Available online at www.sciencedirect.com
ScienceDirect
Journal of Computational Physics 224 (2007) 168–181
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

Some new discretization and adaptation and multigrid methods for 2-D 3-T diffusion equations

Jun Jiang [a], Yunqing Huang [a,*], Shi Shu [a], Shi Zeng [b]

[a] *Hunan Key Laboratory for Computation and Simulation in Science and Engineering, Xiangtan University, Xiangtan City, YangGuTang, Xiangtan, Hunan 411105, PR China*
[b] *Department of Engineering Physics, Tsinghua University, Beijing 100084, PR China*

Received 1 September 2006; received in revised form 10 January 2007; accepted 11 January 2007
Available online 27 January 2007

Abstract

In the simulation of laser-driven implosion of a fuel capsule in inertial confinement fusion experiments, a system of two-dimensional diffusion equations coupled with electron, iron and photon temperature are widely used to approximately describe the process of energy across multiple materials and the exchange of energy among electrons, irons and photons. The numerical solution of such equations is always challenging because of its strong nonlinear phenomena and strong discontinuous interfaces. In this article, we design a symmetric finite volume method, develop the corresponding preconditioning technique, and propose a mesh adaptation algorithm based on Hessian matrix and a two-grid method. Using these new methods, we demonstrate that the energy conservation error and computation efficiency of the integrated algorithm are much better than classical method.

Keywords: Inertial confinement fusion; Finite volume element method; Precondition; Mesh adaptation; Two-grid method

1. Introduction

Unsteady radiation fluid dynamics equations derived from the conservational of mass, momentum and energy are the basic control partial differential equations for laser-driven implosion of a fuel capsule with the goal of igniting a self-sustained reaction in inertial confinement experiments [1–4]. In the course of solving these equations, we found that the energy equation is the most time consuming, yet in the computation of the whole system, solving the energy equation is an very important one. Because of the strong nonlinearity and strong discontinuous interfaces, it is critical to design efficient numerical algorithms for the energy equations. Fortunately, in most cases we can simplify the radiation fluid dynamics equations into 3-T diffusion equations without losing essential properties , which describe the radiation evolution of energy across multiple materials and perceive the exchange of energy among electrons, ions and photons [5].

* Corresponding author. Tel.: +86 732 8292187; fax: +86 732 8292351.
E-mail address: huangyq@xtu.edu.cn (Y. Huang).

0021-9991/$ - see front matter
doi:10.1016/j.jcp.2007.01.013

In this paper, we design a symmetric finite volume element method (SFVEM) and develop the corresponding preconditioning technique, and then propose a mesh adaptation algorithm based on Hessian matrix and a two-grid method. Using the new methods, we design an integrated algorithm to solve the 2-D 3-T diffusion equations, and demonstrated by numerical examples that the integrated algorithm is efficient.

The rest of the paper proceeds as follows. We present 2-D 3-T diffusion equations in Section 2. We propose some new numerical methods and improved processes for solving the model equations in Section 3. We report some numerical examples for the new approach in Section 4 and we finish the paper by a short discussion in Section 5.

2. 2-D 3-T diffusion equations

Two-dimensional diffusion equations coupled with electron, iron and photon are defined by

$$c_{\mathrm{e}}\frac{\partial T_{\mathrm{e}}}{\partial t}-\frac{1}{\rho}\nabla(K_{\mathrm{e}}\nabla T_{\mathrm{e}})=\omega_{\mathrm{ei}}(T_{\mathrm{i}}-T_{\mathrm{e}})+\omega_{\mathrm{er}}(T_{\mathrm{r}}-T_{\mathrm{e}}), \tag{1}$$

$$c_{\mathrm{i}}\frac{\partial T_{\mathrm{i}}}{\partial t}-\frac{1}{\rho}\nabla(K_{\mathrm{i}}\nabla T_{\mathrm{i}})=\omega_{\mathrm{ei}}(T_{\mathrm{e}}-T_{\mathrm{i}}), \tag{2}$$

$$\frac{4}{\rho}c_{\mathrm{r}}T_{\mathrm{r}}^{3}\frac{\partial T_{\mathrm{r}}}{\partial t}-\frac{1}{\rho}\nabla(K_{\mathrm{r}}\nabla T_{\mathrm{r}})=\omega_{\mathrm{er}}(T_{\mathrm{e}}-T_{\mathrm{r}}). \tag{3}$$

In the above equations, $T_{\mathrm{e}},\ T_{\mathrm{i}},\ T_{\mathrm{r}}$ are the temperature functions of electron, iron and photon, respectively. ρ denotes the density of the medium, which is a constant within each subdomain and discontinuous across interfaces of subdomains. $K_{\mathrm{e}},\ K_{\mathrm{i}},\ K_{\mathrm{r}}$ are conductive coefficients of electron, iron, photon, where

$$K_{\alpha}=A_{\alpha}T_{\alpha}^{5/2},\quad \alpha=\mathrm{e},\mathrm{i},$$
$$K_{\mathrm{r}}=A_{\mathrm{r}}T_{\mathrm{r}}^{3+\beta}.$$

ω_{ei} is the energy exchange coefficient between electron and iron, ω_{er} is the energy exchange coefficients between electron and photon, where

$$\omega_{\mathrm{ei}}=\rho A_{\mathrm{ei}}T_{\mathrm{e}}^{-2/3},$$
$$\omega_{\mathrm{er}}=\rho A_{\mathrm{er}}T_{\mathrm{e}}^{-1/2}.$$

Parameter $c_{\alpha},\ A_{\alpha}\ (\alpha=\mathrm{e},\mathrm{i},\mathrm{r}),\ \beta,\ A_{\mathrm{ei}},\ A_{\mathrm{er}}$ are constant within each subdomain, but they are discontinuous across interfaces of subdomains. The system energy of unit mass is defined by

$$E=E_{\mathrm{e}}+E_{\mathrm{i}}+E_{\mathrm{r}},\quad E_{\mathrm{e}}=c_{\mathrm{e}}T_{\mathrm{e}},\quad E_{\mathrm{i}}=c_{\mathrm{i}}T_{\mathrm{i}},\quad E_{\mathrm{r}}=\frac{1}{\rho}c_{\mathrm{r}}T_{\mathrm{r}}^{4}. \tag{4}$$

According to physical experiments, we define the computation domain, boundary conditions and initial conditions as follows.

Computation domain:

$\{(x,y,t)\mid(x,y)\in\Omega_{xy},\ 0\leqslant t\leqslant T\}$, where $\Omega_{xy}(=\bigcup_{i=1}^{3}\Omega_{\mathrm{i}})$ is a half upper circle with radius R under two-dimensional Cartesian coordinate system, and the circle center is overlapped with the coordinate origin, diameter is aligned with the X coordinate axis. Especially, three types of materials are included such that the innermost subdomain (Ω_1) is filled with deuterium gas (DT) and covers the area with $0\leqslant r<R_1$, the middle subdomain (Ω_2) is filled with glass (SiO_2) and covers the area with $R_1\leqslant r<R_2$ and the outside subdomain (Ω_3) is filled with plastic foam (CH). Γ_1 denotes the free boundary and Γ_2 denotes the wall boundary. (see Fig. 1).

Boundary conditions:

1. Wall: $K_{\alpha}\frac{\partial T_{\alpha}}{\partial n}|_{\Gamma_2}=0,\ \alpha=\mathrm{e},\mathrm{i},\mathrm{r}$, where $\frac{\partial T_{\alpha}}{\partial n}$ is the outer normal vector along boundary,
2. Free: $K_{\alpha}\frac{\partial T_{\alpha}}{\partial n}|_{\Gamma_1}=0,\ \alpha=\mathrm{e},\mathrm{i},$

$$T_{\mathrm{r}}=T_{\mathrm{r}}(x,y,t)|_{\Gamma_1}=g_1(x,y).$$

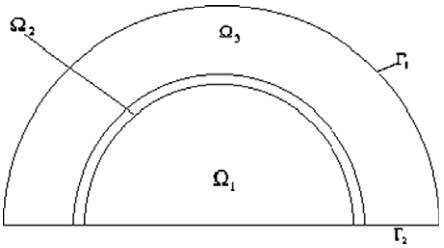

Fig. 1. Computation domain Ω_{xy}.

