Journal of Computational Physics 228 (2009) 3268–3287

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

On choosing a nonlinear initial iterate for solving the 2-D 3-T heat conduction equations ☆

Heng-Bin An [a,*], Ze-Yao Mo [a], Xiao-Wen Xu [a], Xu Liu [b]

[a] High Performance Computing Center, Institute of Applied Physics and Computational Mathematics, Beijing 100088, PR China
[b] Graduate School of China Academy of Engineering Physics, Beijing 100088, PR China

ARTICLE INFO

Article history:
Received 27 September 2006
Received in revised form 21 December 2007
Accepted 20 January 2009
Available online 31 January 2009

Keywords:
3-T heat conduction equations
Nonlinear equations
Inexact Newton method
Initial iterate

ABSTRACT

The 2-D 3-T heat conduction equations can be used to approximately describe the energy broadcast in materials and the energy swapping between electron and photon or ion. To solve the equations, a fully implicit finite volume scheme is often used as the discretization method. Because the energy diffusion and swapping coefficients have a strongly nonlinear dependence on the temperature, and some physical parameters are discontinuous across the interfaces between the materials, it is a challenge to solve the discretized nonlinear algebraic equations. Particularly, as time advances, the temperature varies so greatly in the front of energy that it is difficult to choose an effective initial iterate when the nonlinear algebraic equations are solved by an iterative method. In this paper, a method of choosing a nonlinear initial iterate is proposed for iterative solving this kind of nonlinear algebraic equations. Numerical results show the proposed initial iterate can improve the computational efficiency, and also the convergence behavior of the nonlinear iteration.

1. Introduction

In the numerical simulation of inertial constraint fusion (ICF), it is very important to solve the two-dimensional radiation fluid dynamics equations. In the numerical solution of the equations, over 80% computational cost has been used to solve the 2-D 3-T energy equations [16,25]. Therefore it is urgent to find a high efficient method to solve the 2-D 3-T energy equations which will dominate the whole numerical simulation.

The 2-D 3-T energy equations are used to approximately describe the way of energy diffusion of electron, ion and photon. As time advances, the energy not only broadcasts, but also exchanges between photon and electron, and between electron and ion. The coupled temperature of electron, ion and photon are unknown and therefore are needed to be calculated. In most cases, the 2-D 3-T energy equations can be simplified into the 2-D 3-T heat conduction equations, and they are essentially same [25]. In fact, the 2-D 3-T heat conduction system is a type of radiation diffusion model. For other related radiation diffusion models, see [3,7,8,19,21,22,24,26,27].

For solving the discretized 2-D 3-T heat conduction equations, some nonlinear iterative method must be employed. At present, Newton–Krylov method, in which the Krylov subspace method is employed to solve the Newton linear equations, is widely used in many areas to solve all kinds of large scale nonlinear problems [4–6,20]. One of the most important advantages of the Newton–Krylov method is that it can be implemented without explicitly forming and saving the Jacobian matrix.

☆ This project was supported by Chinese NSF (No. 10701015), Chinese NSF for DYS (No. 60425205), National Basic Key Research Special Fund (No. 2005CB321702) and Chinese NSF (No. 60533020).

* Corresponding author. Tel.: +86 10 5987 2452.
E-mail addresses: an_hengbin@iapcm.ac.cn (H.-B. An), zeyao_mo@iapcm.ac.cn (Z.-Y. Mo), xwxu@iapcm.ac.cn (X.-W. Xu), ninad@sohu.com (X. Liu).

0021-9991/$ - see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2009.01.024

This will greatly improve the computational efficiency and save the memory space. In this paper, the Newton–Krylov method will be used to solve the discretized 2-D 3-T heat conduction equations.

For any iterative method, a suitable initial iterate must be provided. The initial iterate has strong influence on the method, including its convergence, computational efficiency and robustness. To solve the linear equations discretized from time dependent partial differential equations, Fischer proposed a kind of projection technique to produce an effective initial iterate in [15]. More recently, for solving linear and nonlinear equations arising in fluid flow simulation, Tromeur-Dervout and Vassilevski proposed some techniques for choosing better initial guesses [30]. Basing on the history of the evolution problem solving, they proposed a better initial guess for two iterative linear solvers; and by using a reduced model technique, they proposed a better initial guess for iterative nonlinear solvers. For solving the 2-D 3-T heat conduction equations, the temperature at the current time, which is referred to as the *usual initial value*, is often used as the initial guess for computing the temperature at the next time. This is a reasonable choice when the time step size is not very large. If the time step size is relatively large, however, the usual initial value is not a good choice. In particular, for the location where the energy varies acutely, the temperature at the current time is usually not a good choice of initial iterate for computing the temperature at the next time. A prediction/correction method is commonly used for solving ordinary differential equations and computational fluid dynamics, see [17, Chapter 10] and [23]. This method can specially be used to produce an initial iterate for any iterative method. For the 2-D 3-T heat conduction equations, however, the prediction/correction method is not effective when it is used to produce an initial iterate. Actually, the nonlinear iteration does not converge when a prediction/correction initial iterate is used in our numerical experiments.

In this paper, for solving the discretized 2-D 3-T heat conduction equations, a new method of choosing a nonlinear initial iterate is proposed. In this method, first a predicted temperature is obtained through predicting the energy at the next time; second the predicted temperature is further modified on the subregion where the energy varies most acutely. The modification is finished by solving a nonlinear system defined on the subregion. Numerical results show that the proposed initial iterate is very effective. By using the proposed initial iterate, not only is the computational efficiency increased by about twice, but also the convergence behavior is improved in some cases.

The rest of the paper is organized as follows. In Section 2, the 2-D 3-T heat conduction equations and discretization scheme are given, and in Section 3, the Newton–Krylov method and the Kinsol package are introduced. In Section 4, the strategy for choosing the new initial iterate is provided, and in Section 5, some numerical results are presented. Finally, a brief conclusion and some remarks about the method are given in Section 6.

2. 2-D 3-T heat conduction equations and their discretization

In this section, the 2-D 3-T heat conduction equations are firstly introduced, and then the equations are discretized by a finite volume method on a rectangular domain.

2.1. 2-D 3-T heat conduction equations

The 2-D 3-T heat conduction equations are defined as

$$\begin{cases} C_{ve}\frac{\partial T_e}{\partial t} - \frac{1}{\rho}\nabla\cdot(K_e\nabla T_e) = \omega_{ei}(T_i - T_e) + \omega_{er}(T_r - T_e), \\ C_{vi}\frac{\partial T_i}{\partial t} - \frac{1}{\rho}\nabla\cdot(K_i\nabla T_i) = \omega_{ei}(T_e - T_i), \\ C_{vr}\frac{\partial T_r}{\partial t} - \frac{1}{\rho}\nabla\cdot(K_r\nabla T_r) = \omega_{er}(T_e - T_r) \end{cases} \tag{1}$$

with $(x, y) \in \Omega_{xy}$ and $0 \leqslant t \leqslant T_{\max}$. Eq. (1) are respectively the electron, ion and photon temperature equations. The computational domain consists of several subdomains, each contains one specific kind of material. In Eq. (1), the relevant physical parameters are as below:

- T_α $(\alpha = e, i, r)$ are electron, ion and photon temperatures.
- $C_{v\alpha}$ $(\alpha = e, i, r)$ are isochore specific heat coefficients.
- ρ is the density of the material.
- $K_\alpha = K_\alpha(\rho, T_\alpha)$ $(\alpha = e, i, r)$ are heat conduction coefficients.
- ω_{ei} and ω_{er} are respectively the energy exchanging coefficients between electron and ion, and that between electron and photon.

Specifically, $C_{v\alpha}$, K_α and $\omega_{e\alpha}$ are defined respectively as follows:

$$C_{v\alpha} = \begin{cases} c_e, & \alpha = e \\ c_i, & \alpha = i \\ c_r T_r^3, & \alpha = r \end{cases}, \quad K_\alpha = \begin{cases} A_e T_e^{5/2}, & \alpha = e \\ A_i T_i^{5/2}, & \alpha = i \\ A_r T_r^{3+\beta}, & \alpha = r \end{cases}, \quad \omega_{e\alpha} = \begin{cases} A_{ei}\rho T_e^{-2/3}, & \alpha = i \\ A_{er}\rho T_e^{-1/2}, & \alpha = r \end{cases},$$

where c_α are constants, A_α, $A_{e\alpha}$, β and ρ are continuous in the interior of the material.

The boundary and initial conditions are:

- boundary conditions:
  (a) on rigid walls: $K_\alpha \nabla T_\alpha \cdot \mathsf{n} = 0$, $\alpha = e, i, r$, where n is the outer normal vector of the boundary;
  (b) on free surfaces: $K_\alpha \nabla T_\alpha \cdot \mathsf{n} = 0,\ \alpha = e, i;\ T_r = T_r(t,x,y)|_{(x,y)\in\partial\Omega_{xy}}$.
- concatenation conditions: across the material interfaces, T_α, $K_\alpha \nabla T_\alpha \cdot \mathsf{n}$ are continuous, $\alpha = e, i, r$, n is the outer normal vector of the interface;
- initial conditions:

$$T_\alpha(0,x,y) = T^0_\alpha(x,y), \quad \alpha = e, i, r.$$

The system energy of unit mass is defined by

$$E = E_e + E_i + E_r, \quad \text{where } E_e = c_e T_e,\ E_i = c_i T_i,\ E_r = \frac{1}{4} c_r T_r^4.$$

The 2-D 3-T heat conduction equations are highly nonlinear because the heat conduction coefficients K_α and energy exchanging coefficients $\omega_{e\alpha}$ are strongly nonlinear functions of the temperature. Besides, the computational region is composed of multiple materials and some physical parameters are strongly discontinuous across the interfaces of the materials, which further makes the equations hard to be solved. All these properties will be transferred to the discretized nonlinear algebraic equations.

2.2. Discretization scheme for 2-D 3-T heat conduction equations

In this section and the following ones, the rectangular computing domain $\Omega_{xy} = [0,A] \times [0,B]$ is considered, where the up boundary is the free surface, and the rest are the rigid walls. The 2-D 3-T heat conduction equations (1) are discretized by a fully implicit finite volume method. Specifically, the domain Ω_{xy} is zoned into $L \times M$ meshes by some lines paralleling the axes, where the step size in x direction is $h_1 = A/L$ and in y direction is $h_2 = B/M$. The temperature, energy and density are defined at the center points of the meshes.

