An Adaptive Moving Mesh Discontinuous Galerkin Method for the Radiative Transfer Equation

Min Zhang[1], Juan Cheng[2,3], Weizhang Huang[4] and Jianxian Qiu[5,*]

[1] *School of Mathematical Sciences, Xiamen University, Xiamen, Fujian 361005, China.*

[2] *Laboratory of Computational Physics, Institute of Applied Physics and Computational Mathematics, Beijing 100088, China.*

[3] *Center for Applied Physics and Technology, Peking University, Beijing 100871, China.*

[4] *Department of Mathematics, University of Kansas, Lawrence, Kansas 66045, USA.*

[5] *School of Mathematical Sciences and Fujian Provincial Key Laboratory of Mathematical Modeling and High-Performance Scientific Computing, Xiamen University, Xiamen, Fujian 361005, China.*

Abstract. The radiative transfer equation models the interaction of radiation with scattering and absorbing media and has important applications in various fields in science and engineering. It is an integro-differential equation involving time, frequency, space and angular variables and contains an integral term in angular directions while being hyperbolic in space. The challenges for its numerical solution include the needs to handle with its high dimensionality, the presence of the integral term, and the development of discontinuities and sharp layers in its solution along spatial directions. Its numerical solution is studied in this paper using an adaptive moving mesh discontinuous Galerkin method for spatial discretization together with the discrete ordinate method for angular discretization. The former employs a dynamic mesh adaptation strategy based on moving mesh partial differential equations to improve computational accuracy and efficiency. Its mesh adaptation ability, accuracy, and efficiency are demonstrated in a selection of one- and two-dimensional numerical examples.

Corresponding author. Email addresses: `minzhang2015@stu.xmu.edu.cn` (M. Zhang), `cheng_juan@iapcm.ac.cn` (J. Cheng), `whuang@ku.edu` (W. Huang), `jxqiu@xmu.edu.cn` (J. Qiu)

1 Introduction

The radiative transfer equation (RTE) models the interaction of radiation with scattering and absorbing media, which has important applications in fields such as astrophysics, high energy density physics, nuclear physics, inertial confinement fusion, heat transfer, stellar atmospheres, optical molecular imaging, infrared and visible light in space and the atmosphere, and biomedicine. The RTE is an integro-differential equation with seven independent variables (i.e., time, frequency, space, and angles) for a time-dependent three-spatial-dimensional problem. Containing an integral term and with its high dimensionality, the RTE presents a challenge in the development of efficient numerical algorithms. On the other hand, the efficient solution of the RTE plays an important role in the study of radiation hydrodynamics where the RTE is often coupled with the Euler equations, the energy equation, and the equation of state.

In the past, a number of methods have been developed for the numerical solution of the RTE. Those methods can be divided roughly into two categories: stochastic and deterministic approaches. The Monte Carlo method is a widely used method in the former category [24,31]. On the other hand, deterministic approaches involve discrete approximations of the variables in the RTE. In particular, the discretization needs to be applied to all coordinates in space and angles. For angular coordinates, the P_N method, first introduced in [25] and also known as the spherical harmonics method, uses an orthogonal, harmonic basis to approximate the solution. Another approach called the discrete ordinate method (DOM) [6,26] employs spectral collocation and the Legendre-Chebyshev quadrature to discretize the integro-differential equation in angular coordinates. DOM is widely used for the numerical solution of the transport equation [28,35] due to its high accuracy, flexibility, and relatively low computational cost. The angle-discretized RTE forms a system of linear hyperbolic equations with a numerical integral term, which can be discretized in space using a standard method such as a finite difference, finite volume, or finite element method. The discontinuous Galerkin (DG) method is employed for this purpose in the current work.

The DG method is known to be a particularly powerful numerical tool for the simulation of hyperbolic transport problems. It was first used for the RTE by Reed and Hill [36] and theoretically studied by Lesaint and Raviart [27]. The method was later extended to nonlinear conservation laws by Cockburn and Shu [7–10]. The DG method has the advantages of high-order accuracy, geometric flexibility, suitability for handling *h*- and *p*-adaptivity, extremely local data structure, high parallel efficiency, and a good theoretical foundation for stability and error estimates. Over the last few decades, the DG method has been used widely in scientific and engineering computation.

The objective of the current work is to study an adaptive moving mesh DG method (for spatial discretization) combined with DOM (for angular discretization) for the numerical solution of the RTE. Due to its hyperbolic nature, the solution of the RTE can develop discontinuities or sharp layers along spatial directions, which makes mesh adaptation an indispensable tool for use in improving computational accuracy and efficiency.

Mesh adaptation methods can be classified roughly into three groups. The first one is *h*-methods, which generate a new mesh by adding or removing points to an existing mesh. Typically, mesh points are added in regions where the solution variation or error is large, and mesh points are removed in regions where the solution is smooth. The second group is *p*-methods where the order of polynomial approximation varies from place to place according to a certain error estimate or indicator. The third group is *r*-methods, also called moving mesh methods, which relocate mesh point positions while maintaining the total number of mesh points and the mesh connectivity.

Several works have been done in using mesh adaptation for the numerical solution of the RTE. In [22], an adaptive mesh refinement (AMR) algorithm has been formulated and implemented for the RTE by minimizing the spatial discretization error. In [13], *hp*-adaptive DG methods have been developed for the numerical solution of simplified P_N approximations of radiative transfer in non-grey semitransparent media and it has been found that it is possible to approximate the radiative field with a significantly lower computational cost than solving the equations using the conventional finite element method. In [12], *hp*-adaptive methods have been developed based on a short-characteristics approach embedded in the discontinuous finite element framework. No work has been done so far in using moving mesh methods for the numerical solution of the RTE.

In this paper, we study a moving mesh method based on moving mesh partial differential equations (MMPDEs) [18–20] to solve the radiative transfer equation. An MMPDE moves the mesh continuously in time and orderly in space and is formulated as the gradient flow equation of a meshing functional. We use a newly developed discrete approach [16] that makes the implementation of the MMPDE method not only significantly simpler but also much more reliable in the sense that there is a theoretical guarantee for mesh nonsingularity at semi and fully discrete levels [17]. The MMPDE method determines the mesh adaptivity for the size, shape, and orientation of mesh elements using a metric tensor (also called a monitor function) which is matrix-valued function defined throughout the physical domain.

The full discretization of the RTE includes the discretization in angular directions using DOM, in space using an adaptive moving mesh DG method [29], and in time using the backward Euler scheme. The DOM discretization leads to radiative intensity functions for different angular directions which can have discontinuities or sharp layers at different location in space. To take this into account, we compute a metric tensor based on the Hessian for each of these functions and then combine them via matrix intersection (see its definition in Section 4) into a single metric tensor to be used with the MMPDE method. Numerical results in Section 5 show that the adaptive moving mesh DG method with this strategy works well for problems with single or multiple sharp layers and discontinuities in the sense that it is able to automatically concentrate the mesh points in the regions of discontinuities or steep transition layers and is more efficient than its fixed mesh counterpart.

An outline of the paper is as follows. In Section 2, the RTE and DOM are described. The adaptive moving mesh DG method for solving the two-dimensional RTE is presented

in Section 3. The generation of adaptive moving meshes using a new implementation of the MMPDE method is discussed in Section 4. A selection of one- and two-dimensional examples are presented in Section 5 to demonstrate the mesh adaptation ability of the adaptive moving mesh DG method and its accuracy and efficiency. Finally, Section 6 contains conclusions and further comments.

2   The radiative transfer equation

The RTE is an integro-differential equation modeling the conservation of photons [32]. We consider a case with one-group, isotropically scattering radiative transfer. The governing equation for this case reads as

$$\frac{1}{c}\frac{\partial I(\boldsymbol{r},\Omega,t)}{\partial t}+\Omega\cdot\nabla I(\boldsymbol{r},\Omega,t)+\sigma_t I(\boldsymbol{r},\Omega,t)=\frac{\sigma_s}{4\pi}\int_S I(\boldsymbol{r},\tilde{\Omega},t)d\tilde{\Omega}+q(\boldsymbol{r},\Omega,t), \qquad (2.1)$$

where c is the speed of photons, $\boldsymbol{r}$ is the spatial variable, ∇ is the gradient operator with respect to $\boldsymbol{r}$, Ω is the unit angular variable, S is the unit sphere, t is time, $I(\boldsymbol{r},\Omega,t)$ is the radiative intensity in the direction Ω, $\sigma_s \geq 0$ is the scattering coefficient of the medium, $\sigma_t \geq \sigma_s$ is the extinction coefficient of the medium due to both absorption and scattering, and $q(\boldsymbol{r},\Omega,t)$ is a given source term. The vector $\boldsymbol{r}$ is described by the Cartesian coordinates x,y,z while Ω is usually described by a polar angle β measured with respect to a fixed axis in space (such as the z axis) and a corresponding azimuthal angle φ. Letting $\mu=\cos\beta$, then

$$d\boldsymbol{r}=dxdydz, \quad d\Omega=\sin\beta d\beta d\varphi=-d\mu d\varphi.$$

In this work we consider the numerical solution of (2.1) in one and two spatial dimensions. Since the numerical algorithm is similar in one and two dimensions, we describe it only in two dimensions. The equation (2.1) reads in two dimensions as

$$\frac{1}{c}\frac{\partial I(x,y,\Omega,t)}{\partial t}+\Omega\cdot\nabla I(x,y,\Omega,t)+\sigma_t I(x,y,\Omega,t)$$
$$=\frac{\sigma_s}{4\pi}\int_S I(x,y,\tilde{\Omega},t)d\tilde{\Omega}+q(x,y,\Omega,t), \quad (x,y)\in\mathbb{D}, \qquad (2.2)$$

where $\Omega=(\zeta,\eta)$ and

$$\zeta=\sin\beta\cos\varphi=\sqrt{1-\mu^2}\cos\varphi\in(-1,1), \quad \eta=\sin\beta\sin\varphi=\sqrt{1-\mu^2}\sin\varphi\in(-1,1).$$

Denote by $\boldsymbol{n}(x,y)$ the unit outward normal vector of the domain boundary $\partial\mathbb{D}$ at the point (x,y) and define $\partial\mathbb{D}_{in}=\{(x,y)\in\partial\mathbb{D}\,|\,\boldsymbol{n}(x,y)\cdot\Omega<0\}$. Then, the boundary condition can be expressed as

$$I(x,y,\Omega,t)=g(x,y,\Omega,t), \quad (x,y)\in\partial\mathbb{D}_{in}, \qquad (2.3)$$

and the initial condition is

$$I(x,y,\Omega,0)=I_0(x,y,\Omega), \quad (x,y)\in\mathbb{D}, \qquad (2.4)$$

where $g(x,y,\Omega,t)$ and $I_0(x,y,\Omega)$ are given functions. Note that no boundary condition is needed in Ω directions.

The RTE (2.2) needs to be discretized in angular, spatial, and time variables. A challenge for solving (2.2) is due to its high dimensionality: it has two dimensions totally in both angular and spatial coordinates for a one-spatial-dimensional problem and four dimensions for a two-spatial-dimensional problem. To tackle this challenge, a common strategy is to use a high accuracy discretization in angular coordinates. We use the discrete-ordinate method [6] for this purpose. DOM will be described later in this section. Another challenge for solving (2.2) is due to the fact that it is hyperbolic in space. This means that its solution can develop discontinuities and sharp layers across the physical domain, which requires a spatial discretization that can handle those structures and mesh adaptation that can provide high resolution in regions around them. An adaptive moving mesh DG method to be presented in the next section will be used to tackle this challenge.

DOM [6] is a spectral collocation-type method [5]. Indeed, (2.2) is collocated for a finite number of angular directions while the integral in the angular variable is approximated by the Legendre-Chebyshev quadrature [26]. The collocation points in ζ and η are given by

$$\zeta_m = \sqrt{1-\mu_i^2}\cos\varphi_j, \quad \eta_m = \sqrt{1-\mu_i^2}\sin\varphi_j, \quad m=(i-1)N_c+j,$$

where μ_i, $i=1,\cdots,N_l$ denote the roots of the Legendre polynomial of degree N_l and $\varphi_j = (2j-1)/N_c$, $j=1,\cdots,N_c$ are the nodes based on a Chebyshev polynomial. Specifically, the discrete-ordinate approximation of (2.2) is given by

$$\frac{1}{c}\frac{\partial I_m(x,y,t)}{\partial t}+\Omega_m\cdot\nabla I_m(x,y,t)+\sigma_t I_m(x,y,t)$$
$$=\sigma_s\sum_{m'=1}^{N_a} w_{m'}I_{m'}(x,y,t)+q_m(x,y,t), \quad m=1,\cdots,N_a, \tag{2.5}$$

where $\Omega_m = (\zeta_m,\eta_m)$, $m=1,\cdots,N_a \equiv N_lN_c$, are the discrete angular directions, $I_m(x,y,t)$ is an approximation of $I(x,y,\Omega_m,t)$, $q_m(x,y,t) = q(x,y,\Omega_m,t)$, and $\sum_{m=1}^{N_a} w_m I_m(x,y,t)$ is a Legendre-Chebyshev quadrature rule with weights $w_m > 0$ for $(1/4\pi)\int_S I(x,y,\tilde{\Omega},t)d\tilde{\Omega}$.

3   An adaptive moving mesh DG method for the two-dimensional DOM RTEs

We notice that (2.5) is a system of hyperbolic equations and its solution can develop discontinuities and sharp layers. The DG method has been known to be a powerful numerical tool for the simulation of hyperbolic problems with discontinuous solutions [27,36]. Mesh adaptation is also crucial to provide accurate resolution of discontinuities and sharp layers in the solution and improve computational efficiency. We describe a

DG method for (2.5) on a general adaptive moving mesh in this section and the adaptive mesh movement in the next section.

Specifically, we consider time instants

$$t_0 = 0 < t_1 < \cdots < t_n < t_{n+1} < \cdots.$$

For the moment, we assume that a triangular mesh, which consists of non-overlapping triangles covering $\mathbb{D}$ completely and whose vertices depend on t, is known at the time instants, i.e., $\mathscr{T}_h^n$, $n = 0,1,\cdots$, are given. We also assume that the mesh keeps the same connectivity and the same number of elements and vertices for the whole time period. (The position of the vertices is the only thing that changes with time.) The generation of such a moving mesh is discussed in Section 4.

For $t \in [t_n, t_{n+1}]$, the coordinates and velocities of the vertices of the mesh are defined as

$$
\begin{aligned}
x_j(t) &= x_j^n \frac{t_{n+1}-t}{\Delta t_n} + x_j^{n+1} \frac{t-t_n}{\Delta t_n}, \quad \dot{x}_j(t) = \frac{x_j^{n+1}-x_j^n}{\Delta t_n}, \quad j = 1,\cdots,N_v, \\
y_j(t) &= y_j^n \frac{t_{n+1}-t}{\Delta t_n} + y_j^{n+1} \frac{t-t_n}{\Delta t_n}, \quad \dot{y}_j(t) = \frac{y_j^{n+1}-y_j^n}{\Delta t_n}, \quad j = 1,\cdots,N_v,
\end{aligned}
\tag{3.1}
$$

where N_v is the number of the vertices and $\Delta t_n = t_{n+1} - t_n$. The corresponding mesh is denoted by $\mathscr{T}_h(t)$.