Initial condition:

$$T_\alpha(x,y,0) = T_\alpha^0(x,y) = g_2(x,y), \quad \alpha = \mathrm{e},\mathrm{i},\mathrm{r}.$$

2-D 3-T diffusion equations (1)–(3) approximately describe the process of radiant energy broadcasting in the quiescent medium and energy exchange of electrons with photons and irons.

3. Solution methodology

To solve the 2-D 3-T diffusion system (1)–(3), we will develop an integrated numerical process including discretization, linearization, and precondition for solving the linearized system.

3.1. Common numerical methods

For the property of absolute numerical stability, we use backward *Euler* stencil to discretize the temporal derivative. Then, we get the nonlinear partial differential equations as follows:

$$c_\mathrm{e}T_\mathrm{e} - \frac{\Delta t}{\rho}\nabla\cdot(K_\mathrm{e}\nabla T_\mathrm{e}) - \Delta t w_\mathrm{ei}(T_\mathrm{i}-T_\mathrm{e}) - \Delta t w_\mathrm{er}(T_\mathrm{r}-T_\mathrm{e}) = c_\mathrm{e}T_\mathrm{e}^{(n-1)}, \tag{5}$$

$$c_\mathrm{i}T_\mathrm{i} - \frac{\Delta t}{\rho}\nabla\cdot(K_\mathrm{i}\nabla T_\mathrm{i}) - \Delta t w_\mathrm{ei}(T_\mathrm{e}-T_\mathrm{i}) = c_\mathrm{i}T_\mathrm{i}^{(n-1)}, \tag{6}$$

$$\frac{4}{\rho}c_\mathrm{r}T_\mathrm{r}^4 - \frac{\Delta t}{\rho}\nabla\cdot(K_\mathrm{r}\nabla T_\mathrm{r}) - \Delta t w_\mathrm{er}(T_\mathrm{e}-T_\mathrm{r}) = \frac{4}{\rho}c_\mathrm{r}T_\mathrm{r}^3T_\mathrm{r}^{(n-1)}, \tag{7}$$

where $T_\alpha,\ T_\alpha^{(n-1)},\ \alpha = \mathrm{e},\mathrm{i},\mathrm{r}$ are the temperature functions at time t_n and t_{n-1}, $\Delta t = t_n - t_{n-1}$.

We can use the *Newton method* or *Freezing coefficient method* (FCM) to linearize the nonlinear equations (5)–(7). In general, the *Newton method* seems more desirable than FCM, but *Newton method* requires one order derivative, it will break the conservation of equations and make the discretization more complicated. Solving 2-D 3-T diffusion equations, sometimes FCM works better than the *Newton method*.

Linearizing the nonlinear equations (5)–(7) with FCM, we obtain a linear partial differential equations as follows:

$$-\nabla(d_\mathrm{e}\nabla T_\mathrm{e}) + (d_\mathrm{ei}+d_\mathrm{er}+c_\mathrm{e})T_\mathrm{e} - d_\mathrm{ei}T_\mathrm{i} - d_\mathrm{er}T_\mathrm{r} = f_\mathrm{e}, \tag{8}$$

$$-\nabla(d_\mathrm{i}\nabla T_\mathrm{i}) + (d_\mathrm{ei}+c_\mathrm{i})T_\mathrm{i} - d_\mathrm{ei}T_\mathrm{e} = f_\mathrm{i}, \tag{9}$$

$$-\nabla(d_\mathrm{r}\nabla T_\mathrm{r}) + (d_\mathrm{er}+c_\mathrm{r}')T_\mathrm{r} - d_\mathrm{er}T_\mathrm{e} = f_\mathrm{r}, \tag{10}$$

where

$$\begin{cases} d_\mathrm{e} = \frac{\Delta t}{\rho}\tilde{K}_\mathrm{e}, \\ d_\mathrm{i} = \frac{\Delta t}{\rho}\tilde{K}_\mathrm{i}, \\ d_\mathrm{r} = \frac{\Delta t}{\rho}\tilde{K}_\mathrm{r}, \end{cases} \quad \begin{cases} d_\mathrm{ei} = -\Delta t\tilde{w}_\mathrm{ei}, \\ d_\mathrm{er} = -\Delta t\tilde{w}_\mathrm{er}, \\ c_\mathrm{r}' = \frac{4}{\rho}c_\mathrm{r}\tilde{T}_\mathrm{r}^3, \end{cases} \quad \begin{cases} f_\mathrm{e} = c_\mathrm{e}T_\mathrm{e}^{(n-1)}, \\ f_\mathrm{i} = c_\mathrm{i}T_\mathrm{i}^{(n-1)}, \\ f_\mathrm{r} = c_\mathrm{r}'T_\mathrm{r}^{(n-1)}, \end{cases}$$

$\tilde{T}_\alpha,\ \alpha = \mathrm{e},\mathrm{i},\mathrm{r}$ are the latest solution functions of linear PDEs in the nonlinear iteration. $\tilde{K}_\alpha,\ \tilde{w}_{\mathrm{ei}},\ \tilde{w}_{\mathrm{er}}$ are the conductive coefficients and energy exchange coefficients calculated by $\tilde{T}_\alpha$.

In the recent literature, finite difference and finite volume element method (FVEM) are used to discrete the linear PDEs (8)–(10). The matrix of the discrete system by finite difference or finite volume method is not symmetric although the original PDE system (8)–(10) is symmetric. As a result, many efficient methods for solving large algebraic equations, e.g., the conjugate gradient method (CG), cannot be used here.

For lacking of precondition theory of finite difference and finite volume method, we use the ILU as a preconditioner. The common process of solving 2-D 3T diffusion equations (1)–(3) defined as follows.

Algorithm 3.1.

1: Initial temporal variable Δt, $t = 0,\ t_{\mathrm{end}} = T$ and mesh T^h;
2: Solve the nonlinear PDEs (5)–(7) at time $(t + \Delta t)$ with *FC-FVEM-GMRES*(*ILU*) method base on mesh T^h;
3: Adapt the time step size Δt based on mesh T^h and numerical solution;
4: $t = t + \Delta t$, if $(t < t_{\mathrm{end}})$ goto 2:
5: End.

Here, the adaptive rules for time step size are similar as [5].

3.2. Symmetric finite volume element method (SFVEM)

To overcome the disadvantage of FVEM , we propose a new symmetric finite volume element method (SFVEM), which can preserve the symmetrical discrete system of linear PDEs (8)–(10). [6,7] have developed the similar work on parabolic equation and quadrilateral grids.

Let $T^h = \{\tau\}$ denote a regular and quasi-uniform triangulation of Ω, $\partial^2 T^h = \{P_i,\ i = 1,\ldots,N\}$ be the set of vertices of the triangulation T^h and N be the number of vertices. We can construct a dual mesh B^h based upon T^h, called the box mesh, as follows: for each $\tau \in T^h$, select the barycenter point O, then connect O by straight-line segments to the edge midpoints of τ. These segments decompose each τ into three subregions (see Fig. 2(a)). With each $P_i \in \partial^2 T^h$, we associate the box $b_{\mathrm{i}} \in B^h$ (see Fig. 2(b)), which consists of the union of the subregions which have a P_i as a corner and make up the dual mesh. The elements in the dual mesh are call boxes or control volumes and the dual mesh B^h is the so-called barycenter dual mesh.