Now, take the photon heat conduction equation as an example, the concrete discretization of Eq. (1) are presented here. As shown in Fig. 1, let the current element be $\Omega_{l,m}$ $(l = 1,2,\ldots,L;\ m = 1,2,\ldots,M)$, the common edges of $\Omega_{l,m}$ with its neighbors be γ_j $(j = 1,\ldots,4)$, and p_j be the center point of γ_j. Next use (l,m) to represent the center of $\Omega_{l,m}$. Assume that t_n is the current time, t_{n+1} is the next time, and $\Delta t_n = t_{n+1} - t_n$. Integrating both sides of the photon heat conduction equation over $\Omega_{l,m} \times [t_n, t_{n+1}]$, then

$$\int_{t_n}^{t_{n+1}} \int_{\Omega_{l,m}} C_{vr} \frac{\partial T_r}{\partial t}\, d\Omega\, dt - \int_{t_n}^{t_{n+1}} \int_{\Omega_{l,m}} \frac{1}{\rho} \nabla \cdot (K_r \nabla T_r)\, d\Omega\, dt = \int_{t_n}^{t_{n+1}} \int_{\Omega_{l,m}} \omega_{er}(T_e - T_r)\, d\Omega\, dt. \tag{2}$$

The first term in (2) can be discretized as:

$$\int_{t_n}^{t_{n+1}} \int_{\Omega_{l,m}} C_{vr} \frac{\partial T_r}{\partial t}\, d\Omega\, dt \approx |\Omega_{l,m}| \int_{t_n}^{t_{n+1}} \left(C_{vr} \frac{\partial T_r}{\partial t} \right)_{l,m} dt = |\Omega_{l,m}| \int_{t_n}^{t_{n+1}} \left(c_r T_r^3 \frac{\partial T_r}{\partial t} \right)_{l,m} dt = c_r |\Omega_{l,m}| \left(\frac{(T_r^{n+1})^4 - (T_r^n)^4}{4} \right)_{l,m},$$

where $|\Omega_{l,m}|$ represents the area of the element $\Omega_{l,m}$, the physical quantities with superscript n and $n+1$ represent the corresponding value at the current time and the next time, and the physical quantities with subscript $l,\ m$ denote their value at the point (l,m). By using the fully implicit scheme and the Green formula, the second term in (2) can be discretized as

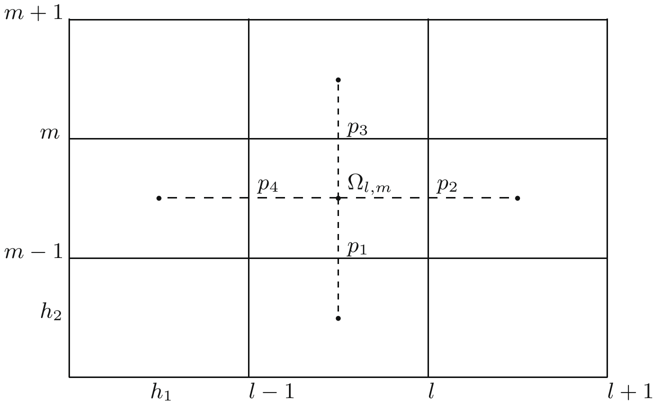


Fig. 1. Schematic diagram of the mesh.

$$\int_{t_n}^{t_{n+1}}\int_{\Omega_{l,m}}\frac{1}{\rho}\nabla\cdot(K_r\nabla T_r)\,d\Omega\,dt\approx\Delta t_n\int_{\Omega_{l,m}}\left(\frac{1}{\rho}\nabla\cdot(K_r\nabla T_r)\right)^{n+1}d\Omega\approx\frac{\Delta t_n}{\rho_{l,m}}\int_{\partial\Omega_{l,m}}(K_r\nabla T_r)^{n+1}\cdot\mathsf{n}\,dl$$
$$\approx\frac{\Delta t_n}{\rho_{l,m}}\sum_{j=1}^{4}|\gamma_j|(K_r^{n+1}\nabla T_r^{n+1}\cdot\mathsf{n})_j,$$

here n is the outer normal vector of the mesh $\Omega_{l,m}$, $|\gamma_j|$ the length of γ_j, and $(K_r^{n+1}\nabla T_r^{n+1}\cdot\mathsf{n})_j$ the value of the heat density at p_j, which can be determined by combining the center point value of K_r and T_r on mesh (l,m) and its neighbor meshes [16]. The last term in (2) can be discretized as

$$\int_{t_n}^{t_{n+1}}\int_{\Omega_{l,m}}[\omega_{er}(T_e-T_r)]\,d\Omega\,dt\approx\Delta t_n|\Omega_{l,m}|(\omega_{er}(T_e-T_r))_{l,m}^{n+1}.$$

The electron and ion heat conduction equations can be discretized in the same way. Thus, the discretized equations at (l,m) are

$$c_e\rho_{l,m}|\Omega_{l,m}|(T_{e,l,m}^{n+1}-T_{e,l,m}^{n})-\Delta t_n\sum_{j=1}^{4}|\gamma_j|(K_e^{n+1}\nabla T_e^{n+1}\cdot\mathsf{n})_j=\rho_{l,m}\Delta t_n|\Omega_{l,m}|(\omega_{ei}(T_i-T_e)+\omega_{er}(T_r-T_e))_{l,m}^{n+1},$$
$$c_i\rho_{l,m}|\Omega_{l,m}|(T_{i,l,m}^{n+1}-T_{i,l,m}^{n})-\Delta t_n\sum_{j=1}^{4}|\gamma_j|(K_i^{n+1}\nabla T_i^{n+1}\cdot\mathsf{n})_j=\rho_{l,m}\Delta t_n|\Omega_{l,m}|(\omega_{ei}(T_e-T_i))_{l,m}^{n+1},$$
$$\frac{1}{4}c_r\rho_{l,m}|\Omega_{l,m}|((T_{r,l,m}^{n+1})^4-(T_{r,l,m}^{n})^4)-\Delta t_n\sum_{j=1}^{4}|\gamma_j|(K_r^{n+1}\nabla T_r^{n+1}\cdot\mathsf{n})_j=\rho_{l,m}\Delta t_n|\Omega_{l,m}|(\omega_{er}(T_e-T_r))_{l,m}^{n+1}.$$

Now let

$$T_{l,m}=(T_{e,l,m}^{n+1},T_{i,l,m}^{n+1},T_{r,l,m}^{n+1})$$

and

$$\mathcal{T}=(T_{1,1},T_{2,1},\ldots,T_{L,1},T_{1,2},T_{2,2},\ldots,T_{L,2},\ldots,T_{1,M},T_{2,M},\ldots,T_{L,M})^T,$$

then the discretized equations at (l,m) are

$$F_{l,m}(\mathcal{T})=(F_{e,l,m}(\mathcal{T}),F_{i,l,m}(\mathcal{T}),F_{r,l,m}(\mathcal{T}))=0,$$

where

$$F_{e,l,m}(\mathcal{T})=c_e\rho_{l,m}|\Omega_{l,m}|(T_{e,l,m}^{n+1}-T_{e,l,m}^{n})-\Delta t_n\sum_{j=1}^{4}|\gamma_j|(K_e^{n+1}\nabla T_e^{n+1}\cdot\mathsf{n})_j-\rho_{l,m}\Delta t_n|\Omega_{l,m}|(\omega_{ei}(T_i-T_e)+\omega_{er}(T_r-T_e))_{l,m}^{n+1},$$
$$F_{i,l,m}(\mathcal{T})=c_i\rho_{l,m}|\Omega_{l,m}|(T_{i,l,m}^{n+1}-T_{i,l,m}^{n})-\Delta t_n\sum_{j=1}^{4}|\gamma_j|(K_i^{n+1}\nabla T_i^{n+1}\cdot\mathsf{n})_j-\rho_{l,m}\Delta t_n|\Omega_{l,m}|(\omega_{ei}(T_e-T_i))_{l,m}^{n+1},$$
$$F_{r,l,m}(\mathcal{T})=\frac{1}{4}c_r\rho_{l,m}|\Omega_{l,m}|((T_{r,l,m}^{n+1})^4-(T_{r,l,m}^{n})^4)-\Delta t_n\sum_{j=1}^{4}|\gamma_j|(K_r^{n+1}\nabla T_r^{n+1}\cdot\mathsf{n})_j-\rho_{l,m}\Delta t_n|\Omega_{l,m}|(\omega_{er}(T_e-T_r))_{l,m}^{n+1}.$$

Let

$$\mathcal{F}(\mathcal{T})=(F_{1,1}(\mathcal{T}),\ldots,F_{L,1}(\mathcal{T}),F_{1,2}(\mathcal{T}),\ldots,F_{L,2}(\mathcal{T}),\ldots,F_{1,M}(\mathcal{T}),\ldots,F_{L,M}(\mathcal{T}))^T,$$

then the equations needed to be solved is

$$\mathcal{F}(\mathcal{T})=0, \tag{3}$$

its scale is $N=3LM$.

Usually, Eq. (3) has the following characteristics:

(i) The expression of the equations is very complicated. Therefore, if Newton method is used to solve the nonlinear equations, it will be very difficult to form the exact Jacobian matrix; instead, only some approximate Jacobian matrix may be produced. Or, it is unnecessary to form it (for example, if a Jacobian free Newton–Krylov method is used).

(ii) The heat conduction coefficients and energy exchanging coefficients have strongly nonlinear dependence on the temperature, and some physical parameters are discontinuous across the interfaces of the materials. Consequently, the discretized nonlinear equations is very ill-conditioned.

Due to the above characteristics, it is a difficult task to solve the equations effectively. In this paper, KINSOL package, which is based on the Newton–Krylov method, will be used to solve this problem.

3. Newton–Krylov method and Kinsol package

Newton–Krylov method [5,6] will be used to solve Eq. (3). This method is one of the most important and effective tools for solving large sparse systems of nonlinear equations, and Kinsol package is a kind of nonlinear solver based on the Newton–Krylov method [10].

