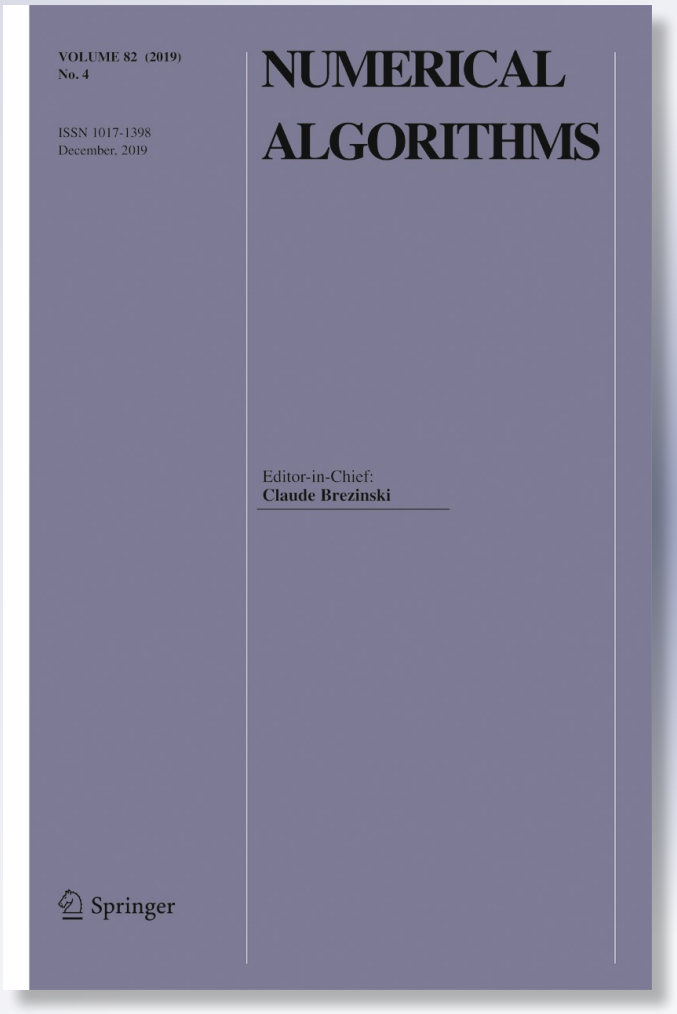
Moving mesh finite difference solution of non-equilibrium radiation diffusion equations

Xiaobo Yang, Weizhang Huang & Jianxian Qiu

Numerical Algorithms

ISSN 1017-1398
Volume 82
Number 4

Numer Algor (2019) 82:1409-1440
DOI 10.1007/s11075-019-00662-5



Springer

Numerical Algorithms (2019) 82:1409–1440
https://doi.org/10.1007/s11075-019-00662-5

ORIGINAL PAPER

Moving mesh finite difference solution of non-equilibrium radiation diffusion equations

Xiaobo Yang[1] · Weizhang Huang[2] · Jianxian Qiu[3]

Received: 26 February 2018 / Accepted: 3 January 2019 / Published online: 6 February 2019

Abstract

A moving mesh finite difference method based on the moving mesh partial differential equation is proposed for the numerical solution of the 2T model for multi-material, non-equilibrium radiation diffusion equations. The model involves nonlinear diffusion coefficients and its solutions stay positive for all time when they are positive initially. Nonlinear diffusion and preservation of solution positivity pose challenges in the numerical solution of the model. A coefficient-freezing predictor-corrector method is used for nonlinear diffusion while a cutoff strategy with a positive threshold is used to keep the solutions positive. Furthermore, a two-level moving mesh strategy and a sparse matrix solver are used to improve the efficiency of the computation. Numerical results for a selection of examples of multi-material non-equilibrium radiation diffusion show that the method is capable of capturing the profiles and local structures of Marshak waves with adequate mesh concentration. The obtained numerical solutions are in good agreement with those in the existing literature. Comparison studies are also made between uniform and adaptive moving meshes and between one-level and two-level moving meshes.

Keywords Moving mesh method · Non-equilibrium radiation diffusion · Predictor-corrector · Positivity · Cutoff · Two-level mesh movement

✉ Jianxian Qiu
jxqiu@xmu.edu.cn

Xiaobo Yang
xwindyb@126.com

Weizhang Huang
whuang@ku.edu

[1] Department of Mathematics, College of Science, China University of Mining Technology, Xuzhou, 221116, Jiangsu, China

[2] Department of Mathematics, University of Kansas, Lawrence, KS 66045, USA

[3] School of Mathematical Sciences, Xiamen University, Xiamen, 361005, Fujian, China

Mathematics Subject Classification (2010) 65M06 · 65M50

1 Introduction

Radiation transport in astrophysical phenomena and inertial confinement fusion can often be modeled using a set of coupled diffusion equations when photon mean free paths are much shorter than characteristic length scales. These equations are highly nonlinear and exhibit multiple time and space scales [25]. Particularly, steep hot wave fronts, called Marshak waves, typically form during radiation transport processes. Energy density and material temperature near the steep fronts can vary dramatically in a short distance. Such complex local structures make mesh adaptation an indispensable tool for use to improve the efficiency in the numerical solution of radiation diffusion equations because the number of mesh points can be prohibitively large when a uniform mesh is used. Research of radiation diffusion has attracted considerable attentions from engineers and scientists [2, 4, 17, 18, 25–29, 31–33, 35, 39–42].

In this work, we are interested in the non-equilibrium situation where the radiation field is not in thermodynamics equilibrium with the material temperature. Marshak [24] develops a time-dependent radiative transfer model, laying the groundwork for the research area. Pomraning [30] obtains an analytic solution to a particular Marshak wave problem, which is analyzed more extensively by Su and Olson [34]. Numerically, Mousseau et al. [26, 27] present a physics-based preconditioning Newton-Krylov method involving Jacobian-free Newton-Krylov (JFNK), operator splitting, and multigrid linear solvers and show that the method can capture the Marshak wave of the thermal transport front properly. Kang [17] proposes a $P1$ nonconforming finite element method for non-equilibrium radiation transport problems. Olson [29] considers a hydrogen-like Saha ionization model for a simplified but physically plausible heat capacity and uses several types of finite difference (FD) schemes to approximate flux-limiting. Sheng et al. [35] construct a monotone finite volume scheme for multi-material, non-equilibrium radiation diffusion equations and show numerically that their method is better than the standard nine-point finite difference scheme and preserves the nonnegativity of energy density.

On the other hand, there exist only a few published studies that have employed mesh adaptation for the numerical solution of radiation diffusion equations. For example, Lapenta and Chacón [18] use a fully implicit moving mesh method to solve a one-dimensional equilibrium radiation diffusion equation. They discretize both the mesh and physical equations using finite volumes and solve the resulting equation with a preconditioned inexact-Newton method. Their results show great improvements in cost-effectiveness with mesh adaptation. Yang et al. [41] study a moving mesh FD method based on the moving-mesh-partial-differential-equation (MMPDE) strategy [9, 11] for equilibrium radiation diffusion equations and show that the method capture Marshak waves accurately and efficiently. Pernice et al. [32] use adaptive mesh refinement to solve three-dimensional non-equilibrium radiation diffusion equations. They use implicit time integration for stiff multi-physics systems as well as the JFNK [15, 16, 26, 33] to solve the resulting nonlinear algebraic equations.

They also use an optimal multilevel preconditioner to provide level-independent solver convergence. Non-equilibrium radiation diffusion equations are challenging to solve, but through the numerical results, they demonstrate their method can efficiently capture the local structures of Marshak waves and can give convincing results with good accuracy.

The objective of this work is to study a moving mesh FD solution of two-dimensional non-equilibrium radiation diffusion systems. The method is based on the MMPDE moving mesh approach [9, 11], which belongs to a type of mesh adaptation method—dynamic mesh adaptation or r-adaptive methods (e.g., see recent works [6, 22, 23, 37, 38, 43]). The MMPDE is used to adaptively move the mesh around evolving features of the physical solution and is defined as the gradient flow equation of a meshing functional based on mesh equidistribution and alignment. The shape, size, and orientation of mesh elements are controlled through a monitor function [8] defined through the Hessian of the energy density. A similar moving mesh FD method has been developed in [41] for equilibrium radiation diffusion equations, and the current work can be considered as a generalization of [41]. However, this generalization is non-trival. Unlike [41], we now need to deal with a system of two coupled equations for the energy density and material temperature. The diffusion coefficients depend on both the energy density and material temperature and it is more sensitive to treat diffusion numerically. Moreover, the system is stiffer, making it more difficult to integrate in time (with smaller time steps) and more expensive to solve overall. Furthermore, it is more delicate to preserve the solution positivity. Like [41], we use here the cutoff strategy to maintain the positivity in the computed solutions. It has been shown in [21] that the strategy retains the accuracy and convergence order of FD approximation for parabolic PDEs. It has been found in [41] that the strategy with a threshold zero (meaning that the computed solutions are kept to be nonnegative) works for equilibrium radiation diffusion equations. For the current situation, on the other hand, we have found that a positive threshold is needed and an empirical choice depending on the mesh size seems to work well for problems we have tested. Numerical results for a selection of examples are presented. They show that the method is capable of capturing the profiles and local structures of Marshak waves with adequate mesh concentration. The obtained numerical solutions are in good agreement with those of [17, 35]. Comparison studies are also made between uniform and adaptive moving meshes and between one-level and two-level moving meshes.

The outline of the paper is as follows. The physical model and governing equations are described in Section!2. The moving mesh FD method and the treatments of nonlinearity as well as the cutoff strategy are discussed in Section 3. In Section 4, numerical results obtained for a selection of examples of multi-material, multiple spot concentration scenarios. Finally, conclusions are drawn in Section 5.

2 The 2T model for non-equilibrium radiation diffusion

Under the assumption of an optically thick medium (short mean free path of photons) a first-principle statement of radiation transport reduces to the radiation diffusion limit. A particular idealized dimensionless form of the governing system, known as

the 2T model, consists of two equations, the radiation diffusion (gray approximation) equation and material energy balance equation, that is,

$$\begin{cases} \frac{\partial E}{\partial t} - \nabla \cdot (D_r \nabla E) = \sigma_a (T^4 - E), \\ \frac{\partial T}{\partial t} - \nabla \cdot (D_t \nabla T) = -\sigma_a (T^4 - E), \end{cases} \tag{2.1}$$

where

$$\sigma_a = \frac{z^3}{T^3}, \quad D_r = \frac{1}{3\sigma_a + \frac{1}{E}|\nabla E|}, \quad D_t = \kappa\, T^{\frac{5}{2}}. \tag{2.2}$$

Here, E represents the photon energy, T is the material temperature, σ_a is the opacity, κ is the material conductivity, and z is the atomic mass number. Notice that a limiting term $|\nabla E|/E$ is added to the diffusion coefficient D_r to avoid a possible unphysical behavior that a flux of energy moves faster than the speed of light in regions of strong gradient where a simple diffusion theory can fail. Moreover, we use the form of the material (plasma) conduction diffusion coefficient D_t from Spitzer and Harm [36] and take $\kappa = 0.01$ in our computation. Furthermore, compared to the equilibrium case, the nonlinear source terms on the right-hand sides of the equations do not vanish in general, reflecting the transfer of energy between the radiation field and material temperature. Additional nonlinearities come from the particular form of diffusion coefficients, which are functions of E and T.

We consider (2.1) in two dimensions on the unit square domain, see Fig. 1. Homogenous Neumann boundary conditions are used for boundary segments $y = 0$

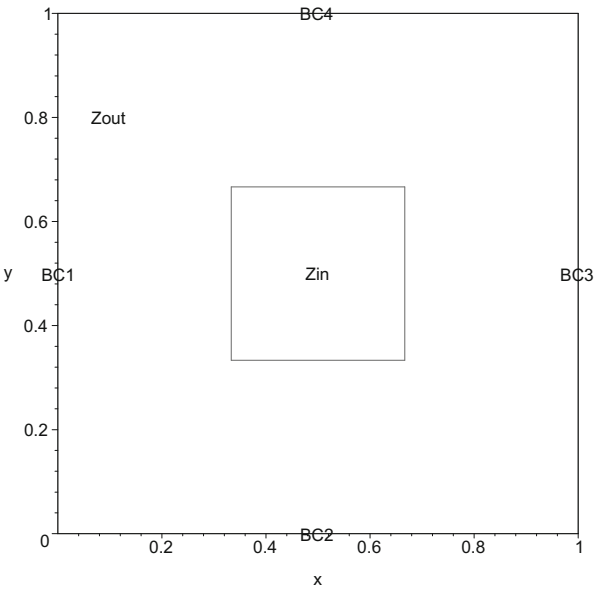


Fig. 1 The physical domain with a middle inset $(1/3, 2/3) \times (1/3, 2/3)$. The value of z is Z_{in} and Z_{out} inside and outside the inset, respectively. BC1 and BC2 are the inflow and outflow boundaries, respectively, while BC3 and BC4 are perfectly insulated

and $y = 1$ and inflow and outflow boundary conditions are employed on $x = 0$ and $x = 1$, respectively. More specifically, we have

$$\begin{cases} \frac{\partial E}{\partial y} = 0, \quad \frac{\partial T}{\partial y} = 0, & \text{on } y = 0 \text{ or } y = 1 \\ \frac{\partial T}{\partial x} = 0, & \text{on } x = 0 \text{ or } x = 1 \\ \frac{1}{4}E - \frac{1}{6\sigma_a}\frac{\partial E}{\partial x} = 1, & \text{on } x = 0 \\ \frac{1}{4}E + \frac{1}{6\sigma_a}\frac{\partial E}{\partial x} = 0, & \text{on } x = 1. \end{cases} \tag{2.3}$$

The initial conditions are

$$\begin{cases} E(x, y, 0) = (1 - \tanh(10x))(1 - 10^{-5}) + 10^{-5}, & (x, y) \in \Omega \\ T(x, y, 0) = E(x, y, 0)^{\frac{1}{4}}, & (x, y) \in \Omega. \end{cases} \tag{2.4}$$

Essentially, $E(x, y, 0)$ is equal to 10^{-5} everywhere except on the boundary $x = 0$ where it is 1. A narrow transition between 10^{-5} and 1 is used to avoid a potentially difficult initial start in numerical computation. (Slightly different initial and boundary conditions are used in the third example in Section 4.)