According to the discretization procedure of FVEM, we firstly take the integral of linear PDEs (8)–(10) over a box b_i, and get the balance equations at vertex P_i as follows.

$$-\int_{\partial b_{\mathrm{i}}} d_{\mathrm{e}} \frac{\partial T_{\mathrm{e}}}{\partial n}\,\mathrm{d}s + \int_{b_{\mathrm{i}}} (d_{\mathrm{ei}} + d_{\mathrm{er}} + c_{\mathrm{e}}) T_{\mathrm{e}} - \int_{b_{\mathrm{i}}} d_{\mathrm{ei}} T_{\mathrm{i}} - \int_{b_{\mathrm{i}}} d_{\mathrm{er}} T_{\mathrm{r}} = \int_{b_{\mathrm{i}}} f_{\mathrm{e}}, \tag{11}$$

$$-\int_{\partial b_{\mathrm{i}}} d_{\mathrm{i}} \frac{\partial T_{\mathrm{i}}}{\partial n}\,\mathrm{d}s + \int_{b_{\mathrm{i}}} (d_{\mathrm{ei}} + c_{\mathrm{i}}) T_{\mathrm{i}} - \int_{b_{\mathrm{i}}} d_{\mathrm{ei}} T_{\mathrm{e}} = \int_{b_{\mathrm{i}}} f_{\mathrm{i}}, \tag{12}$$

$$-\int_{\partial b_{\mathrm{i}}} d_{\mathrm{r}} \frac{\partial T_{\mathrm{r}}}{\partial n}\,\mathrm{d}s + \int_{b_{\mathrm{i}}} (d_{\mathrm{er}} + c'_{\mathrm{r}}) T_{\mathrm{r}} - \int_{b_{\mathrm{i}}} d_{\mathrm{er}} T_{\mathrm{e}} = \int_{b_{\mathrm{i}}} f_{\mathrm{r}}, \tag{13}$$

where ∂b_{i} denotes the boundary of b_{i}.

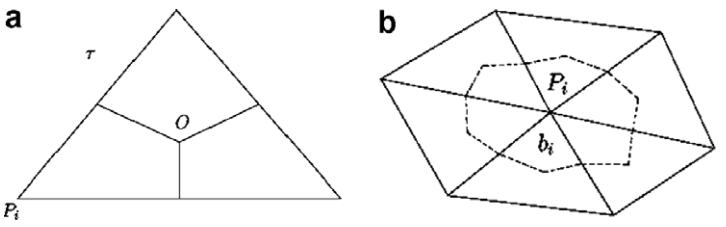

Fig. 2. Barycenter dual mesh: (a) τ and three subcontrol volumes, (b) box b_i and vertex P_i.

For approximating equations (11)–(13), we define three linear finite element spaces and two constant finite element spaces associated with T^h and B^h, respectively, by

$$V_0 = \{v \in L^2 : v|_\tau \in P^0 \ \forall \tau \in T^h\},$$
$$V_B = \{v \in L^2 : v|_{b_\mathrm{i}} \in P^0 \ \forall b_\mathrm{i} \in B^h\},$$
$$V_h = \{v \in C(\overline{\Omega}) : v|_\tau \in P^1 \ \forall \tau \in T^h\},$$
$$V_h^r = \{v \in C(\overline{\Omega}) : v|_\tau \in P^1 \ \forall \tau \in T^h, v|_{\Gamma_1} = g\},$$

where P^k denotes the set of polynomial functions that the order is no more than k, g is a known function. Then, we introduce $I_h : C(\overline{\Omega}) \to V_B$, $I_h^* : C(\overline{\Omega}) \to V_0$, respectively, by

$$I_h v(x) = v(P_i) \quad \forall x \in b_\mathrm{i},\ b_\mathrm{i} \in B^h, \quad I_h^* v(x) = v(O) \quad \forall x \in \tau,\ \tau \in T^h,$$

where O is the barycenter of τ.

Let $T_\mathrm{e} \approx T_\mathrm{e}^h \in V_h$, $T_\mathrm{i} \approx T_\mathrm{i}^h \in V_h$, $T_\mathrm{r} \approx T_\mathrm{r}^h \in V_h^r$, we can get the approximation balance equations as follows.

$$-\int_{\partial b_\mathrm{i}} I_h^*\left(d_\mathrm{e}\frac{\partial T_\mathrm{e}^h}{\partial n}\right)\mathrm{d}s + \int_{b_\mathrm{i}} I_h(d_\mathrm{ei} + d_\mathrm{er} + c_\mathrm{e})I_h T_\mathrm{e}^h - \int_{b_\mathrm{i}} I_h d_\mathrm{ei} I_h T_\mathrm{i}^h - \int_{b_\mathrm{i}} I_h d_\mathrm{er} I_h T_\mathrm{r}^h = \int_{b_\mathrm{i}} I_h f_\mathrm{e}, \tag{14}$$

$$-\int_{\partial b_\mathrm{i}} I_h^*\left(d_\mathrm{i}\frac{\partial T_\mathrm{i}^h}{\partial n}\right)\mathrm{d}s + \int_{b_\mathrm{i}} I_h(d_\mathrm{ei} + c_\mathrm{i})I_h T_\mathrm{i}^h - \int_{b_\mathrm{i}} I_h d_\mathrm{ei} I_h T_\mathrm{e}^h = \int_{b_\mathrm{i}} I_h f_\mathrm{i}, \tag{15}$$

$$-\int_{\partial b_\mathrm{i}} I_h^*\left(d_\mathrm{r}\frac{\partial T_\mathrm{r}^h}{\partial n}\right)\mathrm{d}s + \int_{b_\mathrm{i}} I_h(d_\mathrm{er} + c_\mathrm{r}')I_h T_\mathrm{r}^h - \int_{b_\mathrm{i}} I_h d_\mathrm{er} I_h T_\mathrm{e}^h = \int_{b_\mathrm{i}} I_h f_\mathrm{r}. \tag{16}$$

Next, we will transform the integration over the boundary in Eq. (14) into the integration over the element.

Let τ_{i_j} denote the jth neighboring triangle element of vertex P_i, here $j = 1(1)n_i$ and n_i is the total number of the neighbor element of P_i. In element τ_{i_j}, P_i, P_j^1, P_j^2 are the three corners, M_j^1, M_j^2, M_j^3 denote the three edge midpoints and O_{i_j} is the barycenter (see Fig. 3).

As $I_h^* d_\mathrm{e}(x) = d_\mathrm{e}(O_{i_j})$ is a constant when $x \in \tau_{i_j}$, we have

$$-\int_{\partial b_\mathrm{i}} I_h^*\left(d_\mathrm{e}\frac{\partial T_\mathrm{e}^h}{\partial n}\right)\mathrm{d}s = -\sum_{j=1}^{n_i} I_h^* d_\mathrm{e} \int_{\overset{\frown}{M_j^1 O_{i_j} M_j^2}} \frac{\partial T_\mathrm{e}^h}{\partial n}\,\mathrm{d}s. \tag{17}$$

Since $T_\mathrm{e}^h \in V^h$ is a linear function, $\Delta T_\mathrm{e}^h|_{\tau_{i_j}} = 0$, we find that

$$0 = \int_{b_{i_j}} \Delta T_\mathrm{e}^h\,\mathrm{d}x = \int_{\partial b_{i_j}} \frac{\partial T_\mathrm{e}^h}{\partial n}\,\mathrm{d}s = \int_{\overset{\frown}{M_j^1 O_{i_j} M_j^2}} \frac{\partial T_\mathrm{e}^h}{\partial n}\,\mathrm{d}s + \int_{\overline{M_j^2 P_i}} \frac{\partial T_\mathrm{e}^h}{\partial n}\,\mathrm{d}s + \int_{\overline{P_i M_j^1}} \frac{\partial T_\mathrm{e}^h}{\partial n}\,\mathrm{d}s. \tag{18}$$

Let ϕ_i be the base function of V^h at P_i. Consider

$$\phi_i|_{\overline{P_j^1 P_j^2}} = 0, \quad \frac{\partial T_\mathrm{e}^h}{\partial n}|_{\overline{P_j^l P_i}} \in P^0, \quad \int_{\overline{P_j^l P_i}} \phi_i\,\mathrm{d}s = \frac{1}{2}|\overline{P_j^l P_i}|, \quad l = 1, 2.$$

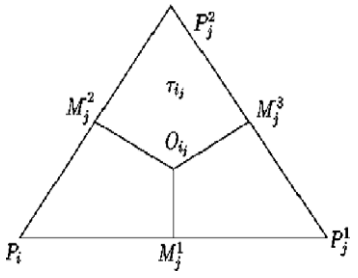

Fig. 3. Triangle element τ_{i_j}.

