Commun. Comput. Phys.
doi: 10.4208/cicp.310110.161010a

Vol. **10**, No. 4, pp. 844-866
October 2011

Picard-Newton Iterative Method with Time Step Control for Multimaterial Non-Equilibrium Radiation Diffusion Problem

Jingyan Yue and Guangwei Yuan*

National Key Laboratory of Science and Technology on Computational Physics, Institute of Applied Physics and Computational Mathematics, P.O. Box 8009, Beijing 100088, China.

Received 31 January 2010; Accepted (in revised version) 16 October 2010

Available online 13 June 2011

Abstract. For a new nonlinear iterative method named as Picard-Newton (P-N) iterative method for the solution of the time-dependent reaction-diffusion systems, which arise in non-equilibrium radiation diffusion applications, two time step control methods are investigated and a study of temporal accuracy of a first order time integration is presented. The non-equilibrium radiation diffusion problems with flux limiter are considered, which appends pesky complexity and nonlinearity to the diffusion coefficient. Numerical results are presented to demonstrate that compared with Picard method, for a desired accuracy, significant increase in solution efficiency can be obtained by Picard-Newton method with the suitable time step size selection.

AMS subject classifications: 65M06, 65M08, 65M12

Key words: Non-equilibrium radiation diffusion, Picard, Picard-Newton based on linearization-discretization, nonlinear iterative method, time step control.

1 Introduction

Picard nonlinear iterative method is a globally convergent method and has been used extensively in many applications. However its iterative convergence rate is only one order, and its solution efficiency is low in solving some practical problems. So Newton-like methods with super-linear convergence must be introduced, and these nonlinear Newton-Krylov solution techniques [2–11] have been developed to ensure nonlinearities convergence. It is well-known (see, e.g., [5–8]) that there are two main obstacles which have prevented people from using Newton-like methods for large scale multi-physics simulations.

*Corresponding author. *Email addresses:* yue_jingyan@iapcm.ac.cn (J. Yue), yuan_guangwei@iapcm.ac.cn (G. Yuan)

The first obstacle in Newton-like methods for solving some practical multi-physical problems is the need for evaluating the entries of the Jacobian matrix. To overcome this obstacle we have proposed in [20, 21] a new nonlinear iterative method named as Picard-Newton (P-N) iterative method for solving numerically nonlinear diffusion equations and multimaterial non-equilibrium radiation diffusion problems. To construct a new nonlinear iteration method, we directly design iterative method for the time discretized nonlinear PDEs instead of nonlinear algebraic system from implicit discretization scheme. By linearizing the time discretized nonlinear PDEs first and discretizing the resulting linear PDEs next, which is named as the linearization-discretization (LD) approach, we have devised the Picard-Newton iterative scheme which gives a specific procedure to the formation of (exact or approximated) Jacobian matrix. It follows that the first obstacle can be overcome through the LD approach.

The second obstacle is that Newton-like methods cannot be expected always to converge since they are locally convergent. In other words, an initial guess inside the radius of convergence is required for Newton-like methods to converge, while for time-dependent or transient problems the initial guess is usually the converged solution from the previous time step. A reliable time step selection is necessary for nonlinear iterative methods to ensure the methods to operate with desired accuracy. Then time step control is essentially important for fast and accurate numerical solution, especially when Newton-like method is employed.

In addition to the difficulties of evaluating the Jacobian matrix and the radius of convergence being small, on each iteration step of Newton-like method for nonlinear systems the cost of solving the linear systems to arrive at the new iteration values might be very expensive, which depends mainly on the matrix properties of the linear systems. In other words, since the matrix properties of the linear systems corresponding to Newton method are usually worse than those corresponding to Picard method, the computational cost of the original Newton method at each nonlinear iteration step is more expensive than that of Picard method.

We hope our P-N method can accelerate the existing Picard method. The P-N method can be formulated by adding certain convective terms of one order to the Picard method. When the one order terms are discretized by centered scheme, the resulting nonlinear iteration is named as P-NC iterative scheme, while discretized by upwind scheme, the resulting nonlinear iteration is named as P-NU iterative scheme. It is worth to point out that the P-NC may be equivalent to standard Newton method, while the P-NU is new and is an improvement over Picard and Newton methods. From the construction of our P-NU method we can see that the diagonal dominance of the matrix from P-NU is not worse than that from Picard method, and then it is possible to reduce remarkably computational time for each step of nonlinear iteration.

Since our P-N has overcome the first obstacle through the LD approach, it remains to demonstrate our P-N need not introduce more strict time steps than that used for Picard. In this paper, two time step control methods are investigated and a study of temporal accuracy of a first order time integration is presented for P-N iteration. We are

primarily concerned with assessing the CPU cost for a desired accuracy as nonlinearity of the problem concerned increases. Numerical results clearly indicate that compared with Picard, significant increase in solution efficiency can be obtained by P-N with the suitable time step size selection. The extend of increase is remarkably larger for more nonlinear problem. Although we focus on non-equilibrium radiation diffusion, the P-N should also be applicable to other physical systems described by the equations of parabolic type.

The remainder of the paper is structured as follows. First, the mathematical model for the coupled system of multimaterial non-equilibrium radiation diffusion with material conduction equations is introduced in Section 2. Next, Picard-Newton iterative scheme is presented in Section 3. Then in Section 4, two time step control methods are given. In Section 5, numerical tests are provided to show the performance comparison between Picard and P-N methods. Finally, conclusion is given in Section 6.

2 Mathematical model

The mathematical model used in this paper is a system of two-dimensional multimaterial non-equilibrium radiation diffusion coupled to material conduction equations [1,6,7]. Let E be radiation energy density and T material temperature. The coupled system is given by

$$\frac{\partial E}{\partial t}-\nabla\cdot(D\nabla E)=\sigma_a(T^4-E), \tag{2.1a}$$

$$\frac{\partial T}{\partial t}-\nabla\cdot(\kappa\nabla T)=\sigma_a(E-T^4), \tag{2.1b}$$

where

$$\nabla=\begin{bmatrix}\partial/\partial x\\ \partial/\partial y\end{bmatrix}$$

is the gradient operator and it acts on function u as follows:

$$\nabla u=\begin{bmatrix}\partial u/\partial x\\ \partial u/\partial y\end{bmatrix}.$$

The divergence operator $\nabla\cdot$ acts on a vector function

$$\vec{A}=\begin{bmatrix}A_1\\ A_2\end{bmatrix}$$

and gives a scalar function

$$\nabla\cdot\vec{A}=\partial A_1/\partial x+\partial A_2/\partial y.$$

The energy exchange is controlled by the photon absorption cross section

$$\sigma_a(T)=\frac{z^3}{T^3}, \tag{2.2}$$

where z is a function of the material and varies as a function of space (x,y). High values of z and low values of T lead to higher energy exchange and therefore tighter coupling between Eqs. (2.1a) and (2.1b). In the more nonlinear problem in Section 5, the two materials have $z=1$ and 10 respectively. In the less nonlinear problems, the two materials have $z=1$ and 5 or 2.5 respectively.

The model for the radiation diffusion coefficient is:

$$D=\frac{1}{3\sigma_a+\frac{|\nabla E|}{E}}. \tag{2.3}$$

The second part in the denominator, $|\nabla E|/E$, is the flux limiter, which is used to keep the propagation velocity of a radiation wave front less than the speed of light [1].

The following form of the material (plasma) conduction coefficient from Spitzer and Harm [16] is used,

$$\kappa=c_0T^{5/2}, \tag{2.4}$$

where $c_0=1.0\times10^{-2}$.

