PARALLELIZATION METHODS FOR SOLVING THREE-TEMPERATURE RADIATION-HYDRODYNAMIC PROBLEMS

GUANGWEI YUAN AND YANZHONG YAO

Laboratory of Computational Physics
Institute of Applied Physics and Computational Mathematics
P. O. Box 8009, Beijing 100088, China

Dedicated to Professor Lishang Jiang on the occasion of his 80th birthday

Abstract. An efficient parallelization method for numerically solving Lagrangian radiation hydrodynamic problems with three-temperature modeling on structural quadrilateral grids is presented. The three-temperature heat conduction equations are discretized by implicit scheme, and their computational cost are very expensive. Thus a parallel iterative method for three-temperature system of equations is constructed, which is based on domain decomposition for physical space, and combined with fixed point (Picard) nonlinear iteration to solve sub-domain problems. It can avoid global communication and can be naturally implemented on massive parallel computers. The space discretization of heat conduction equations uses the well-known local support operator method (LSOM). Numerical experiments show that the parallel iterative method preserves the same accuracy as the fully implicit scheme, and has high parallel efficiency and good stability, so it provides an effective solution procedure for numerical simulation of the radiation hydrodynamic problems on parallel computers.

1. Introduction. The radiation-hydrodynamic flow problem is a classic multi-physical problem coupling with radiation and hydrodynamics, which occurs in many applications such as inertial confinement fusion (ICF), strong explosion and astrophysical systems. When the electron, ion and radiation field are in local thermo-dynamic equilibrium respectively, each having its own temperature, the so-called three-temperature radiation-hydrodynamic model is formed. In some physical processes, radiation conduction and its interaction with matter (electron) via energy exchange between them have a substantial effect on both the state and the motion of materials. Note that the particle picture of the radiation is photon, we will not distinguish the terminologies “radiation” and “photon” in the following.

Radiative processes happen on time scales that differ by many orders of magnitude from those characterizing hydrodynamic flow. This leads to significant computational challenges in the efficient numerical simulation of radiation hydrodynamic problems. The usual solution procedure for radiation hydrodynamic problems is the
splitting method, which is based on the splitting of two distinct physical processes (radiation and hydrodynamic flow). First the Lagrangian hydrodynamic equations are solved by an explicit time discrete scheme and staggered or cell-centered scheme for space discretization. We refer to \[16\] and \[11\] for staggered scheme and \[7\] for cell-centered scheme. Here the Lagrangian grid is moving with fluid, and distorted cells occur with flow becoming distorted. Then on the Lagrangian grid the three temperature equations for photon, electron and ion conduction are solved by fully implicit time discrete scheme and finite volume schemes, e.g., LSOM \[8\] and the diamond scheme \[21\], for space discretization.

The fully implicit scheme is used in solving three-temperature heat conduction equations, including radiation conduction or diffusion, due to two considerations. The first is based on physical consideration, that is, in the physical processes we concerned, the time scales of a photon flight time over a characteristic structural length \(l\) or over a photon mean free path \(\lambda_p\), i.e., \(l/c\) or \(\lambda_p/c\), are very small, where \(c\) is the photon velocity in vacuum. Moreover the processes of energy exchange among photon, electron and ion are fast and even faster than the heat conduction in some cases, so that the energy exchange processes give rise to strongly stiff terms in the system of radiation hydrodynamic equations. The second is from numerical stability consideration, i.e., the implicit scheme is stable unconditionally, while the explicit is not. The well-known CFL condition for the explicit scheme of hydrodynamic flow is \(c_s \frac{\Delta t}{\Delta x} \leq 1\), where \(c_s\) is a strictly positive number related with the sound speed. A similar condition for the explicit scheme of heat conduction is \(\kappa \frac{\Delta t}{\Delta x}^2 \leq \frac{1}{2}\), where \(\kappa\) is conduction coefficient, so that the time-step will be very small for small \(\Delta x\) and \(1/\kappa\). Note that for multi-material problems the mass density for different materials may differ in several orders of magnitude, and \(\kappa\) may depend on certain powers of density. Therefore the explicit scheme for heat conduction cannot be used in practical simulations.

From the implicit discretization of three-temperature heat conduction equations at each time-step, a (global) nonlinear algebraic system is formed. The computational cost of solving the system is very expensive, and we have to use massive parallel computer in order to get numerical solutions within an acceptable time. However the implicit scheme cannot be implemented directly on parallel computer since there is strong data coupling, moreover a large amount of global communication is required in solving iteratively the nonlinear algebraic system. There are two kinds of approaches for the existing parallelization methods. One is based on algebraic parallelization without any modification to the implicit scheme, but it cannot avoid global communication, see \[13\]. Therefore we consider another approach of the parallelization, i.e., the implicit scheme is modified at the interface among neighboring sub-domains by domain decomposition, and the resulting scheme is called as parallel scheme. It is necessary to construct a parallel scheme of avoiding global communication for solving Lagrangian radiation hydrodynamic problems on massive parallel computer.