We now describe the DG discretization of (2.5) on $\mathscr{T}_h(t)$. For any element $K \in \mathscr{T}_h(t)$, denote its vertices by (x_1^K, y_1^K), (x_2^K, y_2^K), (x_3^K, y_3^K) and its area by $|K|$. Consider a set of local orthogonal polynomials of up to degree k in K,

$$P^k(K,t) = \text{span}\{\phi_0^{(K)}(x,y,t), \phi_1^{(K)}(x,y,t), \cdots, \phi_{L-1}^{(K)}(x,y,t)\}, \tag{3.2}$$

where $L = (k+1)(k+2)/2$ is the dimension of $P^k(K,t)$ and $\{\phi_p^{(K)}, p=0,1,\cdots,L-1\}$ are the orthogonal basis functions over the element K. The first three of the basis functions read as

$$
\begin{aligned}
\phi_0^{(K)}(x,y,t) &= 1, \\
\phi_1^{(K)}(x,y,t) &= \sqrt{2}\left(\hat{x} - \frac{1}{3}\right), \\
\phi_2^{(K)}(x,y,t) &= \frac{\sqrt{2}}{2}\left(\hat{x} - \frac{1}{3}\right) + \sqrt{2}\left(\hat{y} - \frac{1}{3}\right),
\end{aligned}
$$

where

$$
\begin{bmatrix} \hat{x} \\ \hat{y} \end{bmatrix} = \frac{1}{2|K|} \begin{bmatrix} (y_3^K - y_1^K) & -(x_3^K - x_1^K) \\ -(y_2^K - y_1^K) & (x_2^K - x_1^K) \end{bmatrix} \begin{bmatrix} x - x_1^K \\ y - y_1^K \end{bmatrix}.
$$

Then the associated DG finite element space can be defined as

$$V_h^k(t) = \{I_m^h(x,y,t) \in L^2(\mathbb{D}) : I_m^h(x,y,t)|_K \in P^k(K,t), \ \forall K \in \mathscr{T}_h(t)\}. \tag{3.3}$$

Notice that any function in this space can be expressed as

$$I_m^h(x,y,t) = \sum_{p=0}^{L-1} I_{m,K}^{[p]}(t)\phi_p^{(K)}(x,y,t), \quad (x,y) \in K, \tag{3.4}$$

where $I_{m,K}^{[p]}$, $p=0,\cdots,L-1$, are the degrees of freedom. Moreover, its time derivative can be written as

$$\frac{\partial I_m^h(x,y,t)}{\partial t} = \sum_{p=0}^{L-1} \left(\frac{dI_{m,K}^{[p]}(t)}{dt}\phi_p^{(K)}(x,y,t) + I_{m,K}^{[p]}(t)\frac{\partial\phi_p^{(K)}(x,y,t)}{\partial t} \right). \tag{3.5}$$

It is not difficult (e.g., see [23]) to show that

$$\frac{\partial\phi_p^{(K)}(x,y,t)}{\partial t} = -\Pi_1(x,y,t)\cdot\nabla\phi_p^{(K)}(x,y,t), \tag{3.6}$$

where $\Pi_1(x,y,t) = (\dot{X},\dot{Y})$ is the piecewise linear interpolation of the nodal mesh velocities, i.e.,

$$\dot{X} = \frac{1}{3}(\dot{x}_1^K + \dot{x}_2^K + \dot{x}_3^K)\phi_0^{(K)} - \frac{\sqrt{2}}{4}(\dot{x}_1^K - 2\dot{x}_2^K + \dot{x}_3^K)\phi_1^{(K)} - \frac{\sqrt{2}}{2}(\dot{x}_1^K - \dot{x}_3^K)\phi_2^{(K)},$$
$$\dot{Y} = \frac{1}{3}(\dot{y}_1^K + \dot{y}_2^K + \dot{y}_3^K)\phi_0^{(K)} - \frac{\sqrt{2}}{4}(\dot{y}_1^K - 2\dot{y}_2^K + \dot{y}_3^K)\phi_1^{(K)} - \frac{\sqrt{2}}{2}(\dot{y}_1^K - \dot{y}_3^K)\phi_2^{(K)}. \tag{3.7}$$

Combining (3.5) and (3.6), we get

$$\frac{\partial I_m^h(x,y,t)}{\partial t} = \sum_{p=0}^{L-1} \frac{dI_{m,K}^{[p]}(t)}{dt}\phi_p^{(K)}(x,y,t) - \Pi_1(x,y,t)\cdot\nabla I_m^h(x,y,t), \quad (x,y) \in K. \tag{3.8}$$

Multiplying (2.5) by a test function $\phi(x,y,t) \in V_h^k(t)$, integrating the resulting equation over K, replacing $I_m(x,y,t)$ with its approximation $I_m^h(x,y,t)$, and using (3.8), we have

$$\int_K \left(\frac{1}{c}\sum_{p=0}^{L-1} \frac{dI_{m,K}^{[p]}(t)}{dt}\phi_p^{(K)} \right)\phi\,dxdy + \int_K \sigma_t I_m^h(x,y,t)\phi\,dxdy$$
$$+ \int_K \left(\Omega_m - \frac{1}{c}\Pi_1(x,y,t) \right)\cdot\nabla I_m^h(x,y,t)\,\phi\,dxdy$$
$$= \int_K \sigma_s\Psi_K(x,y,t)\phi\,dxdy + \int_K q_m(x,y,t)\phi\,dxdy, \tag{3.9}$$

where

$$\Psi_K(x,y,t) = \sum_{m=1}^{N_a} w_m I_{m,K}^h(x,y,t)$$

and $I_{m,K}^h(x,y,t)$ denotes the restriction of $I_m^h(x,y,t)$ on K. Applying the divergence theorem on the third term, we get

$$
\int_K \left(\frac{1}{c} \sum_{p=0}^{L-1} \frac{dI_{m,K}^{[p]}(t)}{dt} \phi_p^{(K)} \right) \phi\, dxdy + \int_K \sigma_t I_m^h(x,y,t) \phi\, dxdy
$$
$$
- \int_K \left(\nabla \cdot \left(\phi \left(\Omega_m - \frac{1}{c} \Pi_1(x,y,t) \right) \right) \right) I_m^h(x,y,t)) dxdy
$$
$$
+ \int_{\partial K} \mathbf{n}_K \cdot \left(\Omega_m - \frac{1}{c} \Pi_1(x,y,t) \right) I_m^h(x,y,t) \phi\, ds
$$
$$
= \int_K \sigma_s \Psi_K(x,y,t) \phi\, dxdy + \int_K q_m(x,y,t) \phi\, dxdy, \tag{3.10}
$$

where $\mathbf{n}_K$ is the outward unit normal to the boundary ∂K. In the above equation, $I_m^h(x,y,t)$ is discontinuous across the cell boundaries in general and its value thereon is not well defined. To specify the value, we define the outflow boundary ∂K^{m+} and the inflow boundary ∂K^{m-} of the element K by

$$
\partial K^{m+} = \left\{ (x,y) \in \partial K \,\middle|\, \left(\Omega_m - \frac{1}{c} \Pi_1(x,y,t) \right) \cdot \mathbf{n}_K(x,y,t) \geq 0 \right\},
$$
$$
\partial K^{m-} = \left\{ (x,y) \in \partial K \,\middle|\, \left(\Omega_m - \frac{1}{c} \Pi_1(x,y,t) \right) \cdot \mathbf{n}_K(x,y,t) < 0 \right\}. \tag{3.11}
$$

It is useful to point out that, in practice, $(1/c)\Pi_1(x,y,t)$ is much smaller than Ω_m for most situations and ∂K^{m+} and ∂K^{m-} can be computed using the simpler formulas

$$
\partial K^{m+} = \{ (x,y) \in \partial K \,|\, \Omega_m \cdot \mathbf{n}_K(x,y,t) \geq 0 \},
$$
$$
\partial K^{m-} = \{ (x,y) \in \partial K \,|\, \Omega_m \cdot \mathbf{n}_K(x,y,t) < 0 \}. \tag{3.12}
$$

Since each interior edge is shared by two triangular elements, the value of $I_m^h(x,y,t)$ on any edge of K can be defined based on its value on K or on the other element sharing the common edge with K. These values are denoted by $I_m^h(int(K),t))$ and $I_m^h(ext(K),t)$, respectively. For the upwind numerical flux, we use $I_m^h(int(K),t)$ for the outflow boundary and $I_m^h(ext(K),t)$ for the inflow boundary. Thus, we can rewrite (3.10) into

$$
\int_K \left(\frac{1}{c} \sum_{p=0}^{L-1} \frac{dI_{m,K}^{[p]}(t)}{dt} \phi_p^{(K)} \right) \phi\, dxdy + \int_K \sigma_t I_m^h(x,y,t) \phi\, dxdy
$$
$$
- \int_K \left(\nabla \cdot \left(\phi \left(\Omega_m - \frac{1}{c} \Pi_1(x,y,t) \right) \right) \right) I_m^h(x,y,t) dxdy
$$
$$
+ \int_{\partial K^{m-}} \mathbf{n}_K \cdot \left(\Omega_m - \frac{1}{c} \Pi_1(x,y,t) \right) I_m^h(ext(K),t) \phi\, ds
$$
$$
+ \int_{\partial K^{m+}} \mathbf{n}_K \cdot \left(\Omega_m - \frac{1}{c} \Pi_1(x,y,t) \right) I_m^h(int(K),t) \phi\, ds
$$
$$
= \int_K \sigma_s \Psi_K(x,y,t) \phi\, dxdy + \int_K q_m(x,y,t) \phi\, dxdy. \tag{3.13}
$$

Explicit time stepping can cause extremely small time steps due to the high photon speed. To avoid this difficulty, we use the backward Euler scheme for (3.13), i.e.,

$$
\int_K \left(\frac{1}{c} \sum_{p=0}^{L-1} \frac{I_{m,K}^{[p]}(t_{n+1}) - I_{m,K}^{[p]}(t_n)}{\Delta t_n} \phi_p^{(K)} \right) \phi \, dxdy + \int_K \sigma_t I_m^h(x,y,t_{n+1}) \phi \, dxdy
$$
$$
- \int_K \left(\nabla \cdot \left(\phi (\Omega_m - \frac{1}{c} \Pi_1(x,y,t_{n+1})) \right) \right) I_m^h(x,y,t_{n+1}) dxdy
$$
$$
+ \int_{\partial K^{m-}} \boldsymbol{n}_K \cdot \left(\Omega_m - \frac{1}{c} \Pi_1(x,y,t_{n+1}) \right) I_m^h(ext(K),t_{n+1}) \phi \, ds
$$
$$
+ \int_{\partial K^{m+}} \boldsymbol{n}_K \cdot \left(\Omega_m - \frac{1}{c} \Pi_1(x,y,t_{n+1}) \right) I_m^h(int(K),t_{n+1}) \phi \, ds
$$
$$
= \int_K \sigma_s \Psi_K(x,y,t_{n+1}) \phi \, dxdy + \int_K q_m(x,y,t_{n+1}) \phi \, dxdy,
$$
$$
\forall \phi \in V_h^k(t_{n+1}), \quad K \in \mathcal{T}_h(t_{n+1}), \quad m = 1, \cdots, N_a. \tag{3.14}
$$

The above equations form a coupled system for the unknown functions $I_m^h(x,y,t_{n+1})$, $m = 1, \cdots, N_a$ since the function $\Psi_K(x,y,t_{n+1})$ contains all of them. To decouple these functions from the equations, a functional-type iteration called the source iteration (SI) [28,39] (also referred to as the grid sweeping algorithm) has been widely used for solving the system in a Gauss-Seidel-like manner. To be specific, assuming that the ℓ-th iteration solutions $I_{m,K}^{h(\ell)}(x,y,t_{n+1})$ (for $m = 1, \cdots, N_a$ and $K \in \mathcal{T}_h(t_{n+1})$) are known, we compute the new approximations $I_{m,K}^{h(\ell+1)}(x,y,t_{n+1})$ element by element in a sweeping direction [11] and through all angular directions $m = 1, \cdots, N_a$ for each given element. Thus, for $K \in \mathcal{T}_h(t_{n+1})$, we have

$$
\int_K \left(\frac{1}{c} \sum_{p=0}^{L-1} \frac{I_{m,K}^{[p](\ell+1)}(t_{n+1}) - I_{m,K}^{[p]}(t_n)}{\Delta t_n} \phi_p^{(K)} \right) \phi \, dxdy + \int_K \sigma_t I_{m,K}^{h(\ell+1)}(x,y,t_{n+1}) \phi \, dxdy
$$
$$
- \int_K \left(\nabla \cdot \left(\phi (\Omega_m - \frac{1}{c} \Pi_1(x,y,t_{n+1})) \right) \right) I_{m,K}^{h(\ell+1)}(x,y,t_{n+1}) dxdy
$$
$$
+ \int_{\partial K^{m-}} \boldsymbol{n}_K \cdot \left(\Omega_m - \frac{1}{c} \Pi_1(x,y,t_{n+1}) \right) I_{m,K}^{h(\ell+1)}(ext(K),t_{n+1}) \phi \, ds
$$
$$
+ \int_{\partial K^{m+}} \boldsymbol{n}_K \cdot \left(\Omega_m - \frac{1}{c} \Pi_1(x,y,t_{n+1}) \right) I_{m,K}^{h(\ell+1)}(int(K),t_{n+1}) \phi \, ds
$$
$$
= \int_K \sigma_s \Psi_K^*(x,y,t_{n+1}) \phi \, dxdy + \int_K q_m(x,y,t_{n+1}) \phi \, dxdy,
$$
$$
\forall \phi \in V_h^k(t_{n+1}), \quad m = 1, \cdots, N_a, \tag{3.15}
$$

where

$$\Psi_K^*(x,y,t_{n+1}) = \sum_{m=1}^{N_a} w_m I_{m,K}^*(x,y,t_{n+1}),$$

$$I_{m,K}^*(x,y,t_{n+1}) = \begin{cases} I_{m,K}^{h(\ell+1)}(x,y,t_{n+1}), & \text{when available,} \\ I_{m,K}^{h(\ell)}(x,y,t_{n+1}), & \text{otherwise.} \end{cases}$$

The iteration is stopped when the difference between two consecutive iterates is smaller than a given tolerance. In our computation, we use $\max_m \| I_m^{h(\ell+1)} - I_m^{h(\ell)} \|_\infty \leq 10^{-12}$. The source iteration is very effective, taking only a few iterations to achieve convergence for most of the problems tested.

The time integration alternates between solving the physical equation and generating the mesh. Starting with the current mesh $\mathcal{T}_h^n$ and a solution $I_m^h(x,y,t_n)$, $m = 1, \cdots, N_a$, a new mesh $\mathcal{T}_h^{n+1}$ is generated using the MMPDE moving mesh strategy to be described in the next section. Then, the DOM-DG scheme (3.15) at t_{n+1} is solved for the new solution approximation $I_m^h(x,y,t_{n+1})$, $m = 1, \cdots, N_a$.