From (18), we have

$$\begin{aligned}\int_{\widehat{M_j^1 O_{i_j} M_j^2}} \frac{\partial T_e^h}{\partial n}\,\mathrm{d}s &= -\int_{M_j^2 P_i} \frac{\partial T_e^h}{\partial n}\,\mathrm{d}s - \int_{P_i M_j^1} \frac{\partial T_e^h}{\partial n}\,\mathrm{d}s = -\frac{1}{2}\left(\int_{P_j^2 P_i} \frac{\partial T_e^h}{\partial n}\,\mathrm{d}s + \int_{P_i P_j^1} \frac{\partial T_e^h}{\partial n}\,\mathrm{d}s\right)\\ &= -\left(\int_{P_j^1 P_i} \frac{\partial T_e^h}{\partial n}\phi_i\,\mathrm{d}s + \int_{P_j^2 P_i} \frac{\partial T_e^h}{\partial n}\phi_i\,\mathrm{d}s + \int_{P_j^1 P_i^2} \frac{\partial T_e^h}{\partial n}\phi_i\,\mathrm{d}s\right) = -\int_{\partial\tau_{i_j}} \frac{\partial T_e^h}{\partial n}\phi_i\,\mathrm{d}s\\ &= -\int_{\tau_{i_j}} \nabla T_e^h \nabla\phi_i\,\mathrm{d}x.\end{aligned}$$

Using this fact in Eq. (17), It follows that

$$-\int_{\partial b_i} I_h^*\left(d_e \frac{\partial T_e^h}{\partial n}\right)\mathrm{d}s = \sum_{j=1}^{n_i}\int_{\tau_{i_j}} I_h^* d_e \nabla T_e^h \nabla\phi_i\,\mathrm{d}s. \tag{19}$$

By (14) and (19), we obtain

$$\int_{\Omega_i} I_h^* d_e \nabla T_e^h \nabla\phi_i + \int_{b_i} I_h(d_{ei}+d_{er}+c_e) I_h T_e^h - \int_{b_i} I_h d_{ei} I_h T_i^h - \int_{b_i} I_h d_{er} I_h T_r^h = \int_{b_i} I_h f_e, \tag{20}$$

where $\Omega_i = \bigcup_{j=1}^{n_i}\tau_{i_j}$.

Since $\phi_i(P_j) = \delta_{ij}$ and ϕ_i has a local support , we have

$$\int_{\Omega} I_h^* d_e \nabla T_e^h \nabla\phi_i + \int_{\Omega} I_h(d_{ei}+d_{er}+c_e) I_h T_e^h I_h\phi_i - \int_{\Omega} I_h d_{ei} I_h T_i^h I_h\phi_i - \int_{\Omega} I_h d_{er} I_h T_r^h I_h\phi_i = \int_{\Omega} I_h f_e I_h\phi_i. \tag{21}$$

Similarly, Eqs. (15) and (16) have the following equivalent equations:

$$\int_{\Omega} I_h^* d_i \nabla T_i^h \nabla\phi_i\,\mathrm{d}x + \int_{\Omega} I_h(d_{ei}+c_i) I_h T_i^h I_h\phi_i \mathrm{d}x - \int_{\Omega} I_h d_{ei} I_h T_e^h I_h\phi_i\,\mathrm{d}x = \int_{\Omega} I_h f_i I_h\phi_i\,\mathrm{d}x, \tag{22}$$

$$\int_{\Omega} I_h^* d_r \nabla T_r^h \nabla\phi_i\,\mathrm{d}x + \int_{\Omega} I_h(d_{er}+c_r') I_h T_r^h I_h\phi_i \mathrm{d}x - \int_{\Omega} I_h d_{er} I_h T_e^h I_h\phi_i\,\mathrm{d}x = \int_{\Omega} I_h f_r I_h\phi_i\,\mathrm{d}x. \tag{23}$$

Let $\bar{T} = (T_e^h, T_i^h, T_r^h)^t \in V_h \times V_h \times V_h^r$, $\bar{f} = (f_e^h, f_i^h, f_r^h)^t$. Using (21)–(23), we obtain the variational formulation of SFVEM.

$$\tilde{a}(\bar{T}, v) = (\bar{f}, v) \quad \forall v = (v_e^h, v_i^h, v_r^h)^t \in V_h \times V_h \times V_h^{r,0}, \tag{24}$$

where

$$\begin{aligned}(\bar{f}, v) &= (I_h f_e, I_h v_e^h) + (I_h f_i, I_h v_i^h) + (I_h f_r, I_h v_r^r),\\ \tilde{a}(\bar{T}, v) &= \tilde{a}_e(T_e^h, v_e^h) + \tilde{a}_i(T_i^h, v_i^h) + \tilde{a}_r(T_r^h, v_r^h) + (I_h d_{ei} I_h T_e^h, I_h v_e^h) + (I_h d_{er} I_h T_e^h, I_h v_e^h) + (I_h d_{ei} I_h T_i^h, I_h v_i^h)\\ &\quad + (I_h d_{er} I_h T_r^h, I_h v_r^h) - (I_h d_{ei} I_h T_i^h, I_h v_e^h) - (I_h d_{er} I_h T_r^h, I_h v_e^h) - (I_h d_{ei} I_h T_e^h, I_h v_i^h) - (I_h d_{er} I_h T_e^h, I_h v_r^h),\end{aligned}$$

and

$$\begin{aligned}\tilde{a}_e(u, w) &= \int_{\Omega} (I_h^* d_e \nabla u \nabla w + I_h c_e I_h u I_h w)\,\mathrm{d}x,\\ \tilde{a}_i(u, w) &= \int_{\Omega} (I_h^* d_i \nabla u \nabla w + I_h c_i I_h u I_h w)\,\mathrm{d}x,\\ \tilde{a}_r(u, w) &= \int_{\Omega} (I_h^* d_r \nabla u \nabla w + I_h c_r' I_h u I_h w)\,\mathrm{d}x.\end{aligned}$$

Here, $V_h^{r,0}$ denotes the space V_h^r with $g = 0$.

According to the variational equation (24), we get the linear algebraic system of Eqs. (5)–(7) with SFVEM.

$$A^h U^h = F^h, \tag{25}$$

where $U^h,\ F^h \in R^{3N},\ A^h \in R^{3N\times 3N}$.

In the process of approximating balance equations, we defined a special operator I_h^* which is different from FVEM. Then, we get the *Petrov–Galerkin* variational formulation of SFVEM. As the bilinear function $\tilde{a}(\bar{T}, v)$ is symmetric, it follows that matrix A^h is symmetric too. Because we only need to calculate the value of d_α, $\alpha = \mathrm{e}, \mathrm{i}, \mathrm{r}$ at the triangle barycenter to produce the stiff matrix of the element in SFVEM , it is obvious that the discretization of SFVEM is cheaper than FVEM.

3.3. Precondition

Similar to the finite element method and the traditional finite volume element method, FVEM suffers from the ill-condition of its coefficient matrix. In the last decade, some efficient preconditioning techniques have been developed for the FEM [8–10]. Since the test space is different from the trial space in FVEM, it's very difficult to develop the preconditioning of FVEM with the same technique of FEM. Generally, we often use ILU decomposition method to precondition the FVEM, which usually cause trouble when the system become bigger.

In [11], an auxiliary linear element stiffness matrix is chosen to be the preconditioner for higher order finite elements. In this subsection, we develop an algebraic multigrid(AMG) preconditioning for the SFVEM coefficient matrix of 2-D 3-T diffusion equations with the ideal of using linear finite element system to precondition finite volume element system.