3 Picard-Newton iterative method

Traditionally, the nonlinear iterative methods are designed for the fully implicit discretization scheme of nonlinear diffusion equations: first,(1st step) fully implicitly discretize the nonlinear partial differential equations (PDEs); then, (2nd step) linearize the system of discrete equations (by Picard or Newton-like methods), and get a sequence of linear system of algebraic equations (maybe the expression of coefficient matrix isn't explicit); finally, (3rd step) solve the linear problems. We call this kind of methods as DL (discretization-linearization) approach. No matter which nonlinear iterative method is taken, Picard or Newton or some other methods, the discretization has been predetermined independently in the DL approach.

To construct a new nonlinear iterative method (P-N), we firstly design an iteration sequence for the time discretized nonlinear PDEs and then devise the spatial discretization scheme for the linearized PDEs, which is named as LD (linearization-discretization) approach. In the LD approach, the feature of linearized PDEs can be taken into consideration in the construction of discrete schemes. The solution process of our P-N method is as follows:

Step 1 Design nonlinear iteration sequence for the time-discretized nonlinear PDEs by Newton linearization, and get an iteration sequence of linearized PDEs.

Step 2 Design the spatial discretization scheme for the resulting linearized PDEs, and educe a system of linear algebraic equations with an explicit coefficient matrix.

Step 3 Solve the linear problems by linear iteration method.

3.1 Newton linearization of the time discretized PDEs

Consider the first order backward Euler time discretization of Eqs. (2.1a) and (2.1b)

$$\frac{E^{n+1}-E^n}{\Delta t}-\nabla\cdot(D^{n+1}\nabla E^{n+1})-f_1^{n+1}=0, \tag{3.1a}$$

$$\frac{T^{n+1}-T^n}{\Delta t}-\nabla\cdot(\kappa^{n+1}\nabla T^{n+1})-f_2^{n+1}=0, \tag{3.1b}$$

where $D^{n+1}=D(E^{n+1},T^{n+1},\nabla E^{n+1})$ and $\kappa^{n+1}=\kappa(T^{n+1})$ are given by (2.3) and (2.4) respectively, f_1^{n+1} and f_2^{n+1} are given by

$$f_1=\sigma_a(T^4-E), f_2=\sigma_a(E-T^4).$$

We directly design an iteration sequence for the time discretized nonlinear PDEs. Introduce the approximations to D^{n+1}, κ^{n+1}, f_1^{n+1}, f_2^{n+1} by the first order Taylor expansion as follows

$$\begin{aligned}D(E^{n+1},T^{n+1},\nabla E^{n+1})\approx& D(E^{(s)},T^{(s)},\nabla E^{(s)})+\frac{\partial D}{\partial E}(E^{(s)},T^{(s)},\nabla E^{(s)})(E^{(s+1)}-E^{(s)})\\&+\frac{\partial D}{\partial T}(E^{(s)},T^{(s)},\nabla E^{(s)})(T^{(s+1)}-T^{(s)})\\&+\sum_{i=1}^{2}\frac{\partial D}{\partial\partial_i E}(E^{(s)},T^{(s)},\nabla E^{(s)})(\partial_i E^{(s+1)}-\partial_i E^{(s)}),\end{aligned} \tag{3.2a}$$

$$\kappa(T^{n+1})\approx\kappa(T^{(s)})+\frac{\partial\kappa}{\partial T}(T^{(s)})(T^{(s+1)}-T^{(s)}), \tag{3.2b}$$

$$\begin{aligned}f_1(E^{n+1},T^{n+1})\approx& f_1(E^{(s)},T^{(s)})+\frac{\partial f_1}{\partial E}(E^{(s)},T^{(s)})(E^{(s+1)}-E^{(s)})+\frac{\partial f_1}{\partial T}(E^{(s)},T^{(s)})(T^{(s+1)}-T^{(s)})\\=&z^3T^{(s+1)}-\sigma_a^{(s)}E^{(s+1)}+3z^3(T^{(s)})^{-4}E^{(s)}(T^{(s+1)}-T^{(s)}),\end{aligned} \tag{3.2c}$$

$$\begin{aligned}f_2(E^{n+1},T^{n+1})\approx& f_2(E^{(s)},T^{(s)})+\frac{\partial f_2}{\partial E}(E^{(s)},T^{(s)})(E^{(s+1)}-E^{(s)})+\frac{\partial f_2}{\partial T}(E^{(s)},T^{(s)})(T^{(s+1)}-T^{(s)})\\=&-z^3T^{(s+1)}+\sigma_a^{(s)}E^{(s+1)}-3z^3(T^{(s)})^{-4}E^{(s)}(T^{(s+1)}-T^{(s)}).\end{aligned} \tag{3.2d}$$

Inserting (3.2a)-(3.2d) into (3.1a) and (3.1b), and linearizing the resulting equations, we can get the following iteration sequence of linearized coupled PDEs

$$\begin{aligned}&\frac{E^{(s+1)}-E^n}{\Delta t}-\nabla\cdot\Big(D(E^{(s)},T^{(s)},\nabla E^{(s)})\nabla E^{(s+1)}\Big)-(z^3T^{(s+1)}-\sigma_a^{(s)}E^{(s+1)})\\&-\nabla\cdot\Big(D'_E(E^{(s)},T^{(s)},\nabla E^{(s)})(E^{(s+1)}-E^{(s)})\nabla E^{(s)}\Big)\\&-\nabla\cdot\Big(D'_T(E^{(s)},T^{(s)},\nabla E^{(s)})(T^{(s+1)}-T^{(s)})\nabla E^{(s)}\Big)\\&-\nabla\cdot\left(\sum_{i=1}^{2}D'_{\partial_i E}(E^{(s)},T^{(s)},\nabla E^{(s)})(\partial_i E^{(s+1)}-\partial_i E^{(s)})\nabla E^{(s)}\right)\\&-3z^3(T^{(s)})^{-4}E^{(s)}(T^{(s+1)}-T^{(s)})=0,\qquad s=1,2,\cdots,\end{aligned} \tag{3.3a}$$

$$\frac{T^{(s+1)}-T^n}{\Delta t}-\nabla\cdot\Big(\kappa(T^{(s)})\nabla T^{(s+1)}\Big)+(z^3T^{(s+1)}-\sigma_a^{(s)}E^{(s+1)})$$
$$-\nabla\cdot\Big(\kappa_T'(T^{(s)})(T^{(s+1)}-T^{(s)})\nabla T^{(s)}\Big)$$
$$+3z^3(T^{(s)})^{-4}E^{(s)}(T^{(s+1)}-T^{(s)})=0,\qquad s=1,2,\cdots, \tag{3.3b}$$

where $E^{(0)}=E^n$, $T^{(0)}=T^n$, and

$$D_E'=\frac{\partial D}{\partial E}=(3z^3/T^3+|\nabla E|/E)^{-2}|\nabla E|E^{-2},$$
$$D_T'=\frac{\partial D}{\partial T}=9(3z^3/T^3+|\nabla E|/E)^{-2}z^3T^{-4},$$
$$D_{\partial_i E}'=\frac{\partial D}{\partial\partial_i E}=-(3z^3/T^3+|\nabla E|/E)^{-2}|\nabla E|^{-1}E^{-1}\partial_i E,$$
$$\kappa_T'=\frac{\partial\kappa}{\partial T}=2.5c_0T^{3/2}.$$

Note that the time level index, $n+1$, has been omitted for simplicity. For one dimensional single nonlinear parabolic equation $u_t-(\kappa(u)u_x)_x=f(u)$ with Dirichlet boundary conditions, we can prove that as $s\to\infty$, the solution of the iterative sequence $u^{(s)}$, given by the above linearization procedure, can converge to the solution of the time discretization equation, u^{n+1}, quadratically. Refer to [17, 18] for more details about convergence analysis.