4   The MMPDE moving mesh method on triangular meshes

In this section we describe the generation of $\mathcal{T}_h^{n+1}$ based on $\mathcal{T}_h^n$ and numerical solution $I_m^h(x,y,t_n)$, $m = 1, \cdots, N_a$ using the MMPDE moving mesh method [18–20]. The method utilizes a metric tensor (or called a monitor function) to provide the information of the size, shape, orientation of mesh elements throughout the domain that is needed for mesh adaptation. We use here a new implementation of the method proposed in [16]. A unique feature for the numerical solution of the RTE is that the functions $I_m^h(x,y,t_n)$, $m = 1, \cdots, N_a$ which correspond to the radiative intensity at angular directions Ω_m, $m = 1, \cdots, N_a$ may have discontinuities and sharp layers at different locations in space. To take this into account, we first compute the metric tensor for each function and then combine all of the metric tensors into a single one.

We start with noting that $\mathcal{T}_h^{n+1}$ and $\mathcal{T}_h^n$ have the same number of the elements (N), the same number of the vertices (N_v), and the same connectivity. They differ only in the location of the vertices. We assume that a reference computational mesh $\hat{\mathcal{T}}_c = \{(\hat{\xi}_j^1, \hat{\xi}_j^2), j = 1, \cdots, N_v\}$, which also has the same connectivity and the same numbers of vertices and elements as $\mathcal{T}_h^n$, has been chosen. In our computation, we take it as a uniform mesh (in the Euclidean metric) defined on domain $\mathbb{D}$. $\hat{\mathcal{T}}_c$ stays fixed in the computation. The generation of $\mathcal{T}_h^{n+1}$ is through a computational mesh $\mathcal{T}_c = \{(\xi_j^1, \xi_j^2), j = 1, \cdots, N_v\}$ which serves as an intermediate variable.

A key idea of the MMPDE moving mesh method is to view any nonuniform mesh as a uniform one in some metric $\mathbb{M}$ [15, 18]. The metric tensor $\mathbb{M} = \mathbb{M}(x,y,t)$ is a symmetric and uniformly positive definite matrix-valued function defined on $\mathbb{D}$. It provides

the magnitude and direction information needed for determining the size, shape, and orientation of the mesh elements throughout the domain. Various metric tensors have been proposed; e.g., see [18,21]. We here use a metric tensor based on the Hessian of the computed solution. Let $H_K(I_m^h(t_n))$ be the Hessian or a recovered Hessian of $I_m^h(t_n)$ on K. Let the eigen-decomposition of $H_K(I_m^h(t_n))$ be

$$H_K(I_m^h(t_n)) = Q\text{diag}(\lambda_1,\lambda_2)Q^T.$$

Denote

$$|H_K(I_m^h(t_n))| = Q\text{diag}(|\lambda_1|,|\lambda_2|)Q^T.$$

The metric tensor is then defined as

$$\mathbb{M}_{K,m} = \det\left(\mathbb{I} + \frac{1}{\alpha_{h,m}}|H_K(I_m^h(t_n))|\right)^{-\frac{1}{6}}\left(\mathbb{I} + \frac{1}{\alpha_{h,m}}|H_K(I_m^h(t_n))|\right), \quad \forall K \in \mathscr{T}_h, \tag{4.1}$$

where $\mathscr{T}_h$ denotes a general physical mesh, $\mathbb{I}$ is the identity matrix, $\det(\cdot)$ is the determinant of a matrix, and $\alpha_{h,m}$ is a regularization parameter defined through the algebraic equation

$$\sum_{K \in \mathscr{T}_h} |K| \det(\mathbb{M}_{K,m})^{\frac{1}{2}} = 2 \sum_{K \in \mathscr{T}_h} |K| \det(|H_K(I_m^h(t_n))|)^{\frac{1}{3}}.$$

The metric tensor (4.1) is known [21] to be optimal for the L^2-norm of linear interpolation error.

Notice that for each m ($1 \le m \le N_a$), $\mathbb{M}_{K,m}$ provides the mesh adaptation information only for function $I_m^h(x,y,t_n)$. To account for all of the functions in the mesh adaptation, we need to combine the metric tensors into a single one. We define

$$\mathbb{M}_K = \mathbb{M}_{K,1} \cap \mathbb{M}_{K,2} \cap \cdots \cap \mathbb{M}_{K,N_a},$$

where "$\cap$" stands for the intersection of symmetric and positive definite matrices which is defined as follows. Let A and B be two symmetric and positive definite matrices. There exists a nonsingular matrix P such that $PAP^T = \mathbb{I}$ and $PBP^T = \text{diag}(b_1,b_2)$. The intersection of A and B is then defined as $A \cap B = P^{-1}\text{diag}(\max(1,b_1),\max(1,b_2))P^{-T}$. Define

$$\mathbb{D}_C = \{(x,y): (x,y)C(x,y)^T < 1\}$$

for any symmetric and positive definite matrix C. It is not difficult to show that $\mathbb{D}_{A\cap B} \subseteq \mathbb{D}_A \cap \mathbb{D}_B$, which gives the meaning of the name "intersection". Notice that the definition is not optimal in the sense that $\mathbb{D}_{A\cap B}$ is not necessarily the biggest ellipse inscribed in $\mathbb{D}_A$ and $\mathbb{D}_B$.

It is known [15,18] that if $\mathscr{T}_h$ is uniform in the metric $\mathbb{M}$ in reference to the computational mesh $\mathscr{T}_c$, it satisfies

$$|K|\sqrt{\det(\mathbb{M}_K)} = \frac{\sigma_h|K_c|}{|\mathbb{D}_c|}, \quad \forall K \in \mathscr{T}_h, \tag{4.2}$$

$$\frac{1}{2}\text{tr}\left((F_K')^{-1}\mathbb{M}_K^{-1}(F_K')^{-T}\right) = \det\left((F_K')^{-1}\mathbb{M}_K^{-1}(F_K')^{-T}\right)^{\frac{1}{2}}, \quad \forall K \in \mathscr{T}_h, \tag{4.3}$$

where F_K' is the Jacobian matrix of the affine mapping: $F_K : K_c \in \mathscr{T}_c \to K$, $\mathbb{M}_K$ is the average of $\mathbb{M}$ over K, $\text{tr}(\cdot)$ denotes the trace of a matrix, and

$$|\mathbb{D}_c| = \sum_{K_c \in \mathscr{T}_c} |K_c|, \quad \sigma_h = \sum_{K \in \mathscr{T}_h} |K| \det(\mathbb{M}_K)^{\frac{1}{2}}.$$

The condition (4.2), called the equidistribution condition, determines the size of elements through the metric tensor $\mathbb{M}$. On the other hand, (4.3), referred to as the alignment condition, determines the shape and orientation of elements through $\mathbb{M}_K$ and shape of K_c. An energy function associated with these conditions is given by

$$\mathcal{I}_h(\mathscr{T}_h, \mathscr{T}_c) = \frac{1}{3} \sum_{K \in \mathscr{T}_h} |K| \det(\mathbb{M}_K)^{\frac{1}{2}} \big(\text{tr}((F_K')^{-1} \mathbb{M}_K^{-1} (F_K')^{-T}) \big)^2$$

$$+ \frac{4}{3} \sum_{K \in \mathscr{T}_h} |K| \det(\mathbb{M}_K)^{\frac{1}{2}} \Big(\det(F_K') \det(\mathbb{M}_K)^{\frac{1}{2}} \Big)^{-2}, \tag{4.4}$$

which is actually a Riemann sum of a continuous functional developed in [14] based on mesh equidistribution and alignment.

Note that $\mathcal{I}_h(\mathscr{T}_h, \mathscr{T}_c)$ is a function of the vertices $\boldsymbol{\xi}_j = (\xi_j^1, \xi_j^2), j = 1, \cdots, N_v$, of the computational mesh $\mathscr{T}_c$ and the vertices $\boldsymbol{x}_j = (x_j, y_j), j = 1, \cdots, N_v$, of the physical mesh $\mathscr{T}_h$. A straight way of solving the minimization problem is to take $\mathscr{T}_c$ as $\hat{\mathscr{T}}_c$ and then solve the minimization problem of $\mathcal{I}_h(\mathscr{T}_h, \hat{\mathscr{T}}_c)$ for the new physical mesh $\mathscr{T}_h^{n+1}$. However, $\mathcal{I}_h(\mathscr{T}_h, \hat{\mathscr{T}}_c)$ is highly nonlinear in $\boldsymbol{\xi}_j = (\xi_j^1, \xi_j^2), j = 1, \cdots, N_v$. The fact that $\mathbb{M}$ is a function of $\boldsymbol{x}$ and thus $\mathbb{M}_K$ is a function of the coordinates of the physical vertices makes the situation more difficult. Here, we adopt an indirect approach, i.e., to take $\mathscr{T}_h$ as $\mathscr{T}_h^n$, minimize $\mathcal{I}_h(\mathscr{T}_h^n, \mathscr{T}_c)$ with respect to $\mathscr{T}_c$, and then obtain the new physical mesh using the relation between $\mathscr{T}_h^n$ and newly obtained $\mathscr{T}_c$. The minimization is carried out by integrating the mesh equation which is defined as the gradient system of the energy function (the MMPDE approach), viz.,

$$\frac{d\boldsymbol{\xi}_j}{dt} = -\frac{\det(\mathbb{M}(\boldsymbol{x}_j))^{\frac{1}{2}}}{\tau} \left(\frac{\partial \mathcal{I}_h(\mathscr{T}_h^n, \mathscr{T}_c)}{\partial \boldsymbol{\xi}_j} \right)^T, \quad j = 1, \cdots, N_v, \tag{4.5}$$

where $\partial \mathcal{I}_h / \partial \boldsymbol{\xi}_j$ is considered as a row vector, $\tau > 0$ is a parameter used to adjust the response time of mesh movement to the changes in $\mathbb{M}$. Using the notion of scalar-by-matrix differentiation [16], we can rewrite (4.5) as

$$\frac{d\boldsymbol{\xi}_j}{dt} = \frac{\det(\mathbb{M}(\boldsymbol{x}_j))^{\frac{1}{2}}}{\tau} \sum_{K \in \omega_j} |K| v_{j_K}^K, \quad j = 1, \cdots, N_v, \tag{4.6}$$

where ω_j is the element patch associated with the vertex $\boldsymbol{x}_j$, j_K is the local index of $\boldsymbol{x}_j$ in K, and $v_{j_K}^K$ is the local velocity contributed by K. Denote the edge matrices of K and K_c by

$E_K = [\boldsymbol{x}_1^K - \boldsymbol{x}_0^K, \boldsymbol{x}_2^K - \boldsymbol{x}_0^K]$ and $E_{K_c} = [\boldsymbol{\xi}_1^K - \boldsymbol{\xi}_0^K, \boldsymbol{\xi}_2^K - \boldsymbol{\xi}_0^K]$, respectively. Let $\mathbb{J} = (F_K')^{-1} = E_{K_c} E_K^{-1}$ and define

$$G(\mathbb{J}, \det(\mathbb{J})) = \frac{1}{3} \det(\mathbb{M}_K)^{\frac{1}{2}} (\text{tr}(\mathbb{J} \mathbb{M}_K^{-1} \mathbb{J}^T))^2 + \frac{4}{3} \det(\mathbb{M}_K)^{\frac{1}{2}} \left(\frac{\det(\mathbb{J})}{\det(\mathbb{M}_K)^{\frac{1}{2}}} \right)^2.$$

It is not difficult [16] to find the derivatives of G with respect to $\mathbb{J}$ and $\det(\mathbb{J})$ as

$$\frac{\partial G}{\partial \mathbb{J}} = \frac{4}{3} \det(\mathbb{M}_K)^{\frac{1}{2}} (\text{tr}(\mathbb{J} \mathbb{M}_K^{-1} \mathbb{J}^T)) \mathbb{M}_K^{-1} \mathbb{J}^T,$$

$$\frac{\partial G}{\partial \det(\mathbb{J})} = \frac{8}{3} \det(\mathbb{M}_K)^{-\frac{1}{2}} \det(\mathbb{J}). \tag{4.7}$$

Then, the local velocities are expressed as

$$\begin{bmatrix} (\boldsymbol{v}_1^K)^T \\ (\boldsymbol{v}_2^K)^T \end{bmatrix} = -E_K^{-1} \frac{\partial G}{\partial \mathbb{J}} - \frac{\partial G}{\partial \det(\mathbb{J})} \frac{\det(E_{K_c})}{\det(E_K)} E_{K_c}^{-1}, \quad \boldsymbol{v}_0^K = -\boldsymbol{v}_1^K - \boldsymbol{v}_2^K. \tag{4.8}$$

Note that the velocities for the boundary nodes need to be modified properly. For example, the velocities for the corner vertices should be set to be zero. For other boundary vertices, the velocities should be modified such that they only slide along the boundary and do not move out of the domain. Specifically, we assume that the domain boundary can be expressed as $\psi(\boldsymbol{\xi}) = 0$. Consider a boundary vertex $\boldsymbol{\xi}_j$ on the boundary which is not a corner and denote the mesh equation (4.5) by $\frac{d\boldsymbol{\xi}_j}{dt} = \boldsymbol{u}_j$. This equation can be modified into

$$\frac{d\boldsymbol{\xi}_j}{dt} = \boldsymbol{u}_j - \frac{(\nabla \psi \cdot \boldsymbol{u}_j) \nabla \psi}{|\nabla \psi|^2},$$

which satisfies $\nabla \psi \cdot \frac{d\boldsymbol{\xi}_j}{dt} = 0$ and makes the node slide on the boundary.

The mesh equation (4.6) can be integrated from t^n to t^{n+1}, starting with the reference computational mesh $\hat{\mathcal{T}}_c$ as the initial mesh. The obtained new mesh is denoted by $\mathcal{T}_c^{n+1}$. Note that $\mathcal{T}_h^n$ is kept fixed during the integration and it and $\mathcal{T}_c^{n+1}$ form a correspondence, i.e., $\mathcal{T}_h^n = \Phi_h(\mathcal{T}_c^{n+1})$. Then the new physical mesh $\mathcal{T}_h^{n+1}$ is defined as $\mathcal{T}_h^{n+1} = \Phi_h(\hat{\mathcal{T}}_c)$ which can be readily computed using linear interpolation.

To conclude this section we would like to point out that a number of other moving mesh methods have been developed in the past and there is a vast literature in the area. The interested reader is referred to the books/review articles [1–3,18,37] and some recent articles [4,30,38,43].