Each of the three-temperature equations is a parabolic equation. Almost all of the work on parallel schemes are devoted to parabolic problems only on one-dimensional or two-dimensional rectangular grids (see \[18\], \[19\], \[20\]). Usually a parallel scheme is a non-overlapping and non-iterative domain decomposition method with a suitable treatment on inner boundary conditions for sub-domain problems. In \[1\] a finite difference domain decomposition algorithm is constructed, in which an explicit scheme on coarse mesh is used at interface, then it can take larger time step.
than that restricted by the classical explicit scheme. In [2] a unconditional stable domain decomposition method is presented with lower accuracy $O(\sqrt{\Delta t} + h)$, where $\Delta t$ and $h$ are the time and space steps respectively. This method is improved in [5], but the space accuracy keeps first-order. In [3] an anisotropic heat conduction problem in magnetized plasmas in cylindrical coordinate is solved by using a parallel scheme constructed in [22], where the values at the interface of sub-domains are predicted by the Du-Fort Frankel scheme. In [25] a stabilized explicit-implicit domain decomposition method is proposed, which is also unconditional stable. In [6], [12], [14] and [24] some unconditionally stable parallel difference schemes with second order accuracy are presented, but they are mainly adapted to the rectangle grids. A conservative domain decomposition method for nonlinear diffusion problems on quadrilateral grids is proposed in [21], in which the diamond scheme (one of nine-point scheme on quadrilateral grids) with Dirichlet and Neumann conditions at inner boundary is adopted for solving sub-domain problems. In [9] a parallel Newton-Krylov-Schwarz (NKS)-based non-linearly implicit algorithm is presented for the numerical solution of the unsteady non-linear multimaterial radiation diffusion problem in two-dimensional space, moreover the numerical results show that the parallel NKS is CPU-time-scalable and non-linearly scalable, but not linearly scalable in terms of iteration numbers. A comprehensive monograph about domain decomposition method is [15]. Recently a Schwarz iterative algorithm with Robin inner boundary conditions is proposed in [4] for solving the neutron diffusion equation, which is linear and discretized with Raviart-Thomas-Nédélec finite elements. However, in all of the above researches the computational grids are fixed (Eulerian) mesh. As we know, there is no study devoted to the parallel scheme for solving Lagrangian radiation hydrodynamic problems.

Our focus is to construct a parallel iterative method for solving the three temperature equations accurately and efficiently. The contributions of this paper include: (i) The validity and applicability of nonlinear diffusion solver are verified on Lagrangian grid, which might become more and more distorted as flow deteriorates; moreover in this solver there is a distinguished feature: the implementation of LSOM is nested in the Picard nonlinear iteration and combined with the domain decomposition method; (ii) The validity and applicability of parallel iterative method are verified for solving Lagrangian radiation hydrodynamic problems, in particular, a complete simulation of whole physical process of multi-material Lagrangian radiation hydrodynamic problems in cylindrical coordinate is performed efficiently; (iii) Our parallelization of LSOM guarantees that it becomes an applicable method in solving radiation-hydrodynamic problems, since the cost of its sequential computation is too huge to be used in practice; we note that usually a cell-centered diffusion scheme (e.g., diamond scheme) has only cell-centered unknown as primary unknown, but LSOM has both cell-centered and cell-edge unknown as primary unknown, so the cost of LSOM increases at least 3 times compared with cell-centered scheme.

This paper is organized as follows. In section 2 the discretization of three-temperature radiation-hydrodynamic equations are introduced. In section 3 the LSOM method for nonlinear diffusion equation is presented, and in section 4 the Picard iteration method for solving the resulting nonlinear system is described. In section 5 a parallel iteration method for solving the three-temperature radiation-hydrodynamic equations is constructed. The numerical experiments are given in section 6 to demonstrate the accuracy, stability and parallel efficiency of the parallelization method. The conclusion is summarized in section 7.
2. Discretization of three-temperature radiation-hydrodynamic equations.

We consider the following system of radiation-hydrodynamic equations with three-temperature in two-dimensional cylindrical coordinate \((x, r)\):

\[
\frac{d\rho}{dt} + \rho \cdot \nabla \bar{U} = 0, \tag{1}
\]

\[
\frac{d\bar{U}}{dt} + \frac{1}{\rho} \nabla P = 0, \tag{2}
\]

\[
C_{\nu a} \frac{dT_{\alpha}}{dt} + \left( P_{\alpha} + q\chi_{\alpha} + \frac{\partial \varepsilon_{\alpha}}{\partial V} \right) \frac{dV}{dt} + \frac{1}{\rho} \nabla \cdot \bar{F}_{\alpha} = S_{\alpha}, \tag{3}
\]

where \(\frac{d}{dt}\) denotes the material or Lagrangian time derivative, and \(\alpha = e, i, p\) represent electron, ion, and photon respectively. Here \(\rho = \rho(x, r, t)\) is the mass density, \(V = \frac{1}{\rho}\) is the specific volume, \(\bar{U} = \bar{U}(x, r, t) = (u(x, r, t), v(x, r, t))\) the flow velocity, \(P = P_e + P_i + P_p + q\) the total pressure, \(q\) viscosity, and \(\chi_{\alpha} = 1\) for \(\alpha = i, e\), and \(\chi_{\alpha} = 0\) for \(\alpha = p\). \(C_{\nu a}, \varepsilon_{\alpha}, P_{\alpha}, T_{\alpha}\) are the specific heat, internal energy, pressure and temperature of electron, ion and photon respectively. \(\bar{F}_{\alpha} = -\kappa_{\alpha} \nabla T_{\alpha}\) and \(\kappa_{\alpha}\) are heat flux and conductivity coefficients of electron, ion and photon respectively. 