3 The moving mesh FD method

In this section, we describe the moving mesh FD method for solving the initial-boundary value problem (IBVP) (2.1), (2.3), and (2.4). We discretize this problem in space using central finite differences and in time using a Singly Diagonally Implicit Runge-Kutta scheme (SDIRK) [3]. We also discuss linearization of the equations, preservation of solution positivity, and adaptive mesh movement.

3.1 FD discretization on moving meshes

We denote a curvilinear moving mesh for Ω by

$$(x_{m,n}(t),\ y_{m,n}(t)), \quad m = 1, ..., M,\ n = 1, ..., N \tag{3.1}$$

where M and N are positive integers. The generation of such an adaptive moving mesh will be described in §3.4. For the moment, we consider (3.1) as the image of a fixed rectangular mesh under a known coordinate transformation $x = x(\xi, \eta, t)$, $y = y(\xi, \eta, t)$, i.e.,

$$x_{m,n}(t) = x(\xi_m, \eta_n, t), \quad y_{m,n}(t) = y(\xi_m, \eta_n, t), \quad m = 1, ..., M, \quad n = 1, ..., N \tag{3.2}$$

where the reference mesh is taken as

$$(\xi_m, \eta_n) = ((m - 1)\Delta\xi, (n - 1)\Delta\eta), \quad m = 1, ..., M, \quad n = 1, ..., N \tag{3.3}$$

and $\Delta\xi = 1/(M - 1)$ and $\Delta\eta = 1/(N - 1)$. The boundary correspondence between the reference and physical domains is given by

$$x(0, \eta) = 0, \quad x(1, \eta) = 1, \quad y(\xi, 0) = 0, \quad y(\xi, 1) = 1. \tag{3.4}$$

We let

$$\hat{E}(\xi, \eta, t) = E(x(\xi, \eta, t), y(\xi, \eta, t), t), \qquad \hat{T}(\xi, \eta, t) = T(x(\xi, \eta, t), y(\xi, \eta, t), t).$$

The discretization of the 2T model on the moving mesh (3.1) consists of two steps, its transformation from (x, y) to (ξ, η) and discretization on the rectangular reference mesh. First, using the coordinate transformation, we can transform (2.1) (e.g., see [11, §3.1.4]) into the reference domain as

$$\begin{cases} \hat{E}_t - \boldsymbol{b}(t)\cdot\hat{\nabla}\hat{E} = \frac{1}{J(t)}\hat{\nabla}\cdot\left(D_R(\hat{E},\hat{T})A(t)\hat{\nabla}\hat{E}\right) + J(t)\sigma_a\left(\hat{T}^4 - \hat{E}\right), \\ \hat{T}_t - \boldsymbol{b}(t)\cdot\hat{\nabla}\hat{T} = \frac{1}{J(t)}\hat{\nabla}\cdot\left(D_T(\hat{T})A(t)\hat{\nabla}\hat{T}\right) - J(t)\sigma_a\left(\hat{T}^4 - \hat{E}\right), \end{cases} \tag{3.5}$$

where

$$J(t) = x_\xi y_\eta - x_\eta y_\xi, \quad \hat{\nabla} = \begin{bmatrix} \frac{\partial}{\partial\xi} \\ \frac{\partial}{\partial\eta} \end{bmatrix},$$

$$\boldsymbol{b}(t) = \frac{1}{J(t)}\begin{bmatrix} y_\eta x_t - x_\eta y_t \\ -y_\xi x_t + x_\xi y_t \end{bmatrix}, \quad A(t) = \frac{1}{J(t)}\begin{bmatrix} x_\eta^2 + y_\eta^2 & -(x_\xi x_\eta + y_\xi y_\eta) \\ -(x_\xi x_\eta + y_\xi y_\eta) & x_\xi^2 + y_\xi^2 \end{bmatrix}.$$

Similarly, the boundary condition (2.3) can be transformed into

$$\begin{cases} \frac{-x_\eta}{x_\xi y_\eta}\frac{\partial\hat{E}}{\partial\xi} + \frac{1}{y_\eta}\frac{\partial\hat{E}}{\partial\eta} = 0, \quad \frac{-x_\eta}{x_\xi y_\eta}\frac{\partial\hat{T}}{\partial\xi} + \frac{1}{y_\eta}\frac{\partial\hat{T}}{\partial\eta} = 0, & \text{on } \eta = 0 \text{ or } \eta = 1 \\ \frac{-y_\xi}{x_\xi y_\eta}\frac{\partial\hat{T}}{\partial\eta} + \frac{1}{x_\xi}\frac{\partial\hat{T}}{\partial\xi} = 0, & \text{on } \xi = 0 \text{ or } \xi = 1 \\ \frac{1}{4}\hat{E} - \frac{1}{6\sigma_a}\left(\frac{1}{x_\xi}\frac{\partial\hat{E}}{\partial\xi} + \frac{-y_\xi}{x_\xi y_\eta}\frac{\partial\hat{E}}{\partial\eta}\right) = 1, & \text{on } \xi = 0 \\ \frac{1}{4}\hat{E} + \frac{1}{6\sigma_a}\left(\frac{1}{x_\xi}\frac{\partial\hat{E}}{\partial\xi} + \frac{-y_\xi}{x_\xi y_\eta}\frac{\partial\hat{E}}{\partial\eta}\right) = 0, & \text{on } \xi = 1 \end{cases} \tag{3.6}$$

where we have used the fact that $x_\eta = 0$ on $\xi = 0$ and $\xi = 1$ and $y_\xi = 0$ on $\eta = 0$ and $\eta = 1$.

The discretization of (3.5) and (3.6) on the rectangular reference mesh (3.3) using central finite differences is straightforward. To save space, we omit the detail of the derivation and formulation of the FD approximation here and refer the reader to [11, §3.2]. The FD approximation of (3.5) can be expressed as

$$\begin{cases} \frac{dE_h}{dt} - \boldsymbol{b}_h(t)\cdot\hat{\nabla}_h E_h = \frac{1}{J_h(t)}\hat{\nabla}_h\cdot\left(D_R(E_h,T_h)A_h(t)\hat{\nabla}_h E_h\right) + J_h(t)\sigma_a\left(T_h^4 - E_h\right), \\ \frac{dT_h}{dt} - \boldsymbol{b}_h(t)\cdot\hat{\nabla}_h T_h = \frac{1}{J_h(t)}\hat{\nabla}_h\cdot\left(D_T(T_h)A_h(t)\hat{\nabla}_h T_h\right) - J_h(t)\sigma_a\left(T_h^4 - E_h\right), \end{cases} \tag{3.7}$$

where E_h and T_h denote the FD approximations of $\hat{E}(\xi,\eta,t)$ and $\hat{T}(\xi,\eta,t)$ on the mesh (3.3), respectively.

3.2 Linearization and predictor-corrector approximation

Recall that the 2T model (3.7) has nonlinear diffusion coefficients. Integration of nonlinear radiation diffusion equations have been studied extensively in the past (e.g., see [13, 14, 20, 28, 31, 33]). Generally speaking, there are three types of methods for treating nonlinear diffusion terms [20]: the Beaming-Warming method, lagged diffusion, and predictor-corrector method. For the Beaming-Warming method, the diffusion coefficient is expanded up to linear term of E and T at the pervious time step and it is a second-order approximation to the diffusion equations. For lagged diffusion, the diffusion coefficient is simply calculated with the energy and material temperature at the pervious time step and it is only a first-order approximation. The

predictor-corrector method uses the lagged diffusion as the predictor while adding a corrector step so it gives a second-order approximation.

In this paper, we use the predictor-corrector method for solving non-equilibrium systems since it is comparable to the Beam-Warming method in terms of accuracy and stability and to lagged diffusion in terms of simplicity and efficiency. With the method, the linearized equation of (3.7) reads as

$$\begin{cases} \frac{dE_h}{dt} - \boldsymbol{b}_h(t)\cdot\hat{\nabla}_h E_h = \frac{1}{J_h(t)}\hat{\nabla}_h\cdot\left(D_L(E_h^*,T_h^*)A_h(t)\hat{\nabla}_h E_h\right) \\ +J_h(t)\sigma_a\left(T_h^4 - E_h\right), \quad t_n < t \le t_{n+1} \\ \frac{dT_h}{dt} - \boldsymbol{b}_h(t)\cdot\hat{\nabla}_h T_h = \frac{1}{J_h(t)}\hat{\nabla}_h\cdot\left(D_T(T_h^*)A_h(t)\hat{\nabla}_h T_h\right) \\ -J_h(t)\sigma_a\left(T_h^4 - E_h\right), \quad t_n < t \le t_{n+1} \\ E_h(t_n) = E_h^n \\ T_h(t_n) = T_h^n \end{cases} \tag{3.8}$$

where E_h^n and T_h^n are the approximations of the energy density and temperature at $t = t_n$. During the prediction stage, E_h^* and T_h^* are taken as the energy density and material temperature at $t = t_n$, i.e., $E_h^* = E_h^n$ and $T_h^* = T_h^n$. This stage is the same as the lagged diffusion method. The solution obtained in this stage at $t = t_{n+1}$ is used as E_h^* and T_h^* during the correction stage. In both stages, the linear equation (3.8) is integrated with a two-stage SDIRK scheme [3]. The resulting linear systems are solved by the unsymmetric multifrontal sparse LU factorization package UMFPACK [5].

3.3 Preservation of solution positivity and cutoff

It is known that the solutions of IBVP (2.1), (2.3), and (2.4) stay positive for all time. Unfortunately, the scheme described in the previous subsections does not preserve the solution positivity and the computed solutions may become zero or even negative at places. Although these values can be very small in magnitude, they can cause nonphysical oscillations and other problems such as not-a-number (NaN), divergence of nonlinear iterations, too small time steps, and even early blowup of computation [39]. We employ here a cutoff strategy, i.e., replace solution values that are below a positive threshold by the threshold. Unfortunately, no theory exists so far on how to choose such a threshold. An empirical formula is $30/((M-1)(N-1))$ (see Table 1) which has been found to work well for the examples we consider. Noticeably, Lu et al. [21] show that the cutoff procedure can retain accuracy, convergence order, and stability of finite difference schemes for linear or nonlinear parabolic PDEs.

Table 1 The values of the cutoff threshold defined as $\frac{30}{(M-1)(N-1)}$ for various mesh sizes

Mesh ($M \times N$)	41×41	61×61	81×81	121×121
Cutoff threshold	1.87e-2	8.30e-3	4.70e-3	2.10e-3

3.4 The MMPDE approach of mesh movement

The MMPDE approach [9–11] is used here to generate the adaptive moving mesh. The main idea of the approach is to generate the moving mesh as the image of a fixed, reference mesh under a time coordinate transformation. Such a coordinate transformation is determined as the solution of an MMPDE which in turn is defined as the gradient flow equation of a meshing functional. We use a meshing functional formulated in terms of the inverse coordinate transformation $\xi = \xi(x, y, t)$ and $\eta = \eta(x, y, t)$ and based on mesh equidistribution and alignment. A monitor function that is symmetric and uniformly positive definite at each point of the domain is used in the functional to provide the information for the size, shape, and orientation of the mesh elements. Denote the Hessian of the energy density by

$$H = \begin{bmatrix} E_{xx} & E_{xy} \\ E_{xy} & E_{yy} \end{bmatrix}.$$

Given its eigen-decomposition $H = Q\text{diag}(\lambda_1, \lambda_2)Q^T$, we define $|H| = Q\text{diag}(|\lambda_1|, |\lambda_2|)Q^T$. Then, the monitor function is chosen as

$$\mathbb{M} = \det(\alpha I + |H|)^{-\frac{1}{4}} \Big[\alpha I + |H| \Big], \tag{3.9}$$

which is known to be optimal for the H^1 norm of the error of linear interpolation [11, 12]. Here, $\alpha > 0$ is the regularization parameter defined through the equation

$$\int_\Omega \det(\mathbb{M}(\alpha))^{\frac{1}{2}} dxdy = 2 \int_\Omega \det(\mathbb{M}(0))^{\frac{1}{2}} dxdy,$$

where $\mathbb{M}(0)$ denotes the monitor function (3.9) with $\alpha = 0$. In practical computation, the Hessian of E is unknown. It is replaced by an approximation based on E_h (see Section 3.5 for a more detailed description). The meshing functional is given by

$$\begin{aligned} I[\xi, \eta] = 0.1 \int_\Omega \det(\mathbb{M})^{\frac{1}{2}} \left(\nabla\xi^T \mathbb{M}^{-1} \nabla\xi + \nabla\eta^T \mathbb{M}^{-1} \nabla\eta \right)^2 dxdy \\ +3.2 \int_\Omega \frac{\det(\mathbb{M})^{\frac{1}{2}}}{\left(J \det(\mathbb{M})^{\frac{1}{2}} \right)^2} dxdy, \end{aligned} \tag{3.10}$$

where $J = x_\xi y_\eta - x_\eta y_\xi = 1/(\xi_x \eta_y - \xi_y \eta_x)$ is the Jacobian of the coordinate transformation. This functional is proposed in [8] to control mesh equidistribution and alignment.