5   Numerical examples

In this section we present numerical results obtained with the moving mesh DG method described in the previous sections for a number of one- and two-dimensional exam-

ples for the RTE. Unless otherwise stated, we use in the discrete-ordinate approxima-
tion the Gauss-Legendre P_8 and the Legendre-Chebyshev P_8-T_8 rules for one- and two-
dimensional problems, respectively, and take the final time as $T = 0.1$. By trial and error,
we have found that the time step size $\Delta t = 10^{-3}$ is sufficiently small so that the spatial
error dominates the total error in the computation. For mesh movement, we take $\tau = 0.1$
for Example 5.3 and $\tau = 0.01$ for the others. For the cases having an exact solution, the
error in the computed solution is measured in the (global) L^1, L^2, and L^∞ norm, i.e.,

$$\int_0^T \|e_h(\cdot,t)\|_{L^1}dt, \quad \int_0^T \|e_h(\cdot,t)\|_{L^2}dt, \quad \int_0^T \|e_h(\cdot,t)\|_{L^\infty}dt.$$

Example 5.1 (A discontinuous test for the one-dimensional RTE simulating the absorbing-s-
cattering model)**.** We take extinction coefficient $\sigma_t = 1000$, $\sigma_s = 1$, $c = 3.0 \times 10^8$, and

$$\begin{aligned}
q(x,\mu,t) = {}&\sigma_t \sin(2\pi(\tanh(Rx)+5\mu t)) + \frac{10\pi\mu}{c}\cos(2\pi(\tanh(Rx)+5\mu t)) \\
&+ 2\pi\mu R\cos(2\pi(\tanh(Rx)+5\mu t))(1-\tanh^2(Rx)) \\
&+ \frac{\sigma_s}{20\pi t}(\cos(2\pi(\tanh(Rx)+5t)) - \cos(2\pi(\tanh(Rx)-5t))) + a(\sigma_t - \sigma_s),
\end{aligned}$$

where $a = 2$ and $R = 200$. The initial condition is $I(x,\mu,0) = \sin(2\pi\tanh(Rx)) + a$ and the
boundary conditions are

$$\begin{aligned}
I(-1,\mu,t) &= \sin(2\pi(-\tanh(R)+5\mu t)) + a, \quad 0 < \mu \leq 1, \ 0 < t \leq T, \\
I(1,\mu,t) &= \sin(2\pi(\tanh(R)+5\mu t)) + a, \quad -1 \leq \mu < 0, \ 0 < t \leq T.
\end{aligned}$$

The exact solution of this example is $I(x,\mu,t) = \sin(2\pi(\tanh(Rx)+5\mu t)) + a$. The mesh
trajectories for the P^2-DG method with a moving mesh of $N = 80$ are shown in Fig. 1. The
moving mesh solution ($N = 80$) in the direction $\mu = -0.5255$ is compared with the fixed
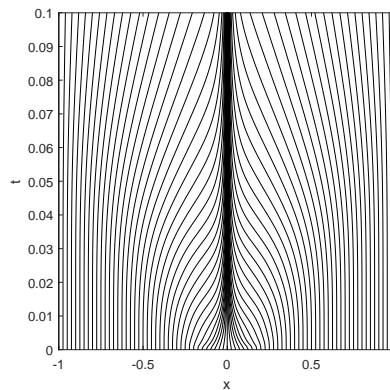

Figure 1: Example 5.1. The mesh trajectories are obtained with the P^2-DG method with a moving mesh of
$N = 80$.

(a) MM N=80, FM N=80

(b) Close view of (a) near x=0

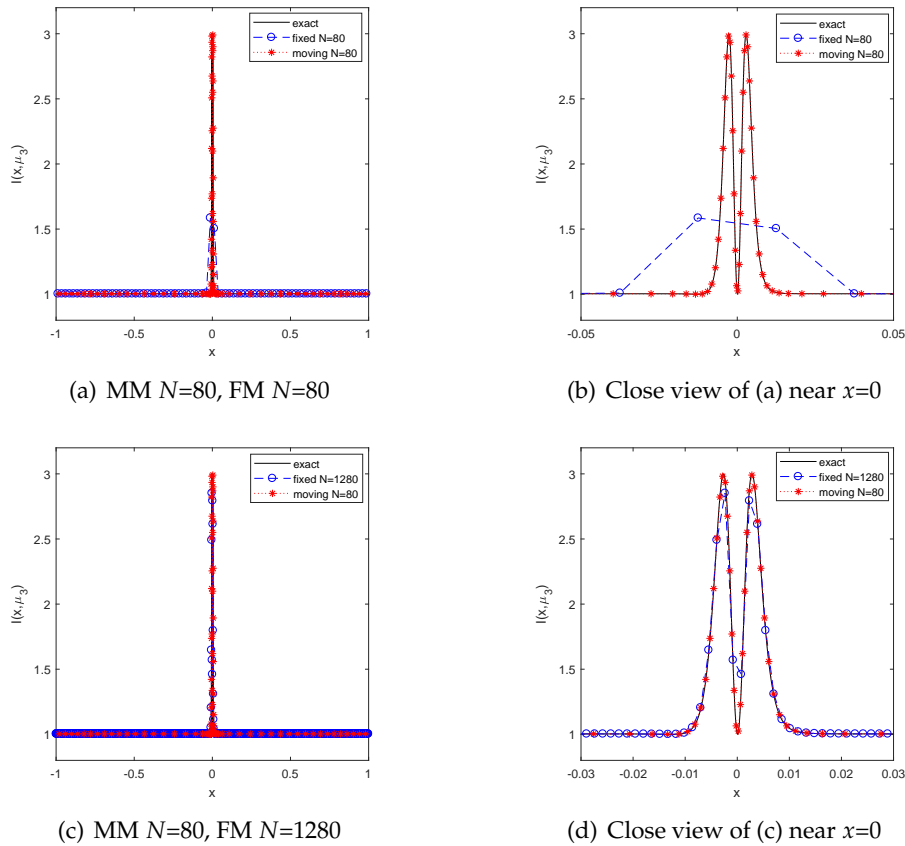(c) MM N=80, FM N=1280

(d) Close view of (c) near x=0

Figure 2: Example 5.1. The solution in the direction $\mu = -0.5255$ obtained with the P^2-DG method with a moving mesh of N=80 is compared with those obtained with fixed meshes of N=80 and N=1280. MM and FM stand for moving mesh and fixed mesh, respectively.

mesh solutions obtained with $N = 80$ and $N = 1280$ in Fig. 2. Similar results are shown in Fig. 3 for the angular direction $\mu = 0.9603$. These results show that the moving mesh solution ($N = 80$) is more accurate than those with fixed meshes of $N = 80$ and $N = 1280$.

The error in the L^1 and L^2 norm is shown in Fig. 4 for both P^1-DG and P^2-DG methods with fixed and moving meshes. It can be seen that both fixed and moving meshes lead to almost the same order of convergence for relatively large N, i.e., 2nd order for P^1-DG and 3rd-order for P^2-DG. However, a moving mesh always produces more accurate solutions than a fixed mesh of the same size for this example.

To show the efficiency of the methods, we plot in Fig. 5 the L^1 norm of the error against the CPU time measured in seconds on a Thinkpad T440 with Matlab 2017a. One can see that moving mesh (MM) P^1-DG (resp. P^2-DG) is more efficient than fixed mesh (FM) P^1-DG (resp. P^2-DG) in the sense that the former leads to a smaller error than the latter for a fixed amount of the CPU time. Moreover, when $N > 50$, MM (resp. FM) P^2-DG is more
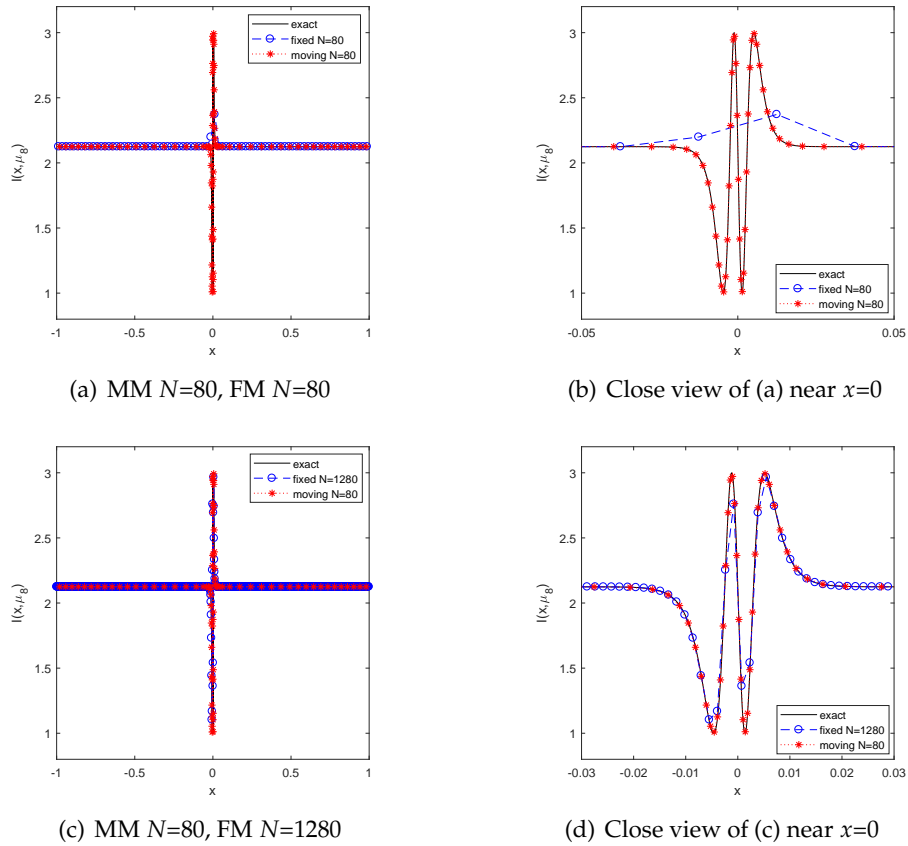
(a) MM N=80, FM N=80

(b) Close view of (a) near x=0

(c) MM N=80, FM N=1280

(d) Close view of (c) near x=0

Figure 3: Example 5.1. The solution in the direction μ=0.9603 obtained with the P^2-DG method with a moving mesh of N=80 is compared with those obtained with fixed meshes of N=80 and N=1280.
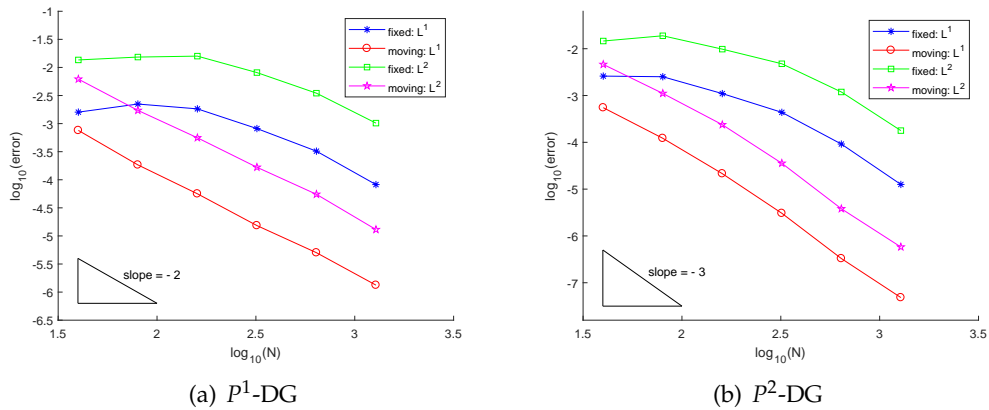

(a) P^1-DG

(b) P^2-DG

Figure 4: Example 5.1. The L^1 and L^2 norm of the error with moving and fixed meshes.
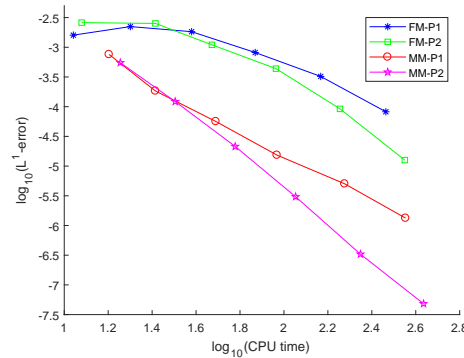
Figure 5: Example 5.1. The L^1 norm of the error is plotted against the CPU time.

efficient than MM (resp. FM) P^1-DG. Thus, a moving mesh improves the computational efficiency and the quadratic DG method has better efficiency than the linear one on both fixed and moving meshes when N is sufficiently large.

Example 5.2 (A discontinuous example of the one-dimensional RTE for the absorbing-scattering model)**.** In this example, we take $\sigma_s = 1$, $c = 3.0 \times 10^8$, and

$$\sigma_t = \begin{cases} 1, & \text{for } 0 \leq x < 0.2, \\ 900, & \text{for } 0.2 \leq x < 0.6, \\ 90, & \text{for } 0.6 \leq x \leq 1, \end{cases}$$

$$q(x,\mu,t) = \begin{cases} 100e^{-t}, & \text{for } 0 \leq x < 0.2, \\ 1, & \text{for } 0.2 \leq x < 0.6, \\ 1000e^{3t}, & \text{for } 0.6 \leq x \leq 1. \end{cases}$$

The initial condition is $I(x,\mu,0) = 15x$ and the boundary conditions are given by

$$I(0,\mu,t) = 0, \quad \text{for } 0 < \mu \leq 1, \ 0 < t \leq T,$$
$$I(1,\mu,t) = 15 + 2t, \quad \text{for } -1 \leq \mu < 0, \ 0 < t \leq T.$$

The solution of this problem has two sharp layers. Since its analytical form is unavailable for comparison purpose we take the numerical solution obtained with the P^2-DG method with a fixed mesh of $N = 20000$ as the reference solution. The mesh trajectories for the P^2-DG method with a moving mesh of $N = 80$ are shown in Fig. 6. The moving mesh solution ($N = 80$) in the direction $\mu = -0.1834$ is compared with the fixed mesh solutions obtained with $N = 80$ and $N = 1280$ in Fig. 7. Similar results are shown in Fig. 8 for the angular direction $\mu = 0.1834$. The results show that the moving mesh solution ($N = 80$) is more accurate than those with fixed meshes of $N = 80$ and $N = 1280$.
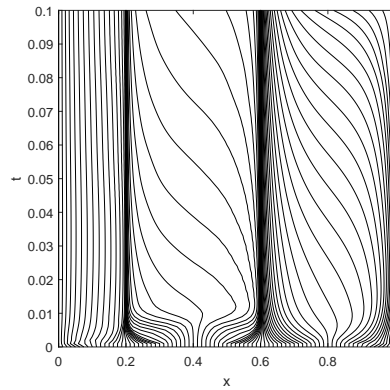
Figure 6: Example 5.2. The mesh trajectories are obtained with the P^2-DG method with a moving mesh of $N=80$.
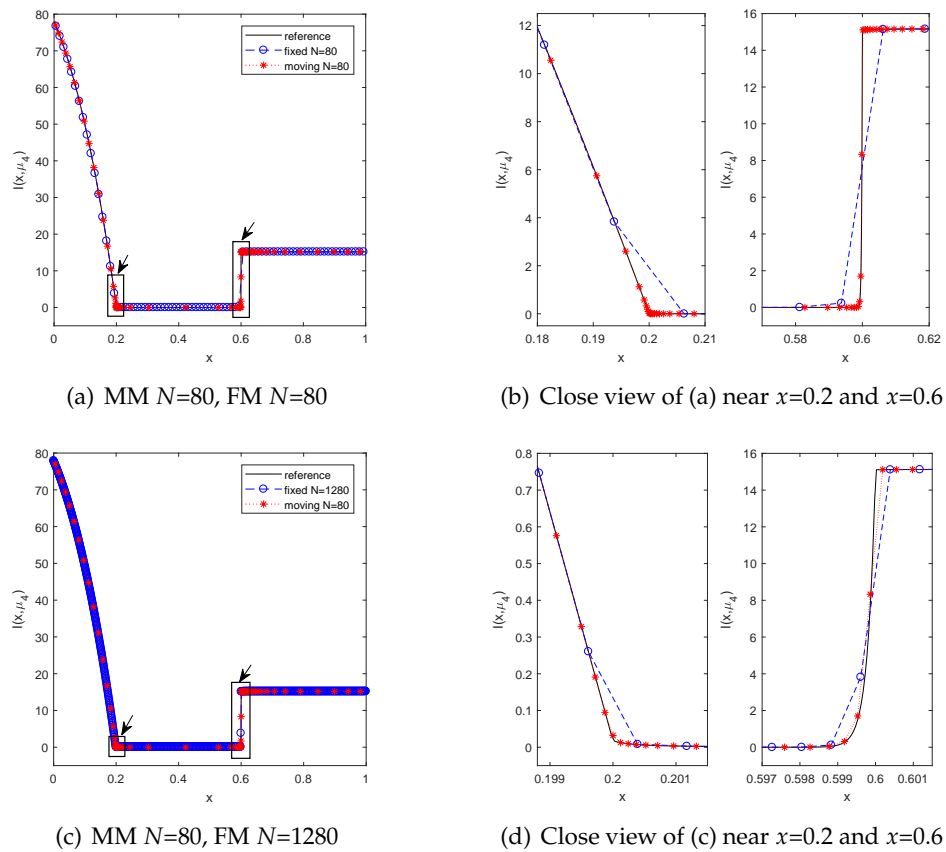