\(S_{\alpha}\) are energy exchange terms plus source terms given as follows:

\[S_e = w_{ei}(T_i - T_e) + w_{ep}(T_p - T_e) + W_e, \quad S_i = w_{ei}(T_e - T_i) + W_i, \quad S_p = w_{ep}(T_e - T_p),\]

where \(w_{ei}\) (or \(w_{ep}\)) is the exchange coefficient of electron-ion (or electron-photon) energy, and \(W_e\) or \(W_i\) is a given source. In general \(w_{ei}, w_{ep}, C_{\nu a}\) and \(\kappa_{\alpha}\) are nonlinear functions of \(\rho\) and \(T_{\alpha}\).

The set of equations (1)-(3) describes two kinds of physical processes coupled tightly: one is nonlinear interaction for hydrodynamics, and the other is heat conduction for electron, ion and photon due to temperature gradient. This system is closed if the equations of state \(P_{\alpha} = P_{\alpha}(\rho, T_{\alpha}), \varepsilon_{\alpha} = \varepsilon_{\alpha}(\rho, T_{\alpha})\) are given. The solution procedure is formulated by a physics-based splitting approach. On each time level, the first step is that the hyperbolic part of hydrodynamics is solved by explicit scheme, which can implemented on parallel computers easily. Then the second step is that the parabolic part of three-temperature is solved by an implicit scheme, where the work terms \(\left(P_{\alpha} + q\chi_{\alpha} + \frac{\partial \varepsilon_{\alpha}}{\partial V} \right) \frac{dV}{dt}\) are given from the 1st step. Moreover, the computational mesh in the 2nd step is determined by the 1st step.

The equation (1) is the mass conservation equation, which leads to an algebraic equation, i.e., the mass on \((n+1)\)-th time level is equal to that on \(n\)-th time level \((M^{n+1} = M^n)\) in each Lagrangian cell.

The equation (2) is the momentum conservation equation, and it is solved by the explicit time discretization and the staggered space discretization, i.e., the thermodynamic unknowns (density, pressure) are defined on the center of cell and the flow velocity is defined on the node of cell. Denote \(X_0 = (x_0, r_0)\) be a node of cell in \(X = (x, r)\) plane, and the node velocity \(\bar{U}_0 = (u_0, v_0)\). Then the node \(X_{0}^{n+\frac{1}{2}} = (x_{0}^{n+\frac{1}{2}}, r_{0}^{n+\frac{1}{2}})\) at time \(t^{n+1}\) is given by

\[X_{0}^{n+\frac{1}{2}} = X_{0}^{n-\frac{1}{2}} + \bar{U}_{0}^{n+\frac{1}{2}} \Delta t,\]

where \(\bar{U}_{0}^{n+\frac{1}{2}}\) is calculated in the following by the Wilkins scheme:
\begin{equation}
\begin{cases}
\frac{u_n^{n+\frac{1}{2}}}{2} = u_0^n - \frac{\Delta t}{2(\rho A)_0^n}
\left[P_1(r_1 - r_4) + P_2(r_2 - r_1) + P_3(r_3 - r_2) + P_4(r_4 - r_3)\right]^n,

\frac{v_n^{n+\frac{1}{2}}}{2} = v_0^n + \frac{\Delta t}{2(\rho A)_0^n}
\left[P_1(x_1 - x_4) + P_2(x_2 - x_1) + P_3(x_3 - x_2) + P_4(x_4 - x_3)\right]^n,
\end{cases}
\end{equation}

where \((x_i, r_i)(1 \leq i \leq 4)\) are cell nodes, \((\rho A)_0^n \equiv \frac{1}{4}\left[(\rho A)_1^n + (\rho A)_2^n + (\rho A)_3^n + (\rho A)_4^n\right]_n\), and \((\rho)_k^n\) and \((A)_k^n\) are the density and area of cell \(k\) respectively. \(4\) comes from the discretization of \((2)\) on a dual cell (1234) with dashed lines as cell-edge in Fig. 1, which depicts the stencil of the Wilkins scheme.

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{wilkins_stencil}
\caption{The stencil of Wilkins scheme}
\end{figure}

The three-temperature energy equations \((3)\) are parabolic equations, which are strongly nonlinear and stiff, moreover the coefficients are discontinuous for multi-material problems. When the implicit time discretization and LSOM space approximation are applied in solving \((3)\), a large nonlinear discrete system is formed whose solution is very time-consuming, moreover it is difficult to implemented naturally on parallel computer. So a parallelization approach will be proposed in section 5, and a new discrete system is formed, which consists of smaller and decoupling discrete systems and can be solved in parallel on parallel computer.

3. LSOM for solving diffusion equations. Consider the following model equation for nonlinear diffusion

\begin{equation}
C(u) \frac{\partial u}{\partial t} + \nabla \cdot \mathbf{F} = Q(X, t, u)u, \quad X \in \Omega, \quad t \in (0, T],
\end{equation}

where \(\mathbf{F} = -\kappa(X, t, u)\nabla u\) is the heat flux, \(u = u(X, t)\) is a function to be solved, \(\Omega\) is a polygonal domain in \(\mathbb{R}^2\), \(X = (x, r) \in \Omega\), \(C(u)\) and \(\kappa\) are positive functions which depend on \(u\) and could be discontinuous on \(\Omega\), and \(Q\) is a given nonlinear function.