The MMPDE is defined as the gradient flow equation of the meshing functional, i.e.,

$$\frac{\partial \xi}{\partial t} = -\frac{1}{\tau} \frac{\delta I}{\delta \xi}, \quad \frac{\partial \eta}{\partial t} = -\frac{1}{\tau} \frac{\delta I}{\delta \eta}, \tag{3.11}$$

where τ is a parameter used to control the response of the mesh movement to the change in the monitor function and $\delta I/\delta\xi$ and $\delta I/\delta\eta$ are the functional derivatives of $I[\xi, \eta]$. It is not difficult to find that

$$\frac{\delta I}{\delta \xi} = -4\theta\nabla\cdot\left(\det(\mathbb{M})^{\frac{1}{2}}\beta\mathbb{M}^{-1}\nabla\xi\right) - 8(1-2\theta)\nabla\cdot\left(\frac{1}{J\det(\mathbb{M})^{\frac{1}{2}}}\begin{bmatrix}\eta_y\\ -\eta_x\end{bmatrix}\right), \tag{3.12}$$

$$\frac{\delta I}{\delta \eta} = -4\theta\nabla\cdot\left(\det(\mathbb{M})^{\frac{1}{2}}\beta\mathbb{M}^{-1}\nabla\eta\right) - 8(1-2\theta)\nabla\cdot\left(\frac{1}{J\det(\mathbb{M})^{\frac{1}{2}}}\begin{bmatrix}-\xi_y\\ \xi_x\end{bmatrix}\right), \tag{3.13}$$

where

$$\beta = \nabla\xi^T\mathbb{M}^{-1}\nabla\xi + \nabla\eta^T\mathbb{M}^{-1}\nabla\eta. \tag{3.14}$$

By interchanging the roles of independent and dependent variables and after some straightforward but lengthy derivations (e.g., see [11, Chapter 6]), we can rewrite the above equation into

$$\frac{\partial}{\partial t}\begin{bmatrix}x\\ y\end{bmatrix} = \frac{1}{\tau}\left(A_{11}\frac{\partial^2}{\partial\xi^2} + (A_{12}+A_{21})\frac{\partial^2}{\partial\xi\,\partial\eta} + A_{22}\frac{\partial^2}{\partial\eta^2} + I_d\, b_1\frac{\partial}{\partial\xi} + I_d\, b_2\frac{\partial}{\partial\eta}\right)\begin{bmatrix}x\\ y\end{bmatrix}, \tag{3.15}$$

where I_d is the 2-by-2 identity matrix and the coefficients A_{ij}, b_1, and b_2 can be found in [11, Chapter 6].

The moving mesh equation (3.15) is supplemented with the one-dimensional version of the MMPDE for the adaptation of boundary points (cf. [8]). They are discretized in space using central finite differences and in time by the backward Euler method with coefficients A_{ij} and b_i calculated at the previous time step. The resulting algebraic systems are solved using the sparse matrix solver UMFPACK [5].

3.5 The solution procedure

We now describe the overall solution procedure of the moving mesh FD method. Assume that the physical solutions E^n and T^n, the mesh (x^n, y^n), and the time step size Δt_n are given at $t = t_n$.

Step 1. The moving mesh step. The monitor function (3.9) is computed using E^n and (x^n, y^n) and smoothed using several sweeps of a low-pass filter. The Hessian of the energy density used in (3.9) is replaced by an approximation obtained using least squares fitting. More specifically, at any mesh point, a local quadratic polynomial is constructed by least squares fitting of the nodal values of E^n at neighboring mesh points. The approximate Hessian at the given mesh point is then obtained by differentiating the quadratic polynomial twice. After the monitor function has been obtained, the mesh equation (3.15) is integrated from t_n to $t_{n+1} = t_n + \Delta t_n$ for the new mesh (x^{n+1}, y^{n+1}).

Step 2. The predictor of the physical PDE solving step. The physical PDE (3.7) is integrated at t_n using the predictor-corrector scheme (3.8) with $E_h^* = E_h^n$ and $T_h^* = T_h^n$. During the integration, the mesh is considered to move linearly in time, viz.,

$$x(t) = \frac{t - t_n}{\Delta t_n} x^{n+1} + \frac{t_n + \Delta t_n - t}{\Delta t_n} x^n, \qquad y(t) = \frac{t - t_n}{\Delta t_n} y^{n+1} + \frac{t_n + \Delta t_n - t}{\Delta t_n} y^n. \tag{3.16}$$

Step 3. The corrector of the physical PDE solving step. The physical PDE (3.7) is integrated from t_n to t_{n+1} using the predictor-corrector scheme (3.8) with E_h^* and T_h^* being taken as the solutions obtained in Step 2.

3.6 Stability

As we have seen from the previous subsection, the whole system consisting of the mesh and physical equations is solved in an alternating manner: the mesh equation (3.15) is integrated using the backward Euler scheme with the coefficients being calculated at the previous step while the physical PDE (3.7) is integrated with the predictor-corrector scheme (3.8). It is extremely difficult, if not impossible, to perform a von Neumann analysis for the stability of the overall scheme due to highly nonlinear couplings between the mesh and the physical solution. With the implicit treatments of each of the mesh and physical equations and due to their parabolic nature, we can expect that the method has a good stability. Indeed, we have not experienced any stability issue in our computation at least for the range of the time step size we have used.

4 Numerical tests

4.1 Conservation analysis

In this subsection, we test the conservation of energy E and temperature T. Define the radiation temperature as $T_r = E^{1/4}$. We want to see if the material temperature T stays close to the radiation temperature T_r as time being.

Example 4.1 For this example, the distribution of the atomic mass number is given by (cf. Fig. 7a)

$$z(x, y) = \begin{cases} 5, & \text{for } (x, y) \in (\frac{1}{3}, \frac{2}{3}) \times (\frac{1}{3}, \frac{2}{3}) \\ 1, & \text{otherwise.} \end{cases} \tag{4.1}$$

The boundary and initial conditions are given by (2.3) and (2.4), respectively. We stop the computation at $t = 3.0$ and use the cutoff method discussed in §3.3. The results obtained with mesh 41×41 and 81×81 are plotted in Fig. 2. Although there are some

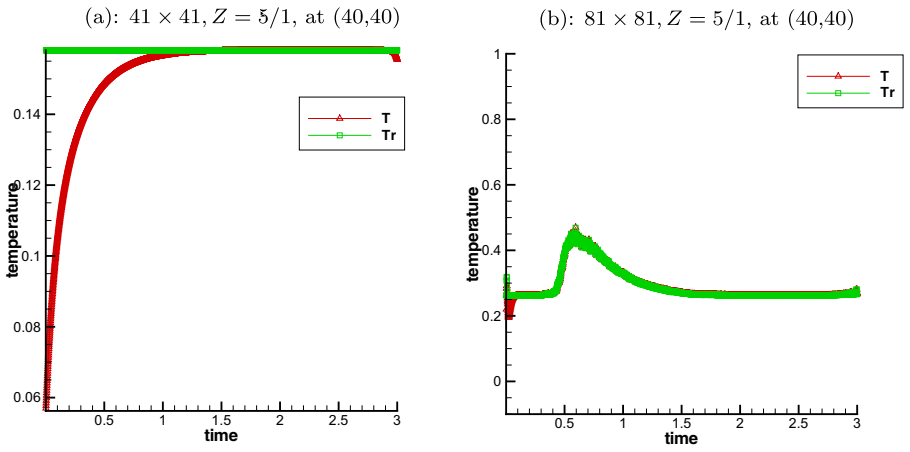


Fig. 2 Example 4.1 The comparison between material temperature T and radiation temperature T_r

oscillations at the beginning and end of the computation, the material temperature T stays close with the radiation temperature T_r for most of the time.

Example 4.2 For this example, the distribution of the atomic mass number is given by (cf. Fig. 7b)

$$z(x, y) = \begin{cases} 10, & \text{for } (x, y) \in (\frac{1}{3}, \frac{2}{3}) \times (\frac{1}{3}, \frac{2}{3}) \\ 1, & \text{otherwise.} \end{cases} \tag{4.2}$$

The setting for the computation is the same as for the previous example. The results obtained with a mesh 81×81 are plotted in Fig. 3. Once again, one can see that with

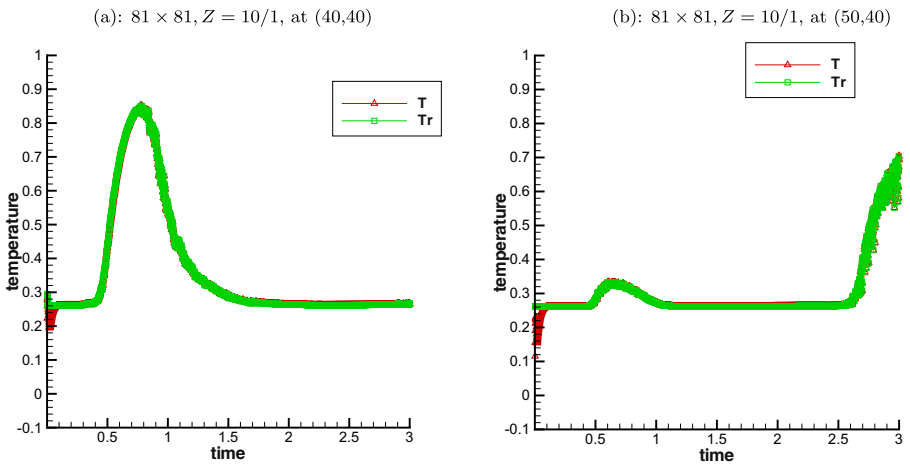


Fig. 3 Example 4.2. The comparison between the material temperature T and the radiation temperature T_r

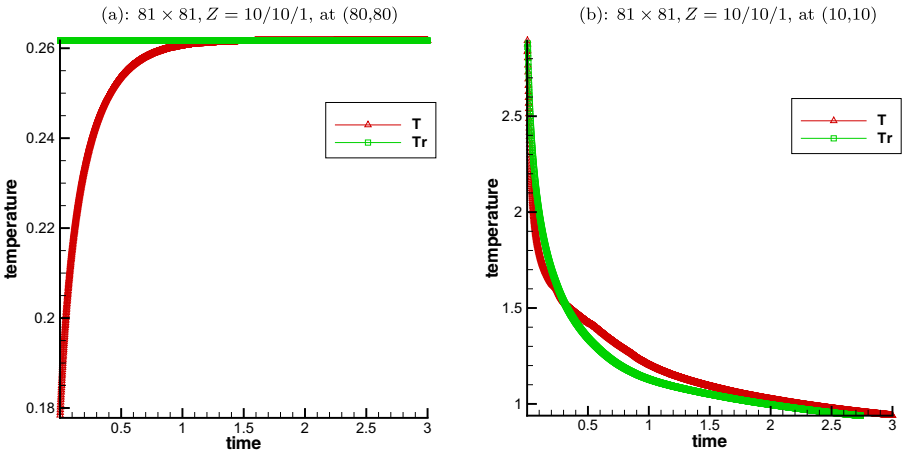


Fig. 4 Example 4.3. The comparison between material temperature T and radiation temperature T_r

the time going on, the material temperature T stays close to the radiation temperature T_r.

Example 4.3 The material configuration for this example is shown in Fig. 18. The insets are $(3/16, 7/16) \times (9/16, 13/16)$ and $(9/16, 13/16) \times (3/16, 7/16)$ and the distribution of the atomic mass number is given as

$$z(x, y) = \begin{cases} 10, & \text{for } (x, y) \in (\frac{3}{16}, \frac{7}{16}) \times (\frac{9}{16}, \frac{13}{16}) \\ 10, & \text{for } (x, y) \in (\frac{9}{16}, \frac{13}{16}) \times (\frac{3}{16}, \frac{7}{16}) \\ 1, & \text{otherwise.} \end{cases} \tag{4.3}$$

The boundary of $\Omega = (0, 1) \times (0, 1)$ is considered as insulated with respect to both radiation and material conduction, i.e.,

$$\frac{\partial E}{\partial n} = \frac{\partial T}{\partial n} = 0, \quad \text{on } \partial\Omega. \tag{4.4}$$

The initial condition is taken as (cf. [35])

$$E(x, y, 0) = 0.001 + 100 \exp\left(-100(x^2 + y^2)\right), \quad T(x, y, 0) = E(x, y, 0)^{\frac{1}{4}}. \tag{4.5}$$

The results can be seen form Fig. 4. Once again, we can see that the material temperature T stays close to the radiation temperature T_r.

4.2 Accuracy test

In this subsection, we study the accuracy of our moving mesh method. We consider two types of initial conditions: smooth conditions and Marshak wave conditions.