The next step is to design spatial discretization for the system of convection-diffusion equations (3.3a) and (3.3b), and it follows that we get a Picard-Newton nonlinear iterative scheme. Different spatial discretization can lead to different nonlinear iterative scheme. The properties of coefficient matrix resulted from Newton linearization in DL are often worse than those from Picard linearization. In our LD approach we can particularly pay more attention to appropriate spatial discretization schemes for Newton correction terms, which haven't been considered in traditional cell-centered discretization schemes in the DL approach.

3.2 Spatial discretization of the resulting linear PDEs

In the linearized PDEs (3.3a), (3.3b), the spacial derivative terms include not only the original diffusion terms, but also Newton correction terms. The main diffusion terms (denoted by $\mathcal{F}_{K,e}$) are

$$-\nabla\cdot\Big(D(E^{(s)},T^{(s)},\nabla E^{(s)})\nabla E^{(s+1)}\Big), \tag{3.4a}$$
$$-\nabla\cdot\Big(\kappa(T^{(s)})\nabla E^{(s+1)}\Big), \tag{3.4b}$$

which appear also in Picard linearization, and can be discretized by the same discretization scheme as that used in the spacial discretization of the implicitly semi-discrete PDEs (3.1a) and (3.1b), so that the resulting scheme is usually Picard nonlinear iterative scheme.

The Newton correction terms, which arise from Newton linearization, consist of two parts. One part is the first order differential with respect to $E^{(s+1)}$ or $T^{(s+1)}$ (denoted by $\mathfrak{F}_{K,e}$) as follows

$$-\nabla\cdot\Big(D'_E(E^{(s)},T^{(s)},\nabla E^{(s)})(E^{(s+1)}-E^{(s)})\nabla E^{(s)}\Big), \tag{3.5a}$$

$$-\nabla\cdot\Big(D'_T(E^{(s)},T^{(s)},\nabla E^{(s)})(T^{(s+1)}-T^{(s)})\nabla E^{(s)}\Big), \tag{3.5b}$$

$$-\nabla\cdot\Big(\kappa'_T(T^{(s)})(T^{(s+1)}-T^{(s)})\nabla T^{(s)}\Big). \tag{3.5c}$$

The other part is the second order differential with respect to $E^{(s+1)}$, which comes from the partial derivative of flux-limited diffusion coefficient with respect to ∇E (denoted by $F_{K,e}$) as follows

$$-\nabla\cdot\left(\sum_{i=1}^{2}D'_{\partial_i E}(E^{(s)},T^{(s)},\nabla E^{(s)})(\partial_i E^{(s+1)}-\partial_i E^{(s)})\nabla E^{(s)}\right). \tag{3.6}$$

We have found that (3.6) can be written as diffusion term with tensor coefficient:

$$-\nabla\cdot\Big(\Lambda(E^{(s)},T^{(s)},\nabla E^{(s)})(\nabla E^{(s+1)}-\nabla E^{(s)})\Big), \tag{3.7}$$

where

$$\Lambda^{(s)}=\begin{pmatrix} D'^{(s)}_{\partial_1 E}\partial_1 E^{(s)} & D'^{(s)}_{\partial_2 E}\partial_1 E^{(s)} \\ D'^{(s)}_{\partial_1 E}\partial_2 E^{(s)} & D'^{(s)}_{\partial_2 E}\partial_2 E^{(s)} \end{pmatrix}. \tag{3.8}$$

We need to design the appropriate discretization schemes for these terms respectively, in particular when multimaterial problems are solved.

For expressing discrete flux, we divide spatial domain by rectangular grid and denote cell by K or L instead of (i,j), and with each cell K we associate one cell center denoted also by K. The area of the cell K is $m(K)$. Denote the vertices by A and B, and the cell side by e (see Fig. 1). If the cell side e is a common edge of cells K and L, and its vertices are A

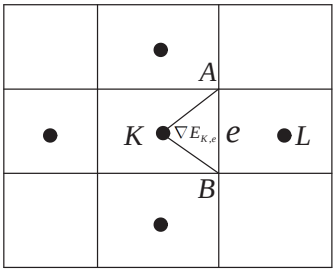

Figure 1: Some geometric notations.

and B, then we denote $e=K|L=BA$. I denotes the midpoint of $e=BA$. Let $\varepsilon(K)$ be the set of all cells, ∂K the set of all the cell sides of cell K, $\vec{n}_{K,e}$ outward unit normal to e on K and $\vec{\tau}_{AB}=\vec{AB}/|AB|$. Sometimes we will omit the iteration index (s) or $(s+1)$ for simplicity when no confusion occurs.

3.2.1 Flux approximation of the diffusion terms (3.4a)-(3.4b)

We take the term (3.4a) for example to introduce the finite volume discretization. By integrating (3.4a) over the cell K and using the Green formula, one obtains

$$\sum_{e\in\partial K}\mathcal{F}_{K,e}=-\sum_{e\in\partial K}\int_e D^{(s)}\nabla E^{(s+1)}.\vec{n}_{K,e}\,dl,$$

where $\mathcal{F}_{K,e}$ is the normal face flux on the edge e, and we can get an approximation of $\mathcal{F}_{K,e}$

$$\mathcal{F}_{K,e}\approx-\tau_{K,e}^{(s)}(E_I^{(s+1)}-E_K^{(s+1)}), \tag{3.9}$$

where

$$\tau_{K,e}^{(s)}=\frac{D_{K,e}^{(s)}|AB|}{|IK|}. \tag{3.10}$$

Similarly, we have

$$\mathcal{F}_{L,e}\approx-\tau_{L,e}^{(s)}(E_I^{(s+1)}-E_L^{(s+1)}), \tag{3.11}$$

where

$$\tau_{L,e}^{(s)}=\frac{D_{L,e}^{(s)}|AB|}{|IL|}.$$

By the continuity of the normal flux component

$$\mathcal{F}_{K,e}=-\mathcal{F}_{L,e}, \tag{3.12}$$

we can obtain

$$E_I^{(s+1)}=\frac{1}{\tau_{K,e}^{(s)}+\tau_{L,e}^{(s)}}(\tau_{K,e}^{(s)}E_K^{(s+1)}+\tau_{L,e}^{(s)}E_L^{(s+1)}). \tag{3.13}$$

Substitute (3.13) into (3.9) to obtain

$$\mathcal{F}_{K,e}\approx-\tau_e^{(s)}(E_L^{(s+1)}-E_K^{(s+1)}), \tag{3.14}$$

where

$$\tau_e^{(s)}=\frac{\tau_{K,e}^{(s)}\tau_{L,e}^{(s)}}{\tau_{K,e}^{(s)}+\tau_{L,e}^{(s)}}.$$

$D_{K,e}^{(s)}$ in (3.10) is defined as the limit of the diffusion coefficient to the edge-center in the cell K. Notice that the energy E and temperature T are continuous and can be defined on the

edge, while the gradient ∇E is discontinuous possibly across edge e so that it is defined on the triangle formed by the cell center and the edge e, and the function of material z is defined on the cell center. So we define

$$D_{K,e}=\frac{1}{\left(3z_K^3/T_e^3+|\nabla E|_{K,e}/E_e\right)}, \tag{3.15}$$

where E_e and T_e are evaluated at the cell-edge, and given by $E_e=\frac{1}{2}(E_A+E_B)$, and $T_e=\frac{1}{2}(T_A+T_B)$. The cell vertex unknowns E_A,E_B are approximated by the cell-centered unknowns. We use the method in [13] to eliminate the cell vertex unknowns. $\nabla E_{K,e}$ is defined as a discrete approximation to the gradient of E on the triangle ΔKBA [14]:

$$\nabla E_{K,e}=\frac{1}{\sin\theta_K}\left[\frac{E_K-E_A}{|KA|}\vec{n}_{KB}+\frac{E_K-E_B}{|KB|}\vec{n}_{AK}\right], \tag{3.16}$$

where θ_K is the angle between KB and KA, $\vec{n}_{KB}$ and $\vec{n}_{AK}$ are outward unit normal to edges $\vec{KB}$ and $\vec{AK}$ respectively on ΔKBA.