For solving the equation \((5)\) we will use the local support-operators method (LSOM) in [8], which satisfies the following properties: (1) it gives almost second-order accuracy on distorted grids even with material discontinuities; (2) it has a local stencil; (3) its coefficient matrix is symmetric positive-definite.
The domain $\Omega$ is divided into a structural quadrilateral mesh with cell index $(i,j)$. The finite volume discretization of 5 can be written as

$$C(u_{i,j}^{n+1}) \frac{u_{i,j}^{n+1} - u_{i,j}^n}{\Delta t} + \frac{1}{S_{i,j}} \sum_{k=1}^{4} F_{i,j,k}^{n+1} = Q_{i,j}^{n+1} u_{i,j}^{n+1},$$

where the unknown variable defined on the cell center is denoted by $u_{i,j}$, and the unknown variable defined at each cell-edge is denoted by $u_{i,j,k}$, $1 \leq k \leq 4$, (see Fig. 2). The outward normal flux crossing each edge is denoted by $F_{i,j,k}$, $1 \leq k \leq 4$, and the subscript $k$ represents the $k$-th edge of the cell $(i,j)$. The area of the cell $(i,j)$ is denoted by $S_{i,j}$.

Figure 2. The stencil for LSOM

The normal flux $F_{i,j,k}$ for LSOM is represented in terms of the unknown variables $u_{i,j}$ and $u_{i,j,k}$ as follows:

$$A_{i,j} \begin{pmatrix} F_{i,j,1} \\ F_{i,j,2} \\ F_{i,j,3} \\ F_{i,j,4} \end{pmatrix} = \kappa(u_{i,j}) \begin{pmatrix} u_{i,j} - u_{i,j,1} \\ u_{i,j} - u_{i,j,2} \\ u_{i,j} - u_{i,j,3} \\ u_{i,j} - u_{i,j,4} \end{pmatrix},$$

where $A_{i,j}$ is a $4 \times 4$ symmetric matrix depending on the geometrical quantities of cell, i.e.,

$$A_{i,j} = S_{i,j} \begin{pmatrix} a_{11} & a_{12} & 0 & a_{14} \\ a_{21} & a_{22} & a_{23} & 0 \\ 0 & a_{32} & a_{33} & a_{34} \\ a_{41} & 0 & a_{43} & a_{44} \end{pmatrix},$$

where $a_{k,k+1} = \frac{w_k \cos \varphi_{k,k+1}}{l_k^2 \sin^2 \varphi_{k,k+1}}$, $a_{k,k} = \frac{w_{k-1}}{l_k^2 \sin^2 \varphi_{k-1,k}} + \frac{w_k}{l_k^2 \sin^2 \varphi_{k,k+1}}$, $w_k$ is weighted coefficient, $l_k$ is length of the $k$-th edge of the cell $(i,j)$, and $\varphi_{k,k+1}$ is the $k$-th internal angle of the cell $(i,j)$, (see Fig. 2). The subscript $k$ loops from 1 to 4, i.e., for $k = 4$, the $k + 1$ is 1. To complete the system (6) two continuity conditions are imposed at each edge, for example, for the first edge of the cell $(i,j)$ there is

$$F_{i,j,1} = -F_{i+1,j,3}, u_{i,j,1} = u_{i+1,j,3}.$$  

LSOM is obtained in [8] by setting

$$w_k = V_k \left( \sum_{l=1}^{4} V_l \right)^{-1}, \quad V_k = \frac{l_k l_{k+1}}{4} \sin \varphi_{k,k+1}, \quad (k = 1, 2, 3, 4).$$
A generalization of LSOM is discussed in [17] by taking different weight coefficient \( w_k \) and is named as the CFM (Cell Functional Minimization).

In the LSOM formulation, the primary unknowns include both the cell-center and the edge-centered unknowns, and the normal flux crossing edge is given by (7). The fact that the edge-centered unknowns are primary ones is helpful in the construction of parallel iterative methods, since the values on the inner boundaries are needed in solving sub-domain problems.

4. Nonlinear iteration method. We use the Picard fixed-point iteration method to solve the nonlinear discrete system (6), (7) and (9).

In the following, the superscript \( n + 1 \) identifying the current time level is omitted for simplicity, so that the discrete function \( u^{n+1}_{i,j} \) is denoted by \( u_{i,j} \) simply. The superscript \( n \) of \( u^n \) at previous time level is preserved for avoiding confusion. Let \( s \) be the nonlinear iteration index. Then Picard nonlinear iteration procedure is devised as follows:

\[
C(u^{(s)}_{i,j}) \frac{u^{(s+1)}_{i,j} - u^n_{i,j}}{\Delta t} + \frac{1}{S_{i,j}} \sum_{k=1}^{4} F^{(s+1)}_{i,j,k} = Q^{(s)}_{i,j} u^{(s+1)}_{i,j},
\]

where

\[
\begin{pmatrix}
F^{(s+1)}_{i,j,1} \\
F^{(s+1)}_{i,j,2} \\
F^{(s+1)}_{i,j,3} \\
F^{(s+1)}_{i,j,4}
\end{pmatrix}
= \kappa(u^{(s)}_{i,j}) A_{i,j}^{-1}
\begin{pmatrix}
u^{(s)}_{i,j,1} - u^{(s+1)}_{i,j,1} \\
u^{(s)}_{i,j,2} - u^{(s+1)}_{i,j,2} \\
u^{(s)}_{i,j,3} - u^{(s+1)}_{i,j,3} \\
u^{(s)}_{i,j,4} - u^{(s+1)}_{i,j,4}
\end{pmatrix},
\]

and the value in all the nonlinear coefficients \( C(u), \kappa(X, t, u) \) and \( Q(X, t, u) \) is taken to be the value of the previous iteration step so as to form a linearized system.