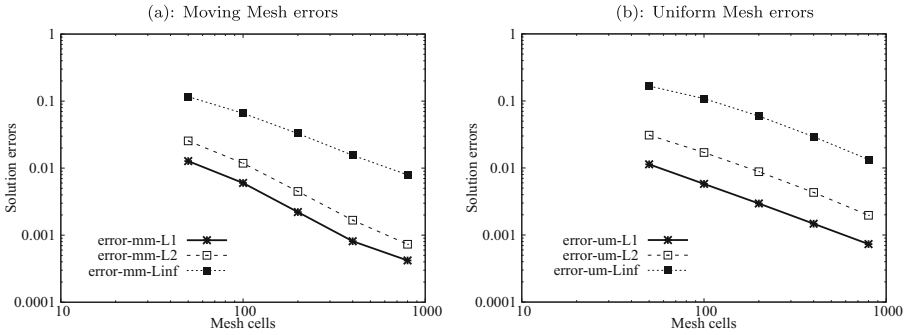


Fig. 5 Example 4.4. The L_1, L_2, L_∞ errors of moving mesh and uniform mesh solutions for smooth initial condition

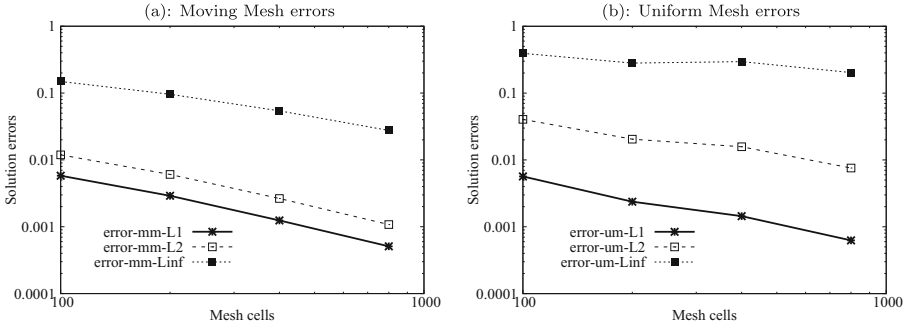


Fig. 6 Example 4.5. The L_1, L_2, L_∞ errors of moving mesh and uniform mesh solutions with Marshak initial condition

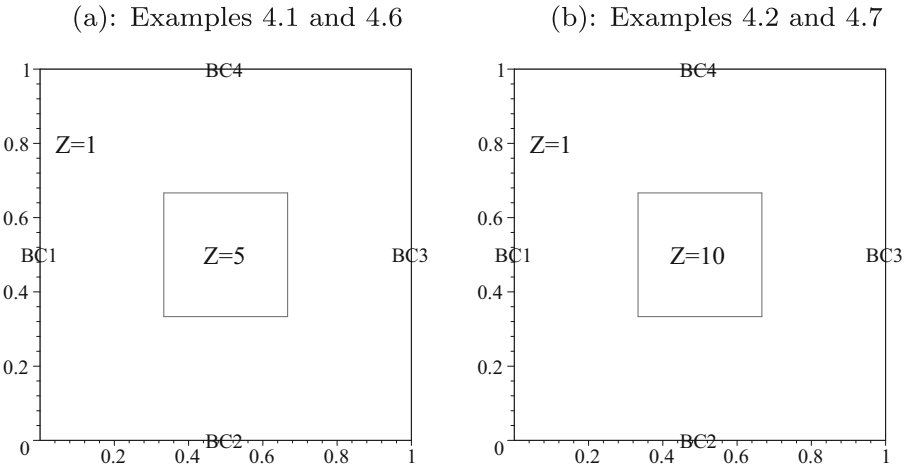


Fig. 7 Material configuration for Examples 4.1, 4.2, 4.6, and 4.7

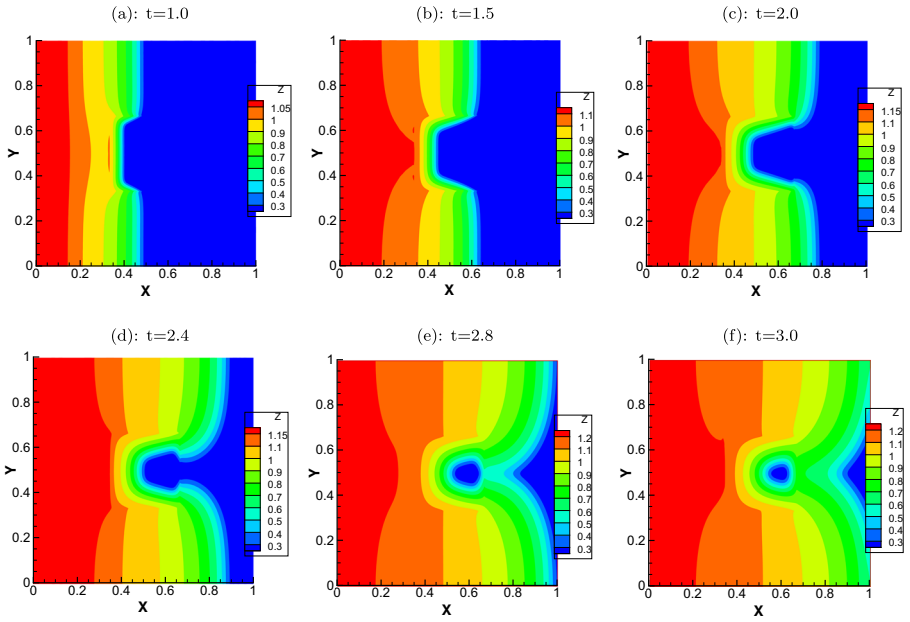


Fig. 8 Example 4.6. The computed solution at $t = 1.0, 1.5, 2.0, 2.4, 2.8, 3.0$ is obtained with a moving mesh of 81×81

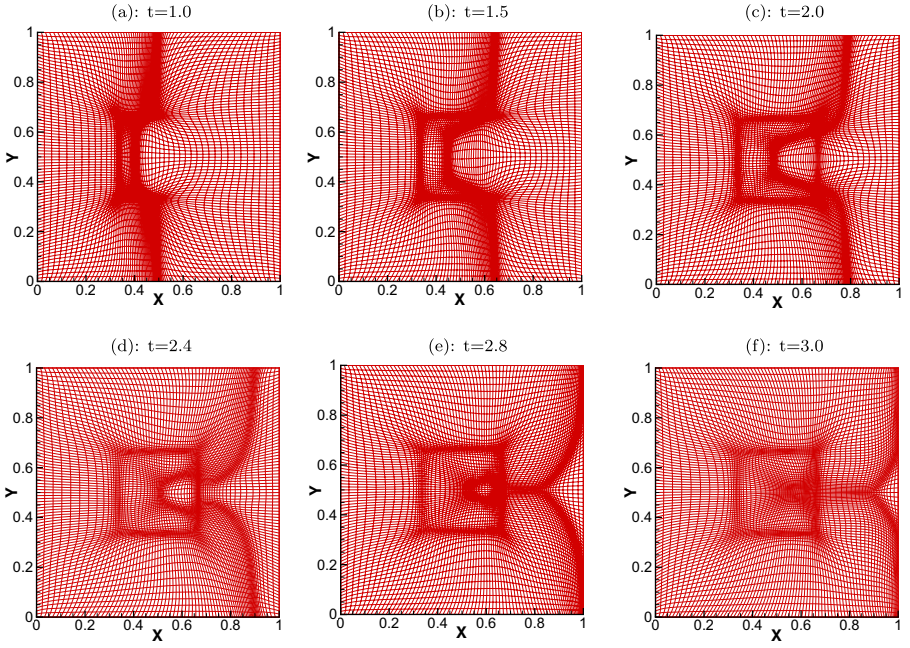


Fig. 9 Example 4.6 The moving mesh (81×81) is shown at $t = 1.0, 1.5, 2.0, 2.4, 2.8, 3.0$

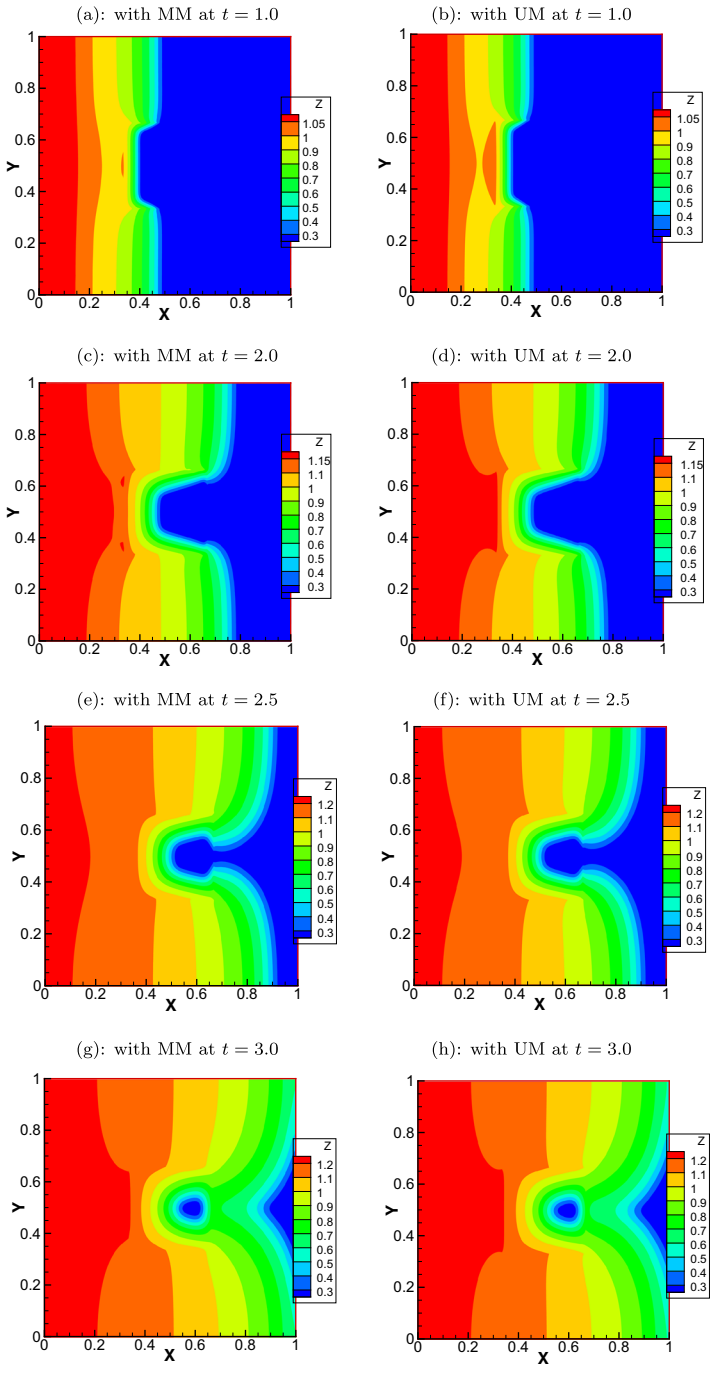


Fig. 10 Example 4.6. The contours of the temperature obtained with a moving mesh (MM) of size 61×61 are compared with those obtained with a uniform mesh (UM) of size 121×121

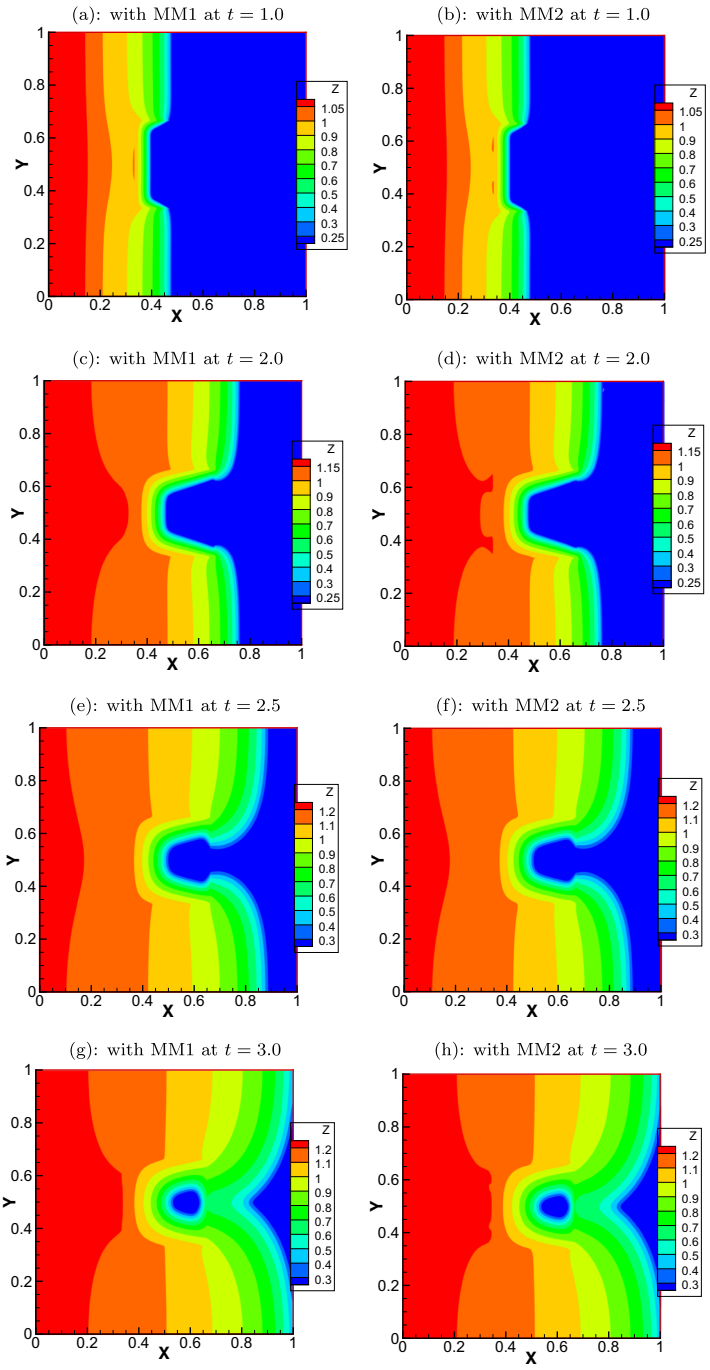


Fig. 11 Example 4.6. The contours of the temperature obtained with an MM1 of size 121 × 121 are compared with those obtained with an MM2 of size 41 × 41 (with the physical PDE being solved on a mesh of size 121 × 121