The derivation of the flux approximation of the term (3.4b) is the same.

3.2.2 Flux approximation of Newton correction terms (3.5a)-(3.5c)

The derivation of flux approximation for Newton correction terms (3.5a)-(3.5c) is similar to (3.14). Noticing that E is continuous, we can obtain flux approximation of (3.5a) on the edge e

$$\mathfrak{F}_{K,e}\approx-\gamma_e^{(s)}[E_L^{(s)}-E_K^{(s)}](E_e^{(s+1)}-E_e^{(s)}), \tag{3.17}$$

where

$$\gamma_e^{(s)}=\frac{\gamma_{K,e}^{(s)}\gamma_{L,e}^{(s)}}{\gamma_{K,e}^{(s)}+\gamma_{L,e}^{(s)}},\qquad \gamma_{K,e}^{(s)}=\frac{|AB|\frac{\partial D_{K,e}^{(s)}}{\partial E}}{|KI|}.$$

In (3.17), $E^{(s+1)}-E^{(s)}$ is approximated on the cell-edge, so the resulting nonlinear iterative scheme is named as edge-centered Picard-Newton (P-NC). Usually the dominance of the diagonal elements of the coefficient matrix from the Picard iterative scheme is better than that from P-NC.

It is well-known that the efficiency for iteratively solving linear system of algebraic equations depends on the properties of coefficient matrix. So we consider a new nonlinear iterative method to avoid coefficient matrix corrupting. Note that Newton correction terms (3.5a)-(3.5c) are convective terms. In order to improve the properties of coefficient matrix and then save CPU time, simultaneously, reflect the character of convective term, we propose an upwind discretization of (3.5a). Denote

$$L_{1,e}=\gamma_e^{(s)}(E_L^{(s)}-E_K^{(s)}).$$

Then the upwind discretization of (3.5a) is defined as

$$\mathfrak{F}_{K,e}=-\begin{cases} L_{1,e}(E_K^{(s+1)}-E_K^{(s)}), & \text{if } L_{1,e}\leq 0, \\ L_{1,e}(E_L^{(s+1)}-E_L^{(s)}), & \text{if } L_{1,e}>0. \end{cases} \tag{3.18}$$

The treatment of (3.5b) and (3.5c) is similar. The nonlinear iterative scheme using upwind scheme to the discretization of correction terms (3.5a)-(3.5c) is named as upwind Picard-Newton (P-NU). Later, we will give numerical comparison among Picard, P-NC and P-NU iterative schemes.

3.2.3 Flux approximation of Newton correction term (3.6)

We use a nine point scheme [13] to discretize the Newton correction term (3.6). Integrating (3.6) over the cell K, we obtain

$$\sum_{e\in\partial K} F_{K,e}=-\sum_{e\in\partial K}\int_e \nabla E^{(s+1)}\cdot(\Lambda^{(s)})^t\vec{n}_{K,\sigma}dl. \tag{3.19}$$

In the following we shall use the same notation $F_{K,e}$ to represent its discrete counterpart. Notice that there exist scalars α and β depending on Λ, such that

$$(\Lambda^{(s)})^t\vec{n}_{K,e}=-\alpha^{(s)}\vec{\tau}_{BA}+\beta^{(s)}\vec{n}_{K,e}, \tag{3.20}$$

where

$$\alpha^{(s)}=\vec{\tau}_{AB}\cdot((\Lambda^{(s)})^t\vec{n}_{K,e}),\qquad \beta^{(s)}=\vec{n}_{K,e}\cdot((\Lambda^{(s)})^t\vec{n}_{K,e}).$$

Inserting (3.20) into (3.19), we can get the approximation of flux

$$F_{K,e}=-\tau_{\Lambda,K,e}^{(s)}\left[E_I^{(s+1)}-E_K^{(s+1)}-D_{\Lambda,K,e}^{(s)}(E_A^{(s+1)}-E_B^{(s+1)})\right], \tag{3.21}$$

where

$$\tau_{\Lambda,K,e}^{(s)}=\frac{|AB|\beta_{K,e}^{(s)}}{|IK|},\qquad D_{\Lambda,K,e}^{(s)}=\frac{|IK|\alpha_{K,e}^{(s)}}{|AB|\beta_{K,e}^{(s)}}.$$

Similarly,

$$F_{L,e}=-\tau_{\Lambda,L,e}^{(s)}\left[E_I^{(s+1)}-E_L^{(s+1)}-D_{\Lambda,L,e}^{(s)}(E_B^{(s+1)}-E_A^{(s+1)})\right]. \tag{3.22}$$

By the continuity of the normal flux component, we can eliminate $E_I^{(s+1)}$ and get the following approximate flux:

$$F_{K,e}=-\tau_{\Lambda,e}^{(s)}\left[E_L^{(s+1)}-E_K^{(s+1)}-D_{\Lambda,e}^{(s)}(E_A^{(s+1)}-E_B^{(s+1)})\right], \tag{3.23}$$

where

$$\tau_{\Lambda,e}^{(s)}=\frac{|AB|}{\frac{|IK|}{\beta_{K,e}^{(s)}}+\frac{|IL|}{\beta_{L,e}^{(s)}}},\qquad D_{\Lambda,e}^{(s)}=\frac{|IK|\alpha_{K,e}^{(s)}}{|AB|\beta_{K,e}^{(s)}}+\frac{|IL|\alpha_{L,e}^{(s)}}{|AB|\beta_{L,e}^{(s)}}.$$

The cell vertex unknowns E_A, E_B in (3.23) are approximated by the cell-centered unknowns [13]. From (3.23) we can see that for the diffusion term with tensor coefficient, there are nine cells in the stencil even on rectangular grid.

For the detailed analysis of the discretization scheme see also [3, 13, 19].