5. Parallel iterative method. In this section, we present a parallel iterative method for solving nonlinear diffusion equations, which is the main part of our parallelization method for solving three-temperature radiation-hydrodynamic equations. For the sake of clarity, the global domain \( \Omega \) is decomposed into two overlapped sub-domains \( \Omega_1 \) and \( \Omega_2 \), see Fig. 3. Denote \( \Gamma_{T2} = \partial \Omega_1 \cap \Omega_2 \) and \( \Gamma_{T1} = \Omega_1 \cap \partial \Omega_2 \), where \( \Gamma_{T2} \) and \( \Gamma_{T1} \) are the inner boundaries for the sub-domains \( \Omega_1 \) and \( \Omega_2 \) respectively.

![Figure 3. The overlapped sub-domains](image)

The parallel iterative method is based on domain decomposition and devised as follows. The diffusion problem to be solved on the sub-domain \( \Omega_1 \) is

\[
C(u^{(s)}_{x,r}) \frac{u^{(s+1)}_{x,r} - u^n_{x,r}}{\Delta t} + \frac{1}{S_{x,r}} \sum_{k=1}^{4} F^{(s+1)}_{k} = Q^{(s)} u^{(s+1)}_{x,r}, \quad (x, r) \in \Omega_1,
\]

\[
\alpha u^{(s+1)} + \kappa(u^{(s)}) \frac{\partial u^{(s+1)}_{x,r}}{\partial n} = \alpha \tilde{u}^{(s)} + \kappa(\tilde{u}^{(s)}) \frac{\partial \tilde{u}^{(s)}_{x,r}}{\partial n}, \quad (x, r) \in \Gamma_{T2},
\]
together with the prescribed boundary condition on \( \partial \Omega \cap \partial \Omega \), and initial condition on \( t = 0 \). Here \( F_k^{(s+1)} \) is the discrete version of \(- \int_{k} \kappa(u(s)) \frac{\partial u^{(s+1)}}{\partial n} dl\) on the \( k \)-th edge of a cell in \( \Omega_1 \). On the sub-domain \( \Omega_2 \), the following diffusion problem is solved

\[
C(\tilde{u}^{(s)}) \frac{\tilde{u}^{(s+1)} - \tilde{u}^{n}}{\Delta t} + \frac{1}{S} \sum_{k=1}^{4} \tilde{F}_k^{(s+1)} = \tilde{Q}^{(s)} \tilde{u}^{(s+1)}, \quad (x, r) \in \Omega_2, \tag{14}
\]

\[
\alpha \tilde{u}^{(s+1)} + \kappa(\tilde{u}^{(s)}) \frac{\partial \tilde{u}^{(s+1)}}{\partial n} = \alpha u^{(s)} + \kappa(u^{(s)}) \frac{\partial u^{(s)}}{\partial n}, \quad (x, r) \in \Gamma_{12}, \tag{15}
\]

together with the prescribed boundary condition on \( \partial \Omega_2 \cap \partial \Omega \), and initial condition on \( t = 0 \). Here \( F_k^{(s+1)} \) is the discrete version of \(- \int_{k} \kappa(u(s)) \frac{\partial u^{(s+1)}}{\partial n} dl\) on the \( k \)-th edge of a cell in \( \Omega_2 \). \( \alpha \) is a given nonnegative constant, and we take \( \alpha = 0.1 \) or \( \alpha = 10 \) in the following test models 1 and 2 respectively.

For each \( s \) both of the problems (12)-(13) and (14)-(15) are linearized and decoupled, which can be solved on parallel computer in parallel. Note that the nonlinear Picard iteration is nested within domain decomposition, and the iteration among sub-domains and the nonlinear Picard iteration are performed synchronously.

In our practical implementation the global domain \( \Omega \) is divided by the structural quadrilateral grids, and partitioned into some sub-domains with shape shown in Fig. 4. The grid cells are averagely distributed in sub-domains, and each sub-domain is assigned to one CPU. Moreover, for the convenience of the parallel computation, each sub-domain will be extended with a row and a column of cells along its inner boundary (see the sub-domains with the dashed borders in Fig. 4). The overlapped domain decomposition strategy is adopted so that it is convenience for using the original code in solving momentum equations, and for implementing various boundary conditions at inner boundary, including Dirichlet, Neumann and Robin conditions in solving three temperature equations.

Now the algorithm procedure for parallel iterative method is described. To state the main idea clearly, here we give just the case of two sub-domains as shown in Fig. 5 with Robin boundary condition at inner boundary. The sub-domain 1 with blue broken line border corresponds one processor numbered 1, and the sub-domain 2 with red solid line border corresponds another processor numbered 2.

The detailed procedure for our parallelization method on \( t^{n+1} \) time level is given as follows.

**Figure 4.** The sub-domains diagram
1. For $s = 0$, take the values of the $t^n$ time level, and compute the values $\alpha u^n + \kappa (u^n) \frac{\partial u^n}{\partial n}$ at $\Gamma_{12}$ on the Processor 1, and $\alpha \bar{u}^n + \kappa (\bar{u}^n) \frac{\partial \bar{u}^n}{\partial n}$ at $\Gamma_{21}$ on the Processor 2, and pass them each other by local communication. Then, they are taken as the Robin boundary conditions for the sub-domains 2 and 1 respectively.