(a) MM $N=80$, FM $N=80$

(b) Close view of (a) near $x=0.2$ and $x=0.6$

(c) MM $N=80$, FM $N=1280$

(d) Close view of (c) near $x=0.2$ and $x=0.6$

Figure 7: Example 5.2. The solution in the direction $\mu=-0.1834$ obtained with the P^2-DG method with a moving mesh of $N=80$ is compared with those obtained with fixed meshes of $N=80$ and $N=1280$.
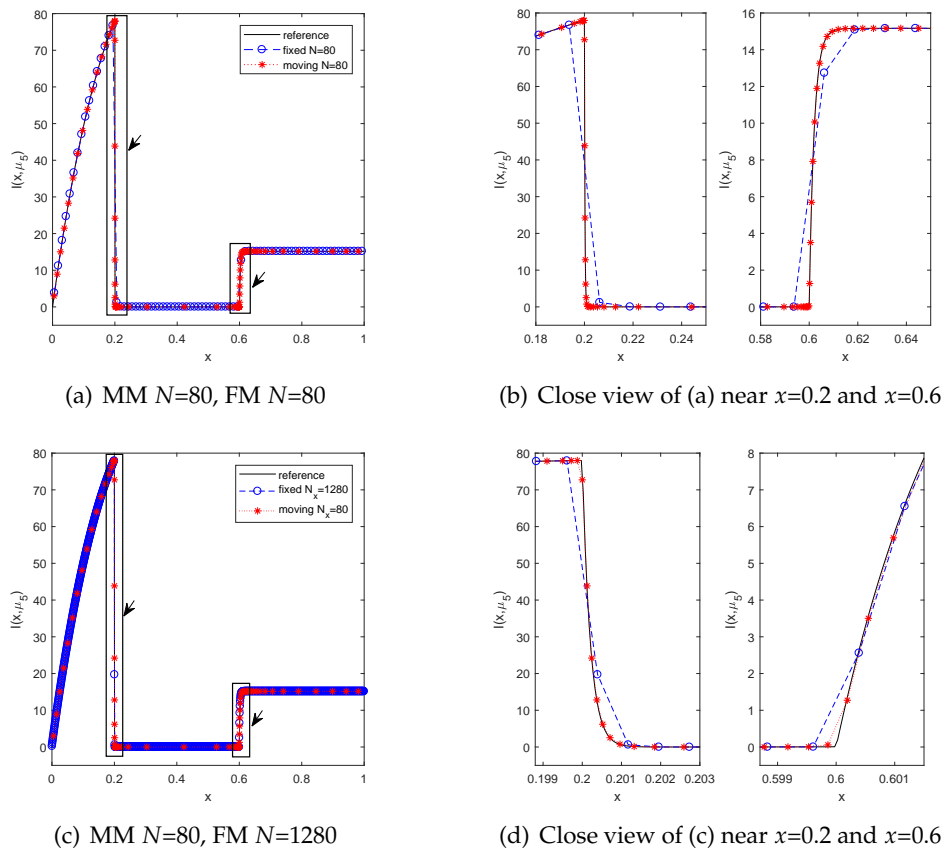
(a) MM $N=80$, FM $N=80$

(b) Close view of (a) near $x=0.2$ and $x=0.6$

(c) MM $N=80$, FM $N=1280$

(d) Close view of (c) near $x=0.2$ and $x=0.6$

Figure 8: Example 5.2. The solution in the direction $\mu=0.1834$ obtained with the P^2-DG method with a moving mesh of $N=80$ is compared with those obtained with fixed meshes of $N=80$ and $N=1280$.

Example 5.3 (An accuracy test for the two-dimensional RTE for the absorbing-scattering model)**.** In this example, we take $\sigma_t=22000$, $\sigma_s=1$, $c=3.0\times10^8$, and

$$q(x,y,\zeta,\eta,t)=e^t\left(-2\pi(\zeta+\eta)(\zeta^2+\eta^2)\cos^3\left(\frac{\pi}{2}(x+y)\right)\sin\left(\frac{\pi}{2}(x+y)\right)\right.$$
$$\left.+\left(\left(\frac{1}{c}+\sigma_t\right)(\zeta^2+\eta^2)-\frac{2}{3}\sigma_s\right)\cos^4\left(\frac{\pi}{2}(x+y)\right)+\left(\frac{1}{c}+\sigma_t-\sigma_s\right)\right).$$

The initial condition is $I(x,y,\zeta,\eta,0)=(\zeta^2+\eta^2)\cos^4(\frac{\pi}{2}(x+y))+1$ and the boundary conditions are given by

$$I(x,0,\zeta,\eta,t)=e^t\left((\zeta^2+\eta^2)\cos^4\left(\frac{\pi}{2}x\right)+1\right),\qquad \eta>0,$$
$$I(x,1,\zeta,\eta,t)=e^t\left((\zeta^2+\eta^2)\cos^4\left(\frac{\pi}{2}(x+1)\right)+1\right),\quad \eta<0,$$
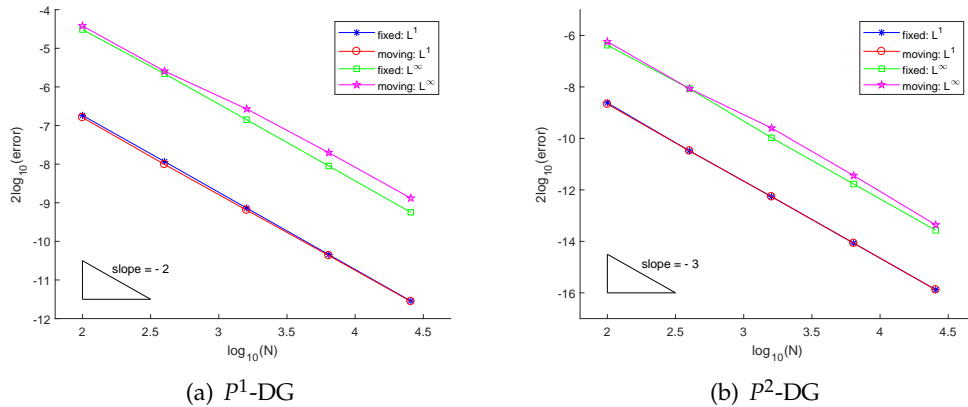
Figure 9: Example 5.3. The L^1 and L^∞ norm of the error with moving and fixed meshes.

$$I(0,y,\zeta,\eta,t) = e^t \left((\zeta^2 + \eta^2) \cos^4 \left(\frac{\pi}{2} y \right) + 1 \right), \qquad \zeta > 0,$$

$$I(1,y,\zeta,\eta,t) = e^t \left((\zeta^2 + \eta^2) \cos^4 \left(\frac{\pi}{2} (1+y) \right) + 1 \right), \quad \zeta < 0.$$

The exact solution of this problem is $I(x,y,\zeta,\eta,t) = e^t \left((\zeta^2 + \eta^2) \cos^4 (\frac{\pi}{2}(x+y)) + 1 \right)$. The error in the L^1 and L^∞ norm is plotted in Fig. 9 for the P^1-DG and P^2-DG methods with fixed and moving meshes. It can be seen that both fixed and moving meshes lead to comparable results and the same convergence order (2nd for P^1-DG and 3rd for P^2-DG), which is consistent with the theoretical prediction.

Example 5.4 (A discontinuous example of the two-dimensional RTE for the transparent model). In this test, we solve the two-dimensional RTE (2.2) with $\sigma_t = 0$, $\sigma_s = 0$, $c = 3.0 \times 10^8$, $q = 0$, $\zeta = 0.3$, and $\eta = 0.5$. The computational domain is $(0,1) \times (0,1)$. The initial condition is

$$I(x,y,\zeta,\eta,0) = \begin{cases} 0, & \text{for } y < \frac{\eta}{\zeta} x, \\ \cos^6 \left(\frac{\pi}{2} y \right), & \text{otherwise.} \end{cases}$$

The boundary conditions are

$$I(0,y,\zeta,\eta,t) = \cos^6 \left(\frac{\pi}{2} y \right) \cos^{10}(t), \qquad I(x,0,\zeta,\eta,t) = 0.$$

The exact solution of this example is

$$I(x,y,\zeta,\eta,t) = \begin{cases} 0, & \text{for } y < \frac{\eta}{\zeta} x, \\ \cos^6 \left(\frac{\pi}{2} (y - \frac{\eta}{\zeta} x) \right) \cos^{10}(t - \frac{x}{c\zeta}), & \text{otherwise.} \end{cases}$$

(a) Radiative intensity on MM $N=1600$

(b) MM $N=1600$ at $t=0.1$

(c) Radiative intensity on FM $N=1600$
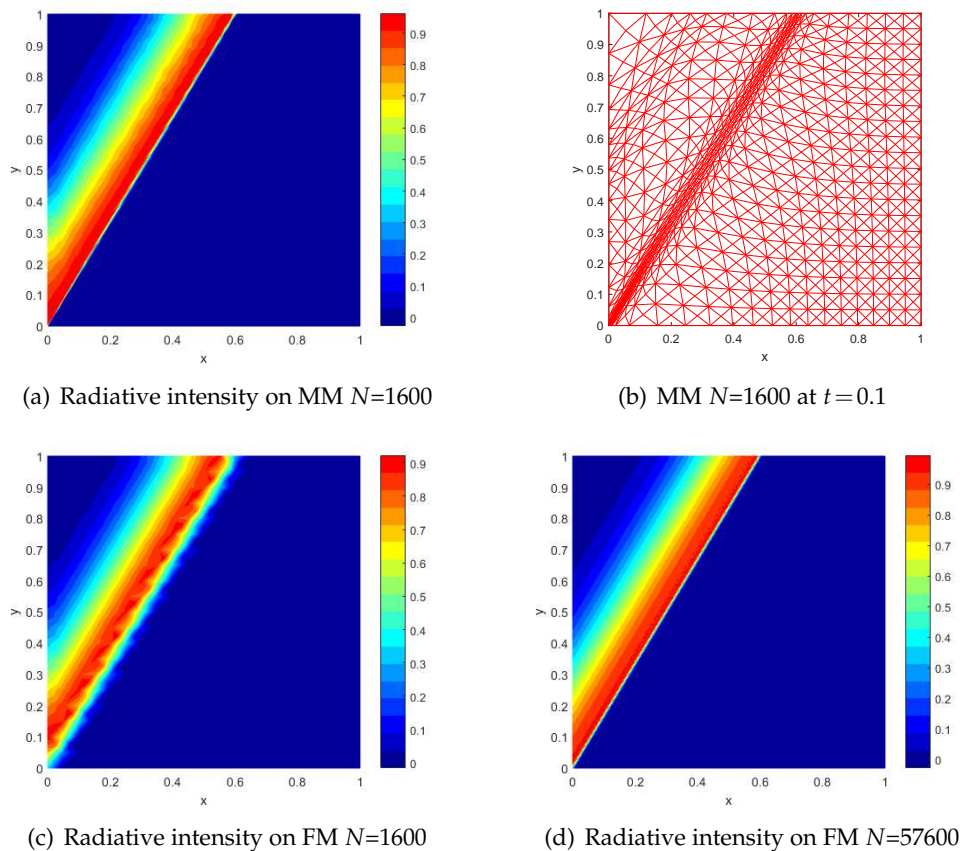

(d) Radiative intensity on FM $N=57600$

Figure 10: Example 5.4. The radiative intensity contours (and mesh) at $t=0.1$ obtained by the P^2-DG method with a moving mesh of $N=1600$ and fixed meshes of $N=1600$ and $N=57600$.

Notice that only a single angular direction is chosen in this example and the integral term in (2.2) is not involved.

The radiative intensity contours obtained with a moving mesh of $N=1600$ and fixed meshes of $N=1600$ and $N=57600$ are shown in Fig. 10. In Fig. 11, we compare the radiative intensity cut along the line $y=0.495$ for the moving mesh of $N=1600$ and the fixed meshes of $N=1600$ and $N=57600$. The results show that the moving mesh solution ($N=1600$) is better than that with the fixed mesh of $N=1600$ and is comparable with that with the fixed mesh of $N=57600$.