By now we have introduced the discretization of the main diffusion terms and Newton correction terms. The finite volume scheme for the discretization of (3.3a) on the cell K is

$$\begin{aligned}
&m(K)\frac{E_K^{n+1,(s+1)}-E_K^n}{\Delta t}-(z_K^3T_K^{n+1,(s+1)}-\sigma_{a,K}^{n+1,(s)}E_K^{n+1,(s+1)})m(K)\\
&\quad-\sum_{e\in\partial K}\tau_e^{n+1,(s)}[E_L^{n+1,(s+1)}-E_K^{n+1,(s+1)}]\\
&\quad-\sum_{e\in\partial K}\gamma_e^{n+1,(s)}[E_L^{n+1,(s)}-E_K^{n+1,(s)}](E_{N1}^{n+1,(s+1)}-E_{N1}^{n+1,(s)})\\
&\quad-\sum_{e\in\partial K}\mu_e^{n+1,(s)}[E_L^{n+1,(s)}-E_K^{n+1,(s)}](T_{N1}^{n+1,(s+1)}-T_{N1}^{n+1,(s)})\\
&\quad-\sum_{e\in\partial K}\tau_{\Lambda,e}^{n+1,(s)}\left[E_L^{n+1,(s+1)}-E_K^{n+1,(s+1)}-D_{\Lambda,e}^{n+1,(s)}(E_A^{n+1,(s+1)}-E_B^{n+1,(s+1)})\right]\\
&\quad-3z_K^3(T_K^{n+1,(s)})^{-4}E_K^{n+1,(s)}(T_K^{n+1,(s+1)}-T_K^{n+1,(s)})m(K)=0,\qquad s=1,2,\cdots,
\end{aligned}\tag{3.24}$$

The discrete scheme of (3.3b) on the cell K is

$$\begin{aligned}
&m(K)\frac{T_K^{n+1,(s+1)}-T_K^n}{\Delta t}+(z_K^3T_K^{n+1,(s+1)}-\sigma_{a,K}^{n+1,(s)}E_K^{n+1,(s+1)})m(K)\\
&\quad-\sum_{e\in\partial K}\overset{\sim}{\tau}_e^{n+1,(s)}[T_L^{n+1,(s+1)}-T_K^{n+1,(s+1)}]\\
&\quad-\sum_{e\in\partial K}\overset{\sim}{\mu}_e^{n+1,(s)}[T_L^{n+1,(s)}-T_K^{n+1,(s)}](T_{N1}^{n+1,(s+1)}-T_{N1}^{n+1,(s)})\\
&\quad+3z_K^3(T_K^{n+1,(s)})^{-4}E_K^{n+1,(s)}(T_K^{n+1,(s+1)}-T_K^{n+1,(s)})m(K)=0,\qquad s=1,2,\cdots,
\end{aligned}\tag{3.25}$$

where

$$\begin{aligned}
&\widetilde{\tau}_e^{n+1,(s)}=\frac{\widetilde{\tau}_{K,e}^{n+1,(s)}\widetilde{\tau}_{L,e}^{n+1,(s)}}{\widetilde{\tau}_{K,e}^{n+1,(s)}+\widetilde{\tau}_{L,e}^{n+1,(s)}},\qquad \overset{\sim}{\tau}_{K,e}^{n+1,(s)}=\frac{|AB|}{|KI|}\kappa_{K,e}^{n+1,(s)},\\
&\mu_e^{n+1,(s)}=\frac{\mu_{K,e}^{n+1,(s)}\mu_{L,e}^{n+1,(s)}}{\mu_{K,e}^{n+1,(s)}+\mu_{L,e}^{n+1,(s)}},\qquad \mu_{K,e}^{n+1,(s)}=\frac{|AB|}{|KI|}\frac{\partial D_{K,e}^{n+1,(s)}}{\partial T},\\
&\widetilde{\mu}_e^{n+1,(s)}=\frac{\widetilde{\mu}_{K,e}^{n+1,(s)}\widetilde{\mu}_{L,e}^{n+1,(s)}}{\widetilde{\mu}_{K,e}^{n+1,(s)}+\widetilde{\mu}_{L,e}^{n+1,(s)}},\qquad \widetilde{\mu}_{K,e}^{n+1,(s)}=\frac{|AB|}{|KI|}\frac{\partial\kappa_{K,e}^{n+1,(s)}}{\partial T}.
\end{aligned}$$

The definitions of $\frac{\partial D_{K,e}}{\partial E}$, $\frac{\partial D_{K,e}}{\partial T}$, $\Lambda_{K,e}$, $\kappa_{K,e}$, $\frac{\partial \kappa_{K,e}}{\partial T}$ are similar to (3.15). The terms on the 3rd, 4th lines of (3.24) and the 3rd line of (3.25) are the discretization corresponding to Newton correction terms (3.5a)-(3.5c). If $E^{(s+1)}-E^{(s)}$ and $T^{(s+1)}-T^{(s)}$ are approximated on the cell-edge as (3.17) ($N1=e$), then the resulting nonlinear iterative scheme is named as edge-centered Picard-Newton (P-NC). While they are approximated as (3.18) ($N1=K$ for $L_{1,e}\leq 0$ or L for $L_{1,e}>0$), the resulting nonlinear iterative scheme is named as upwind Picard-Newton (P-NU). In the following we turn to the emphasis of this work that is to investigate the effect of time step control on Picard-Newton nonlinear iterative method.

4 Time step control

Because the radiation diffusion problems exhibit multiple time scale, time step control algorithms are used to increase solution efficiency. Traditionally, the time step control for methods which do not converge nonlinearities within a time step is based on the relative change in E or T, i.e. $\Delta E/E$ or $\Delta T/T$. If one assumes $\Delta E/E>\Delta T/T$, then one can use the time step control method based on $\Delta E/E$. In this paper the relative change in E is defined by

$$\eta^n=\max_{K\in\varepsilon(K)}\left(\frac{|E_K^n-E_K^{n-1}|}{E_K^{n-1/2}}\right), \tag{4.1}$$

where $E_K^{n-1/2}=(E_K^n+E_K^{n-1})/2$. The superscript n is the time index and the subscript K is the space grid cell index. The time step size is adjusted to achieve a prescribed maximum relative change in E (η_{target})

$$\Delta t_{\text{re}}^{n+1}=\Delta t^n\left[\frac{\eta_{\text{target}}}{\eta^n}\right]^{0.5}. \tag{4.2}$$

For convenience we call this method $\Delta E/E$ method. In this paper, the final time step for $\Delta E/E$ method is chosen to limit the rate of growth by 10%

$$\Delta t^{n+1}=\min(1.1\Delta t^n,\Delta t_{\text{re}}^{n+1},\Delta t_{\text{max}}^{n+1}). \tag{4.3}$$

Therefore, given the initial time step size and "target maximum $\Delta E/E$", the time step sizes can be computed from (4.3).

Another kind of time step control method is a thermal front CFL approach which is originally presented in [12]. The basic idea is to estimate the dominant wave propagation speed in the problem. In one dimension this is the ratio of temporal to spatial derivative of the variable. Briefly, one can assume that the following hyperbolic PDE

$$\frac{\partial E}{\partial t}+v_{\text{rad}}\frac{\partial E}{\partial x}=0 \tag{4.4}$$

models the problem. One can use the numerical approximation of Eq. (4.4) and get

$$v_{\text{rad}} = -\frac{||\triangle E/\triangle t||_1}{||\triangle E/\triangle x||_1}.$$

Then the time step size is adjusted to achieve a target CFL number

$$\Delta t^{n+1}_{CFL} = \frac{CFL\,\|\,\Delta x\,\|}{v^n_{\text{rad}}}. \tag{4.5}$$

In this paper, we let

$$\|\,\Delta x\,\| = \sqrt{2m(K)},$$
$$v^n_{\text{rad}} = \frac{\sum\limits_{K\in\varepsilon(K)} |E^n_k - E^{n-1}_K|/\Delta t^n}{\sum\limits_{K\in\varepsilon(K)} |\nabla E^n_K|}.$$

This is from [6], and the differences are that we include the cells near the boundaries in the sum, which are excluded in [6], and the approximation to $|\nabla E^n_K|$ is different. We use (3.16) to discretize the gradient.