3.1. Newton–Krylov method

Newton–Krylov method is a kind of inexact Newton method [11]. Since the classical inexact Newton method is locally convergent, some global strategies, such as the line search or trust region techniques, are required in practical applications [2,4–6,13]. Particularly, if the line search method is augmented with Newton–Krylov method, then Newton–Krylov with line search method is obtained [1,13,14,28], which can be described by the following algorithm. Note that in the algorithm the function $f(\mathcal{T})$ is defined by

$$f(\mathcal{T}) = \frac{1}{2}\|\mathcal{F}(\mathcal{T})\|_2^2. \tag{4}$$

Algorithm. Newton–Krylov with line search

```
1. Given $\mathcal{T}^{(0)}$, tolerance $\varepsilon$, and two parameters $0 < \alpha < \beta < 1$, let $k := 0$.
2. While $\|\mathcal{F}(\mathcal{T}^{(k)})\| > \varepsilon$
   2.1. Choose $\bar{\eta}_k \in [0, 1)$, solve the equations
        $$\mathcal{F}'(\mathcal{T}^{(k)})s = -\mathcal{F}(\mathcal{T}^{(k)}) \tag{5}$$
        by some Krylov method, and obtain an inexact Newton direction $s^{(k)}$, such that
        $$\|\mathcal{F}(\mathcal{T}^{(k)}) + \mathcal{F}'(\mathcal{T}^{(k)})s^{(k)}\| \leqslant \bar{\eta}_k\|\mathcal{F}(\mathcal{T}^{(k)})\|. \tag{6}$$
   2.2. Perform line search along $s^{(k)}$, obtain a step $\delta_k = \theta_k s^{(k)}$, where $\theta_k \in (0, 1]$, such that
        $$f(\mathcal{T}^{(k)} + \delta_k) \leqslant f(\mathcal{T}^{(k)}) + \alpha\nabla f(\mathcal{T}^{(k)})\delta_k \tag{7}$$
        and
        $$f(\mathcal{T}^{(k)} + \delta_k) \geqslant f(\mathcal{T}^{(k)}) + \beta\nabla f(\mathcal{T}^{(k)})\delta_k. \tag{8}$$
   2.3. $\mathcal{T}_{k+1} := \mathcal{T}_k + \delta_k$.
   2.4. $k := k + 1$.
```

The linear equations (5) is the *Newton equations*, $\bar{\eta}_k$ is the *forcing term*, which is used to control the accuracy for solving the Newton equations, and $s^{(k)}$ is the *inexact Newton direction* of $\mathcal{F}$ at $\mathcal{T}^{(k)}$. In step 2.2 of the algorithm, a step $\delta_k = \theta_k s^{(k)}$ satisfying both (7) and (8) is obtained, and then in step 2.3 the next iterate is formed. The inequalities (7) and (8) are respectively α and β conditions [12]. As for the determination of the parameter θ_k and some other details of the algorithm, see [4–6,12–14,28].

In step 1 of the algorithm, a proper initial iterate $\mathcal{T}^{(0)}$ must be provided. The choice of the initial iterate has strong influence on the computational efficiency of the method. In most cases, a good initial guess cannot be obtained easily. To solve Eq. (3), the temperature at the current time, which is referred as the *usual initial value*, is often used as the initial iterate for computing the temperature at the next time.

In the implementation of the Newton–Krylov method, only the product of the Jacobian matrix with vectors is concerned, which can be approximately replaced by the computation of function evaluations. In fact, if $\mathcal{F}$ is continuously differentiable, then for any $\mathcal{T},\ v \in R^N,\ \mathcal{F}'(\mathcal{T})v$ can be computed approximately by

$$\mathcal{F}'(\mathcal{T})v \approx \frac{\mathcal{F}(\mathcal{T} + \sigma v) - \mathcal{F}(\mathcal{T})}{\sigma},$$

where σ is a finite difference step size. Therefore, this kind of method can be implemented without explicitly forming and saving the Jacobian matrix. As a result, the computational cost and memory space can be greatly saved. Thus, Newton–Krylov method is suitable for solving large scale problems in application areas [4–6,20].

3.2. A nonlinear solver – Kinsol

The numerical experiments in this paper are carried out on a platform – Kinsol package. This package, developed in Lawrence Livermore National Laboratory, is a general-purpose nonlinear solver based on Newton–Krylov subspace technology

(Krylov Inexact Newton SOLver) [10]. Kinsol is a part of the package Sundials, a SUite of Nonlinear and DIfferential/ALgebraic equation Solvers. For details about Sundials, see [18].

4. The strategy for choosing the initial iterate

In this section, a choice of the nonlinear initial iterate for solving Eq. (3) will be presented. Assume that, at time step n (current time), the electron temperature T_e^n, the ion temperature T_i^n and the photon temperature T_r^n are known. Next an initial temperature $T_e^{n+1,0}$, $T_i^{n+1,0}$ and $T_r^{n+1,0}$ for time step $n+1$ (next time) will be predicted.

The procedure for predicting the temperature includes two phases: first, the energy of electron, ion and photon for the next time is predicted, and on this base an initial temperature is obtained; second, the obtained initial temperature is further modified by solving a subregion nonlinear system. Now, the concrete prediction procedure is described as below.

4.1. Prediction of the initial energy value

Because the up surface is free, only the prediction of energy on meshes corresponding to one line $x = x_l \equiv \left(l - \frac{1}{2}\right)h_1$, which parallels to axis y, will be discussed. Furthermore, for the sake of convenience, we will focus on the prediction of the photon energy.

Note that, from time step $n-1$ to n, the variation ratio of photon energy on mesh (l,m) is

$$D_{r,l,m}^n \equiv \frac{E_{r,l,m}^n - E_{r,l,m}^{n-1}}{\Delta t_{n-1}}, \quad m = 1,2,\ldots,M, \tag{9}$$

where $E_{r,l,m}^n$ and $E_{r,l,m}^{n-1}$ respectively denote the photon energy on the mesh (l,m) at time step n and $n-1$, and Δt_{n-1} represents the time step size from time step $n-1$ to n. By (9), define

$$\xi_{r,l,m}^n = \frac{D_{r,l,m}^{n+1}}{D_{r,l,m}^n}, \quad m = 1,2,\ldots,M, \tag{10}$$

which represents the proportion of the variation ratio of the photon energy at three sequent time steps (time step $n-1, n$ and $n+1$). For the sake of convenience, $\xi_{r,l,m}^n$ is called the proportion factor of the variation ratio of photon energy at time step n on the mesh (l,m). It is easy to see that the prediction of the photon energy for time step $n+1$ is equivalent to the prediction of $\xi_{r,l,m}^n$. Therefore, it is only necessary to concentrate on the prediction of $\xi_{r,l,m}^n$ in the following.

Since the photon energy variation ratio $D_{r,l,m}^{n+1}$ and $D_{r,l,m}^n$ near the wave front vary more greatly than at other locations, the same situation occurs to $\xi_{r,l,m}^n$. To predict the proportion factor $\xi_{r,l,m}^n$ along the line x_l, it is necessary to determine the location of wave front; that is, to determine the location where the photon energy varies acutely from the previous time to the current time. For this end, let

$$k_{r,l}^n = \arg \max_{1 \leqslant m \leqslant M} D_{r,l,m}^n,$$

then $(l, k_{r,l}^n)$ is the mesh on which the photon energy varies the most acutely from time step $n-1$ to n along the line $x = x_l$. Divide the meshes $\{(l,m) : 1 \leqslant m \leqslant M\}$ corresponding to the line $x = x_l$ into two parts, that is,

(i) $\{(l,m) : 1 \leqslant m \leqslant k_{r,l}^n\}$, the meshes locates at the area where the front of the photon energy has just or has not reached.
(ii) $\{(l,m) : k_{r,l}^n < m \leqslant M\}$, the meshes locates at the area where the front of the photon energy has already or just swept.

Next, the value of $\xi_{r,l,m}^n$ on the above two parts will be determined respectively, see Fig. 2 for an illustration:

(i) For $1 \leqslant m \leqslant k_{r,l}^n$, the variation of photon energy on different meshes will be greatly different: if the photon energy does not reach some mesh at the next time, then the photon energy on this mesh will not change; otherwise, if the front of the photon energy reaches the mesh at the next time, then the photon energy on this mesh will vary acutely from current time to next time. For both cases, let $\xi_{r,l,m}^n = 1$. Consequently, the photon energy at the next time is predicted by

$$E_{r,l,m}^{n+1,0} = E_{r,l,m}^n + \frac{\Delta t_n}{\Delta t_{n-1}}\left(E_{r,l,m}^n - E_{r,l,m}^{n-1}\right), \quad m = 1,2,\ldots,k_{r,l}^n. \tag{11}$$

In this way, it is assumed that the proportion factors of the variation ratio of photon energy on each mesh are same at two sequent time steps.

(ii) For $k_{r,l}^n + 1 \leqslant m \leqslant M$, the meshes (l,m) are in the location where the photon energy has already or has just past. Therefore, these meshes can be further divided into two parts: one where the front of photon energy has just swept and another where the front of the photon energy has already swept:

(a) Assume that there are k_M meshes where the front of the photon energy has just passed, here k_M is an integer related to M (see numerical experiments in Section 5). For these k_M meshes, the proportion factors of the variation ratio of the photon energy at the current time and the next time are greatly different. For these k_M meshes, the value of $\xi_{r,l,m}^n$ is given by

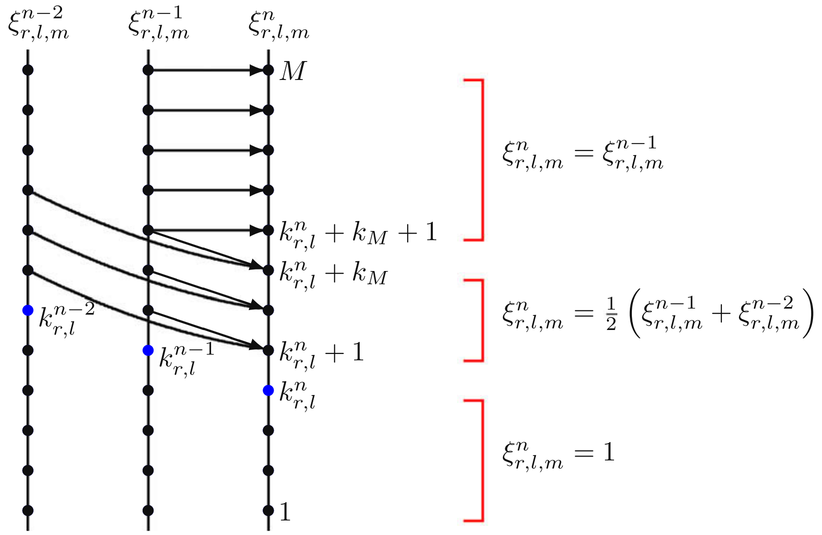


Fig. 2. Schematic diagram for choosing $\xi^n_{r,l,m}$.

$$\xi^n_{r,l,k^n_{r,l}+j}=\frac{1}{2}\left(\xi^{n-1}_{r,l,k^{n-1}_{r,l}+j}+\xi^{n-2}_{r,l,k^{n-2}_{r,l}+j}\right),\quad j=1,2,\ldots,k_M. \tag{12}$$

In fact, in the above way, it is assumed that the proportion factor of the variation ratio of photon energy on the mesh $(l,k^n_{r,l}+j)$ at time step n is approximately equal to the mean of the value on the mesh $(l,k^{n-1}_{r,l}+j)$ at time step $n-1$ and the value on the mesh $(l,k^{n-2}_{r,l}+j)$ at time step $n-2$. By (12), the photon energy on these k_M meshes can be predicted by

$$E^{n+1,0}_{r,l,k^n_{r,l}+j}=E^n_{r,l,k^n_{r,l}+j}+\frac{\Delta t_n\xi^n_{r,l,k^n_{r,l}+j}}{\Delta t_{n-1}}\left(E^n_{r,l,k^n_{r,l}+j}-E^{n-1}_{r,l,k^n_{r,l}+j}\right),$$

where $j=1,2,\ldots,k_M$.