2. Perform nonlinear iteration, and solve the linearized problems on each sub-domain in parallel. First, in the Processors 1 and 2 the coefficients and right hand items are computed, and then the two linear algebraic systems of equations are solved independently to obtain the cell-centered and the cell-edged unknowns $u^{(s+1)}$.

3. Calculate the values $\alpha u^{(s+1)} + \kappa (u^{(s+1)}) \frac{\partial u^{(s+1)}}{\partial n}$ at $\Gamma_{12}$ on the Processor 1, and Calculate the values $\alpha \bar{u}^{(s+1)} + \kappa (\bar{u}^{(s+1)}) \frac{\partial \bar{u}^{(s+1)}}{\partial n}$ at $\Gamma_{21}$ on the Processor 2, and then pass them each other to refresh their inner boundary conditions.

4. Repeat the steps 2 and 3 till the convergence criterion holds.

It is noted that the above procedure is also fit for the case that inner boundary condition is Dirichlet or Neumann boundary condition. Furthermore, for our parallel iterative method only local communication is needed, i.e., message passing happens only among neighboring processors, which is beneficial to get high parallel efficiency.

6. **Numerical experiments.** In this section we present some numerical examples to show the performance of the parallelization method for solving two dimensional, three temperature, Lagrangian radiation hydrodynamic problems.

In our test problems we use the Thomas-Fermi statistic equation of state for electron, and the idea gas equation of state for ion such as $P_i = \Gamma_i \rho T_i$, $\varepsilon_i = \frac{3}{2} \Gamma_i T_i$, where $\Gamma_i$ is a physical constant related with medium, and for photon $P_e = \frac{a}{\rho} T_e^3$, $\varepsilon_r = \frac{5}{2} T_e^4$, where $a$ is the radiation constant.

For the choice of time-step we take $\Delta t = 10^{-5}$ at the initial time, and then its size varies adaptively according to the number of nonlinear iteration in solving three temperature equations and CFL conditions. When the number of nonlinear iteration is larger than 20 or CFL condition does not hold, the time-step decreases. When the number of nonlinear iteration is smaller than 10, the time-step increases and
the increased time-step must satisfy the following constraints: (1) CFL condition holds; (2) it is not larger than a given bound, e.g., $5 \times 10^{-3}$; (3) the frequency of increasing time-step is not larger than a given number, e.g., 0.1, that is the number of increasing time-step is not larger than 40 when 400 time levels are considered to be one cycle. We refer to [10] and [23] for more detail discussion.

The convergence criterion of nonlinear parallel iteration for solving three temperature equations is that the maximum norm of the nonlinear residues of three temperature equations are smaller than $5 \times 10^{-7}$.

6.1. Test Model 1. The initial computation domain is a quarter of circular disc \{(x, r)|\sqrt{x^2 + r^2} \leq 150, x \geq 0, r \geq 0\} with its initial grids as shown in Fig. 6. The boundary conditions are

\[
\begin{align*}
  u &= 0, \quad \frac{\partial T_\alpha}{\partial n} = 0 \quad \text{on the left boundary, } (\alpha = e, i, p) \\
  v &= 0, \quad \frac{\partial T_\alpha}{\partial n} = 0 \quad \text{on the bottom boundary, } (\alpha = e, i, p) \\
  \frac{\partial T_e}{\partial n} = \frac{\partial T_i}{\partial n} = 0, \quad \kappa \frac{\partial T_p}{\partial n} = T^4_b \quad \text{on the free surface (outer boundary),}
\end{align*}
\]

where the $T_b = 0.01 \times (30 - (t/100 - \sqrt{29})^2)$, and its variation curve is shown in Fig. 7. The initial position of the free surface is the quarter of circular arc \{(x, r)|\sqrt{x^2 + r^2} = 150, x \geq 0, r \geq 0\}. The initial conditions are

\[
\begin{align*}
  &T_\alpha = 0.01, (\alpha = e, i, p) \quad \text{on } \{t = 0\} \times \{\sqrt{x^2 + r^2} \leq 150\} \\
  &\rho = 0.014866, \quad \text{for } \sqrt{x^2 + r^2} \leq 120, \\
  &\rho = 0.076817, \quad \text{for } 120 < \sqrt{x^2 + r^2} < 150.
\end{align*}
\]

\[\text{Figure 6. The initial grids } 20 \times 30\]

In order to verify the accuracy of the parallelization method, we compare the result of using 16 CPUs with that computed by 1 CPU. The initial grid is $20 \times 30$, which is radial as shown in Fig. 6. \((m, n)\) denote a cell located at \(m\)-th column (circumferential direction) and \(n\)-th row (axial direction), and $1 \leq m \leq 20, 1 \leq n \leq 30$.

Figs. 8, 9 and 10 are respectively the temperature variation curves of the electron, ion and photon at cells (10, 1), (10, 6), (10, 7), (10, 30). The solid lines are
Figure 7. Variation curve for $T_b$

obtained by the original sequential method on 1 CPU, and the dashed lines are computed by the parallelization method on 16 CPUs. We can see that the both lines are completely coincident in whole computational phase, which show that the parallelization method has same accuracy as the original sequential method.