Similar to (4.3), the final time step based on CFL method employed in this paper is defined by

$$\Delta t^{n+1} = \min(1.1\Delta t^n, \Delta t^{n+1}_{CFL}, \Delta t^{n+1}_{\max}). \tag{4.6}$$

Given the initial time step size and "target CFL", the time step sizes can be computed from (4.6).

5 Numerical results

In this section, we present numerical results from Picard and P-N methods. Time step convergence studies are performed and CPU time will be compared for desired accuracy.

The test problems are solved on the domain $0 \le x \le 1$ and $0 \le y \le 1$. The boundary condition for the radiation equation at the left boundary ($x = 0, 0 \le y \le 1$) is

$$\frac{1}{4}E - \frac{1}{2D}\frac{\partial E}{\partial x} = 1,$$

and at the right boundary ($x = 1, 0 \le y \le 1$) is

$$\frac{1}{4}E + \frac{1}{2D}\frac{\partial E}{\partial x} = 0.$$

At the top and bottom boundaries ($y = 0, y = 1, 0 \le x \le 1$) we use

$$\frac{\partial E}{\partial y} = 0.$$

For the material conduction equation, all four walls are insulated:

$$\left.\frac{\partial T}{\partial x}\right|_{x=0} = \left.\frac{\partial T}{\partial x}\right|_{x=1} = \left.\frac{\partial T}{\partial y}\right|_{y=0} = \left.\frac{\partial T}{\partial y}\right|_{y=1} = 0.$$

The value for z is 1 everywhere except in the rectangle defined by $\frac{1}{3} \le x \le \frac{2}{3}$, and $\frac{1}{3} \le y \le \frac{2}{3}$, where $z = z_{high}$. The initial energy is $E_0 = 1 \times 10^{-5}$, and the initial material temperature is $T_0 = (E_0)^{1/4}$. The problem is initialized to equilibrium with radiation temperature being equal to material temperature. The simulation runs for three units of time ($t_{end} = 3$).

In this paper, we use Picard and P-N methods to solve the nonlinear system and GMRES method with ILUT preconditioner to solve the linear system. The program is written in FORTRAN, and run on a windows system. The nonlinear convergence tolerance within a time step is defined as $||\mathbf{F}(\mathbf{u}^{s+1})|| \le 1 \times 10^{-6}$ and $||\mathbf{F}(\mathbf{u}^{s+1})|| \le 1 \times 10^{-2}||\mathbf{F}(\mathbf{u}^0)||$, where $\mathbf{F}(\mathbf{u})$ is the residual function. The expression of the residual function for P-N is the same as that for Picard. We will take the default maximum number of nonlinear iterations to be 20, and will also consider other numbers of nonlinear iterations for Picard method.

Since Picard is a proven solver and we will compare P-N with Picard, a base solution is computed using Picard for the sake of impartiality. The base solution is computed with a time step of $\eta_{\text{target}} = 0.005$ and the nonlinear convergence tolerance of $||\mathbf{F}(\mathbf{u}^{s+1})|| \le 1 \times 10^{-8}$ and $||\mathbf{F}(\mathbf{u}^{s+1})|| \le 1 \times 10^{-4}||\mathbf{F}(\mathbf{u}^0)||$. For the time-step convergence study, radiation temperature is the variable used to compute the L_2 norm between the base and the computed solution:

$$L_2(Error) = \sqrt{\sum_{K \in \varepsilon(K)} (T_r(K) - T_r^b(K))^2 m(K)},$$

where T_r^b is the radiation temperature from the base solution.

The test problems are chosen to represent more nonlinear problem with $z_{high} = 10$ and less nonlinear problems with $z_{high} = 5$ and $z_{high} = 2.5$.

5.1 More nonlinear problem

In the more nonlinear problem, $z_{high} = 10$. The material is initially cold, and there is the constant energy flux into the left wall. Therefore, the initial constant energy flux is being imparted into the initially cold material. At $t = 3$, it reaches the state shown in Fig. 2. The energy flux spreads from the left wall until it has engulfed the obstacle of high z material. Fig. 2 shows that the high z material prevents the radiation from entering the obstacle, and the center of the obstacle is still cold.

5.1.1 Results for time step of $\Delta E/E$ method

The first result to be presented is a time step convergence study with the $\Delta E/E$ method to control the time step. The change in timescales from the $\Delta E/E$ method can be seen in

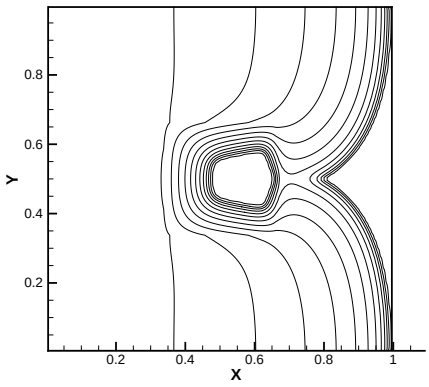

Figure 2: Contours of material temperature at $t=3$.

Figs. 3 and 4. These figures show the time step sizes from P-NU and Picard methods with $\eta_{\text{target}}=0.2$ in (4.3), respectively. The time step sizes from the two methods are almost the same except around $t=1.6$ and $t=2.6$, where time step sizes from Picard decrease significantly.

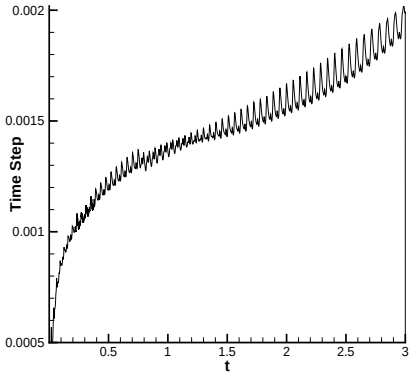

Figure 3: Time steps for P-NU with $\eta_{\text{target}}=0.2$ for 60×60 problem.

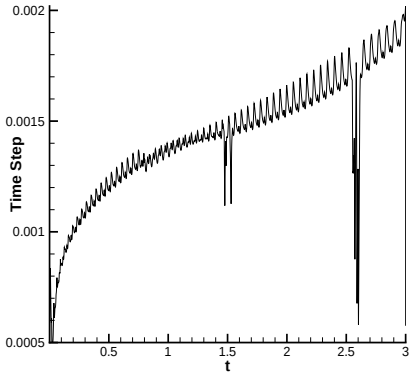

Figure 4: Time steps for Picard with $\eta_{\text{target}}=0.2$ for 60×60 problem.

Fig. 5 shows a time step convergence study for Picard and P-N methods on 60×60 grid with $0.1\leq\eta_{\text{target}}\leq1$, where η_{target} is depicted on the x axis and L_2 norm of error on the y axis. The dash-dot line represents a zero-order line, which shows zero-order time accuracy for Picard method. The dotted line represents a first-order line, which shows first-order time accuracy for P-N. The designed first-order time convergence of P-N is apparent, however, Picard does not achieve the first-order time convergence and produces almost 0th order time convergence. This is because Picard method fails to converge within 20 nonlinear iterations when $0.1\leq\eta_{\text{target}}\leq1$. Even we set a maximum of nonlinear iterations by 100, Picard method fails to converge under this loose tolerance.