For the purpose of this section, we firstly present the FEM variational equations of linear PDEs (8)–(10).

$$a(\bar{T}, v) = (\bar{f}, v), \quad \forall v = (v_{\mathrm{e}}^h, v_{\mathrm{i}}^h, v_{\mathrm{r}}^h)^t \in V_h \times V_h \times V_h^{r,0}, \tag{26}$$

where

$$\begin{aligned}
(\bar{f}, v) &= (f_{\mathrm{e}}, v_{\mathrm{e}}^h) + (f_{\mathrm{i}}, v_{\mathrm{i}}^h) + (f_{\mathrm{r}}, v_{\mathrm{r}}^r),\\
a(\bar{T}, v) &= a_{\mathrm{e}}(T_{\mathrm{e}}^h, v_{\mathrm{e}}^h) + a_{\mathrm{i}}(T_{\mathrm{i}}^h, v_{\mathrm{i}}^h) + a_{\mathrm{r}}(T_{\mathrm{r}}^h, v_{\mathrm{r}}^h) + (d_{\mathrm{ei}}T_{\mathrm{e}}^h, v_{\mathrm{e}}^h) + (d_{\mathrm{er}}T_{\mathrm{e}}^h, v_{\mathrm{e}}^h) + (d_{\mathrm{ei}}T_{\mathrm{i}}^h, v_{\mathrm{i}}^h) + (d_{\mathrm{er}}T_{\mathrm{r}}^h, v_{\mathrm{r}}^h)\\
&\quad - (d_{\mathrm{ei}}T_{\mathrm{i}}^h, v_{\mathrm{e}}^h) - (d_{\mathrm{er}}T_{\mathrm{r}}^h, v_{\mathrm{e}}^h) - (d_{\mathrm{ei}}T_{\mathrm{e}}^h, v_{\mathrm{i}}^h) - (d_{\mathrm{er}}T_{\mathrm{e}}^h, v_{\mathrm{r}}^h).
\end{aligned}$$

According to the properties of coefficient functions and elliptic equation, we have

$$\begin{aligned}
a(\bar{T}, \bar{T}) &\gtrsim \|T_{\mathrm{e}}\|_1^2 + \|T_{\mathrm{i}}\|_1^2 + \|T_{\mathrm{r}}\|_1^2 + (d_{\mathrm{ei}}T_{\mathrm{e}}, T_{\mathrm{e}}) + (d_{\mathrm{er}}T_{\mathrm{e}}, T_{\mathrm{e}}) + (d_{\mathrm{ei}}T_{\mathrm{i}}, T_{\mathrm{i}}) + (d_{\mathrm{er}}T_{\mathrm{r}}, T_{\mathrm{r}})\\
&\quad - 2(d_{\mathrm{ei}}T_{\mathrm{i}}, T_{\mathrm{e}}) - 2(d_{\mathrm{er}}T_{\mathrm{r}}, T_{\mathrm{e}})\\
&\gtrsim \|T_{\mathrm{e}}\|_1^2 + \|T_{\mathrm{i}}\|_1^2 + \|T_{\mathrm{r}}\|_1^2 + (d_{\mathrm{ei}}(T_{\mathrm{e}} - T_{\mathrm{i}}), (T_{\mathrm{e}} - T_{\mathrm{i}})) + (d_{\mathrm{er}}(T_{\mathrm{e}} - T_{\mathrm{r}}), (T_{\mathrm{e}} - T_{\mathrm{r}}))\\
&\gtrsim \|T_{\mathrm{e}}\|_1^2 + \|T_{\mathrm{i}}\|_1^2 + \|T_{\mathrm{r}}\|_1^2 \gtrsim \|T\|_1^2,
\end{aligned}$$

and

$$\begin{aligned}
a(\bar{T}, \bar{T}) &\lesssim \|T_{\mathrm{e}}\|_1^2 + \|T_{\mathrm{i}}\|_1^2 + \|T_{\mathrm{r}}\|_1^2 + \|T_{\mathrm{e}} - T_{\mathrm{i}}\|_0^2 + \|T_{\mathrm{e}} - T_{\mathrm{r}}\|_0^2\\
&\lesssim \|T_{\mathrm{e}}\|_1^2 + \|T_{\mathrm{i}}\|_1^2 + \|T_{\mathrm{r}}\|_1^2 + (\|T_{\mathrm{e}}\|_0 + \|T_{\mathrm{i}}\|_0)^2 + (\|T_{\mathrm{e}}\|_0 + \|T_{\mathrm{r}}\|_0)^2\\
&\lesssim \|T_{\mathrm{e}}\|_1^2 + \|T_{\mathrm{i}}\|_1^2 + \|T_{\mathrm{r}}\|_1^2 + (\|T_{\mathrm{e}}\|_0^2 + \|T_{\mathrm{i}}\|_0^2) + (\|T_{\mathrm{e}}\|_0^2 + \|T_{\mathrm{r}}\|_0^2)\\
&\lesssim \|T_{\mathrm{e}}\|_1^2 + \|T_{\mathrm{i}}\|_1^2 + \|T_{\mathrm{r}}\|_1^2 + (\|T_{\mathrm{e}}\|_1^2 + \|T_{\mathrm{i}}\|_1^2) + (\|T_{\mathrm{e}}\|_1^2 + \|T_{\mathrm{r}}\|_1^2)\\
&\lesssim \|T_{\mathrm{e}}\|_1^2 + \|T_{\mathrm{i}}\|_1^2 + \|T_{\mathrm{r}}\|_1^2 \lesssim \|T\|_1^2,
\end{aligned}$$

which demonstrate that

$$a(\bar{T}, \bar{T}) \eqsim \|\bar{T}\|_1^2, \tag{27}$$

and the function $a(\bar{T}, v)$ is positive definite. In the same way, we can prove

$$\tilde{a}(\bar{T}, \bar{T}) \eqsim \|\bar{T}\|_1^2. \tag{28}$$

The corresponding discretezation linear algebraic system of (26) is

$$A_h U_h = F_h, \tag{29}$$

where $F_h \in R^{3N}$ and $A_h \in R^{3N \times 3N}$. A_h is a SPD matrix obviously.

From (27) and (28), It is easy to prove that the coefficient matrix A^h and A_h satisfy

$$(A^h U, U) \lesssim (A_h U, U) \lesssim (A^h U, U) \quad \forall U \in R^{3N}. \tag{30}$$

It shows that A^h is a SPD matrix, CG method is available for solving the linear system (25). It also demonstrates the spectral equivalence of A_h and A^h. So, we can obtain the following theorem.

Theorem 3.1. *Suppose that the triangulation is quasi-uniform, the condition number of matrix $A_h^{-1}A^h$ satisfy*

$$\kappa(A_h^{-1}A^h) \lesssim 1, \tag{31}$$

where A^h and A_h are the coefficient matrix of equations (8)–(10) with SFVEM and FEM, respectively.

According to Theorem 3.1 it shows that the inverse of FEM coefficient matrix A_h^{-1} is an efficient preconditioner of A^h SFVEM coefficient matrix of linear PDEs (8)–(10). Therefore, preconditioning SFVEM can be realized by preconditioning FEM.

Here, we present an AMG precondition.

Let A_h be a matrix from FEM, $V_k, k = 1(1)J$ be a group of finite element space such that $V_1 \subset V_2 \subset \cdots \subset V_J := S_0^h, N_k = \dim V_k, A_k$ be the FEM coefficient matrix on k-th level space V_k. The essential operations of precondition is that, for any given vector g, calculate the vector $w = Bg$.

Algorithm 3.2 (*AMG*). Let $B_1 = A_1^{-1}$, for given $B_{k-1} : R^{N_{k-1}} \to R^{N_{k-1}}$ and $\forall g_k \in R^{N_k}$, we define $B_k : R^{N_k} \to R^{N_k}$ as follows:

step1: $V^1 = R_k^1 g_k$
step2: $V^2 = V^1 + Q_{k-1}^T B_{k-1} Q_{k-1}(g_k - A_k V^1)$
step3: $B_k g_k = R_k^2 V^2$,

where R_k^l $(l = 1, 2)$ are the pre-smoother and post-smoother of A_k, respectively, Q_{k-1} is restrict operator of $R^{N_k} \to R^{N_{k-1}}$, Q_{k-1}^T is the interpolate operator of $R^{N_{k-1}} \to R^{N_k}$.

According to Algorithm 3.2, we can precondition the FEM coefficient matrix A_h, for most triangulation (e.g. quasi-uniform, shape regular), we have [8,9]

$$\kappa(BA_h) \lesssim 1, \tag{32}$$

where $B := B_J$ is defined by Algorithm 3.2.

Because of the equivalence of A^h and A_h, we have

$$\kappa(BA^h) \lesssim 1.$$

It shows that B is an efficient preconditioner of SFVEM coefficient matrix A^h induced from linear PDEs (8)–(10).

3.4. Mesh adaptation

In recent years, the mesh adaptation technique based on Hessian matrix arises a new view point both in theoretical analysis and adaptive computation [13,14,16,19]. In [17], the relationship between the optimal mesh and Hessian matrix was demonstrated. Here, we briefly describe the relevant theory.