Figure 8. Eletron temperature variation curve

Fig. 11 gives the time-steps and the nonlinear iteration numbers used by the original sequential method on 1 CPU and by the parallelization method on 16 CPUs, respectively. It shows that the parallelization method needn’t decrease the size of time-step, moreover the nonlinear iteration numbers remain same.

Fig. 12 presents the linear iteration numbers used by the original sequential method on 1 CPU and by the parallelization method on multi-CPUs, which are in fact the maximum of all linear iteration numbers used on various CPUs from 2 to 16 in solving those linear systems resulted from the parallelization method. It shows that the linear iteration numbers decrease with increasing the number of CPUs and the scale of linear algebraic system becoming small.
Figure 9. Ion temperature variation curve

Figure 10. Photon temperature variation curve
Table 1 gives the time for the model 1 by using CPUs from 2 to 16, and the computational time is 6922s for 1 CPU. The reason for super-linear speedup includes that more linear iterations are needed to solve a global linear system on 1 CPU, which leads to a larger scale linear system than those linear systems for sub-domain problems on multi-CPUs, while the time step size and the nonlinear iteration numbers are kept to be same, see Fig. 11. The parallel efficiency is estimated by comparing with the time for 2 CPUs, which shows the parallel accelerated effect is remarkable.

Table 1. The parallel efficiency on grid 20 × 30

| CPUs | 2    | 4    | 8    | 16   |
|------|------|------|------|------|
| Time(s) | 2934 | 1473 | 727  | 394  |
| Efficiency | 1    | 99.6% | 101% | 93.1% |

Table 2 gives the numerical results for model 1 with 400 × 800 cells. The largest number of the CPUs is 3200, and the parallel efficiency is approximated 89%. This
result shows that the parallelization method is very efficient and has good scalability for large scale model.

Table 2. The parallel efficiency on grid 400 × 800

| CPUs        | 400(40 × 10) | 800(40 × 20) | 1600(40 × 40) | 3200(40 × 80) |
|-------------|--------------|--------------|---------------|---------------|
| Time(s)     | 4852         | 2187         | 1132          | 677           |
| Efficiency  |              |              |               |               |

6.2. Test Model 2. The initial computation domain is a rectangle \{(x,r)|0 \leq x \leq 150, 0 \leq r \leq 150\} as shown in Fig. 13 (left) with the initial grid 20 × 20. The boundary conditions are

\[
\vec{U} \cdot \vec{n} = 0, \quad \frac{\partial T_\alpha}{\partial n} = 0 \quad (\alpha = e, i, p), \quad \text{on the left, top and bottom boundaries},
\]

\[
\frac{\partial T_e}{\partial n} = \frac{\partial T_i}{\partial n} = 0, \quad \kappa \frac{\partial T_r}{\partial n} + aT_r^4 = 0, \quad \text{on the right boundary}
\]

and the initial conditions at \( t = 0 \) are

\[
T_\alpha = 0.01 \quad (\alpha = e, i, p), \quad 0 \leq x \leq 150, 0 \leq r \leq 150,
\]

\[
\rho = 0.001 + ran \times 0.0001, \quad 0 \leq x \leq 150, 0 \leq r \leq 120,
\]

\[
\rho = 0.02 + ran \times 0.002, \quad 0 \leq x \leq 150, 120 < r < 150,
\]

where \( ran \) is a disturbance factor, which is a random number from \(-1\) to \(1\). The initial density is shown as Fig. 13 (right). \((m,n)\) denotes a cell located at \(m\)-th column \((x\) direction) and \(n\)-th row \((r\) direction), and \(1 \leq m \leq 20, 1 \leq n \leq 20\).

In the most left column of grids, the source \( S(t) = 0.001 \times (10 - (t/100 - 3)^2)t \) is added, and its time variation curve is shown as Fig. 14.

![Figure 13. The initial grids 20 × 20 and the initial density](image)
our future work we will consider how to improve the prediction of inner interface boundary conditions in order to reduce the nonlinear iteration numbers. Table 3 gives the time for the model 2 by using CPUs from 2 to 16, and the computational time is 3912s for 1 CPU. The parallel efficiency is estimated by comparing with the time for 2 CPUs. Note that the increase of nonlinear iteration numbers can reduce the parallel efficiency, moreover when grid keeps to be 20 × 20 and 16 CPUs are used, the cell number is not larger than 36 in each CPU, so the computational granularity in each CPU is very small.

| CPUs | 2    | 4    | 8    | 16   |
|------|------|------|------|------|
| Time(s) | 2792 | 1490 | 783  | 553  |
| Efficiency | 1    | 93.7% | 89.1% | 63.1% |
adopts the overlapped domain decomposition method and combines with nonlinear Lagrangian radiation-hydrodynamic problems with three-temperature model. It

This paper presents a parallel iterative method for solving the

Figure 16. Electron temperature variation curve in left cells

Figure 17. Ion temperature variation curve in middle cells

7. Conclusions. This paper presents a parallel iterative method for solving the Lagrangian radiation-hydrodynamic problems with three-temperature model. It adopts the overlapped domain decomposition method and combines with nonlinear
iteration. The computational mesh is moving with fluid, and then the interface among sub-domains is also moving. The three-temperature system of nonlinear diffusion equations is solved by the LSOM method on arbitrary quadrilateral grids. The sub-domain problems are linearized and can be solved on massive parallel computers in parallel. Numerical results show the parallel iterative method is accurate, which gives the same numerical results as the fully implicit scheme, moreover it is CPU-time scalable and linear parallel speedup is obtained in our numerical tests.