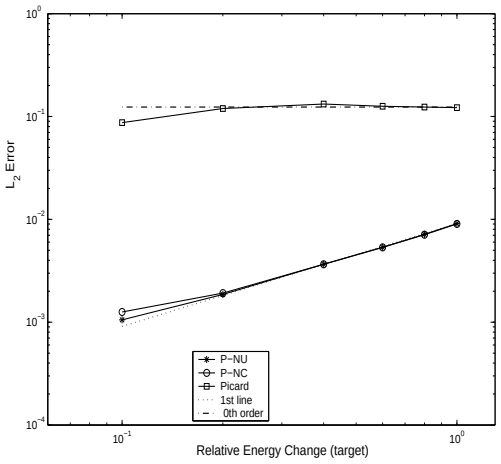

Figure 5: Time step convergence study with $0.1 \leq \eta_{\text{target}} \leq 1$ for 60×60 problem.

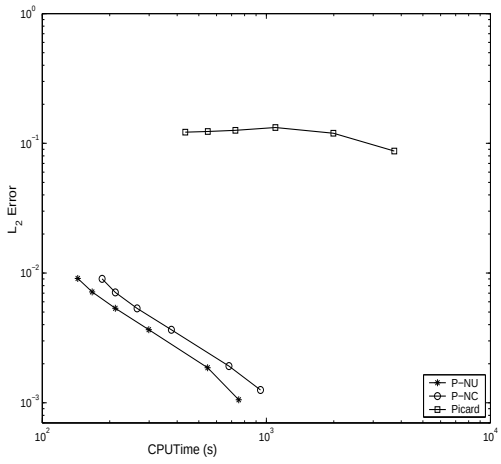

Figure 6: Efficacy with $0.1 \leq \eta_{\text{target}} \leq 1$ for 60×60 problem.

Therefore the errors build up and the solution drifts away from the correct solution. For a given η_{target}, the L_2 error of Picard is an order of magnitude larger than that of P-N, and P-NU is slightly more accurate than P-NC.

Fig. 6 shows efficacy for Picard and P-N methods with $0.1 \leq \eta_{\text{target}} \leq 1$ on 60×60 grid. In this plot, the same error values shown in Fig. 5 are now plotted against the CPU time of the numerical computation. An efficacy plot can directly show which method gives the highest level of accuracy for the smallest amount of CPU time. As was expected, P-NU is the most effective. The properties of coefficient matrix of linear system from P-NU are better than those from P-NC, so the number of linear iterations of P-NU should be less than that of P-NC. Table 1 compares average nonlinear iterations per time step ($\frac{nonlinear}{time-step}$), average linear iterations per nonlinear iteration step ($\frac{linear}{nonlinear}$) and average linear iterations per time step ($\frac{linear}{time-step}$). From the table we can see that the numbers of linear iterations ($\frac{linear}{nonlinear}$) from P-NU with different η_{target} are less than those from P-NC. Simultaneously, good properties of linear system result in slight decrease on nonlinear iterations.

Table 1: Algorithm performances as a function of time step (η_{target}) at $t = 3$.

Method and η_{target}	$\frac{nonlinear}{time-step}$	$\frac{linear}{nonlinear}$	$\frac{linear}{time-step}$
P-NU, 0.2	1.852	3.264	6.045
P-NC, 0.2	1.856	3.282	6.092
P-NU, 0.6	2.272	3.516	7.988
P-NC, 0.6	2.273	3.733	8.487
P-NU, 1.0	2.837	3.691	10.472
P-NC, 1.0	2.851	3.863	11.012

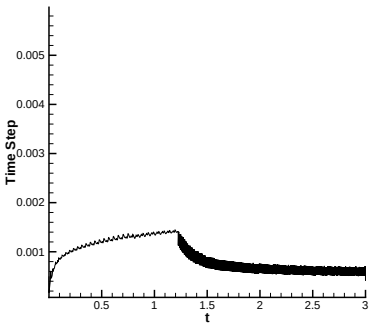

Figure 7: Time steps from Picard method with time step control (5.2) of $\eta_{\text{target}} = 0.2$ for 60×60 problem.

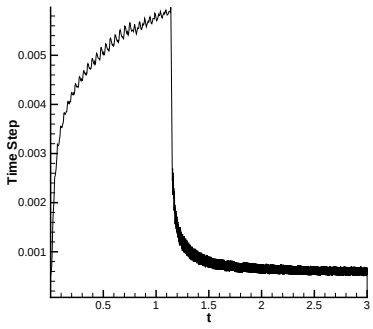

Figure 8: Time steps from Picard method with time step control (5.2) of $\eta_{\text{target}} = 0.8$ for 60×60 problem.

When $0.1 \leq \eta_{\text{target}} \leq 1$ and the maximum number of nonlinear iterations is set by 20, Picard method can not converge at all time levels under the time step control (4.3). To ensure the convergence of Picard method, we set a maximum of nonlinear iterations by 50, and once the number of nonlinear iterations is more than 20 we cut down the time step size by 20%:

$$\Delta t^{n+1}_{\text{Picard}} = 0.8\Delta t^n, \tag{5.1}$$

and the final time step size is computed by

$$\Delta t^{n+1} = \min(\Delta t^{n+1}_{\text{re}}, \Delta t^{n+1}_{\text{Picard}}). \tag{5.2}$$

In this case, about after $t = 1.2$, time step size is mainly determined by (5.1) instead of relative change in E , and time step sizes from different η_{target} are almost the same. The time step sizes from $\eta_{\text{target}} = 0.2$ and $\eta_{\text{target}} = 0.8$ are shown in Figs. 7 and 8 respectively.

In order to compare accuracy and efficiency of P-N with Picard in the case that these methods are all convergent, firstly we solve the more nonlinear problem by Picard with time step determined by (5.2), and keep the time step sizes record; then use the time step sizes recorded to solve the same problem by P-N. Fig. 9 shows an efficacy plot for

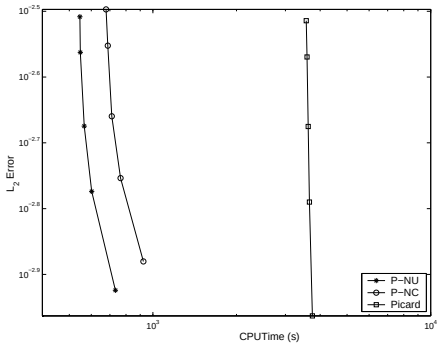

Figure 9: Efficacy of Picard and P-N methods with the same time step size.

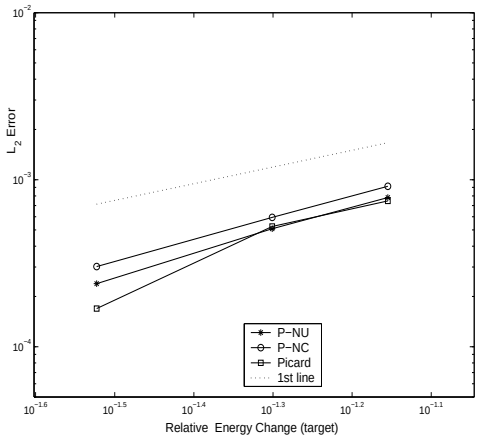

Figure 10: Time step convergence study with $0.03 \leq \eta_{\text{target}} \leq 0.07$ for 60×60 problem.