The error in the L^1 and L^2 norm is shown in Fig. 12 for the P^1-DG and P^2-DG methods with fixed and moving meshes. It is worth pointing out that we cannot expect P^1-DG and P^2-DG can achieve their optimal order for this problem since the solution is discontinuous. One can see from the figure that both fixed and moving meshes lead to almost the same convergence order. The order of P^1-DG is about 0.72 in L^1 norm and 0.36 in L^2 norm while that of P^2-DG is about 0.81 in L^1 norm and 0.41 in L^2 norm. Moreover, the

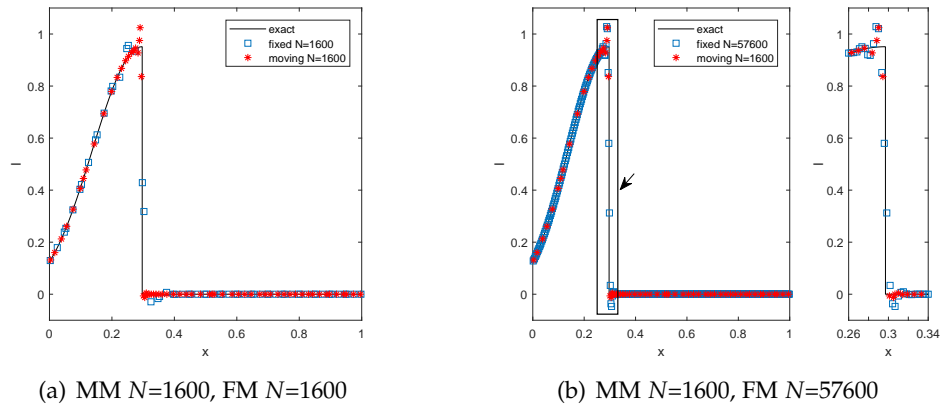(a) MM $N=1600$, FM $N=1600$

(b) MM $N=1600$, FM $N=57600$

Figure 11: Example 5.4. The comparison of the radiative intensity cut along the line $y=0.495$ obtained by the P^2-DG method with a moving mesh of $N=1600$ and fixed meshes of $N=1600$ and $N=57600$.
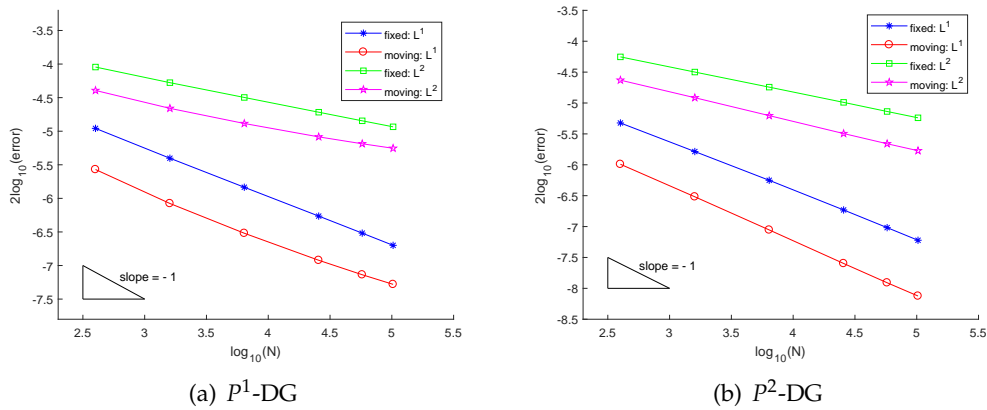

(a) P^1-DG

(b) P^2-DG

Figure 12: Example 5.4. The L^1 and L^2 norm of the error with moving and fixed meshes.

figure shows that a moving mesh produces more accurate solutions than a fixed mesh of the same number of elements for this example.

Example 5.5 (A discontinuous example of the two-dimensional RTE for the purely absorbing model). In this example, we choose $\sigma_t=1$, $\sigma_s=0$, $c=3.0\times10^8$, $q=0$, $\zeta=0.4$, and $\eta=0.9$. The computational domain is $(0,1)\times(0,1)$. The initial condition is

$$I(x,y,\zeta,\eta,0)=\begin{cases}\tanh(500(x-0.5))+1, & \text{for } y<\frac{\eta}{\zeta}x,\\ 1, & \text{otherwise.}\end{cases}$$

The boundary conditions are

$$I(0,y,\zeta,\eta,t)=e^{(y^2t)}, \qquad I(x,0,\zeta,\eta,t)=\tanh(500(x-0.5))+1.$$

(a) Radiative intensity on MM N=6400

(b) MM N=6400 at t=0.1

(c) Radiative intensity on FM N=6400
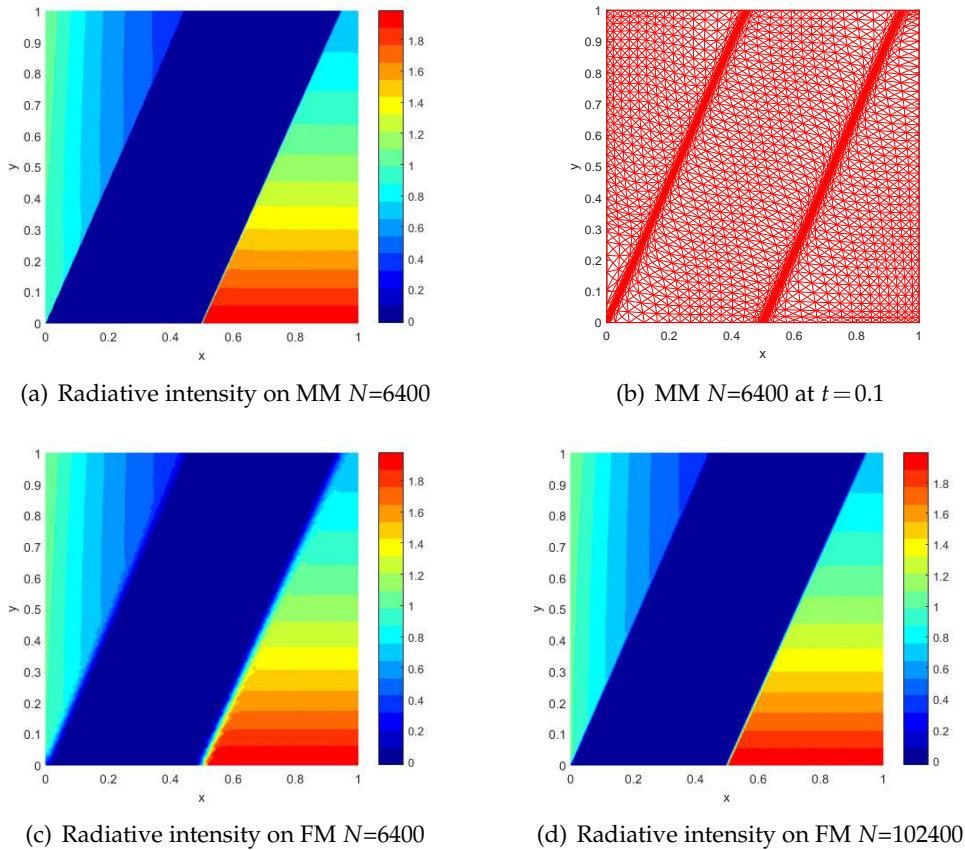

(d) Radiative intensity on FM N=102400

Figure 13: Example 5.5. The radiative intensity contours (and mesh) at t=0.1 obtained by the P^2-DG method with a moving mesh of N=6400 and fixed meshes of N=6400 and N=102400.

The exact solution of this example is

$$I(x,y,\zeta,\eta,t)=\begin{cases} (\tanh(500(x-\frac{\zeta}{\eta}y-0.5))+1)e^{-\frac{\sigma_t}{\eta}y}, & \text{for } y<\frac{\eta}{\zeta}x, \\ e^{\left((y-\frac{\eta}{\zeta}x)^2(t-\frac{x}{c\zeta})-\frac{\sigma_t}{\zeta}x\right)}, & \text{otherwise,} \end{cases}$$

which exhibits a discontinuity along with $y=\frac{\eta}{\zeta}x$ and a sharp layer along with $x=\frac{\zeta}{\eta}y+0.5$. Like the previous example, only a single angular direction is chosen in this example and the integral term in (2.2) is not involved.

The radiative intensity contours obtained with the P^2-DG method with a moving mesh of $N=6400$ and fixed meshes of $N=6400$ and $N=102400$ are shown in Fig. 13. In Fig. 14, the radiative intensity cut along the line $y=0.495$ is compared for moving and fixed meshes. The advantage of using a moving mesh is clear. The error in the L^1 and L^2 norm is plotted as a function of N in Fig. 15 for fixed and moving meshes. The P^1-DG method shows an order of about 0.87 in L^1 norm and 0.57 in L^2 norm while P^2-DG has

(a) MM N=6400, FM N=6400

(b) Close view of (a) near x=0.22 and x=0.72

(c) MM N=6400, FM N=102400
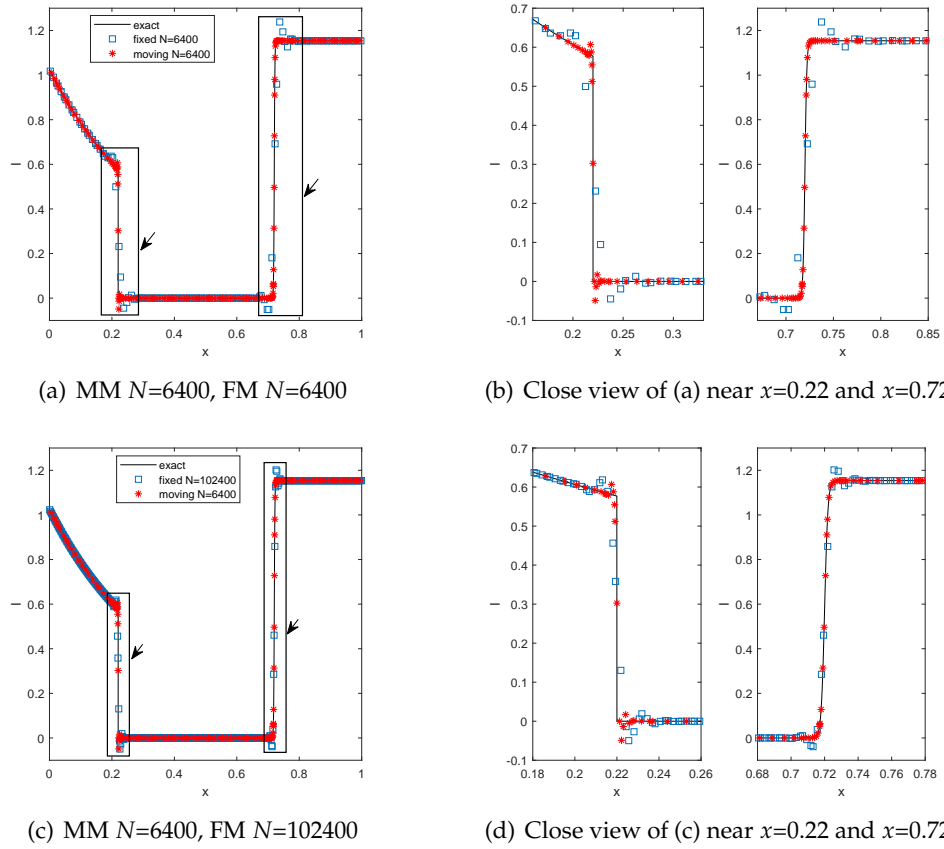
(d) Close view of (c) near x=0.22 and x=0.72

Figure 14: Example 5.5. The radiative intensity cut along the line $y=0.495$ obtained a moving mesh of $N=6400$ is compared with those obtained with fixed meshes of $N=6400$ and $N=102400$.
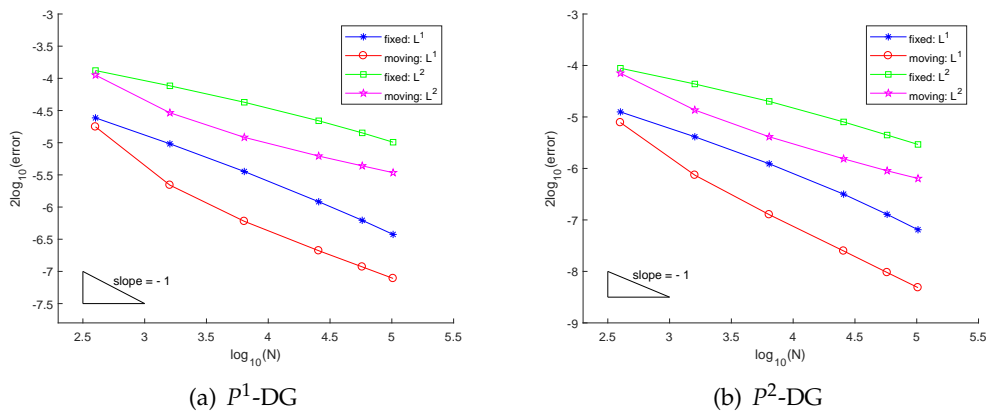

(a) P^1-DG

(b) P^2-DG

Figure 15: Example 5.5. The L^1 and L^2 norm of the error with moving and fixed meshes.
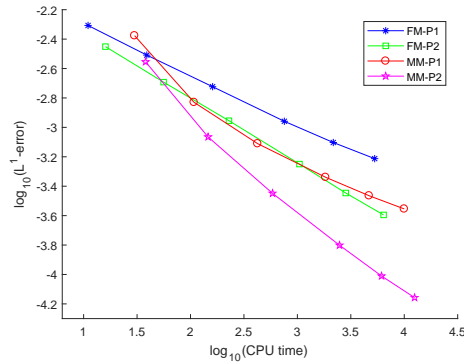
Figure 16: Example 5.5. The L^1 norm of the error is plotted against the CPU time.

an order of 1.2 in L^1 norm and 0.73 in L^2 norm for both fixed and moving meshes. A moving mesh produces more accurate solutions than a fixed mesh of the same number of elements for this example.

To show the efficiency of the methods for this two-dimensional example, we plot the L^1 norm of the error against the CPU time in Fig. 16. We can see that the error is smaller for moving mesh P^1-DG (resp. P^2-DG) than fixed mesh P^1-DG (resp. P^2-DG) for a fixed amount of the CPU time. Moreover, the better efficiency of a higher-order method is more obvious in this example than the one-dimensional Example 5.1: Fixed mesh P^2-DG is nearly equally or more efficient than moving mesh P^1-DG.