Acknowledgments. The authors thank the colleagues in the team of numerical methods for radiation hydrodynamics for fruitful discussions, and Prof. J. M. Wu for providing us with sequential LSOM code.
REFERENCES

[1] C. N. Dawson, Q. Du and T. F. Dupont, A finite difference domain decomposition algorithm for numerical solution of the heat equation, Math. Comp., 57 (1991), 63–71.
[2] M. Dryja, Substructuring methods for parabolic problems, in Fourth International Symposium on Domain Decomposition Methods for Partial Differential Equations (Moscow, 1990), SIAM, Philadelphia, PA, (1991), 264–271.
[3] S. Gunter and K. Lackner, A mixed implicit-explicit finite difference scheme for heat transport in magnetized plasmas, Journal of Computational Physics, 228 (2009), 282–293.
[4] E. Jamelot and P. C. Jr, Fast non-overlapping Schwarz domain decomposition methods for solving the neutron diffusion equation, Journal of Computational Physics, 241 (2013), 445–463.
[5] Yu. M. Laevsky and O. V. Rudenko, Splitting methods for parabolic problems in nonrectangular domains, Appl. Math. Lett., 8 (1995), 9–14.
[6] H. L. Liao, H. S. Shi and Z. Z. Sun, Corrected explicit-implicit domain decomposition algorithms for two-dimensional semilinear parabolic equations, Science in China Series A: Mathematics, 52 (2009), 2362–2388.
[7] P. H. Mairle, R. Abgrall, J. Breil and J. Oudij, A centered Lagrangian scheme for multidimensional compressible flow problems, SIAM Journal on Scientific Computing, 29 (2007), 1781–1824.
[8] J. E. Morel, R. M. Roberts and M. J. Shashkov, A local support-operators diffusion discretization scheme for quadrilateral $r-z$ meshes, Journal of Computational Physics, 144 (1998), 17–51.
[9] S. Ovtchinnikov and X. C. Cai, One-level Newton-Krylov-Schwarz algorithm for unsteady nonlinear radiation diffusion problem, Numerical Linear Algebra with Applications, 11 (2004), 867–881.
[10] W. J. Rider and D. A. Knoll, Time step size selection for radiation diffusion calculations, Journal of Computational Physics, 152 (1999), 790–795.
[11] M. Shashkov, Conservative Finite Difference Methods, CRC Press, Boca Raton, FL, 1996.
[12] Z. Q. Sheng, G. W. Yuan and X. D. Hang, Unconditional stability of parallel difference schemes with second order accuracy for parabolic equation, Applied Mathematics and Computation, 184 (2007), 1015–1031.
[13] A. Shrestakova, J. Milovich and D. Kershaw, Parallelization of an unstructured-grid, laser fusion design code, SIAM News, 32 (1999), 6–10.
[14] H. S. Shi and H. L. Liao, Unconditional stability of corrected explicit-implicit domain decomposition algorithms for parallel approximation of heat equations, SIAM J. Numer. Anal., 44 (2006), 1584–1611.
[15] A. Toselli and O. Widlund, Domain Decomposition Methods-Algorithms and Theory, Springer-Verlag, Berlin Heidelberg, 2005.
[16] M. L. Wilkins, Computer Simulation of Dynamic Phenomena, Springer-Verlag, Berlin Heidelberg, 1999.
[17] L. Yin, J. M. Wu and Y. Z. Yao, A cell functional minimization scheme for parabolic problem, Journal of Computational Physics, 229 (2010), 8935–8951.
[18] G. W. Yuan, X. D. Hang and Z. Q. Sheng, Parallel difference schemes with interface extrapolation terms for quasi-linear parabolic systems, Science in China Series A: Mathematics, 50 (2007), 253–275.
[19] G. W. Yuan and X. D. Hang, Conservative parallel schemes for diffusion equations, Chinese Journal of Computational Physics, 27 (2010), 475–491.
[20] G. W. Yuan, L. G. Shen and Y. L. Zhou, Parallel Difference Schemes for Parabolic Problem, in Proceeding of 2002 5th International Conference on Algorithms and Architectures for Parallel Processing, IEEE Computer Society, (2002), 238-242.
[21] G. W. Yuan, Y. Z. Yao and L. Yin, Conservative domain decomposition procedure for nonlinear diffusion problems on arbitrary quadrilateral grids, SIAM J. Sci. Comput., 33 (2011), 1352–1368.
[22] G. W. Yuan and F. L. Zuo, Parallel differences schemes for heat conduction equations, International Journal of Computer Mathematics, 80 (2003), 995–999.
[23] J. Y. Yue and G. W. Yuan, Picard-Newton iterative method with time step control for multimaterial non-equilibrium radiation diffusion problem, Commun. Comput. Phys., 10 (2011), 841–866.
[24] S. H. Zhu, Conservative domain decomposition procedure with unconditional stability and second-order accuracy, *Applied Mathematics and Computation*, 216 (2010), 3275–3282.

[25] Y. Zhuang and X. Sun, Stabilized explicit-implicit domain decomposition methods for the numerical solution of parabolic equations, *SIAM J. Sci. Comput.*, 24 (2002), 335–358.

Received November 2013; 1st revision January 2014; 2nd revision March 2014.

*E-mail address: yuan.guangwei@iapcm.ac.cn*

*E-mail address: yao.yanzhong@iapcm.ac.cn, corresponding author*