Let Ω be an open set of R^n. Given a function $u \in C^2(\overline{\Omega})$, we define a symmetric positive definite matrix function $H(x)$ and a scaled Hessian matrix as follows:

$$|\xi^t(\nabla^2 u)(x)\xi| \leqslant c_0 \xi^t H(x) \xi, \quad \xi \in R^n, \ x \in \Omega, \tag{33}$$

$$H_p = (\det H)^{-\frac{1}{2p+n}} H, \quad p \geqslant 1, \tag{34}$$

here, c_0 is positive constant. The matrix H is called a majorizing Hessian matrix of u and H_p defines a Riemanian metric on Ω. In [17], the relationship between the optimal mesh and Hessian matrix was established.

When we use the theory to adapt the mesh, it is very difficult to get the Hessian matrix ($(\nabla^2 u)(x)$) from the numerical solution, especially, using the linear finite element of which the piecewise second order derivative is zero. Here, we introduce the following method to calculate the Hessian matrix with linear finite element.

First, we calculate the numerical gradient of each node with some recovery technique from numerical solution. Replacing the numerical solution u_h with the element ($\frac{\partial u_h}{\partial x}$ or $\frac{\partial u_h}{\partial y}$) of ∇u_h in turn, we can obtain $\frac{\partial^2 u_h}{\partial x^2}$, $\frac{\partial^2 u_h}{\partial y^2}$, $\frac{\partial^2 u_h}{\partial x \partial y}$ with the same recovery technique. Then, we have the Hessian matrix $(\nabla^2 u)(x)$ for

$$(\nabla^2 u)(x) \approx \begin{pmatrix} \frac{\partial^2 u_h}{\partial x^2} & \frac{\partial^2 u_h}{\partial x \partial y} \\ \frac{\partial^2 u_h}{\partial y \partial x} & \frac{\partial^2 u_h}{\partial y^2} \end{pmatrix}. \tag{35}$$

In [12,15,18], some recovery methods have been developed to approximate the gradient. Here, we use a simple average method as follows:

$$(\nabla u_h)(\mathbf{x}_i) = \frac{\sum_{\tau \in \Omega_i} |\tau| (\nabla u_h)_\tau}{\sum_{\tau \in \Omega_i} |\tau|}. \tag{36}$$

According to the theory of mesh adaptation based on Hessian, we design the following algorithm for Eqs. (1)–(3).

Algorithm 3.3. Let T_0^h be the initial triangulation and T_k^h $(k = 0, 1, 2 \ldots)$ be the kth adaptive grid, then find the adaptive grid T_{k+1}^h:

step1: Restrict the numerical solution of grid T_k^h to grid T_0^h;
step2: Calculate $H_{p(x)}$ on each node of T_0^h;
step3: Calculate $\bar{d}$ the average length of edge in T_0^h under the new metric;
step4: Mark the edge to refine, of which the length is longer than $\lambda \bar{d}$ under the new metric. Here, λ is a parameter used to control the scale of adaptive grid;
step5: Create adaptive grid T_{k+1}^h by refining the marked edge of T_0^h, the numerical solution of T_{k+1}^h is interpolated from T_0^h.

Let $u_\alpha^0, u_\alpha^{k+1}$, $\alpha = \mathrm{e}, \mathrm{i}, \mathrm{r}$ be numerical solutions on grid T_0^h and T_{k+1}^h, $P_i \in \partial^2 T_{k+1}^h$ be a node of grid T_{k+1}^h. In Algorithm 3.3, we use the following method to implement the interpolation from T_0^h to T_{k+1}^h.

(1) $P_i \in (\partial^2 T_{k+1}^h \cap \partial^2 T_0^h)$, then
$$u_\alpha^{k+1}(P_i) = u_\alpha^0(P_i). \tag{37}$$
(2) If $P_i \notin \partial^2 T_0^h$, the father points $P_{i1}, P_{i2} \in \partial^2 T_0^h$ of P_i can be found. According to (4), the density of energy be follows

$$\varepsilon_{\mathrm{r}}^0(P_{i1}) = \frac{c_{\mathrm{r}}}{\rho}(u_{\mathrm{r}}^0(P_{i1}))^4, \quad \varepsilon_{\mathrm{r}}^0(P_{i2}) = \frac{c_{\mathrm{r}}}{\rho}(u_{\mathrm{r}}^0(P_{i2}))^4, \quad \varepsilon_{\mathrm{r}}^{k+1}(P_i) = \frac{\varepsilon_{\mathrm{r}}^0(P_{i1}) + \varepsilon_{\mathrm{r}}^0(P_{i2})}{2}.$$

Then, we have

$$\begin{cases} u_{\mathrm{e}}^{k+1}(P_i) = \frac{u_{\mathrm{e}}^0(P_{i1}) + u_{\mathrm{e}}^0(P_{i2})}{2}, \\ u_{\mathrm{i}}^{k+1}(P_i) = \frac{u_{\mathrm{i}}^0(P_{i1}) + u_{\mathrm{i}}^0(P_{i2})}{2}, \\ u_{\mathrm{r}}^{k+1}(P_i) = \sqrt[4]{\rho \varepsilon_{\mathrm{r}}^{k+1}(P_i)/c_{\mathrm{r}}}. \end{cases} \tag{38}$$

3.5. Two-grid method

In [20–22], some two-grid methods were proposed to solve the nonsymmetric, indefinite and nonlinear problem, which solve the original problem on a coarser grid and solve a simplified problem on a finer grid.

Because of the efficiency, two-grid method has engaged scientist's attention in the recent years. Some people have used this method to solve nonlinear parabolic equations, nonlinear schrodinger equations, reaction–diffusion equations, elasticity problem, two-dimension incompressibility Navier–Stokes equation, stationary MHD equation and integral equation [23–26], etc. Here, we will design a two-grid algorithm to solve 2-D 3-T diffusion equations.

Let T^H, T^h be two quasi-uniform triangulation, H, h be the mesh size of them, respectively, $H \gg h$. Assuming that $\tilde{T}_e^h$, $\tilde{T}_i^h$, $\tilde{T}_r^h$ are the approximate solution functions of nonlinear equations (5)–(7) on the fine grid T^h, we derive the following independent linear PDEs from the coupled linear partial differential equations (8)–(10).

$$-\nabla(d_e\nabla T_e) + (d_{ei} + d_{er} + c_e)T_e = f_e, \tag{39}$$

$$-\nabla(d_i\nabla T_i) + (d_{ei} + c_i)T_i = f_i, \tag{40}$$

$$-\nabla(d_r\nabla T_r) + (d_{er} + c_r')T_r = f_r, \tag{41}$$

where d_e, d_i, d_r, d_{ei}, d_{er}, and c_r' are same as Eqs. (8)–(10),

$$f_e = c_e T_e^{(n-1)} + d_{ei}\tilde{T}_i^h + d_{er}\tilde{T}_r^h,$$

$$f_i = c_i T_i^{(n-1)} + d_{ei}\tilde{T}_e^h,$$

$$f_r = c_r' T_r^{(n-1)} + d_{er}\tilde{T}_e^h.$$

$\tilde{K}_\alpha$, $\alpha = e, i, r$, $\tilde{w}_{ei}$, $\tilde{w}_{er}$ are the conduction coefficients and energy exchange coefficients calculated by $\tilde{T}_\alpha^h$. Then, we can proposed the two-grid algorithm of nonlinear PDEs (5)–(7).

Algorithm 3.4 (*Two-grid method*).

step1: Find the numerical solution functions T_e^H, T_i^H, T_r^H of the nonlinear PDEs (5)–(7) on the coarse grid T^h;
step2: Get the approximate solution functions $\tilde{T}_e^h$, $\tilde{T}_i^h$, $\tilde{T}_r^h$ of fine grid T^h with interpolation of solution functions on coarse grid T^h;
step3: Find the numerical solutions $\bar{T}_e^h$, $\bar{T}_i^h$, $\bar{T}_r^h$ of electron linear PDE (39), iron linear PDE (40) and photon linear PDE (41) on the fine grid T^h.
step4: Solve the coupled linear PDEs (8)–(10) on the fine grid T^h, the coefficients of which are calculated by $\bar{T}_e^h$, $\bar{T}_i^h$, $\bar{T}_r^h$.

Though we solve the couple linear PDEs (8)–(10) on the fine grid T^h in *Algorithm 3.3* to control the energy conservation error of system, our two-grid algorithm is still related to so-called *mesh independence principle* (MIP) for solving nonlinear differential equations (5)–(7).