(b) If $k^n_{r,l}+k_M+1\leqslant m\leqslant M$, then the front of photon energy has already swept the meshes. For these meshes, choose

$$\xi^n_{r,l,m}\approx\xi^{n-1}_{r,l,m},\quad m=M,M-1,\ldots,k^n_{r,l}+k_M+1, \tag{13}$$

where $\xi^n_{r,l,m}$ is defined by (10), and

$$\xi^{n-1}_{r,l,m}=\begin{cases}D^n_{r,l,m}/D^{n-1}_{r,l,m}, & m=M,M-1,\ldots,k^{n-1}_{r,l}+k_M,\\ \xi^{n-1}_{r,l,k^{n-1}_{r,l}+k_M}, & m=k^{n-1}_{r,l}+k_M-1,\ldots,k^n_{r,l}+k_M+1.\end{cases} \tag{14}$$

In this way, it is assumed that the proportion factors of the variation ratio of photon energy on a fixed mesh are almost same at two sequent time steps. Now, based on the above assumption, the photon energy on these meshes can be predicted by

$$E^{n+1,0}_{r,l,m}=E^n_{r,l,m}+\frac{\Delta t_n\xi^n_{r,l,m}}{\Delta t_{n-1}}(E^n_{r,l,m}-E^{n-1}_{r,l,m}),$$

where $m=M,M-1,\ldots,k^n_{r,l}+k_M+1$.

Once the above value of photon energy is predicted, a predicted photon temperature can be obtained. For electron and ion, in a similar way $D^n_{\alpha,l,m}$, $\xi^n_{\alpha,l,m}$ and $k^n_{\alpha,l}$ $(\alpha=e,i)$ can be defined, and the temperature at the next time step can be predicted.

4.2. Further modification of the temperature

For the initial temperature of electron, ion and photon obtained in Section 4.1, numerical experiments show that it is relatively accurate (the predicted temperature is very close to the temperature at time step $n+1$) on most of the meshes except the ones where the energy varies acutely. That is, the obtained initial temperature is not so effective near the meshes where the energy varies acutely (see the numerical results in Section 5). To make the ultimate initial value more effective, it is necessary to further modify the initial temperature on these meshes. For this purpose, assume that, along the line $x=x_l$, it is necessary to modify k_f meshes in front of the mesh on which the energy varies most acutely and k_b at the back of it. That is, the temperature on the meshes $(l, k^n_{\alpha,l} - k_f), \ldots, (l, k^n_{\alpha,l} + k_b)$ should be modified. To deal with the cases of electron, ion and photon uniformly, let

$$k^n_{u,l} = \min\{\max\{k^n_{\alpha,l} + k_b : \alpha = e, i, r\}, M\}$$

and

$$k^n_{d,l} = \max\{\min\{k^n_{\alpha,l} - k_f : \alpha = e, i, r\}, 1\},$$

then it is only necessary to modify the initial temperature of electron, ion and photon on $k^n_{u,l} - k^n_{d,l} + 1$ meshes $\{(l, m) : k^n_{d,l} \leqslant m \leqslant k^n_{u,l}\}$. Consider all the meshes corresponding to the L lines, it is necessary to modify the initial temperature on the subregion

$$\Omega^n = \bigcup_{l=1}^{L} \bigcup_{m=k^n_{d,l}}^{k^n_{u,l}} \Omega_{l,m}.$$

See Fig. 3 for a simple illustration.

To modify the temperature on Ω^n, we restrict the original equations (3) onto this subregion, and obtain a small scale nonlinear equations

$$\widetilde{\mathcal{F}}(\widetilde{\mathcal{T}}) = 0, \tag{15}$$

where $\widetilde{\mathcal{F}}$ and $\widetilde{\mathcal{T}}$ represent respectively the restriction of $\mathcal{F}$ and $\mathcal{T}$ onto the subregion Ω^n, that is,

$$\widetilde{\mathcal{F}} = \mathcal{F}|\Omega^n \quad \text{and} \quad \widetilde{\mathcal{T}} = \mathcal{T}|\Omega^n.$$

To define the system (15) properly, the following boundary conditions are supplied:

- For the boundary in the interior of Ω, the temperature obtained in Section 4.1 is used to supply a boundary condition. This is a reasonable choice since the temperature obtained in $\Omega \setminus \Omega^n$ is relatively accurate.
- For the physical boundary conditions, the original conditions for Eq. (1) are used.

In fact, with above boundary conditions, the subregion system (15) is a part of the full system (3), where the temperature in Ω^n is unknown, and that outside Ω^n is known.

For solving Eq. (15), the Kinsol package is also used. Besides, the temperature obtained previously on Ω^n is used as the initial iterate for solving this subregion nonlinear equations. After the solution of Eq. (15), a relatively accurate temperature on the subregion is obtained. The temperature on Ω^n and the previously predicted temperature on $\Omega \setminus \Omega^n$ comprise the initial iterate to the nonlinear iteration of the full, coupled nonlinear equations.

It is necessary to point out that the idea of modifying the subregion temperature is similar to the additive Schwarz preconditioned inexact Newton (ASPIN) method in [9], but there is some difference between them. In the ASPIN method, many subregion nonlinear systems are needed to be solved, and then the correction to the approximate solution on each subregion is summed up to form a new nonlinear system on the whole region. Here, only one subregion nonlinear system is needed to be solved and the approximate solution on the subregion will be used to form an initial iterate for solving the full nonlinear system. For details about the ASPIN method and its convergence results, see [1,9].

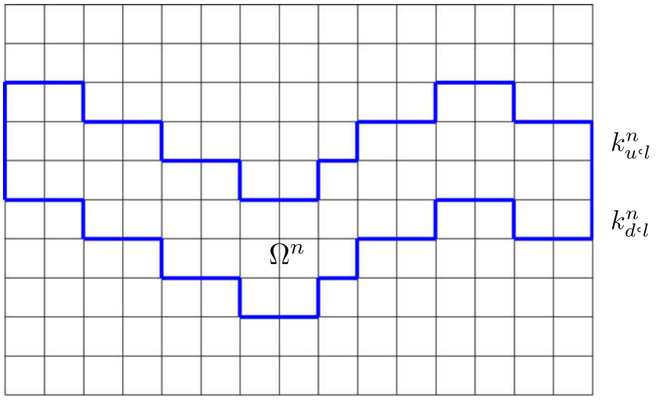


Fig. 3. Schematic diagram for subregion Ω^n.

5. Numerical results

In this section, some numerical results about the improved initial value will be presented.

In all the experiments, the maximal simulation physical time is 10. The initial temperature of electron, ion and photon in the domain is 3.0×10^{-4}. The parameters for computing domain are $A = 10,\ B = 10$, the upper half of the domain is plastic foam (CH), and the lower half part is glass (SiO_2). For convenience of reference, here is a list of the relevant physical parameters cited from [25]. For each material, the parameters c_α are defined by

$$c_e = 1.5\Gamma_e, \quad c_i = 1.5\Gamma_i, \quad c_r = 0.25\Gamma_r,$$

where Γ_α together with some other parameters are listed in Table 1.

5.1. Set up for solver

Kinsol package is employed to solve both the full nonlinear system and subregion nonlinear system, and GMRES method [29] is used to solve the Newton equation. Kinsol iterations are terminated when an absolute tolerance $\|\mathcal{F}(\mathcal{T}^{(k)})\|_\infty \leqslant \varepsilon_{residual}$ or a step tolerance $\|\delta^{(k)}\|_\infty \leqslant \varepsilon_{step}$ is satisfied. For the GMRES iteration, the controlling parameters include the forcing term $\bar{\eta}_k$, the maximal iteration number *max_gmr_it* and the maximal restart number *max_gmr_restart*. The concrete values of these parameters for solving both the full and subregion systems are listed in Table 2.

At each time step, for both the full and subregion nonlinear equations, the block diagonal part P^n of the Jacobian matrix at *initial guess* is formed, and P^n is used as the preconditioning matrix for all linear systems at this time step. Therefore, for different initial values, the corresponding preconditioning matrices have the same structure, but the corresponding entries of the matrices may be different. Note that each diagonal block of P^n is 3×3 sub-block with the following structure:

$$\begin{pmatrix} * & * & * \\ * & * & 0 \\ * & 0 & * \end{pmatrix}.$$

5.2. Control of time steps

Usually, the variation of energy is used to control the time step size in the numerical simulation of radiation diffusion problems [19,22,26,27]. In our numerical experiments, based on the variation of energy, the following self-adaptive rule is used to control the time step size Δt_n:

```
0. Let $\Delta t_{n-1}$ be known, give $\beta$ and $\gamma$, such that $0 < \beta < \gamma < 1$.
1. If the nonlinear iteration at time step $n$ succeeds, then
   1.1. if $\Delta E^{n-1}_{\text{relative}} \leqslant \beta$, then $\Delta t_n = 1.2\Delta t_{n-1}$;
   1.2. if $\beta < \Delta E^{n-1}_{\text{relative}} \leqslant \gamma$, then $\Delta t_n = \Delta t_{n-1}$;
   1.3. if $\Delta E^{n-1}_{\text{relative}} > \gamma$, then $\Delta t_n = 0.8\Delta t_{n-1}$.
2. $\Delta t_n \geqslant \max\{\Delta t_n, \Delta t_{\text{low}}\}$, $\Delta t_n \leqslant \min\{\Delta t_n, \Delta t_{\text{up}}\}$.
3. If the nonlinear iteration at time step $n$ fails, then $\Delta t_n = \frac{1}{2}\Delta t_n$, and solve the equations at time step $n$
   again.
4. If the time step size has reached the minimal threshold $\Delta t_{\text{low}}$ or if the time step size has been short-
   ened continuously five times, but the nonlinear iteration still fails, then exit the whole
   simulation.
```

In the above rule, the following symbols and parameters are used:

- $\Delta E^{n-1}_{\text{relative}} \equiv (E^n - E^{n-1})/E^{n-1}$ is the relative variation of the system energy from time step $n-1$ to n, where E^j represents the system energy at time j.
- β and γ are respectively the low and up threshold for the relative variation of the energy. In numerical experiments, $\gamma = 20\%$ is fixed, and β is set four different values: $\beta = 1\%$, 2%, 5% or 10%.
- Δt_{up} and Δt_{low} are respectively the maximal and minimal threshold for the time step size. In numerical experiments, $\Delta t_{\text{up}} = 1.0 \times 10^{-2}$ and $\Delta t_{\text{low}} = 1.0 \times 10^{-6}$.

Table 1
Physical parameters.

Parameter	ρ	Γ_e	Γ_i	Γ_r	A_e	A_i	A_r	β	A_{ei}	A_{er}
CH	1.1	45	70	0.007568	81	0.02	$2.1e2/\rho^2$	3.0	7000	79
SiO_2	2.5	40	40	0.007568	60	0.00017	$9.0e2/\rho^{1.5}$	2.4	4000	140

Table 2
KINSOL parameters for solving the full and subregion systems.