Example 5.6 (A steep transition layer example of the two-dimensional RTE for the absorbing-scattering model). In this example, the parameters are taken as $\sigma_t = 10000$, $\sigma_s = 1$, $c = 3.0 \times 10^8$, and

$$
\begin{aligned}
q(x,y,\zeta,\eta,t) = e^t \bigg(& \left(\frac{1}{c} + \sigma_t \right) \left(a - \tanh \left(R(x^2 + y^2 - \sqrt{2(1 - \zeta^2 - \eta^2)}) \right) \right) \\
& - 2R(\zeta x + \eta y) \left(1 - \tanh^2 \left(R(x^2 + y^2 - \sqrt{2(1 - \zeta^2 - \eta^2)}) \right) \right) \\
& + \frac{\sqrt{2}\sigma_s}{2R} \left(\ln \left(\cosh(R(\sqrt{2} - x^2 - y^2)) \right) - \ln \left(\cosh(R(x^2 + y^2)) \right) \right) - \sigma_s a \bigg),
\end{aligned}
$$

where $R = 200$ and $a = 10$. The initial condition is

$$
I(x,y,\zeta,\eta,0) = \left(a - \tanh \left(R(x^2 + y^2 - \sqrt{2(1 - \zeta^2 - \eta^2)}) \right) \right)
$$

and the boundary conditions are

$$
\begin{aligned}
I(x,0,\zeta,\eta,t) &= e^t \left(a - \tanh \left(R(x^2 - \sqrt{2(1 - \zeta^2 - \eta^2)}) \right) \right), & \eta > 0, \\
I(x,1,\zeta,\eta,t) &= e^t \left(a - \tanh \left(R(x^2 + 1 - \sqrt{2(1 - \zeta^2 - \eta^2)}) \right) \right), & \eta < 0,
\end{aligned}
$$

(a) Radiative intensity on MM N=6400

(b) MM $N = 6400$ at $t = 0.1$

(c) Radiative intensity on FM N=6400
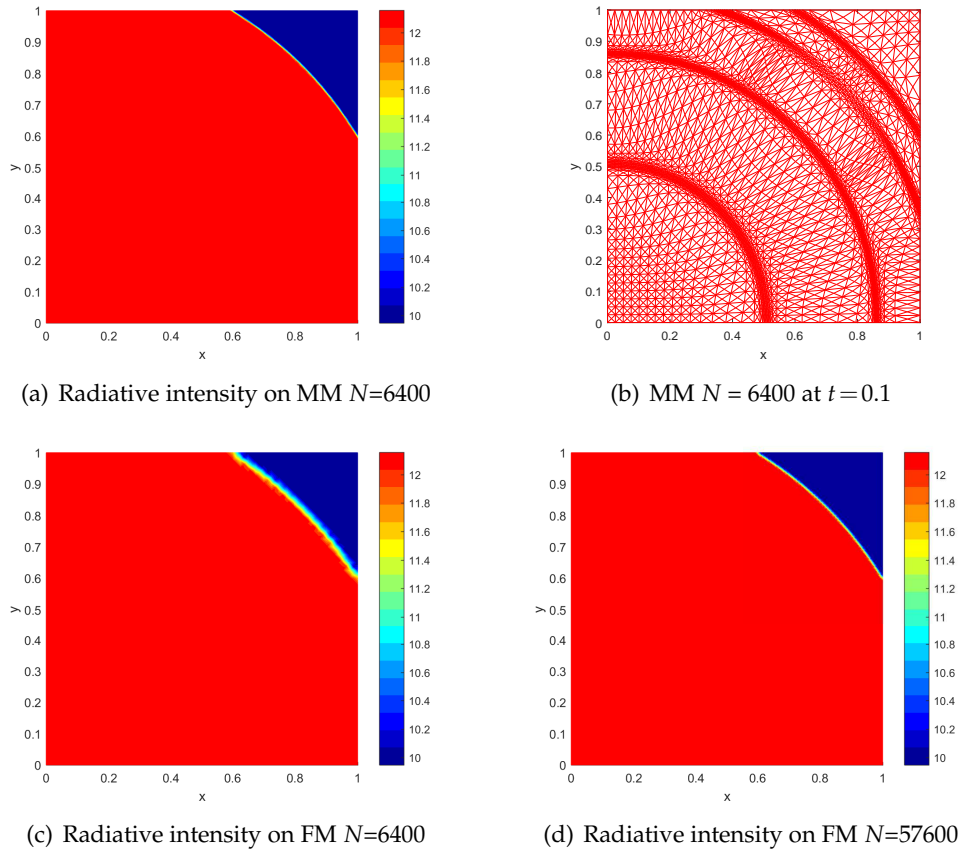

(d) Radiative intensity on FM N=57600

Figure 17: Example 5.6. The radiative intensity contours in the direction $\Omega = (-0.2578, -0.1068)$ (and mesh) at $t = 0.1$ obtained by P^2-DG method with a moving mesh of $N = 6400$ and fixed meshes of $N = 6400$ and $N = 57600$.

$$I(0,y,\zeta,\eta,t) = e^t \left(a - \tanh \left(R(y^2 - \sqrt{2(1-\zeta^2-\eta^2)}) \right) \right), \qquad \zeta > 0,$$

$$I(1,y,\zeta,\eta,t) = e^t \left(a - \tanh \left(R(1+y^2 - \sqrt{2(1-\zeta^2-\eta^2)}) \right) \right), \quad \zeta < 0.$$

This problem has the exact solution

$$I(x,y,\zeta,\eta,t) = e^t \left(a - \tanh \left(R(x^2+y^2 - \sqrt{2(1-\zeta^2-\eta^2)}) \right) \right),$$

for which the location of the steep transition layers changes with the angular variable $\Omega = (\zeta,\eta)$. The radiative intensity contours for the directions $\Omega = (-0.2578, -0.1068)$ and $\Omega = (0.7860, 0.3256)$ are shown in Figs. 17 and 18, respectively, for the P^2-DG method with moving and fixed meshes. It can be seen that the elements of the moving mesh are concentrated in the regions of the sharp transition layers in the radiative intensities for all
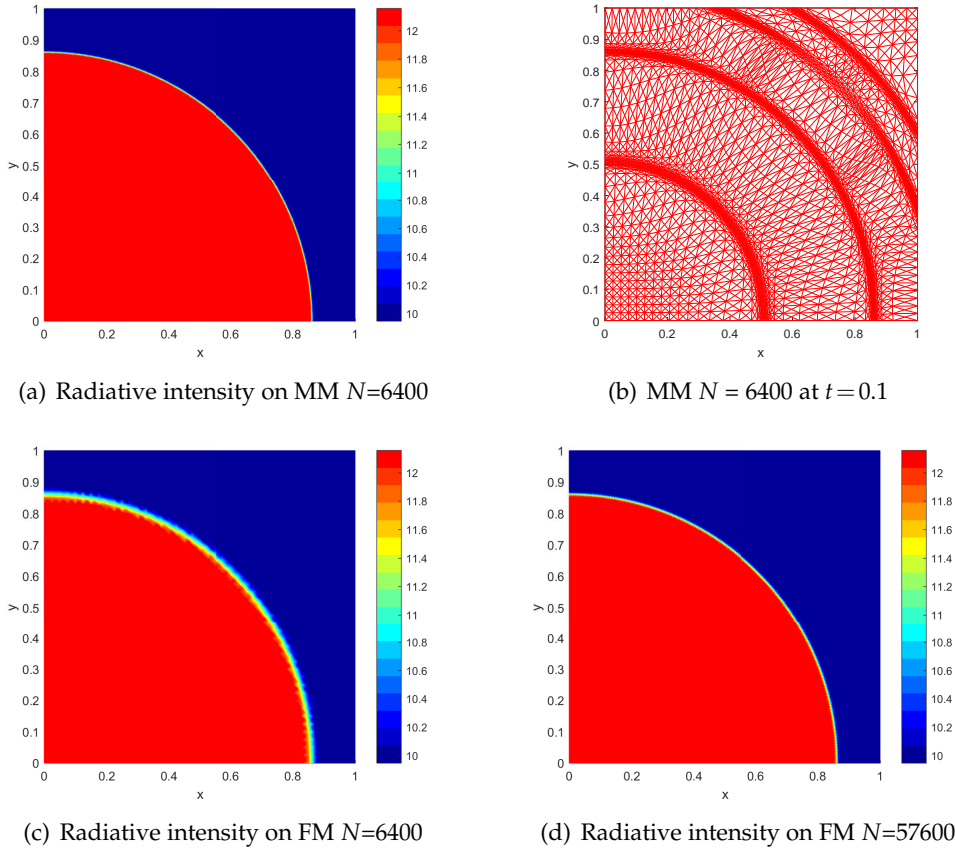
(a) Radiative intensity on MM N=6400

(b) MM N = 6400 at t=0.1

(c) Radiative intensity on FM N=6400

(d) Radiative intensity on FM N=57600

Figure 18: Example 5.6. The radiative intensity contours in the direction $\Omega = (0.7860, 0.3256)$ (and mesh) at t=0.1 obtained by P^2-DG method with a moving mesh of N=6400 and fixed meshes of N=6400 and N=57600.

angular directions. (The mesh shows four layers while only two layers are shown in the intensity contours in Figs. 17 and 18. The figures for the intensities for other directions are omitted to save space.) It can be seen that the moving mesh ($N = 6400$) provides a better resolution of the layers than the fixed mesh of N=6400 and is comparable with the fixed mesh of N=57600.

The error in the L^1 and L^2 norm is plotted in Fig. 19 as a function of N. The convergence order is similar for both fixed and moving meshes, i.e., the order of P^1-DG is about 1.7 in L^1 norm and 1.6 in L^2 norm and that of P^2-DG is 2.2 in both L^1 and L^2 norm.

Example 5.7 (A steep transition layer example of the two-dimensional RTE for the absorbing-scattering model). In this final example, we take $\sigma_t = 33$, $\sigma_s = 3$, and $c = 3.0 \times 10^8$. The computational domain is $(-1,1) \times (-1,1)$. Define

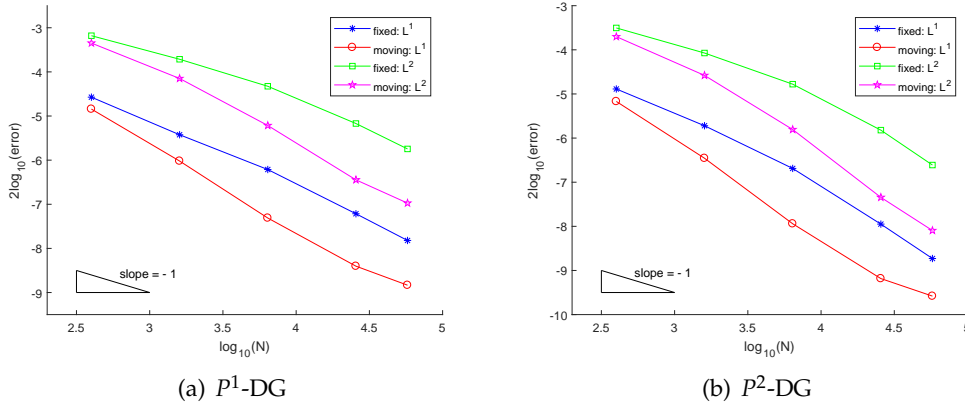$$C_0(x,y) = \tanh\left(R\left(x^2 + y^2 - \frac{1}{8} \right) \right),$$

Figure 19: Example 5.6. The L^1 and L^2 norm of the error with moving and fixed meshes.

$$C_1(x,y) = \tanh\left(R\left((x-0.5)^2+(y-0.5)^2-\frac{1}{8}\right)\right),$$

$$C_2(x,y) = \tanh\left(R\left((x-0.5)^2+(y+0.5)^2-\frac{1}{8}\right)\right),$$

$$C_3(x,y) = \tanh\left(R\left((x+0.5)^2+(y-0.5)^2-\frac{1}{8}\right)\right),$$

$$C_4(x,y) = \tanh\left(R\left((x+0.5)^2+(y+0.5)^2-\frac{1}{8}\right)\right),$$

where $R = 200$. The source term is taken as

$$
\begin{aligned}
&q(x,y,\zeta,\eta,t)\\
&= e^t(\zeta^2+\eta^2)\left(\left(\frac{1}{c}+\sigma_t\right)\left(5a-\sum_{i=0}^{4}C_i(x,y)\right)-R\big(2(\zeta x+\eta y)(1-C_0(x,y)^2)\right.\\
&\quad +(\zeta(2x-1)+\eta(2y-1))(1-C_1(x,y)^2)+(\zeta(2x-1)+\eta(2y+1))(1-C_2(x,y)^2)\\
&\quad \left.+(\zeta(2x+1)+\eta(2y-1))(1-C_3(x,y)^2)+(\zeta(2x+1)+\eta(2y+1))(1-C_4(x,y)^2))\right)\\
&\quad -\frac{2}{3}e^t\sigma_s\left(5a-\sum_{i=0}^{4}C_i(x,y)\right),
\end{aligned}
$$

where $a = 2$. The initial condition is $I(x,y,\zeta,\eta,0) = (\zeta^2+\eta^2)\left(5a-\sum_{i=0}^{4}C_i(x,y)\right)$ and the boundary conditions are

$$I(x,0,\zeta,\eta,t) = e^t(\zeta^2+\eta^2)\left(5a-\sum_{i=0}^{4}C_i(x,0)\right), \quad \eta > 0,$$

(a) Radiative intensity on MM N=6400

(b) MM $N = 6400$ at $t = 0.1$

(c) Radiative intensity on FM N=6400
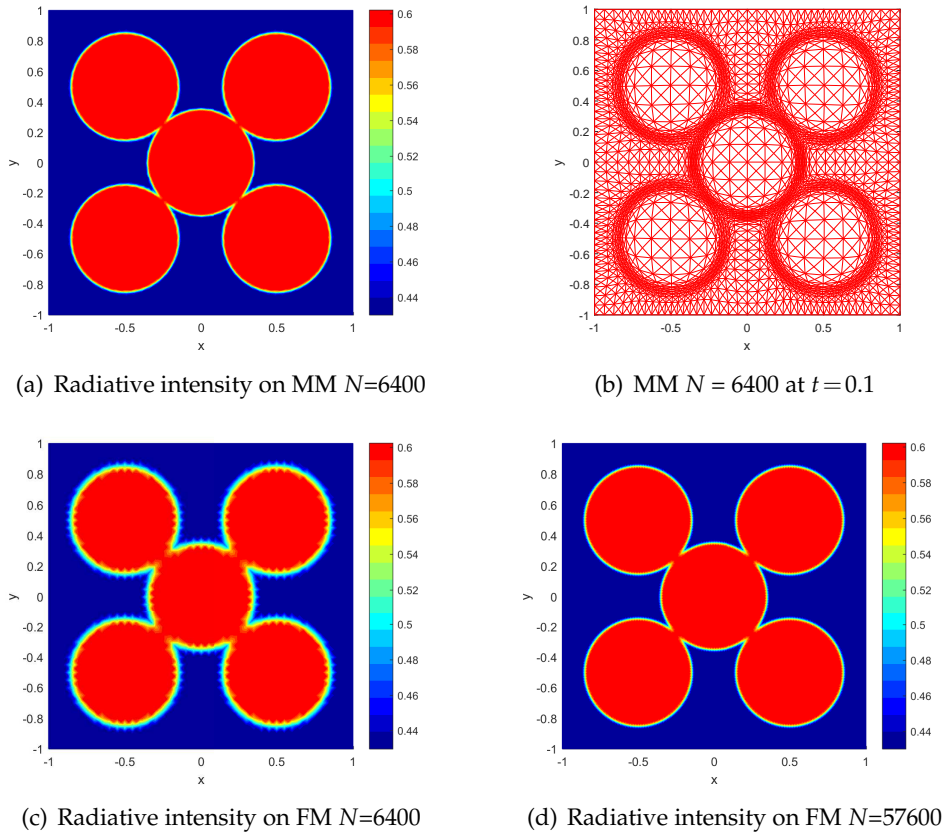

(d) Radiative intensity on FM N=57600

Figure 20: Example 5.7. The radiative intensity contours in the direction $\Omega = (0.2578, 0.1068)$ (and mesh) at $t = 0.1$ obtained by P^2-DG method with a moving mesh of $N = 6400$ and fixed meshes of $N = 6400$ and $N = 57600$.

$$I(x,1,\zeta,\eta,t) = e^t(\zeta^2 + \eta^2)\left(5a - \sum_{i=0}^{4} C_i(x,1)\right), \quad \eta < 0,$$

$$I(0,y,\zeta,\eta,t) = e^t(\zeta^2 + \eta^2)\left(5a - \sum_{i=0}^{4} C_i(0,y)\right), \quad \zeta > 0,$$

$$I(1,y,\zeta,\eta,t) = e^t(\zeta^2 + \eta^2)\left(5a - \sum_{i=0}^{4} C_i(1,y)\right), \quad \zeta < 0.$$

The exact solution of the problem is

$$I(x,y,\zeta,\eta,t) = e^t(\zeta^2 + \eta^2)\left(5a - \sum_{i=0}^{4} C_i(x,y)\right),$$

which exhibits a sharp layer of five-ring shape, independent of ζ and η. The radiative intensity contours in the direction $\Omega = (0.2578, 0.1068)$ are shown in Fig. 20 for the P^2-