If we use finite element method to solve the linear and nonlinear equations in Algorithm 3.3, it is very easy to prove that the convergence error with $\|\cdot\|_1$ is $O(h + H^2)$ by the FEM theory. Considering the spectra equivalence of FEM and SFVEM, we apply the SFVEM to solve the PDE in Algorithm 3.4.

3.6. Integrated algorithm

By integrating all new methods mentioned above, we propose the following algorithm to solve 2-D 3-T diffusion equations (1)–(3).

Algorithm 3.5.

step1: Initial temporal variable Δt, $t = 0, t_{end} = T$ and mesh $T^H = T^h$;
step2: Update grid T^h with algorithm 3.3;
step3: Solve nonlinear PDEs (5)–(7) with two-grid algorithm 3.4;
step4: $t = t + \Delta t$
step5: Adapt the time step size Δt based on the coarse grid T^H and numerical solution;
step6: if $(t < t_{end})$ goto step2
step7: End.

4. Numerical examples

In this section, we apply our algorithms to 2-D 3-T diffusion equations (1)–(3). We take the typical model of laser-driven implosion of inertial confinement fusion in paper [5] as a benchmark. Deuterium gas (DT), glass (SiO_2) and plastic foam (CH) are filled into subdomain Ω_1, Ω_2, Ω_3 respectively.

Fig. 4 depicts the three triangulations of the computational domain. *Mesh0* consists of 2356 triangle elements. *Mesh1* is a consistent refinement of *Mesh0* and *Mesh2* is a consistent refinement of *Mesh1*. The element numbers of *Mesh1* , *Mesh2* are 9424, 37,696, respectively.

We use energy conservation error to evaluate the efficiency of different algorithms, which is defined as follows:

$$Err = \left| \frac{E^n_{\text{enter}} - (E^n_{\text{own}} - E^0_{\text{own}})}{E^n_{\text{enter}}} \right|,$$

where E^n_{enter} denotes the total radiation energy importing from free boundary, E^n_{own} the system energy at time t_n, E^0_{own} the initial energy of the system.

4.1. Efficiency of SFVEM

In Section 3, we have indicated that the discritization of SFVEM is cheaper than FVEM and FEM by theoretical analysis. Here, we calculate the 2-D 3-T diffusion equations on *mesh0* with FEM, FVEM and SFVEM. Table 1 shows the average discrete speed of three different schemes. It demonstrates that SFVEM discretization is faster than the classical FVEM and FEM. Table 2 shows the energy conservation error at five different physical time of FVEM and SFVEM. These results verify the efficiency of SFVEM.

4.2. Efficiency of preconditioning

In order to check efficiency of preconditioning for SFVEM, we calculate 2-D 3-T diffusion equations on *mesh0* with Algorithm 3.1, which discretize the equations with SFVEM. *GMRES* and *CG* iteration methods are used to solve linear system. *BILU* and *AMG* are applied to precondition the SFVEM, respectively.

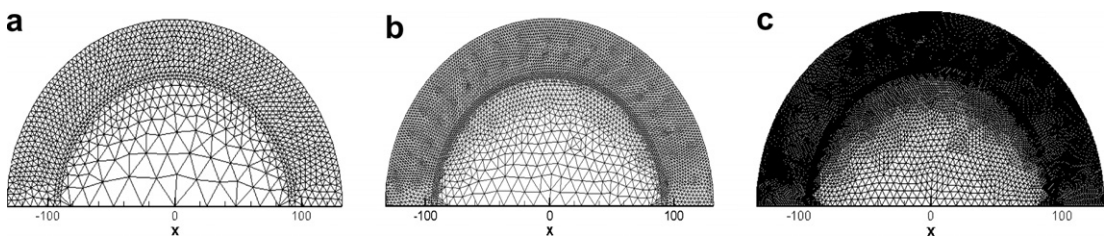

Fig. 4. The triangulations in the numerical example: (a) *mesh0*, (b) *mesh1*, (c) *mesh2*.

Table 1
Average time for one discretization (s)

FEM	FVEM	SFVEM
6.85×10^{-2}	4.42×10^{-2}	3.63×10^{-2}

Table 2
Energy conservation error (%) of different method

Physical time	1.0	5.0	10.0	20.0	100.0
FVEM	9.37	6.71	5.85	5.32	4.40
SFVEM	8.69	6.50	5.68	5.17	4.32

From Table 3, we find that AMG is an effective preconditioner of SFVEM, but BILU has very little effect on the simulation. We will use CG(AMG) method to solve the linear system in the examples that follow.

4.3. Efficiency of adaptive

Here, we finish the simulation of 2-D 3-T diffusion equations with FC-SFVEM-CG(AMG) method and adaptive Algorithm 3.3. We name the example "*Adaptive(Hess)*".

Fig. 5 shows the contours of photon temperatures at time of 3.08, 11.85 in the numerical example. It indicates that the front of radiation evolution of photon energy is located at the subdomain of CH, SiO_2 and DT, respectively, and the approximate photon temperature is originally symmetrical. Similarly, the approximate electron and iron temperatures are all originally symmetrical. The phenomena certify the original symmetry of Eqs. (1)–(3), which can be found in all examples.

For comparison, we finish the same simulation with an ordinary adaptive method based on gradient and call the example *Adaptive(grad)*, and fill the energy conservation error of "Mesh0" and "Mesh1" into Table 3, which calculate the equations on *Mesh0* and *Mesh1* with FC-SFVEM-CG(AMG) method.

From Table 4, we can find that the energy conservation of *Adaptive(Hess)* is much better then Mesh1, of which the number of elements is no more than 3600. It is obvious that *Adaptive(Hess)* is more efficient than *Adaptive(Grad)*.

Table 3
Comparison of precondition (BILU and AMG)

Item	BILU_GMRES	BILU_CG	AMG_GMRES	AMG_CG
$t = 1.0$	8.70	8.70	8.70	8.70
$t = 5.0$	6.50	6.50	6.50	6.50
$t = 10.0$	**	5.69	5.69	5.69
$t = 20.0$	**	5.17	5.17	5.16
$t = 100.0$	**	**	4.32	4.32
Time iteration	**	**	5832	1961
CPU time (s)	**	**	19,514.03	6,813.77

"**" Means that the time iteration is more than 10^6.

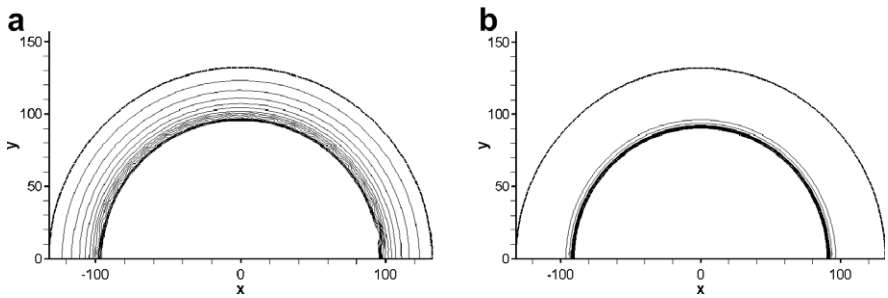

Fig. 5. Contours of photon temperature: (a) $t = 3.08$, (b) $t = 11.85$.

