpha_{\mathbf{i} \pm \frac{1}{2} \mathbf{e}_d}$. Finally, the EB centroid is denoted by $\bar{\mathbf{x}}_{\mathbf{i}}^{\text{EB}}$ with a corresponding area fraction $\alpha_{\mathbf{i}}^{\text{EB}}$ and an outward normal vector $\mathbf{n}_{\mathbf{i}}$.

To derive the above quantities, let $V_{\mathbf{i}}$ denote the part of the Cartesian cell inside the computational domain, $A_{\mathbf{i}}^{\pm, d}$ be the un-cut part of a face $f_d^{\pm}(\mathbf{i})$, and $A_{\mathbf{i}}^{\text{EB}}$ be the part of the EB that intersects with the

cell. The above quantities are given by

$$\bar{\mathbf{x}}_{\mathbf{i}} = \frac{1}{\Delta x^D} \int_{V_{\mathbf{i}}} \mathbf{x} dV_{\mathbf{i}}, \quad (13a)$$

$$\bar{\mathbf{x}}_{\mathbf{i} \pm \frac{1}{2} \mathbf{e}_d} = \frac{1}{\Delta x^{D-1}} \int_{A_{\mathbf{i}}^{\pm, d}} \mathbf{x} dA_{\mathbf{i}}^{\pm, d}, \quad (13b)$$

$$\bar{\mathbf{x}}_{\mathbf{i}}^{\text{EB}} = \frac{1}{\Delta x^{D-1}} \int_{A_{\mathbf{i}}^{\text{EB}}} \mathbf{x} dA_{\mathbf{i}}^{\text{EB}}, \quad (13c)$$

$$\kappa_{\mathbf{i}} = \frac{1}{\Delta x^D} \int_{V_{\mathbf{i}}} dV_{\mathbf{i}}, \quad (13d)$$

$$\alpha_{\mathbf{i} \pm \frac{1}{2} \mathbf{e}_d} = \frac{1}{\Delta x^{D-1}} \int_{A_{\mathbf{i}}^{\pm, d}} \mathbf{x} dA_{\mathbf{i}}^{\pm, d}, \quad (13e)$$

$$\alpha_{\mathbf{i}}^{\text{EB}} = \frac{1}{\Delta x^{D-1}} \int_{A_{\mathbf{i}}^{\text{EB}}} dA_{\mathbf{i}}^{\text{EB}}, \quad (13f)$$

$$\mathbf{n}_{\mathbf{i}} = \frac{1}{\Delta x^{D-1}} \int_{A_{\mathbf{i}}^{\text{EB}}} \mathbf{n} dA_{\mathbf{i}}^{\text{EB}}. \quad (13g)$$

For brevity's sake, we will from now on omit the dependency on \mathbf{i} , and re-specify this when necessary.

3.2. Poisson & Eddington equations

We use the finite volume method for the Poisson equation and Eddington equations. Since the Eddington equations contain an additional diagonal term, it is sufficient to restrict our discussion to the Poisson equation.

Integrating the Poisson equation over a cell volume $V_{\mathbf{i}}$ yields

$$\frac{1}{\kappa_{\mathbf{i}} \Delta x} \left[\sum_{f \in f_d^+(\mathbf{i})} \alpha_f \epsilon_f g_f^d - \sum_{f \in f_d^-(\mathbf{i})} \alpha_f \epsilon_f g_f^d - \alpha_{\mathbf{i}}^{\text{EB}} \epsilon_{\mathbf{i}}^{\text{EB}} (\mathbf{g}_{\mathbf{i}}^{\text{EB}} \cdot \mathbf{n}_{\mathbf{i}}) \right] = \rho_{\mathbf{i}} + \sigma_{\mathbf{i}}, \quad (14)$$

where g_f^d is the flux through the face f . This is discretized by using second-order approximations for the fluxes g_f^d through each cell face. Note that each Cartesian face may consist of several sub-faces. For example, cells that are cut by the dielectric surface will contain two faces; one of the gas phase and one for the solid phase. This is strictly speaking not the case for faces that are cut by conducting surfaces because there is no flux passing into the volume through the part of the face that lies within the conductor. Note also that in Eq. (14), the embedded boundary is only the conductor surfaces. The contribution of the EB fluxes for the dielectric are contained entirely within $\sigma_{\mathbf{i}}$, thus avoid the need for specialized boundary condition stencils for such cells.

For un-cut faces we discretize using

$$g_f^d = \frac{1}{\Delta x} (\phi_{\mathbf{i}^+} - \phi_{\mathbf{i}^-}), \quad (15)$$

where \mathbf{i}^{\pm} denotes the control volumes on the high (+) and low (−) sides of the face f . In order to maintain a second order scheme, cut-face fluxes are interpolated to their respective centroids

3.2.1. Multigrid solver

The discretized Poisson equation is solved by embedding it in a geometric multigrid solver in residual-correction form. Multigrid involves iterative relaxation on progressively coarsened grids and is also compatible with AMR. On the coarsest level, the discretized system is solved directly with a biconjugate gradient stabilized method. Typically, better convergence for each cycle is achieved the further one coarsens; for

domains without embedded boundaries one may coarsen up to a level where the domain only consists of two cells in one of the coordinate direction. However, for embedded boundary applications the convergence rate is usually harmed by coarsening too far because the coarsened stencils near embedded boundary may no longer be a good approximation of the finer stencils. Moreover, over-coarsening may also lead to geometrical under-resolution. We avoid both of these issues by dropping to the bottom solver before the convergence rate drops.

3.2.2. Boundary conditions

For the Poisson equation we use either Dirichlet or Neumann boundary conditions on the domain edges, and Dirichlet on the conductors. Neumann boundary conditions are straightforward to implement since they specify the boundary flux directly, whereas Dirichlet is more involved. To compute Dirichlet boundary conditions we use the Johansen-Colella method by casting a ray from the EB centroid along the EB normal into the computational domain. In two dimensions, this ray will cross a coordinate line connecting two cell-centers, and the interpolated value at this point may be used to approximate the derivative $\partial_n \phi$ on the EB centroid by polynomial interpolation and differentiation. In practice, we use a second order stencil if we can, but we remark that even with a first order stencil the scheme is second order due to the residual-correction form of the multigrid solver.

4. Numerical examples

5. Concluding remarks

Finally, we wish to provide a few concluding remarks regarding streamer simulations. In our opinion, we anticipate that there is much to be gained by community-driven improvements of the radiative transfer in streamers. Firstly, let us remark that the stationary simplification that is frequently used for radiative transport codes in streamer simulations is only weakly applicable, and certainly not valid for atmospheric discharge which take place on length scales of hundreds or even thousands of meters. In addition, all diffusion-based RTE methods neglect even qualitative features such as geometrical shading, and we broadly believe that improved radiative transport codes should be implemented into existing streamer code frameworks. Since direct integration of the RTE is unpractical in three dimensions, the obvious candidates for RTE simplification are the discrete ordinates method (DOM) and moment-based RTE. Discrete ordinates essentially solve the RTE by sampling the propagation angles Ω on a finite angular grid and can be implemented straightforwardly using existing CDR framework since the angularly discretized RTE is essentially, a set of advection equations. Unfortunately, the DOM is likely to become a computational bottleneck in both CPU and memory since the numerical sampling of angular space must be large in order avoid numerical artifacts. We therefore suggest that the community investigates moment-based RTE with improved closure relations are. In this case, the RTE reduces to three (four) hyperbolic conservation equations in 2D (3D) for each frequency group. For causal reasons those equations should be solved using higher order methods, such as those of Godunov.