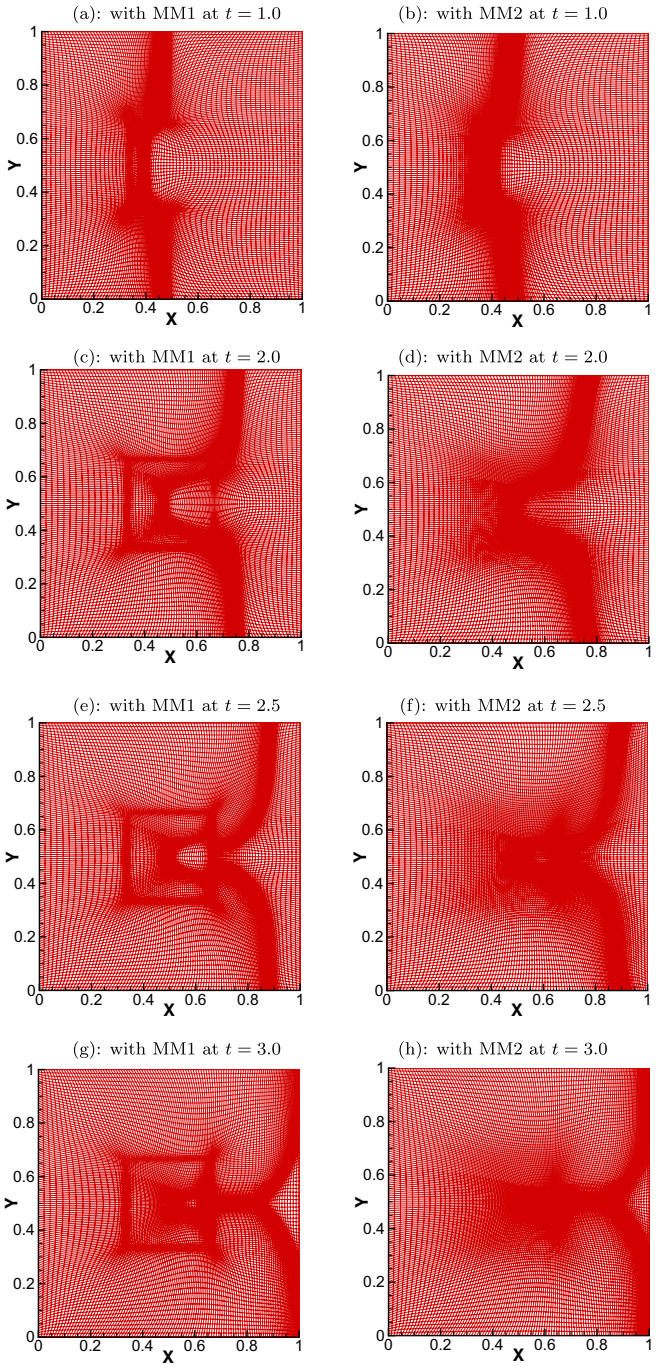


Fig. 12 Example 4.6. Moving meshes of size 121 × 121 obtained with MM1 and MM2 moving mesh strategies. MM2 is obtained by uniformly interpolating a 41 × 41 moving mesh

Table 2 CPU time comparison among one-level and two-level moving mesh methods and the uniform mesh method for Example 4.6. The CPU time is measured in seconds. The last column is the ratio of the used CPU time to that used with a uniform mesh of the same size

	Fine mesh	Coarse mesh	Total CPU time	Ratio
One-level MM	41×41	41×41	2544	5.32
	81×81	81×81	33724	14.81
	121×121	121×121	356720	63.91
Tow-level MM	41×41	41×41	2544	5.32
	81×81	41×41	8549	3.76
	121×121	41×41	20325	3.64
Fixed mesh	41×41	n/a	478	1
	81×81	n/a	2276	1
	121×121	n/a	5581	1

4.2.1 Smooth conditions in one-dimensional case

Example 4.4 We first consider the smooth initial condition as

$$\begin{cases} E(x,0) = E_L + (E_R - E_L)(1 + \tanh(50(x - 0.25)))/2, & x \in [0,1] \\ T(x,0) = E(x,0)^{1/4}, & x \in [0,1] \end{cases} \tag{4.6}$$

and the boundary condition as

$$\begin{cases} \frac{1}{4}E(0,t) - \frac{1}{6\sigma}(\partial_x E)(0,t) = \frac{1}{4}E_L, & x = 0 \\ \frac{1}{4}E(0,t) + \frac{1}{6\sigma}(\partial_x E)(0,t) = \frac{1}{4}E_R, & x = 1 \end{cases} \tag{4.7}$$

where $E_L = 4$ and $E_R = 0.004$. The exact solution for this example is not available, and we use the numerical solution obtained with a uniform mesh of 4000 as the reference solution. The error is plotted in Fig. 5.

Here, we use the number of mesh cells as 50, 100, 200, 400, and 800 for both uniform and moving meshes. From Fig. 5, we can see that both uniform mesh and moving mesh solutions are convergent. Upon a closer look at the figure, one can find that the L_1, L_2, and L_∞ norm of the error of moving mesh solutions are much smaller than those of uniform mesh solutions. The former seems to decrease faster too.

4.2.2 Marshak wave conditions in one-dimensional case

Example 4.5 In this example, we use the Marshak wave initial condition

$$\begin{cases} E(x,0) = E_L + (E_R - E_L)(1 + \tanh(50(x - 0.08)))/2, & x \in [0,1] \\ T(x,0) = E(x,0)^{1/4}, & x \in [0,1] \end{cases} \tag{4.8}$$

and the boundary condition (4.7) with $E_L = 4.0, E_R = 10^{-5}$. The solution of this problem is called a Marshak wave solution. We use the mesh of 100, 200, 400, and 800 cells and plot the error in Fig. 6. One can see that the error is smaller and converges faster with moving meshes than uniform meshes.

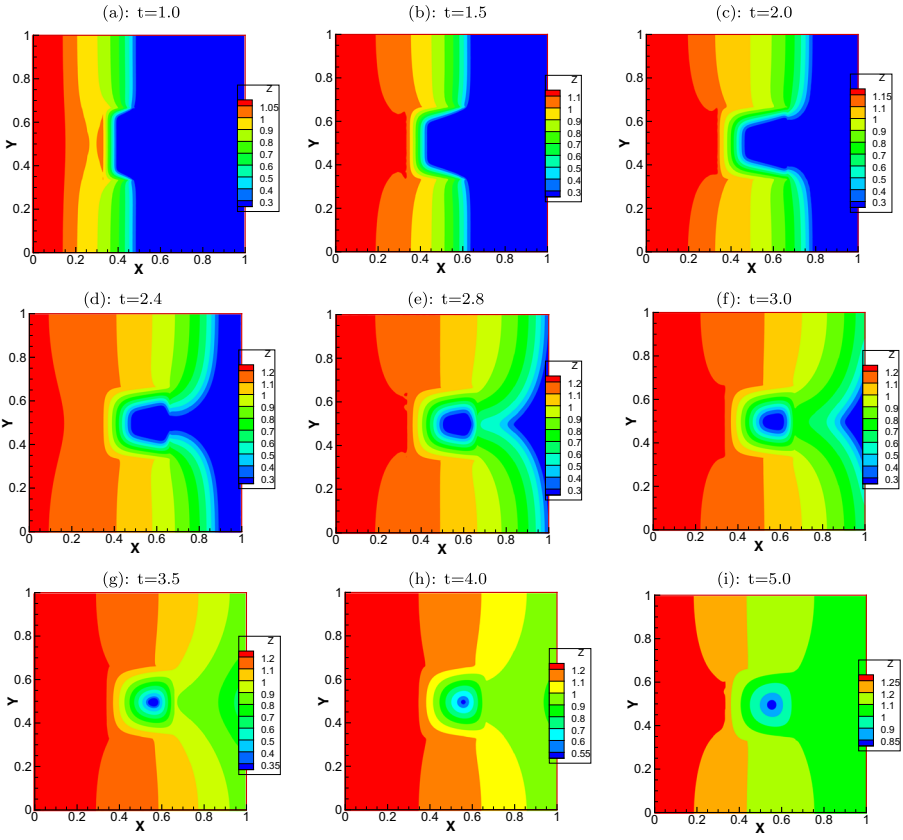


Fig. 13 Example 4.7. The computed solution with a moving mesh of 81 × 81 is shown at $t = 1.0, 1.5, 2.0, 2.4, 2.8, 3.0, 3.5, 4.0, 5.0$

4.3 Discontinuous Test

In this section, we present numerical results obtained by the moving mesh FD method described in the previous section for three examples of multi-material radiation diffusion. The material configuration is given in Fig. 7 for the first two examples and in Fig. 18 for the third one (which also has a slightly different boundary condition than (2.3)). In the results, MM, MM1, and MM2 stand for moving mesh, one-level moving mesh, and two-level moving mesh, respectively.

Example 4.6 For this example, the distribution of the atomic mass number is given by (4.1). The initial and boundary conditions are given in (2.3) and (2.4), respectively.

A typical moving mesh of 81 × 81 and the computed solution thereon are shown in Figs. 8 and 9. From the figures, we can see that the hot wave front propagates from left to right and meets the central obstacle and then a Marshak wave is formed. The profile of the Marshak wave has been captured accurately by the moving mesh and

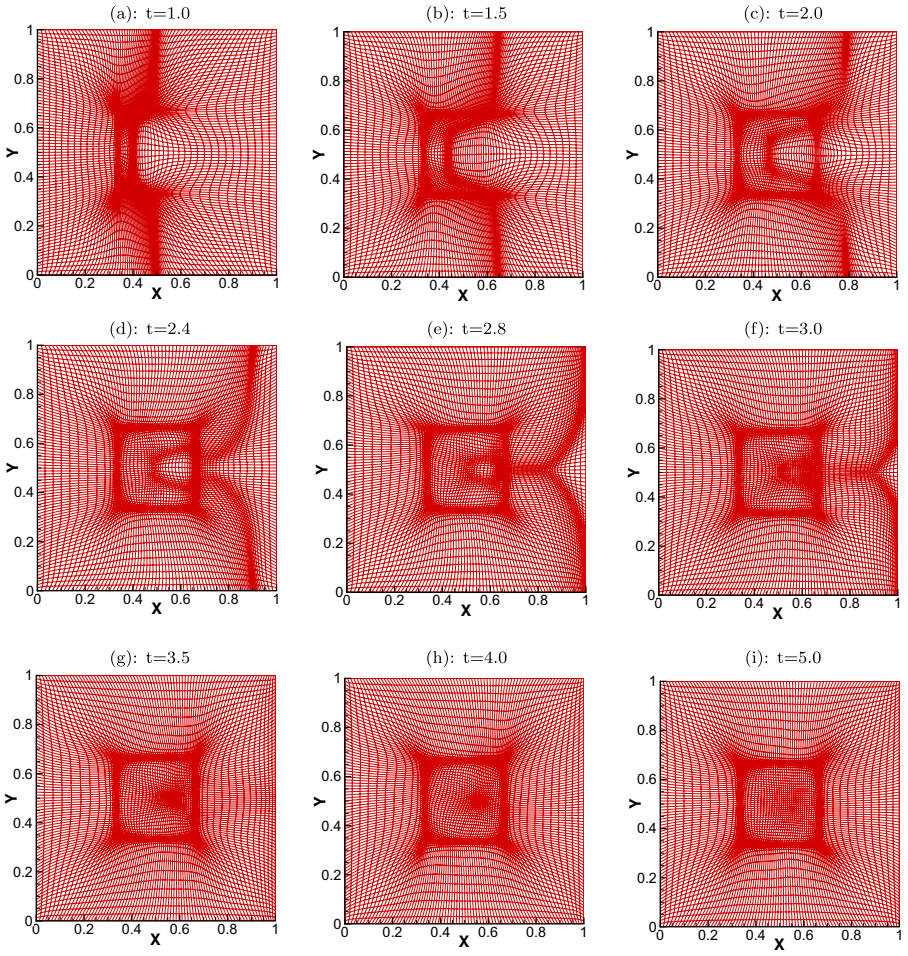


Fig. 14 Example 4.7. The moving mesh of 81×81 is shown at $t = 1.0, 1.5, 2.0, 2.4, 2.8, 3.0, 3.5, 4.0, 5.0$

the nodes concentrate around the front of the wave. This demonstrates the mesh concentration ability of the moving mesh method. Fig. 10 shows the solutions obtained with a moving mesh of 61×61 and a uniform mesh of 121×121, which are comparable.

The results obtained with a moving mesh of 121×121 are compared in Figs. 11 and 12 to those obtained with a two-level moving mesh strategy (MM2) [7] where a mesh of size 41×41 is moved using the moving mesh method, but the physical PDEs are solved on a mesh of 121×121 that is generated by uniformly refining the moving mesh. Interestingly, MM2 leads to results with comparable accuracy but saves significant CPU time. The CPU times for one-level and two-level moving meshes and uniform meshes are listed in Table 2. From the table, one can see that the moving mesh method is more costly than the method with a uniform mesh of the same size.