Parameters	$\varepsilon_{residual}$	ε_{step}	$\bar{\eta}_k$	*max_gmr_it*	*max_gmr_restart*
Full	10^{-7}	10^{-13}	10^{-2}	200	10
Subregion	10^{-4}	10^{-6}	10^{-4}	100	5

5.3. One-dimensional case

First consider a relative simple case: at each time, the photon temperature along the free surface is a constant. This is a one-dimensional problem in essence. In the experiments of this case, the photon temperature on the free surface is set

$$T_{frb}(x,t) = \begin{cases} \frac{1}{2} + \frac{3}{4}\sqrt{t}, & t \leqslant 4, \\ 2, & t > 4. \end{cases}$$

The partition number along axis x is set $L = 40$, and along axis y is set three different cases: $M = 160$, 320 and 640. Moreover, for the improved initial value, $k_M = M/40 + 1,\ k_f = M/40 + 1$ and $k_b = M/20 + 1$ are set in the numerical experiments.

To illustrate the effectiveness of the different initial values, the predicted photon temperature on five meshes along a line $x = x_l$ are listed in Table 3, where $M = 160$ and $\beta = 1\%$, and the forward/backward Euler method is employed to produce a predicted/corrected initial value. In the table, $T^{n+1,0}_{r,usua}$ represents the usual initial value, that is, $T^{n+1,0}_{r,usua} = T^n_r$; T^{n+1}_r is computed with the usual initial value; $T^{n+1,0}_{r,impr}$ represents the predicted temperature obtained by the method in this paper; and $T^{n+1,0}_{r,prcr}$ represents the predicted temperature obtained by the prediction/correction method. Besides, to show clearly the difference between an initial value and T^{n+1}_r, the relative error for different initial values on each mesh is listed in Table 4. For example, for the improved initial value, the relative error on each mesh is defined as

$$\varepsilon^{n+1}_{r,impr} = \frac{|T^{n+1,0}_{r,impr} - T^{n+1}_r|}{T^{n+1}_r}.$$

From Table 3, it can be seen that on the meshes 156 and 157, all the predicted photon temperature is 3.0e−4, which equals exactly the temperature at the next time. Apart from these two meshes, it can be seen from Tables 3 and 4 that the predicted/corrected initial value is the worst among all initial values. In particular, on mesh 159, the predicted/corrected initial value is 9.398861e−3, which is about 31.3237 times as high as the corresponding value of T^{n+1}_r. On this mesh, the relative error of the corrected initial value reaches 30.31374. The improved initial value is the best on all meshes. That is, on all meshes, the improved initial value is the one closest to T^{n+1}_r. The usual initial value is better than the predicted/corrected initial value, but worse than the improved initial value.

It should be pointed out that the nonlinear iteration is divergent when the above predicted/corrected initial iterate is used. Since the 2-D 3-T heat conduction equations have strong nonlinearities, therefore the initial iterate obtained by the prediction/correction method differs greatly from the temperature at the next time step in the vicinity of the front of energy. Thus the prediction/correction method is not suitable for choosing initial iterate for this problem. In the following discussion, the predicted/corrected initial value will no longer be concerned. Attention will be paid on comparing the effectiveness of the improved initial value with the usual initial value.

Fig. 4 is the curves of the photon temperature at current time, the corresponding next time and the initial temperature given by the improved initial value on a line $x = x_l$. In the figure, T^n_r and T^{n+1}_r denote respectively the photon temperature at

Table 3
Comparison of photon temperature for different initial values ($M = 160,\ \beta = 1\%,\ n = 9$).

m	156	157	158	159	160
T^{n+1}_r	3.0e−4	3.0e−4	3.0e−4	3.001513e−4	1.120870e−1
$T^{n+1,0}_{r,usua}$	3.0e−4	3.0e−4	3.0e−4	3.001128e−4	1.101993e−1
$T^{n+1,0}_{r,impr}$	3.0e−4	3.0e−4	3.0e−4	3.001598e−4	1.129541e−1
$T^{n+1,0}_{r,prcr}$	3.0e−4	3.0e−4	1.392022e−3	9.398861e−3	5.635451e−1

Table 4
Relative error of photon temperature for different initial values ($M = 160,\ \beta = 1\%,\ n = 9$).

m	156	157	158	159	160
$\varepsilon^{n+1}_{r,usua}$	0.0	0.0	0.0	1.282686e−4	1.684138e−2
$\varepsilon^{n+1}_{r,impr}$	0.0	0.0	0.0	2.831905e−5	7.735955e−3
$\varepsilon^{n+1}_{r,prcr}$	0.0	0.0	3.640073	3.031374e+1	4.027747e+0

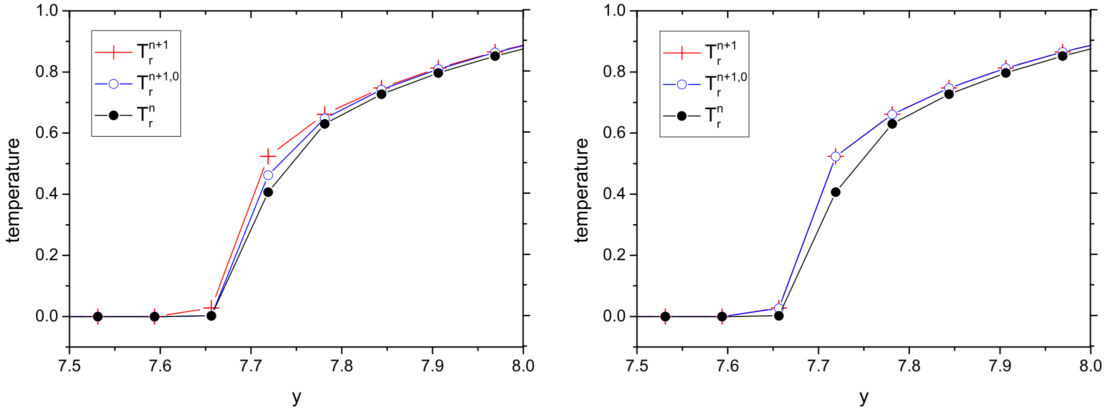


Fig. 4. Photon temperature at time step n and $n+1$, and the improved initial value of photon temperature for time step $n+1$ $(M = 160,\ \beta = 1\%)$. (left) No modified; (right) modified.

current time and next time, and $T_r^{n+1,0}$ the initial photon temperature determined by the improved initial value. The initial photon temperature in the left figure is obtained in the first phase of the prediction method, that in the right figure is obtained after the subregion nonlinear equations is solved. From the left figure it can be seen that the initial temperature obtained in the first phase is very near to the temperature at the next time on all the meshes except the part where energy (temperature) varies acutely. The right figure shows that the ultimate initial value is very close to the temperature of the next time on all meshes. Particularly, on the meshes where the energy varies acutely, the ultimate initial value is much improved by solving the subregion nonlinear equations.

It should be pointed out that the cost for computing the improved initial value is not expensive. In specific, Table 5 shows the CPU time (minute) for solving the full nonlinear equations and for computing the improved initial value. In the table, $\mathtt{CPU}_{total}$ represents the total CPU time for solving the full nonlinear equations, $\mathtt{CPU}_{impr}$ represents the CPU time for computing the improved initial value, and $\mathtt{CPU}_{phase\_one}$ and $\mathtt{CPU}_{phase\_two}$ represent respectively the CPU time spent on the first and second phases for computing the improved initial value. All the CPU time in the table is the sum spent on all the time steps. From this table, it can be seen that the total CPU time for computing the improved initial value is less than 0.05 of the total time for solving the full system. Furthermore, with the increase of the problem scale, the relative cost for computing the improved initial value is decreasing. For the total cost spent on computing the improved initial value, more than half time is used to solve the subregion nonlinear equations, and less than half is spent on predicting the energy for each mesh.

Next to discuss about the influence of β, one of the time step size controlling parameters, on the variation of the energy and time step size. The curves of the relative system energy variation and time step size for different β are shown in Fig. 5 when the usual initial value is used, where $M = 160$. Take the case of $\beta = 10\%$ as an example, it can be seen from the figure that the relative variation of the system energy rises continuously at about the first 50 time steps, and the corresponding time step size is enlarged step by step. At several time steps after that, the variation of the system energy gets near 10% with the maximal variation not larger than 12%, and the corresponding time step size is enlarged or remains unchanged. Around time step 60 ($t = 0.1155$), the time step size reaches the allowed maximal value, and the variation of the system energy reduces continuously. When $\beta = 5\%$, 2% and 1%, the curves of the system energy variation and time step size are similar to the case of $\beta = 10\%$. About after $t = 1.7655$, the curves of the system energy variation and time step size for all cases are flat, with the time step size reaching the allowed maximal value and remaining to the end of the simulation. When $M = 320$ and 640, the curves of the corresponding variation of the system energy and time step size are similar to those of the case $M = 160$.

It is necessary to point out that for the usual initial value in three cases of $M = 640$ (β = 1%, 2% and 5%), there exist nonlinear iteration failures and consequent reduction of the time step size, see Table 6. The curves of energy variation and time step size for these three cases are a little different from those of the improved initial value. Except these three cases, there is no nonlinear iteration failure in the simulation for all cases of the two initial values, and the curves of energy variation and time step size are totally same.

Table 5
CPU time for computing the improved initial value (one dimension, $\beta = 1\%$).

M	$\mathtt{CPU}_{total}$	$\mathtt{CPU}_{impr}$	$\mathtt{CPU}_{phase\_one}$	$\mathtt{CPU}_{phase\_two}$	$\mathtt{CPU}_{impr}/\mathtt{CPU}_{total}$
160	177.4920	8.5851	4.0868	4.4983	0.0484
320	590.9220	19.1995	8.7647	10.4348	0.0325
640	2592.0433	51.1832	23.8207	27.3625	0.0197

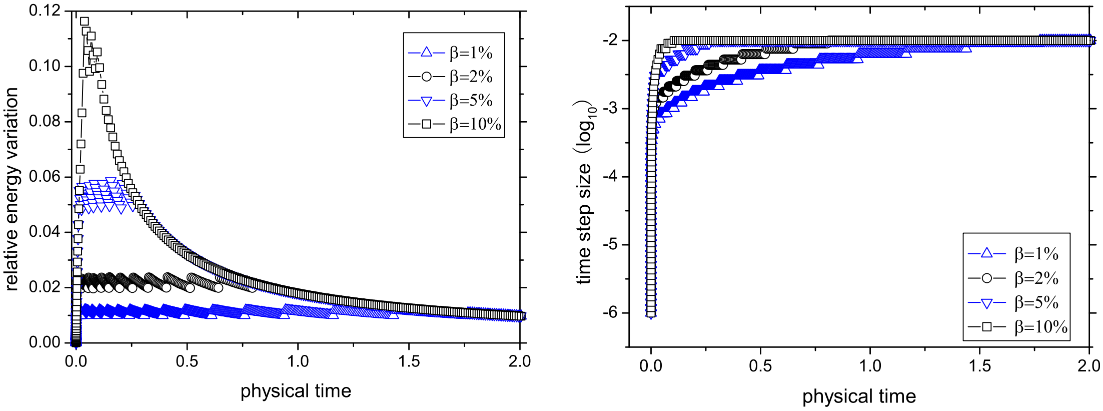


Fig. 5. Curves of the energy variation and time step size ($M = 160$). (left) Energy variation; (right) time step size.