(a) MM $N=6400$, FM $N=6400$

(b) Close view of (a) near $x \in (-0.35, 0.35)$

(c) MM $N=6400$, FM $N=57600$

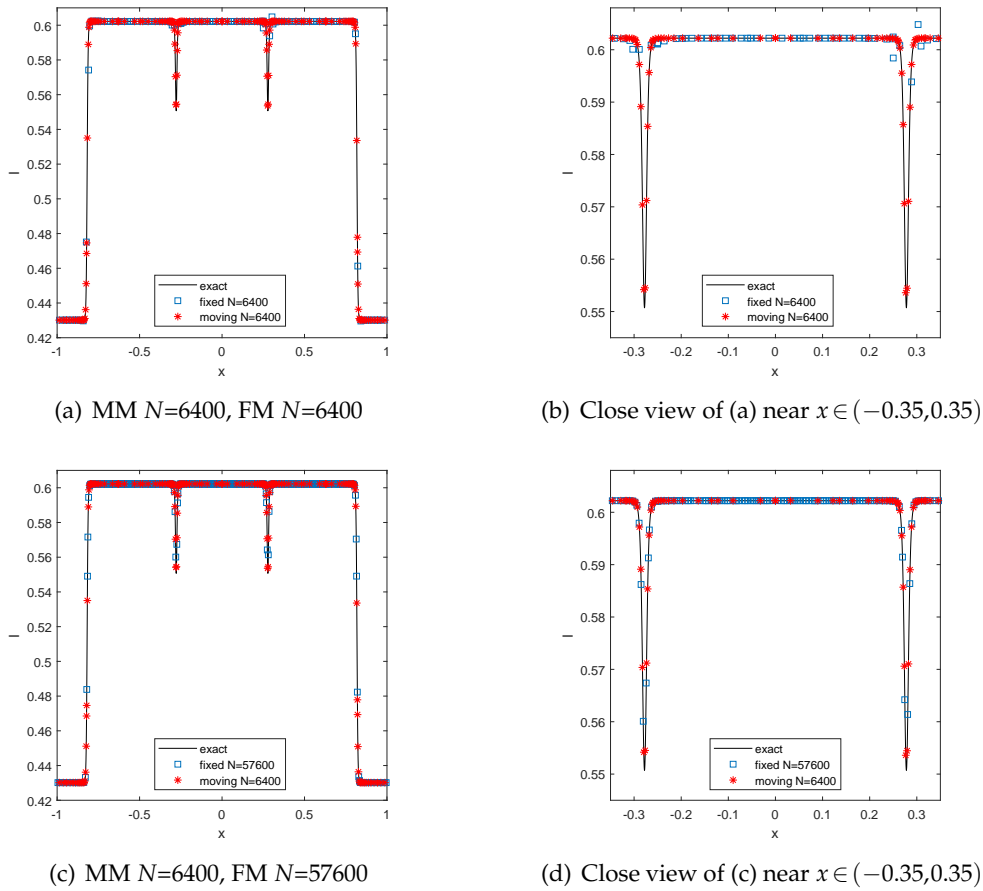(d) Close view of (c) near $x \in (-0.35, 0.35)$

Figure 21: Example 5.7. The radiative intensity in the direction $\Omega = (0.2578, 0.1068)$ cut along the line $y = 0.8x$ obtained a moving mesh of $N = 6400$ is compared with those obtained with fixed meshes of $N = 6400$ and $N = 57600$.

DG method with moving and fixed meshes. In Fig. 21, the radiative intensity cut along the line $y = 0.8x$ for the direction $\Omega = (0.2578, 0.1068)$ is compared for moving and fixed meshes. The error in L^1 and L^2 norm is plotted as a function of N in Fig. 22.

6 Conclusions

In the previous sections an adaptive moving mesh DG method has been presented for the numerical solution of the radiative transfer equation. The RTE is an integro-differential equation modeling the conservation of photons. It involves an integral term in the angular directions while being hyperbolic in space. The challenges for its numerical solution include the needs to handle with its high dimensionality, the presence of the integral

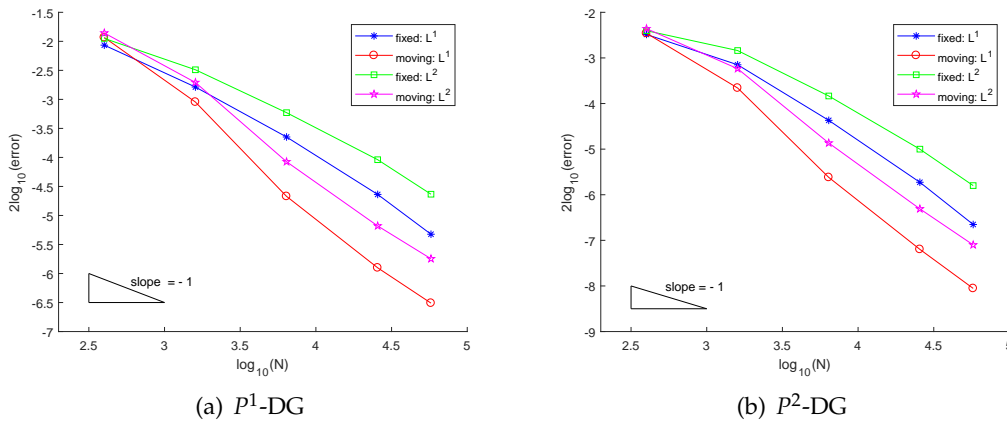(a) P^1-DG   (b) P^2-DG

Figure 22: Example 5.7. The L^1 and L^2 norm of the error with moving and fixed meshes.

term, and the development of discontinuities and sharp layers in its solution along spatial directions. In the current work, the RTE is discretized first in the angular directions with the discrete ordinate method and then with a DG method in space on a moving mesh. The mesh is moved adaptively using the MMPDE strategy to provide better resolution of sharp layers or discontinuities and thus better efficiency. The source iteration is used to avoid the coupling of the radiative intensities for all angular directions in the integral term.

A selection of one- and two-dimensional examples have been presented to demonstrate the accuracy and efficiency of the method. It has been shown that the method is able to automatically concentrate the mesh points in regions of discontinuities or steep transition layers and is more efficient than its fixed mesh counterpart. Particularly, the combination of the metric tensors for radiative intensities for different angular directions into a single metric tensor using the matrix intersection seems to work well for the tested problems with single or multiple sharp layers and discontinuities. Interestingly, the results also show that the quadratic DG method has better efficiency than the linear DG for both fixed and moving meshes.

It should be pointed out that we have not considered positivity-preserving limiters [39, 42] nor nonoscillatory limiters such as TVB limiter [7, 9], the WENO limiter [33, 40], or HWENO limiter [34, 41] in the current work. One may observe that localized spurious oscillations occur in numerical solutions containing discontinuities. How to combine limiters with our moving mesh DG method for the RTE or more general integro-differential equations will be an interesting research topic for the near future. Other future work will include extending the method to the numerical solution of the RTE coupled with the Euler equations, the material equation or the energy equation for real situations.

Acknowledgments

References

[1] M. J. Baines. *Moving Finite Elements*. Oxford University Press, Oxford, 1994.

[2] M. J. Baines, M. E. Hubbard, and P. K. Jimack. Velocity-based moving mesh methods for nonlinear partial differential equations. *Commun. Comput. Phys.*, 10 (2011), 509-576.

[3] C. J. Budd, W. Huang, and R. D. Russell. Adaptivity with moving grids. *Acta Numer.*, 18 (2009), 111-241.

[4] W. Boscheri, M. Semplice, and M. Dumbser, Central WENO subcell finite volume limiters for ADER discontinuous Galerkin schemes on fixed and moving unstructured meshes, *Commun. Comput. Phys.*, 25 (2019), 311-346.

[5] C. Canuto, M. Y. Hussaini, A. Quarteroni, and T. A. Zang. Spectral Methods in Fluid Dynamics, Springer-Verlag, New York (1988).

[6] B. G. Carlson, Solution of the Transport Equation by S_n Approximations, Los Alamos Scientific Laboratory Report LA-1599, 1953.

[7] B. Cockburn and C.-W. Shu, TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws II: general framework, *Math. Comput.*, 52 (1989), 411-435.

[8] B. Cockburn, S.-Y. Lin, and C.-W. Shu, TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws III: one dimensional systems, *J. Comput. Phys.*, 84 (1989), 90-113.

[9] B. Cockburn, S. Hou, and C.-W. Shu, The Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws IV: the multidimensional case, *Math. Comput.*, 54 (1990), 545-581.

[10] B. Cockburn and C.-W. Shu, The Runge-Kutta discontinuous Galerkin method for conservation laws V: multidimensional systems, *J. Comput. Phys.*, 141 (1998), 199-224.

[11] G. Colomer, R. Borrell, F. X. Trias, and I. Rodriguez, Parallel algorithms for S_n transport sweeps on unstructured meshes, *J. Comput. Phys.*, 232 (2013), 118-135.

[12] A. Dedner and P. Vollmller, An adaptive higher order method for solving the radiation transport equation on unstructured grids, *J. Comput. Phys.*, 178 (2002), 263-289.

[13] S. Giani and M. Seaid, *hp*-adaptive discontinuous Galerkin methods for simplified P_N approximations of frequency-dependent radiative transfer, *Comput. Methods Appl. Mech. Engrg.*, 301 (2016), 52-79.

[14] W. Huang, Variational mesh adaptation: isotropy and equidistribution, *J. Comput. Phys.*, 174 (2001), 903-924.

[15] W. Huang, Mathematical principles of anisotropic mesh adaptation, *Commun. Comput. Phys.*, 1 (2006), 276-310.

[16] W. Huang and L. Kamenski, A geometric discretization and a simple implementation for variational mesh generation and adaptation, *J. Comput. Phys.*, 301 (2015), 322-337.

[17] W. Huang and L. Kamenski, On the mesh nonsingularity of the moving mesh PDE method, *Math. Comput.*, 87 (2018), 1887-1911.

[18] W. Huang and R. D. Russell, Adaptive Moving Mesh Methods, Springer, New York, Applied Mathematical Sciences Series, Vol. 174 (2011).

[19] W. Huang, Y. Ren, and R. D. Russell, Moving mesh methods based on moving mesh partial differential equations, *J. Comput. Phys.*, 113 (1994), 279-290.

[20] W. Huang, Y. Ren, and R. D. Russell, Moving mesh partial differential equations (MMPDEs) based upon the equidistribution principle, *SIAM J. Numer. Anal.*, 31 (1994), 709-730.

[21] W. Huang and W. Sun, Variational mesh adaptation II: error estimates and monitor functions, *J. Comput. Phys.*, 184 (2003), 619-648.

[22] J. P. Jessee, W. A. Fiveland, L. H. Howell, P. Colella, and R. B. Pember, An adaptive mesh refinement algorithm for the radiative transport equation, *J. Comput. Phys.*, 139 (1998), 380-398.

[23] P. K. Jimack and A. J. Wathen, Temporal derivatives in the finite-element method on continuously deforming grids, *SIAM J. Numer. Anal.*, 28 (1991), 990-1003.

[24] R. Kong, M. Ambrose, and J. Spanier, Efficient automated Monte Carlo methods for radiation transport, *J. Comput. Phys.*, 227 (2008), 9463-9476.

[25] M. Krook, On the solution of equations of transfer, *Astrophys. J.*, 122 (1955), 488-497.

[26] K. D. Lathrop and B. G. Carlson, Discrete Ordinates Angular Quadrature of the Neutron Transport equation, Tech. Report LA-3186, Los Alamos Scientific Laboratory, 1965.

[27] P. LeSaint and P.-A. Raviart, On a Finite Element Method for Solving the Neutron Transport Equation, in: C. de Boor (Ed.), Mathematical Aspects of Finite Elements in Partial Differential Equations, Academic Press, New York, 1974, 89-123.

[28] E. E. Lewis and W. F. Miller, Computational Methods of Neutron Transport, New York, 1984.

[29] D. Luo, W. Huang, and J. Qiu, A quasi-Lagrangian moving mesh discontinuous Galerkin method for hyperbolic conservation laws, *J. Comput. Phys.*, 396 (2019), 544-578.

[30] X. Li and W. Huang, Anisotropic mesh adaptation for 3D anisotropic diffusion problems with application to fractured reservoir simulation, *Numer. Math. Theor. Meth. Appl.*, 10 (2017), 913-940.

[31] R. G. McClarren and T. J. Urbatsch, A modified implicit Monte Carlo method for time-dependent radiative transfer with adaptive material coupling, *J. Comput. Phys.*, 228 (2009), 5669-5686.

[32] G. C. Pomraning, The Equations of Radiation Hydrodynamics, Pergamon Press, New York (1973).

[33] J. Qiu and C.-W. Shu, Runge-Kutta discontinuous Galerkin method using WENO limiters, *SIAM J. Sci. Comput.*, 26 (2005), 907-929.

[34] J. Qiu and C.-W. Shu, Hermite WENO schemes and their application as limiters for Runge-Kutta discontinuous Galerkin method: one-dimensional case, *J. Comput. Phys.*, 193 (2004), 115-135.

[35] J. C. Ragusa, J. L. Guermond, and G. Kanschat, A robust S_N-DG-approximation for radiation transport in optically thick and diffusive regimes, *J. Comput. Phys.*, 231 (2012), 1947-1962.

[36] W. H. Reed and T. R. Hill, Triangular mesh methods for the neutron transport equation, Report LA-UR-73-479, Los Alamos Scientific Laboratory, Los Alamos, NM, 1973.

[37] T. Tang. Moving mesh methods for computational fluid dynamics flow and transport. *Recent Advances in Adaptive Computation (Hangzhou, 2004)*, Volume 383 of *AMS Contemporary Mathematics*, pages 141–173. Amer. Math. Soc., Providence, RI, 2005.

[38] E. S. Wise, B. T. Cox, and B. E. Treeby, mesh density functions based on local bandwidth applied to moving mesh methods, *Commun. Comput. Phys.*, 22 (2017), 1286-1308.

[39] D. Yuan, J. Cheng, and C.-W. Shu, High order positivity-preserving discontinuous Galerkin

methods for radiative transfer equations, *SIAM J. Sci. Comput.*, 38 (2016), A2987-A3019.

[40] J. Zhu, J. Qiu, C.-W. Shu, and M. Dumbser, Runge-Kutta discontinuous Galerkin method using WENO limiters II: Unstructured meshes, *J. Comput. Phys.*, 227 (2008), 4330-4353.

[41] J. Zhu and J. Qiu, Hermite WENO schemes and their application as limiters for Runge-Kutta discontinuous Galerkin method III: Unstructured meshes, *J. Sci. Comput.*, 39 (2009), 293-321.

[42] M. Zhang, J. Cheng, and J. Qiu, High order positivity-preserving discontinuous Galerkin schemes for radiative transfer equations on triangular meshes, *J. Comput. Phys.*, 397 (2019), https://doi.org/10.1016/j.jcp.2019.07.010.

[43] H. Zhang and P. A. Zegeling, A moving mesh finite difference method for non-monotone solutions of non-equilibrium equations in porous media, *Commun. Comput. Phys.*, 22 (2017), 935-964.