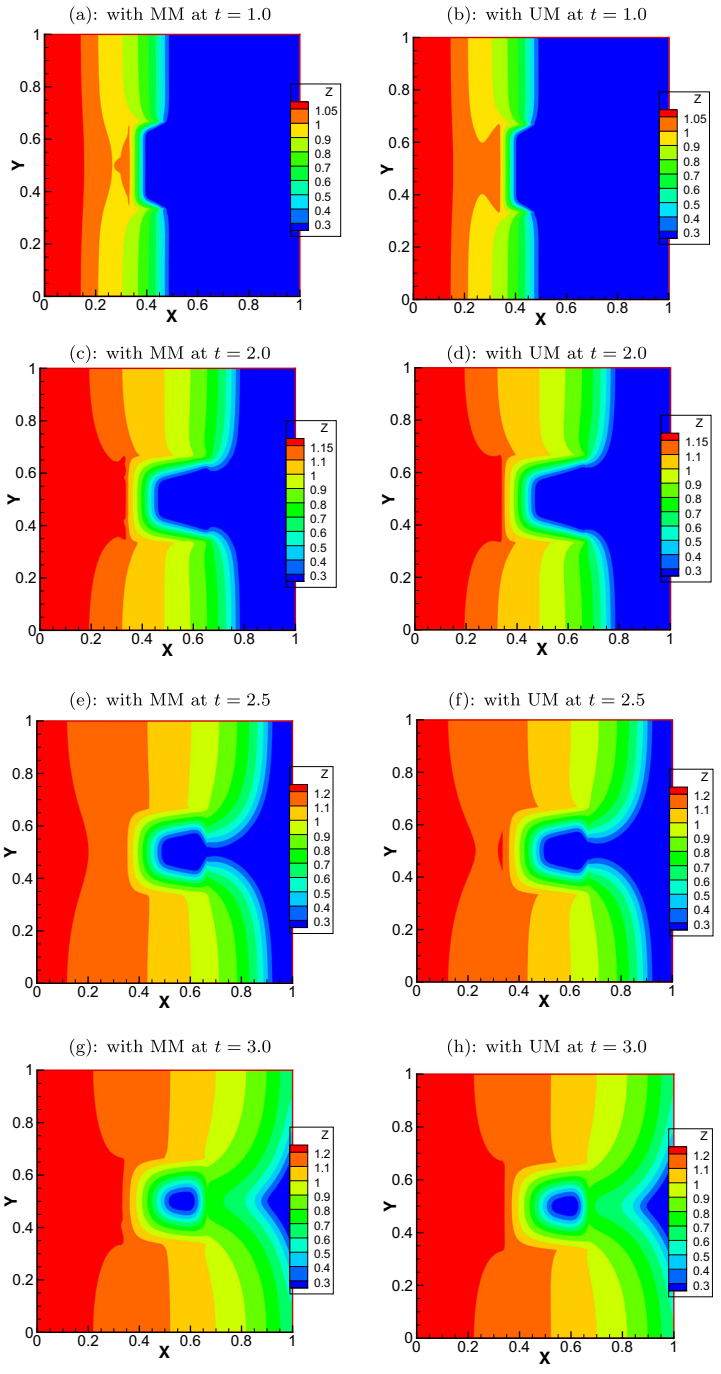


Fig. 15 Example 4.7. The contours of the temperature obtained with a moving mesh (MM) of size 61×61 are compared with those obtained with a uniform mesh (UM) of size 121×121

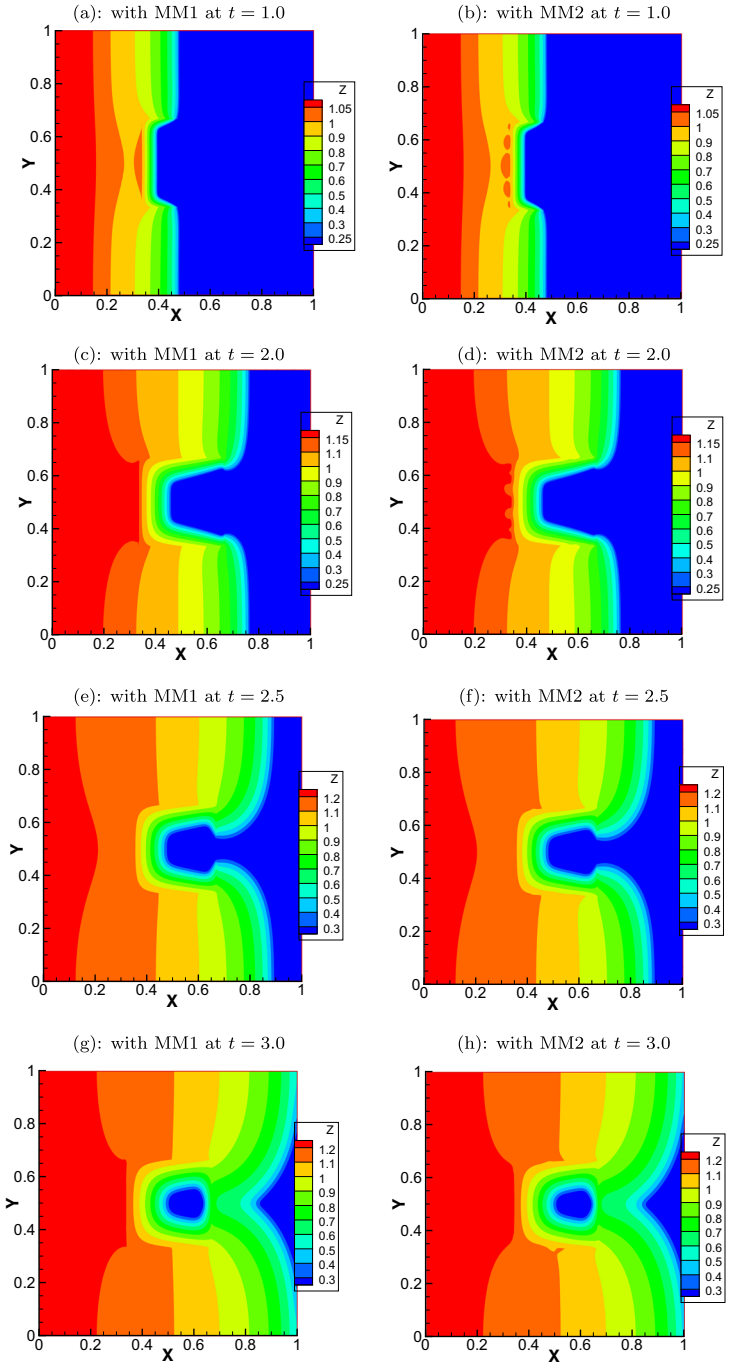


Fig. 16 Example 4.7. The contours of the temperature obtained with an MM1 of size 121 × 121 are compared with those obtained with an MM2 of size 41 × 41 (with the physical PDE being solved on a uniformly refined mesh of size 121 × 121)

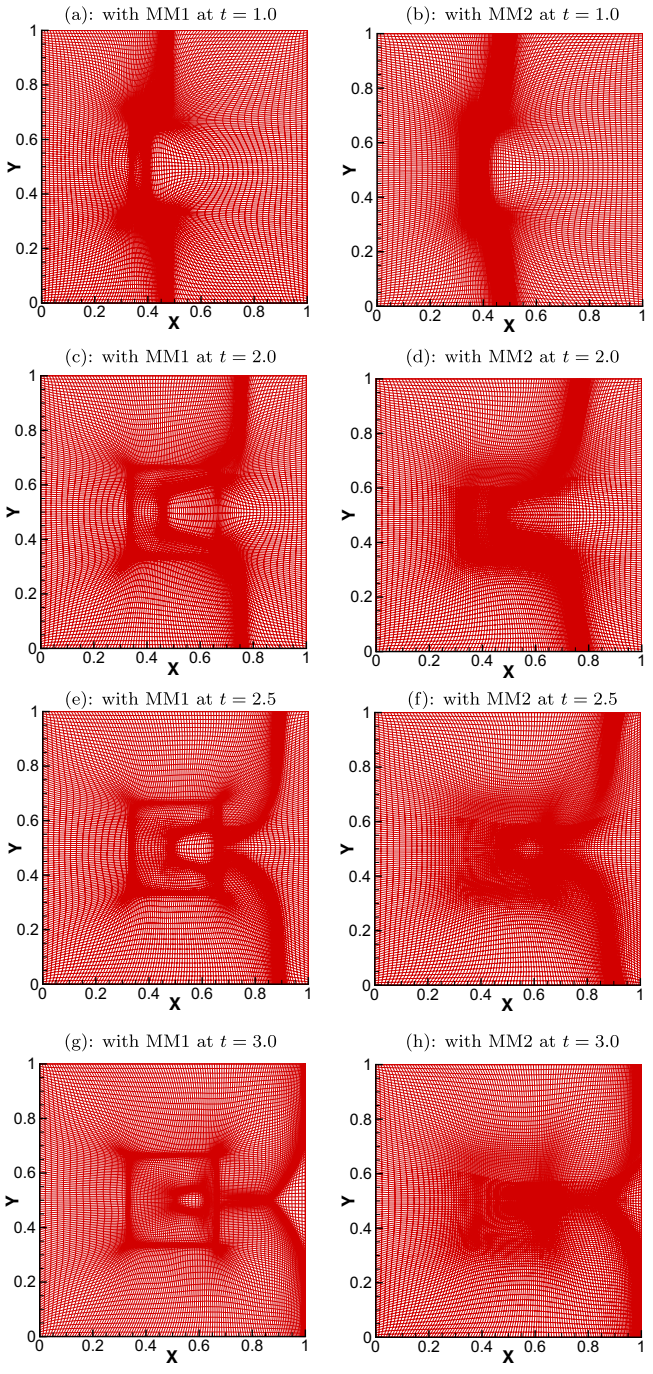


Fig. 17 Example 4.7. Moving meshes of size 121 × 121 obtained with MM1 and MM2 moving mesh strategies. MM2 is obtained by uniformly interpolating a 41 × 41 moving mesh

Table 3 CPU time comparison among one-level, two-level moving mesh and uniform mesh methods for Example 4.7. The CPU time is measured in seconds. The last column is the ratio of the used CPU time to that used with a uniform mesh of the same size

	Fine mesh	Coarse mesh	Total CPU time	Ratio
One-level MM	41×41	41×41	2951	5.85
	81×81	81×81	139374	58.76
	121×121	121×121	1786732	333.03
Tow-level MM	41×41	41×41	2951	5.85
	81×81	41×41	9581	4.03
	121×121	41×41	21888	4.08
Fixed mesh	41×41	n/a	498	1
	81×81	n/a	2372	1
	121×121	n/a	5365	1

This is not surprising since the moving mesh method solves more equations. The efficiency of the moving mesh method can be improved significantly using the two-level moving mesh strategy. For example, for the case with mesh 81×81, the CPU time of MM2 (with the coarse mesh 41×41) is about 25.3% of that with the one-level moving mesh (MM1). For the case 121×121, the CPU time for MM2 is only about 5.7% of that of MM1. Moreover, when the mesh size increases from 41×41 to 81×81 the CPU time increases about 13.3 times for MM1. This number is about 10.6 times

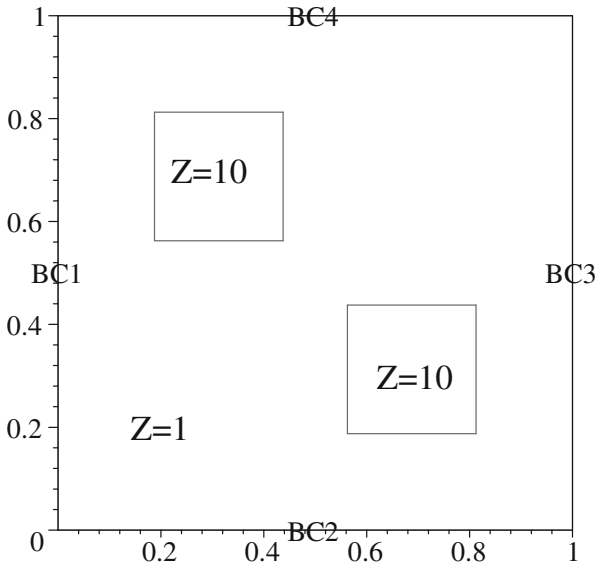


Fig. 18 The material configuration for Examples 4.3 and 4.8. The walls BC1, BC2, BC3, BC4 are insulated with respect to radiation diffusion and material conduction

when the mesh size increases from 81×81 to 121×121. For MM2, the corresponding number is only 3.36 and 2.38, respectively. Finally, we compare MM2 with the uniform mesh method. From Table. 2, we can see that the difference between the two is getting smaller as the mesh becomes finer.

Example 4.7 The setting of this example is the same as the previous example except that the distribution of the atomic mass number is given by (4.2). Note that the jump in the values of z is more significant than that in the previous example.