Table 6
Total time steps for different initial values in one-dimensional case.

M	Usual initial value				Improved initial value			
	1%	2%	5%	10%	1%	2%	5%	10%
160	1389	1156	1062	1047	1389	1156	1062	1047
320	1389	1158	1063	1047	1389	1158	1063	1047
640	1393*	1161*	1064*	1047	1391	1158	1063	1047

The reliability of the computational result will be checked by verifying the system energy conservation. Let $\Delta E_{\text{enter}}^{n-1,n}$ and $\Delta E_{\text{system}}^{n-1,n}$ represent respectively the energy entered from the free surface and the increased energy of the system from time step $n-1$ to n. Define the energy conservation error from time step $n-1$ to n as

$$\mathcal{E}^n = \frac{|\Delta E_{\text{enter}}^{n-1,n} - \Delta E_{\text{system}}^{n-1,n}|}{\Delta E_{\text{enter}}^{n-1,n}}.$$

Take the cases of $M = 160$ and 320 as examples, where $\beta = 2\%$, the curves of the energy conservation error at all time steps for the usual initial value are shown in Fig. 6. From this figure, it can be seen that the energy conservation error is always less than 4×10^{-4}, and at most time steps less than 1×10^{-4} when $M = 160$. When $M = 320$, it is always less than 6×10^{-4}, and in most period of the simulation time less than 2×10^{-4}. This shows that the energy is conservative when the usual initial value is used and thus the computational result is reliable. In a similar way, the computational result of the improved initial value can also reflect the energy conservation. Besides, in the following it will be shown that at each time step the computed temperature is same for the two initial values.

Since the nonlinear problem may have multiple solutions, it is necessary to check if the two sequences starting from different initial values converge to the same solution. Let $T_{\alpha,usua}^t$ and $T_{\alpha,impr}^t$ $(\alpha = e, i, r)$ denote respectively the computed temperature at time t when the usual and improved initial values are used, and let

$$\tau_\alpha^t = \frac{\|T_{\alpha,usua}^t - T_{\alpha,impr}^t\|_2}{\|T_{\alpha,usua}^t\|_2}.$$

Take the case of $M = 160$, $\beta = 5\%$ as an example. The curves of τ_α^t $(\alpha = e, r)$ for all computational time are plotted in Fig. 7.

From Fig. 7, it can be seen that the relative error of photon temperature computed with the two initial values at all computing time is always less than 2×10^{-3}, and the corresponding relative error of electron always less than 7×10^{-4}. At most of the computing time, the error of photon and electron temperature computed by the two initial values is almost 0. Therefore, the photon temperature computed with the two initial values is same. So is electron temperature. In a similar way, it can be shown that the ion temperature computed with the two initial values is same. Therefore, when the usual and the improved initial values are employed in the computation, the sequences generated from these two initial values will converge to the same solution of the nonlinear system. Thus, as discussed above for the usual initial value, the computational result of the improved initial value is also reliable.

By the self-adaptive rule of the time step size, if all the nonlinear iterations succeed in the whole simulation when different initial values are employed, then the total time steps for different initial values should be same. To illustrate this point, the total time steps for different initial values are listed in Table 6. The data with a star means that there is nonlinear iteration failure in the simulation.

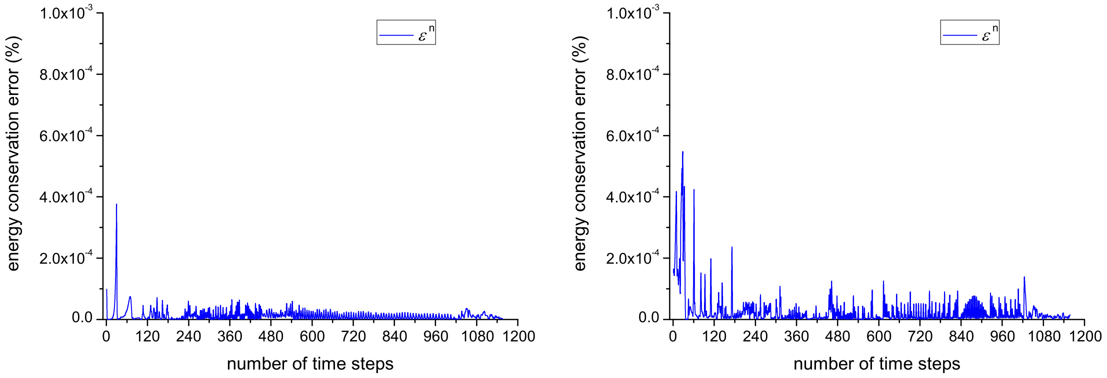


Fig. 6. Curves of the energy conservation error for usual initial value ($\beta = 2\%$). (left) $M = 160$; (right) $M = 320$.

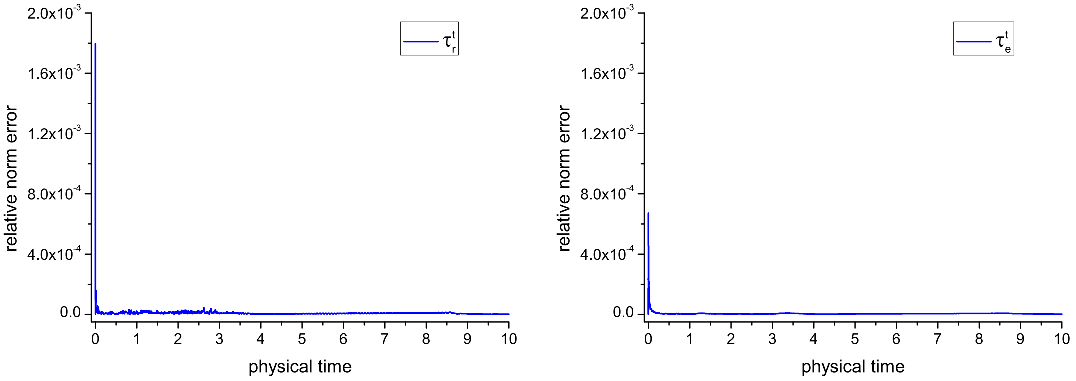
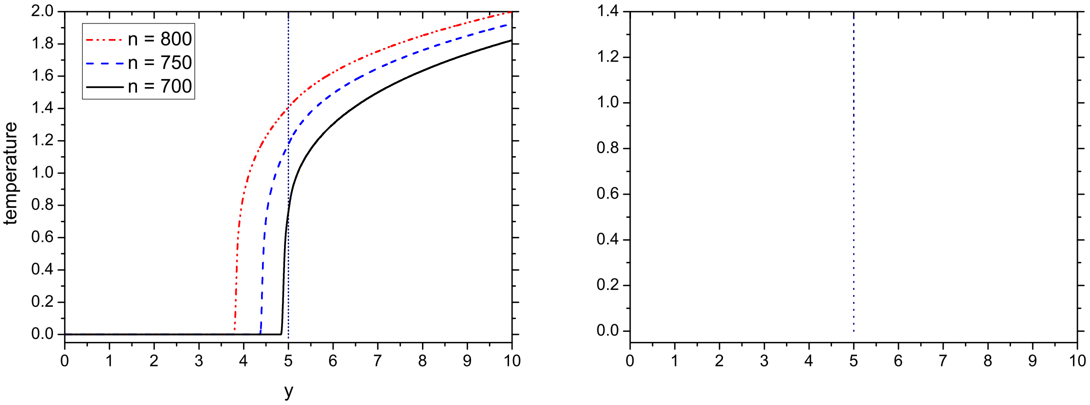


Fig. 7. Relative error curves of photon and electron temperature computed with two initial values ($M = 160,\ \beta = 5\%$). (left) Photon; (right): electron.

Table 6 shows that if there is no nonlinear iteration failure for the two initial values, then the total time steps for the two initial values are always same. This is consistent with the previous analysis. For the improved initial value, there is no nonlinear iteration failure for all cases; for the usual initial value, some nonlinear iteration failures occurred in three cases when $M = 640$, and consequently the total time steps of the usual initial value is more than those of the improved initial value. All the failures of the usual initial value are caused by the divergence of the nonlinear iteration. It should be pointed out that for all the failed cases, all the linear iterations converged. Thus the failures are caused by the initial value. The above analysis shows that the improved initial value can improve the convergence behavior of the nonlinear iteration. Besides, it is easy to see from Table 6 that the total time steps needed to complete the simulation are mainly determined by the time step size controlling parameter β, but have little relationship with the scale of the problem.

Next to discuss the law and character of electron energy diffusion and exchanging with photon and ion near the interface of the two materials. By Table 1, the electron energy diffusion coefficient in CH ($A_e = 81$) is larger than that in SiO_2 ($A_e = 60$), the photon/electron energy exchanging coefficient in CH ($A_{er} = 79$) is smaller than that in SiO_2 ($A_{er} = 140$), and the electron/ion energy exchanging coefficient in CH ($A_{ei} = 7000$) is larger than that in SiO_2 ($A_{ei} = 4000$). Thus when the energy crosses the interface between CH and SiO_2, the phenomenon will appear as follows: near the interface, the electron energy in SiO_2 diffuses slower than in CH, while by energy exchanging with photon and ion, electron gets more energy in SiO_2 than in CH. Therefore, near the interface, the electron temperature in SiO_2 will be higher than in CH. This phenomenon is observed in the computational tests, see Fig. 8.

In Fig. 8, the curves of electron temperature at three time steps along the line $x = x_c \equiv \left(l_c - \frac{1}{2}\right)h_1$ are plotted, where $l_c = \left[\frac{L}{2}\right]$. For comparison, the curves of photon temperature are also plotted in the figure. It can be seen that at time step 700, the electron temperature near the interface in SiO_2 is higher than in CH. In SiO_2, from interior to interface, the electron temperature rises acutely and reaches the highest near the interface; then in CH material, the electron temperature drops acutely and reaches the lowest near the interface. As time advances, the highest point and the lowest point of the electron temperature curves near the interface are farther and farther away from the interface. Thus the electron temperature varies more and more mildly near the interface. Besides, by observing the curves of electron temperature at the three time steps, it can be seen that the highest point is nearer to the interface than the lowest point, this is due to the difference of the electron energy diffusion coefficients in the two materials. The ion temperature has the similar nature to electron near the interface. It should be pointed out that Mo et al. also observed this phenomenon [25] where the computing domain is a half disc.