Table 4
Energy conservation error (%)

Physical time	Mesh0	Mesh1	*Adaptive(Hess)*	*Adaptive(Grad)*
1.0	8.69	4.16	3.19	4.87
5.0	6.50	3.14	2.52	3.96
10.0	5.68	2.74	1.72	3.38
20.0	5.16	2.48	1.56	3.41
100.0	4.32	2.08	1.59	3.16

Table 5
Comparison of energy conservation error (%) and CPU consuming

Item	Mesh0	Mesh1	*two-grid(Mesh1)*	*two-grid(Mesh2)*
$t = 1.0$	8.69	4.16	4.18	2.07
$t = 5.0$	6.50	3.14	3.16	1.56
$t = 10.0$	5.68	2.74	2.77	1.37
$t = 20.0$	4.92	2.37	2.42	1.20
$t = 100.0$	4.32	2.08	2.11	1.05
Time iteration	1961	2925	1961	1961
CPU time	1.89 h	16.49 h	2.74 h	8.44 h

Table 6
Comparison of energy conservation error (%) and consuming of CPU

Item	Mesh0	Mesh1	*Adaptive(Hess)*	*two-grid* (*Mesh1*)	*Integrated*
$t = 1.0$	8.69	4.16	3.19	4.18	3.16
$t = 5.0$	6.50	3.14	2.52	3.16	2.43
$t = 10.0$	5.68	2.74	1.72	2.77	1.81
$t = 20.0$	5.16	2.48	1.56	2.42	1.53
$t = 100.0$	4.32	2.08	1.59	2.11	1.60
Time iteration	1961	2925	3569	1961	1961
CPU time	1.89 h	16.49 h	3.43 h	2.74 h	0.99 h

4.4. Efficiency of two-grid method

In order to check the efficiency of two-grid method, we accomplish two numerical examples "*two-grid (Mesh1)*" and "*two-grid(Mesh2)*" with Algorithm 3.4. The fine mesh of *two-grid(Mesh1)* is *Mesh1*. *Mesh2* is used as the fine mesh of *two-grid(Mesh2)*. The coarse grid of two examples are both *Mesh0*.

From Table 5, we get the following observations.

(1) The energy conservation error of *two-grid(Mesh1)* has a little difference with Mesh1. It demonstrates that the two-grid method has no effect on improving the conservation error.
(2) The CPU time of *two-grid(Mesh1)* is only 17% of Mesh1, which have the same fine grid. Two-grid method is effective on reducing the computation time.
(3) The energy conservation error of *two-grid(Mesh2)* is 50% of *two-grid(Mesh1)*. The reducing scale of energy conservation error is still consistent with the order of the whole numerical method.

4.5. Efficiency of integrated algorithm

Algorithm 3.5 integrate the all new method developed in this paper. We use the algorithm to finish the numerical example "*Integrated*". For comparison, we fill the energy conservation error and CPU consuming of example Mesh0, Mesh1, *Adaptive(Hess) two-grid(Mesh1)* and Example *Integrated* into Table 6.

From the result of Table 6, we can find that the energy conservation error of *Integrated* is similar as *Adaptive(Hess)*, and the CPU time is only 29% of the later. It demonstrates that Algorithm 3.5 is the most efficient method for 2-D 3-T diffusion equations.

5. Conclusions

In this paper, we proposed a symmetric finite volume method, an AMG preconditioner, a mesh adaptive algorithm and a two-grid algorithm. With the new methods, we improved the common process of solving 2-D 3-T diffusion equations, and obtain a new algorithm that integrated all the methods developed in the paper. Numerical examples demonstrate that the new algorithms are effective. Considering the relation between the 2-D 3-T diffusion equations and radiation fluid dynamics equations, we expect that our new methods will provide efficient numerical approaches in the simulation of ICF.

Acknowledgments

The authors wish to thank Prof. Jinchao XU, Prof. Longjun SHEN and Dr. Long CHEN for their valuable suggestions. This work was subsidized by the NSFC for Distinguished Young Scholars (10625106), National Basic Research Program of China under the Grants 2005CB321701, 2005CB321702 and NSAF (10376031).

References

[1] Karnig O. Mikaelian, LASNEX simulations of the classical and laser-driven Rayleigh–Taylor instability, Phys. Rev. (1990) 4944–4951.
[2] J.A. Harte, W.E. Alley, D.S. Bailey, J.L. Eddleman, G.B. Zimmerman, Lasnex-a 2-d physics code for modeling icf. UCRL-LR-105821-96-4, 1996, pp. 150–164.
[3] Shangwu Fu, Shuke Huang, Yunsheng Li, Numerical simulation of indirectly driven high-convergence implosions, Chinese J. Comput. Phys. 16 (2) (1999) 162–166.
[4] Steven H. Langer, Howard A. Scott, Michael M. Marinak, Otto L. Landen, Comparisons of line emission from 2- and 3-dimension simulations of icf capsules to experiments, J. Quant. Spectrosc. Radiat. Transfer (2003) 275–286.
[5] Zeyao Mo, Longjun Shen, Wittum Gabriel, Parallel adaptive multigrid algorithm for 2-d 3-t diffusion equations, Int. J. Comput. Math. 81 (3) (2004) 361–374.
[6] X. Ma, S. Shu, A. Zhou, Symmetric finite volume discretization for parabolic problems, Comput. Methods Appl. Mech. Eng. 192 (2003) 4467–4485.
[7] Shi Shu, Haiyuan Yu, Yunqing Huang, Cunyun Nie, A symmetric finite volume element scheme on quadrilateral grids and superconvergence, J. Numer. Anal. Modeling 1 (2004) 1–18.
[8] J.H. Bramble, J.E. Pasciak, J. Xu, Parallel multilevel preconditioners, Math. Comp. 55 (1990) 1–22.
[9] Jinchao Xu, Iterative methods by space decomposition and subspace correction, SIAM Rev. 34 (4) (1992) 581–613.
[10] Jinchao Xu, C. Cai, A preconditioned GMRES method for nonsysmmetric or indefinite problems, Math. Comp. 59 (1992) 311–319.
[11] Yunqing Huang, Shi Shu, Xijun Yu, Preconditioning higher order finite element system by algebraic multigrid method of linear elements, J. Comput. Math. 24 (2006) 657–664.
[12] O.C. Zienkiewicz, J.Z. Zhu, The superconvergence patch recovery and a posteriori error estimates. part 1: The recovery techniques [J], Int. J. Numer. Methods Eng. 33 (1992) 1331–1364.
[13] Ruo Li, Tao Tang, Pingwen Zhang, Moving mesh methods in multiple dimensions based on harmonic maps, J. Comput. Phys. 170 (2) (2001) 562–588.
[14] Ruo Li, Tao Tang, Pingwen Zhang, A moving mesh finite element algorithm for singular problems in two and three space dimensions, J. Comput. Phys. 177 (2) (2002) 365–393.
[15] C. Carstensen, S. Bartels, Each averaging technique yields reliable a posteriori error control in FEM on unstructured grids. I. Low order conforming, nonconforming, and mixed FEM [J], Math. Comput. 71 (239) (2002) 945–969.
[16] Weizhang Huang, Weiwei Sun, Variational mesh adaptation II: error estimates and monitor functions, J. Comput. Phys. 184 (2) (2003) 619–648.
[17] Long Chen, Pengtao Sun, Jinchao Xu, Optimal anisotropic simplicial meshes for minimizing interpolation errors in L^p-norm [J], Math. Comp. 76 (257) (2007) 179–204.
[18] R.E. Bank, J. Xu, Asymptotically exact a posteriori error estimators, part II: General unstructured grids [J], SIAM J. Numer. Anal. 41 (6) (2003) 2313–2332.
[19] Weizhang Huang, Mathematical principles of anisotropic mesh adaptation, Commun. Comput. Phys. 1 (2006) 276–310.
[20] Jinchao Xu, A New class of iterative methods for nonselfadjoint or indefinite problems, SIAM J. Numer. Anal. 29 (1992) 303–319.
[21] Y. Huang, Y. Chen, A multilevel iterate correction method for solving nonlinear singular problems, Natural Science J. Xiangtan Univ. 16 (1994) 23–26.
[22] Jinchao Xu, Two-grid discretization techniques for linear and nonlinear PDE, SIAM J. Numer. Anal. 5 (1996) 1759–1777.
[23] S.N. Chandler-Wilde, M. Rahman, C.R. Ross, A fast two-grid and finite section method for a class of integral equations on the real line with application to an acoustic scattering problem in the half-plane, Numer. Math. 93 (2002) 151.
[24] Y. Huang, Z.C. Shi, T. Tang, W. Xue, A multilevel successive iteration method for nonlinear elliptic problems, Math. Comp. 57 (2004) 525–539.
[25] Chun-feng Ren, Yi-chen Ma, A two-grid method for the steady penalized Navier–Stokes equations, J. Comput. Math. 22 (2004) 101–112.
[26] L. Ignaty, E. Zuazua, A two-grid approximation scheme for nonlinear schrodinger equations: dispersive properties and convergence, C. R. Math. Acad. Sci. Paris 341 (6) (2005) 381–386.