The moving mesh of 81×81 and the solution are shown in Figs. 13 and 14. From the figures, we can see that the shape of the central obstacle and the profile of the Marshak wave have been captured and reflected accurately by the mesh concentration. It is also worth mentioning that the solutions obtained here are comparable to those obtained by Kang [17, Fig. 6 on page 15] but with more mesh points. Comparison results are shown in Fig. 15 for a moving mesh of 61×61 versus a uniform mesh of 121×121 and in Figs. 16 and 17 for a one-level moving mesh of 121×121 versus a two-level moving mesh of 41×41 (with the physical PDE being solved on

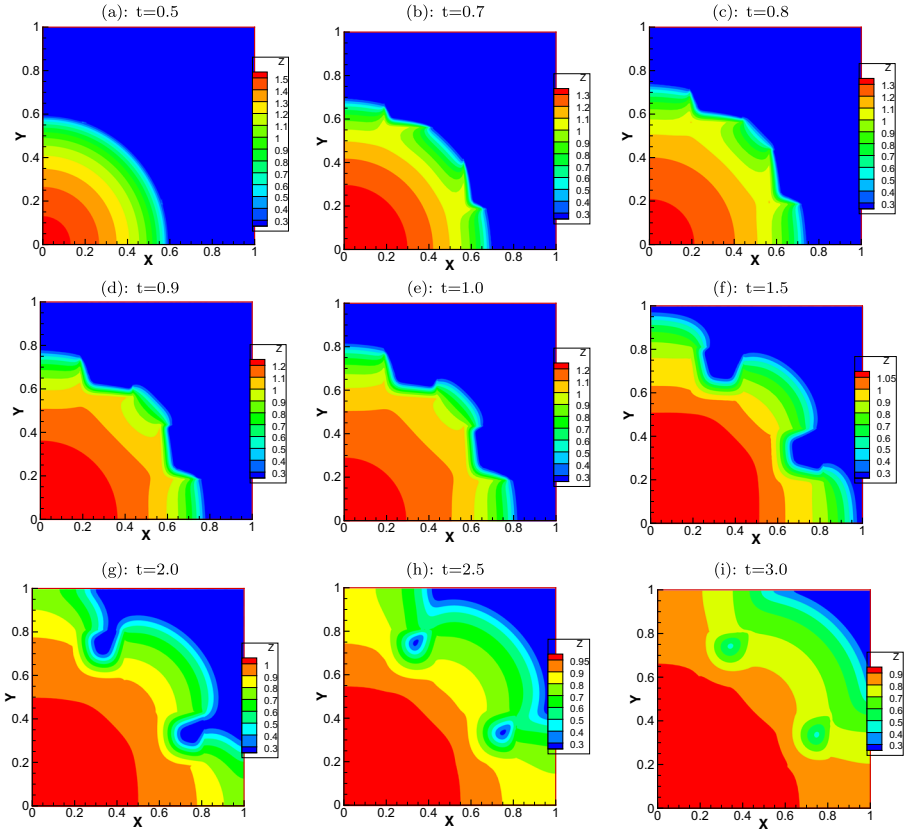


Fig. 19 Example 4.8. The computed solution on a moving mesh of 81×81 is shown at $t = 0.5, 0.7, 0.8, 0.9, 1.0, 1.5, 2.0, 2.5, 3.0$

a uniformly refined mesh of 121 × 121). The results are all comparable. Moreover, the CPU time is listed in Table 3. It can be seen that the two-level moving mesh strategy can significantly improve the efficiency of the moving mesh method without compromising the accuracy.

Example 4.8 The material configuration for this example is shown in Figs. 18. The insets are $(3/16, 7/16) \times (9/16, 13/16)$ and $(9/16, 13/16) \times (3/16, 7/16)$ and the distribution of the atomic mass number is given by (4.3). The boundary of $\Omega = (0, 1) \times (0, 1)$ is considered as insulated with respect to both radiation and material conduction; see (4.4). The initial condition is taken as (4.5).

A typical moving mesh of size 81 × 81 and the computed solution thereon are shown in Figs. 19 and 20. Once again, we can see that our moving mesh method is able to capture the Marshak wave accurately. The results are in good agreement with

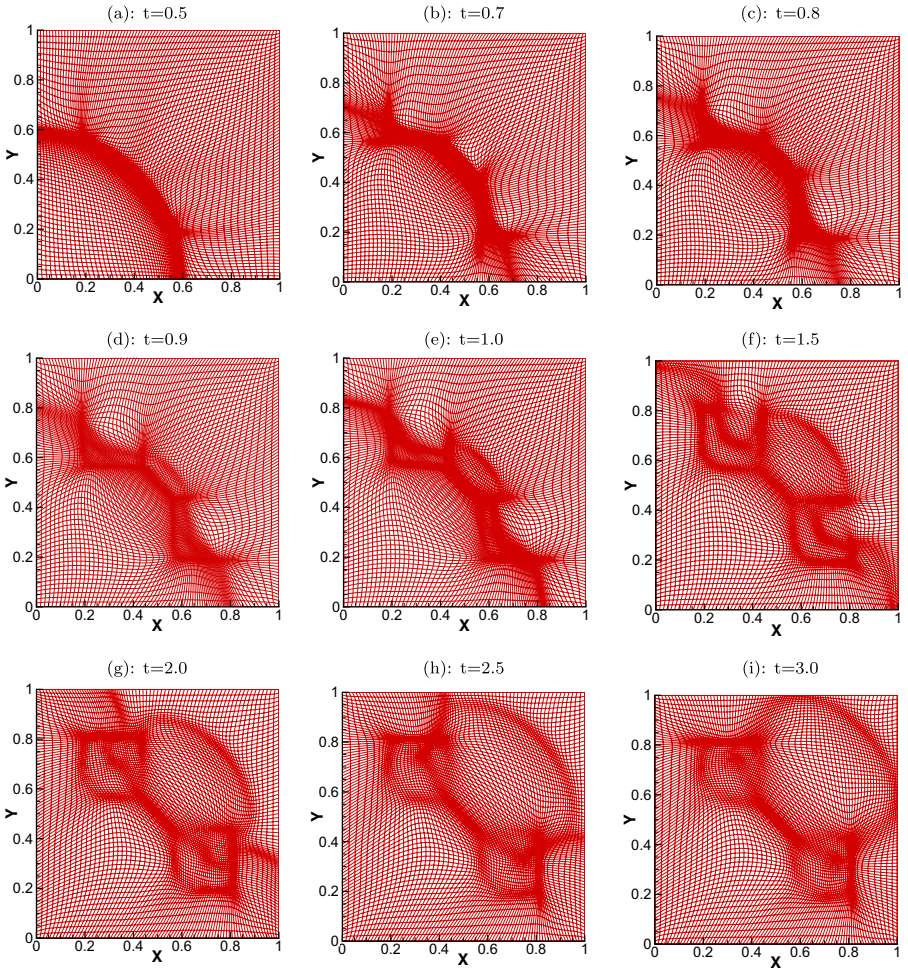


Fig. 20 Example 4.8. The moving mesh of 81 × 81 is shown at $t = 0.5, 0.7, 0.8, 0.9, 1.0, 1.5, 2.0, 2.5, 3.0$

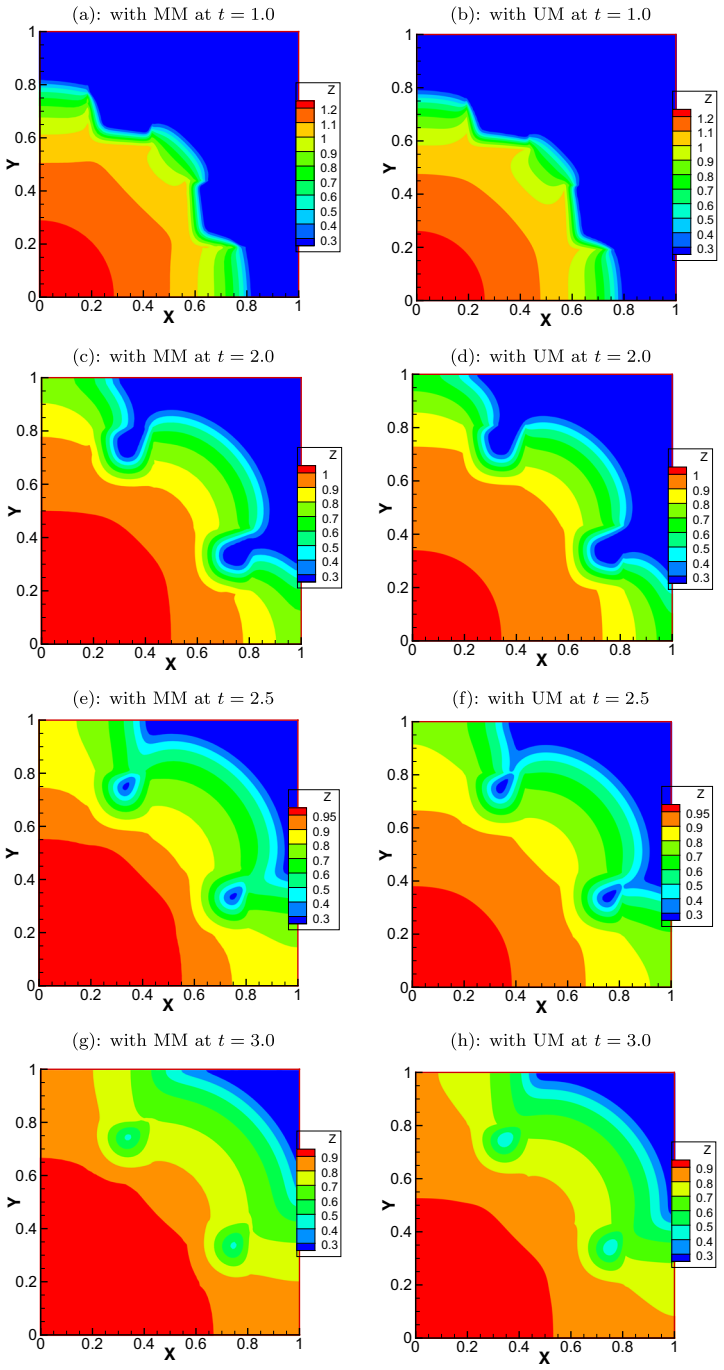


Fig. 21 Example 4.8. The contours of the temperature obtained with a moving mesh (MM) of size 61×61 are compared with those obtained with a uniform mesh (UM) of size 121×121

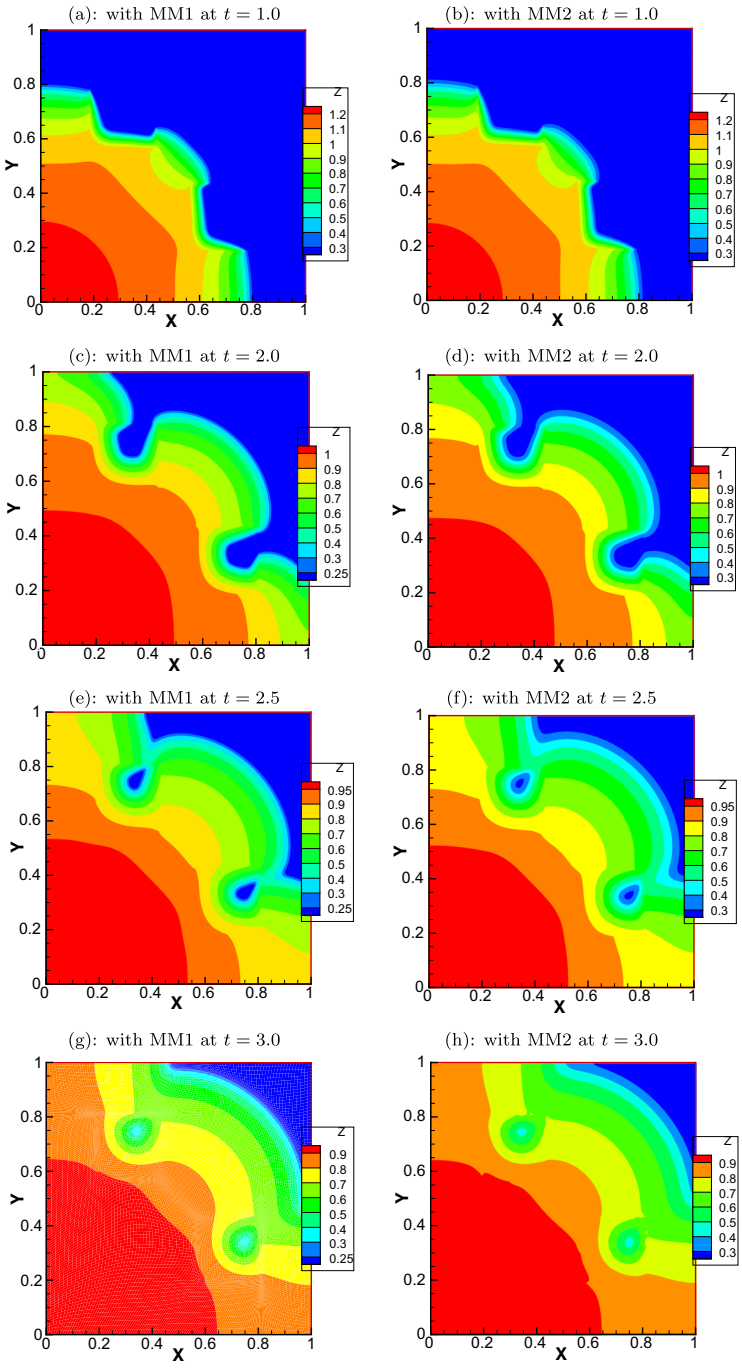


Fig. 22 Example 4.8.The contours of the temperature obtained with an MM1 of size 121 × 121 are compared with those obtained with an MM2 of size 41 × 41 (with the physical PDE being solved on a uniformly refined mesh of size 121 × 121)