Next to analyze the variation law of the proportion factors of variation ratio of energy defined by (10). Consider the case of $M = 320$ and $\beta = 10\%$. In Fig. 9, the factor curves of photon and electron energy along the line $x = x_c$ at time step 200 are plotted. At the same time, for convenience of comparison, the concrete factors and temperature of electron and photon energy near the mesh where the factor is the largest are listed in Tables 7 and 8. In the tables, $\xi^{n}_{\alpha,m}$ represent the factors corresponding to the computed temperature (energy) at time step $n-1,\ n$ and $n+1$, i.e.,

$$\xi^{n}_{\alpha,m} = \frac{E^{n+1}_{\alpha,m} - E^{n}_{\alpha,m}}{E^{n}_{\alpha,m} - E^{n-1}_{\alpha,m}} \cdot \frac{\Delta t_n}{\Delta t_{n-1}},$$

$\xi^{n,0}_{\alpha,m}$ denote the factors corresponding to the computed temperature (energy) at time step $n-1$ and n and the predicted temperature (energy) at time step $n+1$, i.e.,

$$\xi^{n,0}_{\alpha,m} = \frac{E^{n+1,0}_{\alpha,m} - E^{n}_{\alpha,m}}{E^{n}_{\alpha,m} - E^{n-1}_{\alpha,m}} \cdot \frac{\Delta t_n}{\Delta t_{n-1}}.$$

In the above two expressions, $E^{j}_{\alpha,m}(j = n-1, n, n+1)$ represent the energy on the mesh (l_c, m) at time step j, $E^{n+1,0}_{\alpha,m}$ represent the energy corresponding to the predicted temperature at time step $n+1$. Particularly, if $E^{n}_{\alpha,m} = E^{n-1}_{\alpha,m}$, then let $\xi^{n}_{\alpha,m} = \xi^{n,0}_{\alpha,m} = 1$.

From Fig. 9 and Table 7, it can be seen that, from time step $n-1$ to n and then to $n+1$, the proportion factor of the variation ratio of the photon energy on mesh 251 is 32.2369, but the factor on mesh 252 is only 0.8013. The factors on these two meshes differ by 40.23 times. This is because mesh 251 is in the front location of the energy broadcasting, while mesh 252 is in the location where the front of energy has just passed. The meshes after 252 are in the location where the front of photon energy has just or already swept, therefore the difference of the factors on these meshes are not so distinct. With the increment of the mesh indices, the factors get close to 1. As shown by Fig. 9 and Table 7, the predicted photon temperature is very close to the photon temperature at time step $n+1$, and the corresponding $\xi^{n,0}_{r,m}$ and $\xi^{n}_{r,m}$ are very close. Besides, by observing the photon temperature on mesh 250, it can be seen that, at time step 200, the photon temperature is 3.0e−4, which is same as the initial temperature in the region; at time step 201, the photon temperature has increased a bit. For the improved initial

Table 7
Comparison of the proportion factors of the variation ratio of photon energy ($n = 200,\ M = 320,\ \beta = 10\%$).

m	250	251	252	253	254	255
$\xi^{n}_{r,m}$	1.0	32.2369	0.8013	0.9127	0.9442	0.9599
$\xi^{n,0}_{r,m}$	1.0	32.1989	0.8042	0.9101	0.9417	0.9575
$T^{n}_{r,m}$	3.0e−4	3.0212e−4	1.3183e−1	4.8784e−1	5.7732e−1	6.4129e−1
$T^{n+1}_{r,m}$	3.000003e−4	1.3820e−2	4.0120e−1	5.3602e−1	6.1259e−1	6.6916e−1
$T^{n+1,0}_{r,m}$	3.000002e−4	1.3446e−2	4.0123e−1	5.3592e−1	6.1253e−1	6.6912e−1

Table 8
Comparison of the proportion factors of the variation ratio of electron energy ($n = 200,\ M = 320,\ \beta = 10\%$).

m	250	251	252	253	254	255
$\xi^{n}_{e,m}$	1.0	1.5902	0.9260	0.9557	0.9671	0.9734
$\xi^{n,0}_{e,m}$	1.0	1.5897	0.9262	0.9555	0.9670	0.9733
$T^{n}_{e,m}$	3.0e−4	3.0052e−4	8.1559e−3	3.2487e−2	5.4037e−2	7.3932e−2
$T^{n+1}_{e,m}$	3.000002e−4	1.7652e−3	2.1731e−2	4.4435e−2	6.5030e−2	8.4254e−2
$T^{n+1,0}_{e,m}$	3.000001e−4	1.7577e−3	2.1732e−2	4.4433e−2	6.5029e−2	8.4254e−2

value, by solving the subregion nonlinear system, the subtle increment of the photon temperature on this mesh has been reflected (the computed photon temperature is 3.000003e−4 and the predicted photon temperature is 3.000002e−4). The proportion factors of the variation ratio of electron energy are similar.

The total iteration numbers, function evaluation numbers and CPU time for different β and M are listed in Table 9. In this table and afterward ones, the following symbols are used:

- NNI represents the nonlinear iteration number for solving the nonlinear equations;
- NLI represents the linear iteration number for solving the linear equations;
- NFE represents the function evaluation number; and
- CPU represents the CPU time (minute).

Besides, to show the computational efficiency of the improved initial value, the relative reduction of the total nonlinear iterations, function evaluations and CPU time of the improved initial value to those of the usual initial value are listed in Table 10. For example, the improved efficiency of the nonlinear iterations is given by $100(\mathtt{NNI}_{usua\_total} - \mathtt{NNI}_{impr\_total})/\mathtt{NNI}_{usua\_total}$, where $\mathtt{NNI}_{usua\_total}$ is the total nonlinear iteration number of the usual initial value and $\mathtt{NNI}_{impr\_total}$ the total nonlinear iteration number of the improved initial value.

From Table 9, it can be seen that for all cases, the total nonlinear iteration numbers, linear iteration numbers, function evaluation numbers and CPU time of the improved initial value are always less than those of the usual initial value. Table 10 shows that, for all cases, the total nonlinear iteration numbers, function evaluations and CPU time of the improved initial value are less at least 50% than those of the usual initial value. Furthermore, the larger the scale of the problem is, the more

Table 9
Total iterations and CPU time for different initial values in one-dimensional case.

β (%)	M	Usual initial value				Improved initial value			
		NNI	NLI	NFE	CPU	NNI	NLI	NFE	CPU
1	160	3632	190,697	199,350	371.52	1805	81,922	86,921	177.49
	320	4120	401,530	411,159	1792.55	1581	130,694	135,445	590.92
	640	5938	1,054,129	1,068,656	12,325.19	1864	225,711	230,878	2592.04
2	160	3252	187,719	195,379	362.39	1522	79,039	83,239	168.35
	320	3779	396,024	404,740	1730.16	1449	128,356	132,412	568.53
	640	5691	1,070,097	1,083,917	12,712.33	1794	231,501	236,341	2649.14
5	160	3081	186,582	193,806	356.66	1419	79,096	82,996	165.30
	320	3631	396,142	404,467	1735.68	1335	126,241	129,974	569.53
	640	5470	1,044,835	1,058,072	12,186.26	1654	221,551	225,994	2473.13
10	160	3060	186,695	193,862	357.74	1375	77,652	81,449	164.34
	320	3619	397,186	405,471	1729.55	1309	125,257	128,922	563.48
	640	5444	1,045,010	1,058,240	11,623.80	1658	226,541	230,998	2538.86

Table 10
Improved efficiency of the improved initial value in one-dimensional case (%).

β (%)	M								
	160			320			640		
	NNI	NFE	CPU	NNI	NFE	CPU	NNI	NFE	CPU
1	50.30	56.40	52.23	61.63	67.06	67.03	68.61	78.40	78.97
2	53.20	57.40	53.54	61.66	67.28	67.14	68.48	78.20	79.16
5	53.94	57.18	53.65	63.23	67.87	67.19	69.76	78.64	79.71
10	55.07	57.99	54.06	63.83	68.20	67.42	69.54	78.17	78.16

computational cost can be saved. This shows that the improved initial value can improve the computational efficiency greatly. In particular, the improved initial value has the potential and advantage for solving large scale problem.

5.4. Two-dimensional case

In this subsection, we consider the two-dimensional case. For this case, the photon temperature on the free surface is set as

$$T_{frb}(x,t) = \begin{cases} \frac{1}{8}(2+\sqrt{t})\sin((2\pi x)/A) + \frac{3}{4}\sqrt{t} + \frac{1}{2}, & t \leqslant 4, \\ \frac{1}{2}\sin((2\pi x)/A) + 2, & t > 4. \end{cases}$$

For the domain partition, altogether three cases are considered: $L = M = 64$, $L = M = 128$ and $L = M = 256$. Besides, the parameters for the improved initial value are set as $k_M = M/32 + 1$, $k_f = M/32 + 1$ and $k_b = M/16 + 1$.

In Table 11, the CPU time (minute) for solving the full nonlinear equations and computing the improved initial value is listed. The symbols in this table have same meanings as in Table 5. It can be seen from this table that the total cost of CPU time for computing the improved initial value is less than 0.07 of the total time for solving the full nonlinear equations. Furthermore, with the increase of the problem scale, the relative cost for computing the improved initial value has a tendency of decrease. For the cost spent on computing the improved initial value, more than half time is used to solve the subregion nonlinear equations, and less than half time is spent at the first phase to predict the energy value.

As in the case of one dimension, the reliability of the computational result can be verified by testing the energy conservation. Here the energy conservation error curves for the usual initial value are plotted in Fig. 10, where $\beta = 2\%$. The left and right figures are respectively the error curves corresponding to $M = 64$ and $M = 128$. It is easy to see from Fig. 10 that the energy conservation error for $M = 64$ is less than 1.5×10^{-3}, and for $M = 128$ less than 2.5×10^{-3}. This means that the energy is conservative in the process of simulation when the usual initial value is used. For the improved initial value, it can be shown in the similar way that the energy is also conservative.

Table 11
CPU time for computing the improved initial value (two dimension, $\beta = 1\%$).

M	CPU_{total}	CPU_{impr}	CPU_{phase_one}	CPU_{phase_two}	$\mathrm{CPU}_{impr}/\mathrm{CPU}_{total}$
64	98.6923	6.6922	3.3064	3.3858	0.0678
128	528.3387	25.1565	9.9032	15.2535	0.0476
256	5905.7103	155.5728	72.4207	83.1521	0.0263

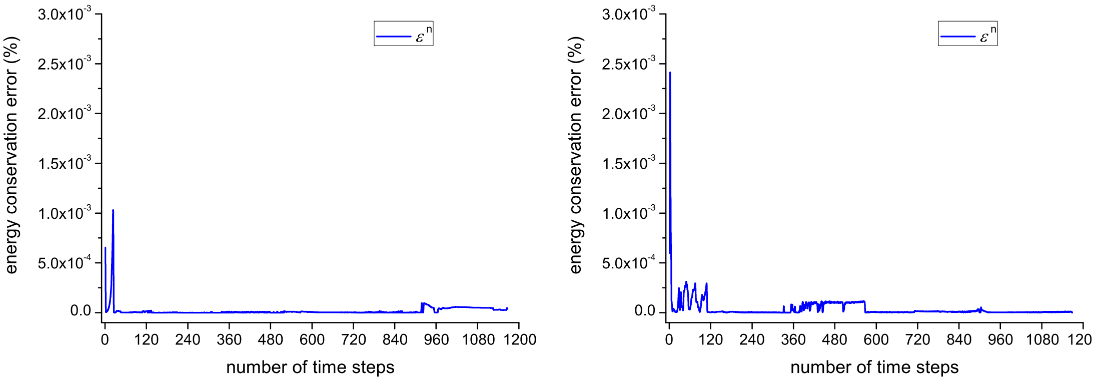


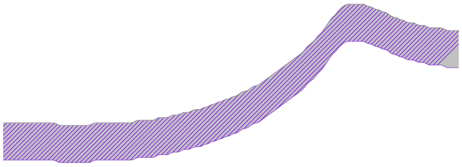
Fig. 10. Energy conservation error curves for the usual initial value (two dimension, $\beta = 2\%$). (left) $M = 64$; (right) $M = 128$.