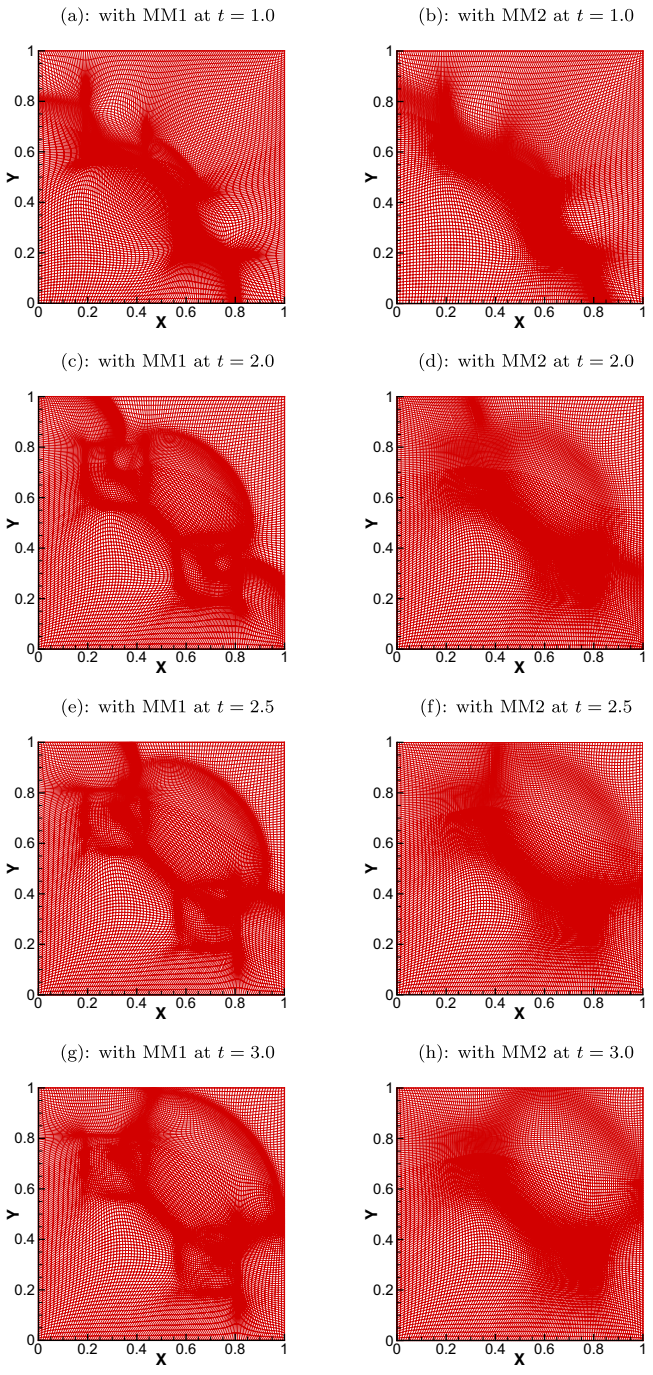


Fig. 23 Example 4.8. Moving meshes of size 121 × 121 obtained with MM1 and MM2 moving mesh strategies. MM2 is obtained by uniformly interpolating a 41 × 41 moving mesh

Table 4 CPU time comparison among one-level, two-level moving mesh and uniform mesh methods for Example 4.8. The CPU time is measured in seconds. The last column is the ratio of the used CPU time to that used with a uniform mesh of the same size

	Fine Mesh	Coarse Mesh	Total CPU time	Ratio
One-level MM	41×41	41×41	2359	4.72
	81×81	81×81	38828	17.25
	121×121	121×121	983286	174.16
Tow-level MM	41×41	41×41	2359	4.72
	81×81	41×41	8912	3.96
	121×121	41×41	22836	4.04
Fixed mesh	41×41	n/a	500	1
	81×81	n/a	2251	1
	121×121	n/a	5646	1

those by Sheng et al. [35]. Comparison results are shown in Fig. 21 for a moving mesh of 61×61 and a uniform mesh of 121×121 and in Figs. 22 and 23 for one-level and two-level moving meshes of 121×121. The CPU time is recorded in Table 4.

5 Conclusions

In the previous sections we have studied the moving mesh finite difference solution of the 2T model for multi-material, non-equilibrium radiation diffusion equations based on the MMPDE moving mesh strategy. The model involves nonlinear diffusion coefficients and its solutions stay positive for all time when they are positive initially. Nonlinear diffusion and preservation of solution positivity pose challenges in the numerical solution of the model. A coefficient-freezing predictor-corrector method has been used for treating nonlinear diffusion while a cutoff strategy with a positive threshold [21] has been employed to keep the solutions positive. A two-level moving mesh strategy and the sparse matrix solver UMFPACK with the MAC OSX acceleration have been used to improve the efficiency of the computation.

The method has been applied to three examples of multi-material non-equilibrium radiation diffusion. The numerical results show that the method is able to capture the profiles and local structures of Marshak waves with adequate mesh concentration. The numerical solutions are in good agreement with those in the existing literature. Comparison studies have also been made between uniform and adaptive moving meshes and between one-level and two-level moving meshes. It is shown that the two-level moving mesh strategy can significantly improve the computational efficiency with only a mild accuracy compromise. Extending the current method to three-dimensional radiation diffusion models [19] and more realistic three-temperature models [1] will be an interesting research topic for near future.

Acknowledgments The work was supported in part by NSFC (China) (Grant No. 11701555), NSAF (China) (Grant No. U1630247), and Science Challenge Project (China) (Grant No. TZ2016002).

References

1. An, H., Jia, X., Walker, H.F.: Anderson acceleration and application to the three-temperature energy equations. J. Comput. Phys. **347**, 1–19 (2017)
2. Bowes, R.L., Wilson, J.R.: Numerical Modeling in Applied Physics and Astrophysics. Jones and Bartlett, Boston (1991)
3. Cash, J.R.: Diagonally implicit Runge-Kutta formulate with error estimate. J. Inst. Math. Appl. **24**, 293–301 (1979)
4. Castor, J.I.: Radiation Hydrodynamics. Cambridge University Press, Cambridge (2004)
5. Davis, T.A.: Algorithm 832: UMFPACK, an unsymmetric-pattern multifrontal method. ACM Trans. Math. Software **30**, 196–199 (2004)
6. Dvinsky, A.S.: Adaptive grid generation from harmonic maps on Riemannian manifolds. J. Comput. Phys. **95**, 450–476 (1991)
7. Huang, W.: Practical aspects of formulation and solution of moving mesh partial differential equations. J. Comput. Phys. **171**, 753–775 (2001)
8. Huang, W.: Variational mesh adaptation:isotropy and equidistribution. J. Comput. Phys. **174**, 903–924 (2001)
9. Huang, W., Ren, Y., Russell, R.D.: Moving mesh methods based on moving mesh partial differential equations. J. Comput. Phys. **113**, 279–290 (1994)
10. Huang, W., Russell, R.D.: A high dimensional moving mesh strategy. Appl. Numer. Math. **26**, 63–76 (1998)
11. Huang, W., Russell, R.D.: Adaptive Moving Mesh Methods. Springer, New York (2011). Applied Mathematical Sciences Series, Vol. 174
12. Huang, W., Sun, W.: Variational mesh adaptation II: error estimates and monitor functions. J. Comput. Phys. **184**, 619–648 (2003)
13. Knoll, D.A., Chacon, L., Margolin, L.G., Mousseau, V.A.: On balanced approximations for time integration of multiple time scale systems. J. Comput. Phys. **185**, 583–611 (2003)
14. Knoll, D.A., Lowrie, R.B., Morel, J.E.: Numerical analysis of time integration errors for non-equilibrium radiation diffusion. J. Comput. Phys. **226**, 1332–1347 (2007)
15. Knoll, D.A., Rider, W.J., Olson, G.L.: An efficient nonlinear solution method for non-equilibrium radiation diffusion. J. Quant. Spect. **63**, 15–29 (1999)
16. Knoll, D.A., Rider, W.J., Olson, G.L.: Nonlinear convergence, accuracy, and time step control in non-equilibrium radiation diffusion. J. Quant. Spect. **70**, 25–36 (2001)
17. Kang, K.S.: P1 Nonconforming finite element method for the solution of radiation transport problems, ICASE Report No. 2002-28
18. Lapenta, G., Chacón, L.: Cost-effectiveness of fully implicit moving mesh adaptation: a practical investigation in 1D. J. Comput. Phys. **219**, 86–103 (2006)
19. Lai, X., Sheng, Z., Yuan, G.: Monotone finite volume scheme for three dimensional diffusion equation on tetrahedral meshes. Commun. Comput. Phys. **21**, 162–181 (2017)
20. Lowrie, R.B.: A comparison of implicit time integration methods for nonlinear relaxation and diffusion. J. Comput. Phys. **196**, 566–590 (2004)
21. Lu, C., Huang, W., Van Vleck, E.S.: The cutoff method for numerical computation of nonnegative solutions of parabolic PDEs with application to anisotropic diffusion and lubrication-type equations. J. Comput. Phys. **242**, 24–36 (2013)
22. Li, X., Huang, W.: Anisotropic mesh adaptation for 3D anisotropic diffusion problems with application to fractured reservoir simulation. Numer. Math. Theor. Meth. Appl. **10**, 913–940 (2017)
23. Liu, J., Qiu, J., Goman, M., Li, X., Liu, M.: Positivity-preserving Runge-Kutta discontinuous Galerkin method on adaptive Cartesian grid for strong moving shock. Numer. Math. Theor. Meth. Appl. **9**, 87–110 (2016)
24. Marshak, R.E.: Effect of radiation on shock wave behavior. Phys. Fluids **1**, 24–29 (1958)
25. Mihalas, D., Mahalas, B.W.: Foundations of Radiation Hydrodynamics. Oxford University Press, Oxford (1984)

26. Mousseau, V.A., Knoll, D.A., Rider, W.J.: Physical-based preconditioning and the Newton-Krylov method for non-equilibrium radiation diffusion. J. Comput. Phys. **160**, 743–765 (2000)
27. Mousseau, V.A., Knoll, D.A.: New physics-based preconditioning of implicit methods for non-equilibrium radiation diffusion. J. Comput. Phys. **190**, 42–51 (2003)
28. Ovtchinnikov, S., Cai, X.C.: One-level Newton-Krylov-Schwarz algorithm for unsteady nonlinear radiation diffusion problems. Numer. Lin. Alg. Appl. **11**, 867–881 (2004)
29. Olson, G.L.: Efficient solution of multi-dimensional flux-limited non-equilibrium radiation diffusion coupled to material conduction with second order time discretization. J. Comput. Phys. **226**, 1181–1195 (2007)
30. Pomraning, G.C.: The non-equilibrium Marshak wave problem. J. Quant. Spectrosc. Radiat. Transf. **21**, 249–261 (1979)
31. Pernice, M., Philip, B.: Solution of equilibrium radiation diffusion problems using implicit adaptive mesh refinement. SIAM J. Sci. Comput. **27**(electronic), 1709–1726 (2006)
32. Philip, B., Wang, Z., Brrill, M.A., Rodriguez, M.R., Pernice, M.: Dynamic implicit 3D adaptive mesh refinement for non-equilibrium radiation diffusion. J. Comput. Phys. **262**, 17–37 (2014)
33. Rider, W.J., Knoll, D.A., Olson, G.L.: A multigrid Newton-Krylov method for multimaterial equilibrium radiation diffusion. J. Comput. Phys. **152**, 164–191 (1999)
34. Su, B., Olson, G.L.: Benchmark results for the non-equilibrium Marshak diffusion problem. J. Quant. Spectrosc. Radiat. Transf. **56**, 337–351 (1996)
35. Sheng, Z., Yue, J., Yuan, G.: Monotone finite volume schemes of non-equilibrium radiation diffusion equations on distorted meshes. SIAM J. Sci. Comput. **31**, 2915–293 (2009)
36. Spitzer, L., Harm, R.: Transport phenomena in a completely ionized gas. Phys. Rev. **89**, 977–981 (1953)
37. Wise, E.S., Cox, B.T., Treeby, B.E.: Mesh density functions based on local bandwidth applied to moving mesh methods. Commun. Comput. Phys. **22**, 1286–1308 (2017)
38. Wu, Y., Wang, H.: An AMG preconditioner for solving the Navier-Stokes equations with a moving mesh finite element method. East. Asia. J. Appl. Math. **6**, 353–366 (2016)
39. Yuan, G., Hang, X., Sheng, Z., Yue, J.: Progress in numerical methods for radiation diffusion equations. Chinese J. Comput. Phys. **26**, 475–500 (2009)
40. Yue, J., Yuan, G.: Picard-newton iterative method with time step control for multimaterial non-equilibrium radiation diffusion problem. Commun. Comput. Phys. **10**, 844–866 (2011)
41. Yang, X., Huang, W., Qiu, J.: A moving mesh finite difference method for equilibrium radiation diffusion equations. J. Comput. Phys. **298**, 661–677 (2015)
42. Zhao, X., Chen, Y., Gao, Y., Yu, C., Li, Y.: Finite volume element methods for non-equilibrium radiation diffusion equations. Int. J. Numer. Meth. Fluids **73**, 1059–1080 (2013)
43. Zhang, H., Zegeling, P.A.: A moving mesh finite difference method for non-monotone solutions of non-equilibrium equations in porous media. Commun. Comput. Phys. **22**, 935–964 (2017)