To determine the improved initial value, a nonlinear system on a subregion Ω^n must be solved at each time step. In the whole simulation, Ω^n can always reflect the location where the energy varies most acutely. The subregions at two time steps are given as examples in Fig. 11.

It can be seen from the left figure of Fig. 11 that, at time step 600, the low boundary of the subregion Ω^n has crossed the interface between CH and SiO_2, while the up boundary of the subregion is close to the free surface. The outline of the subregion approximately reflects the energy distribution on the free surface. It can be observed from the right figure that at time step 900, the subregion has completely lain in the SiO_2 region, and the low boundary of the subregion almost contacts the low boundary of the computing region. Because of the energy diffusion, the outline of the subregion determined at this time step is "flatter" than that at time step 600.

In Table 12, the total time steps for the two initial values are listed. The simulation is not completed when the usual initial value is used for all cases of $M = 256$. For the improved initial value, although there exist nonlinear iteration failures in the four cases of $M = 256$, the simulation is completed successfully. When $M = 64$ and 128, the time steps for the two initial values are totally same because there is no nonlinear iteration failure for these two initial values.

In Table 13, the total nonlinear iterations, linear iterations, function evaluations and CPU time for the two initial values are presented. Besides, in Table 14 the improved efficiency of the improved initial value is given for different cases. It can be



are not so obvious; after time step 50, the linear iterations of the improved initial value are less than those of the usual initial value. In particular, about from time step 470, the linear iterations of the usual initial value are at least two times of the improved initial value. This shows that the improved initial value can effectively save the linear iterations. Consequently, the computational efficiency can be improved by using the improved initial value.

To show the whole trend of temperature in the domain, the following Fig. 13 for photon and electron temperature at $t = 1.049$ are plotted. It is clear that near the free surface, the photon temperature can reflect the distribution of the energy on the free surface. The electron temperature on the free surface can also reflect the distribution of the energy, but not so clearly. The electron temperature in SiO_2 region near the interface has a clear protuberance. Furthermore, it is easy to see that the variation of the electron temperature is not so acute as the photon temperature. The ion temperature is similar to that of electron.

6. Conclusion and remarks

The 2-D 3-T heat conduction equations (3) are a kind of system with strong nonlinearities. For solving this kind of equations, one nonlinear iteration method must be employed and an effective nonlinear initial iterate should be supplied. The nonlinear initial iterate has important influence on the efficiency and convergence property of the method. For solving the discretized 2-D 3-T heat conduction equations, usually the temperature at the current time is used as the initial iterate for computing the temperature at the next time. This kind of initial iterate is not effective in many cases. In this paper, a kind of method of choosing an effective nonlinear initial iterate for solving the discretized 2-D 3-T heat conduction equations is proposed.

In the proposed method, if the temperature at the current time is known, then the energy of electron, ion and photon at the next time is predicted firstly for obtaining the temperature; then, the obtained initial value of temperature is further modified by solving a subregion nonlinear system. Numerical results show that the initial iterate designed in this paper can greatly improve the computational efficiency. At the same time, it can improve the robustness of the nonlinear solver in many cases.

At last we point out that the idea of choosing the initial iterate in this paper can be generalized to the cases of other kind of diffusion equations. In fact, when the numerical solution at each time step n is obtained, a subregion containing the wave front of time step $n+1$ may be predicted by detecting the gradient of the solution at time step n or using a front tracking method. Then the spacial computing domain Ω can be divided into two parts

$$\Omega = \Omega_1^n \cup \Omega_2^n,$$

where Ω_1^n contains the location of wave front, and $\Omega_2^n = \Omega \setminus \Omega_1^n$. Since the wave front is contained in the subregion Ω_1^n, the solution at time step $n+1$ will not vary acutely on Ω_2^n. Therefore by using some interpolation techniques, an approximate solution on Ω_2^n may be given. Then the approximate solution on Ω_2^n can supply a boundary condition for Ω_1^n, and a subregion system on Ω_1^n can be defined. By solving the subregion system, an new initial value can be obtained, or an effective preconditioner may be constructed.

Acknowledgments

The authors gratefully acknowledge the anonymous referees and Prof. Yabe for many valuable suggestions and remarks which helped to improve the paper.

References

[1] H.-B. An, On convergence of the additive Schwarz preconditioned inexact Newton method, SIAM J. Numer. Anal. 43 (5) (2005) 1850–1871.
[2] H.-B. An, Z.-Z. Bai, A globally convergent Newton-GMRES method for large sparse systems of nonlinear equations, Appl. Numer. Math. 57 (2007) 235–252.
[3] C. Baldwin, P.N. Brown, R. Falgout, F. Graziani, J. Jones, Iterative linear solvers in 2D radiation-hydrodynamics code: methods and performance, J. Comput. Phys. 154 (1999) 1–40.
[4] S. Bellavia, B. Morini, A globally convergent Newton-GMRES subspace method for systems of nonlinear equations, SIAM J. Sci. Comput. 23 (2001) 940–960.
[5] P.N. Brown, Y. Saad, Hybrid Krylov methods for nonlinear systems of equations, SIAM J. Sci. Stat. Comput. 11 (1990) 450–481.
[6] P.N. Brown, Y. Saad, Convergence theory of nonlinear Newton–Krylov algorithms, SIAM J. Optim. 4 (1994) 297–330.
[7] P.N. Brown, D.E. Shumaker, C.S. Woodward, Fully implicit solution of large-scale non-equilibrium radiation diffusion with high order time integration, J. Comput. Phys. 204 (2005) 760–783.
[8] P.N. Brown, C.S. Woodward, Preconditioning strategies for fully implicit radiation diffusion with material-energy transfer, SIAM J. Sci. Comput. 23 (2001) 499–516.
[9] X.-C. Cai, D.E. Keyes, Nonlinearly preconditioned inexact Newton algorithms, SIAM J. Sci. Comput. 24 (2002) 183–200.
[10] A.M. Collier, A.C. Hindmarsh, R. Serban, C.S. Woodward, User Documentation for KINSOL v2.4.0, Technical Report UCRL-SM-208116, Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, Livermore, 2006.
[11] R.S. Dembo, S.C. Eisenstat, T. Steihaug, Inexact Newton methods, SIAM J. Numer. Anal. 19 (1982) 400–408.
[12] J.E. Dennis, R.B. Schnabel, Numerical Methods for Unconstrained Optimization and Nonlinear Equations, SIAM, Philadelphia, 1996.
[13] S.C. Eisenstat, H.F. Walker, Globally convergent inexact Newton methods, SIAM J. Optim. 4 (1994) 393–422.
[14] S.C. Eisenstat, H.F. Walker, Choosing the forcing term in an inexact Newton method, SIAM J. Sci. Comput. 17 (1996) 16–32.
[15] P.F. Fischer, Projection techniques for iterative solution of $A\underline{x} = b$ with successive right-hand sides, Comput. Method Appl. Mech. Eng. 163 (1998) 193–204.

[16] S.-W. Fu, H.-Q. Fu, L.-J. Shen, S.-K. Huang, G.-N. Chen, A nine point difference scheme and iteration solving for two-dimensional energy equations with three temperatures, Chinese J. Comput. Phys. 15 (1998) 489–497.

[17] P.M. Gresho, R.L. Lee, R.L. Sani, On the time dependent solution of the incompressible Navier–Stokes equations in two and three dimensions, Rec. Adv. Numer. Method Fluid 1 (1980) 27–81.

[18] A.C. Hindmarsh, P.N. Brown, K.E. Grant, S.L. Lee, R. Serban, D.E. Shumaker, C.S. Woodward, SUNDIALS: suite of nonlinear and differential/algebraic equation solvers, ACM Trans. Math. Softw. 31 (2005) 363–396.

[19] L.H. Howell, J.A. Greenough, Radiation diffusion for multi-fluid Eulerian hydrodynamics with adaptive mesh refinement, J. Comput. Phys. 184 (2003) 53–78.

[20] D.A. Knoll, D.A. Keys, Jacobian-free Newton–Krylov method: a survey of approaches and applications, J. Comput. Phys. 193 (2004) 357–397.

[21] D.A. Knoll, W.J. Rider, G.L. Olson, An efficient nonlinear solution method for non-equilibrium radiation diffusion, J. Quant. Spectrosc. Radiat. Transf. 63 (1999) 15–29.

[22] D.A. Knoll, W.J. Rider, G.L. Olson, Nonlinear convergence, accuracy, and time step control in nonequilibrium radiation diffusion, J. Quant. Spectrosc. Radiat. Transf. 70 (2001) 25–36.

[23] R. Kress, Numerical Analysis, Springer-Verlag, New York, 1998.

[24] R.B. Lowrie, A comparison of implicit time integration methods for nonlinear relaxation and diffusion, J. Comput. Phys. 196 (2004) 566–590.

[25] Z.-Y. Mo, L.- J. Shen, G. Wittum, Parallel adaptive multigrid algorithm for 2-D 3-T diffusion equations, Int. J. Comput. Math. 81 (3) (2004) 361–374.

[26] C.C. Ober, J.N. Shadid, Studies on the accuracy of time-integration methods for the radiation-diffusion equations, J. Comput. Phys. 195 (2004) 743–772.

[27] M. Pernice, B. Philip, Solution of equilibrium radiation diffusion problems using implicit adaptive mesh refinement, SIAM J. Sci. Comput. 27 (2006) 1709–1726.

[28] M. Pernice, H.F. Walker, NITSOL: a Newton iterative solver for nonlinear systems, SIAM J. Sci. Comput. 19 (1998) 302–318.

[29] Y. Saad, M.H. Schultz, GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems, SIAM J. Sci. Comput. 7 (1986) 856–869.

[30] D. Tromeur-Dervout, Y. Vassilevski, Choice of initial guess in iterative solution of series of systems arising in fluid flow simulations, J. Comput. Phys. 219 (2006) 210–227.