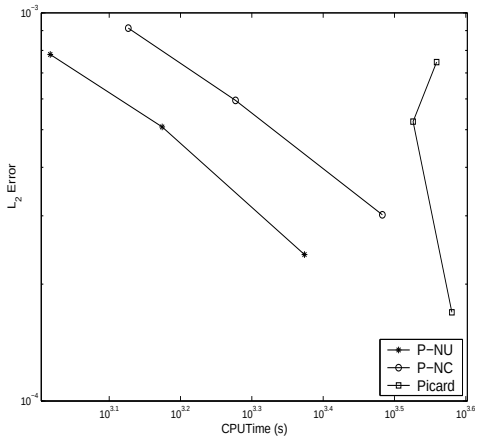

Figure 11: Efficacy with $0.03 \leq \eta_{\text{target}} \leq 0.07$ for 60×60 problem.

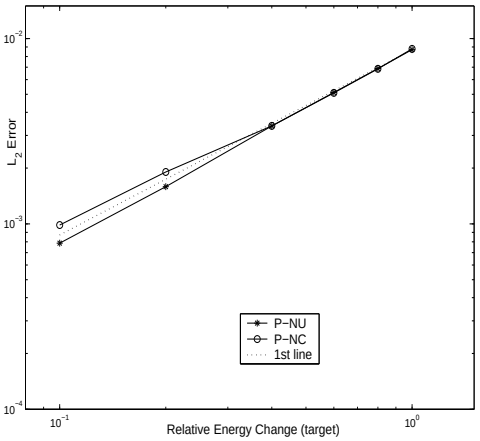

Figure 12: Time step convergence study with $0.1 \leq \eta_{\text{target}} \leq 1$ for 120×120 problem.

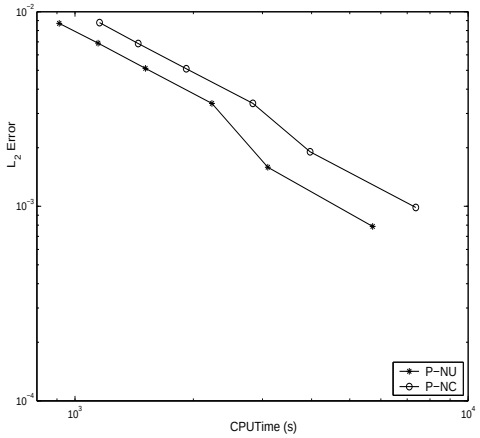

Figure 13: Efficacy with $0.1 \leq \eta_{\text{target}} \leq 1$ for 120×120 problem.

the three schemes with the same time step sizes obtained from Picard method. P-NU produces the same level of accuracy for the least CPU time.

When time step is determined by $\Delta E/E$ method (4.3) with smaller η_{target}, Picard can converge within 20 nonlinear iterations. Fig. 10 shows a time step convergence study for Picard and P-N methods on 60×60 grid with $0.03 \leq \eta_{\text{target}} \leq 0.07$. The dotted line represents a first-order line. Picard is slightly more accurate than P-N when η_{target} is small. The slopes for P-N are very close to the expected value of 1, while Picard does not have straight lines on the convergence plot. Fig. 11 shows an efficacy plot corresponding to Fig. 10. P-NU is still the most effective.

Fig. 12 shows a time step convergence study for P-N on 120×120 grid. P-N schemes achieve the 1st order convergence. Fig. 13 shows the corresponding efficacy plot. The

two plots show that P-NU is still more accurate and efficacious as the mesh is refined. Under the same time step, Picard still fails to converge within 20 iterations.

5.1.2 Results for time step of CFL method

In this section, the thermal wave CFL method is used to control the time step. Table 2 compares the CFL number with the corresponding average value of η^n for selected runs.

Table 2: A comparison of CFL number and corresponding η.

CFL	0.01	0.02	0.04	0.08	0.1
η	0.1839	0.3603	0.6766	1.1332	1.2770

Fig. 14 shows the time step sizes from P-NU with $CFL=0.01$ in (4.6).

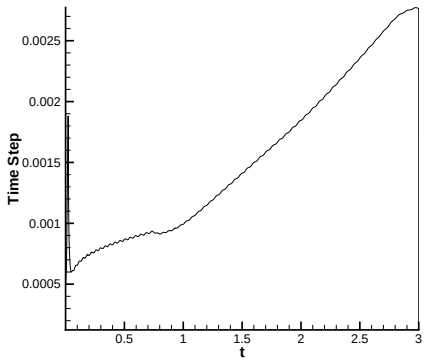

Figure 14: Time steps for P-NU with $CFL=0.01$ for 60×60 problem.

Fig. 15 shows a time step convergence study for P-N on two grids (60×60, 120×120), where L_2 norm of error (on the y axis) is plotted as a function of CFL number (on the x axis). The dotted line represents a first-order line. P-NU is slightly more accurate than P-NC. Fig. 16 shows an efficacy plot, which illustrates that P-NU is more effective.

Similar to the results in the above section, Picard method fails to converge within 20 nonlinear iterations with CFL method (4.6) of the same target CFL numbers used in Figs. 15 and 16. In order to obtain convergent solution for Picard method, we introduce time step control (5.1) and the final time step size is rewritten by

$$\Delta t^{n+1}=\min(\Delta t^{n+1}_{CFL},\Delta t^{n+1}_{\text{Picard}}). \tag{5.3}$$

Table 3 presents comparisons for various methods and target CFL numbers. In Table 3, Picard-(4.6) and P-NU-(4.6) denote Picard and P-NU with time step control (4.6) respectively, and Picard-(5.3) denotes Picard with time step control (5.3). P-NU-(5.3) denotes P-NU with the same time step sizes as what is obtained by (5.3) from Picard. P-NU is more efficacious than Picard under the two time step control methods.

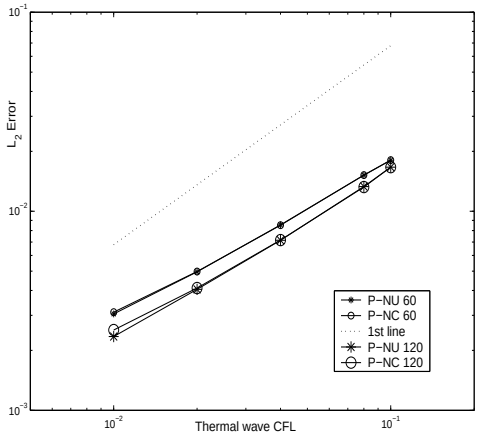

Figure 15: Time step convergence study of P-N with CFL method.

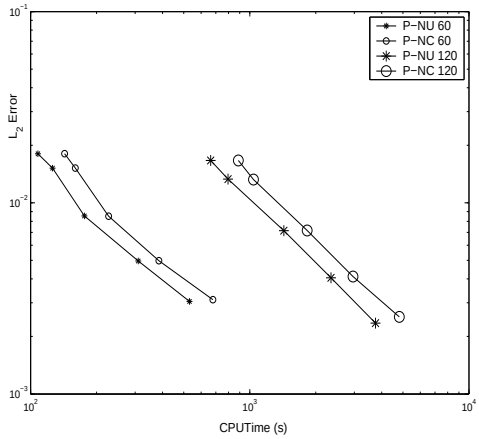

Figure 16: Efficacy of P-N with CFL method.

Table 3: Algorithm performances as a function of CFL at $t=3$.

Method and CFL	Num.of time step	$\frac{nonlinear}{time-step}$	$\frac{linear}{time-step}$	L_2 error	CPU
P-NU-(4.6), 0.02	1196	2.02	6.77	4.96d-3	310
P-NU-(4.6), 0.08	324	3.44	13.31	1.52d-2	126
P-NU-(5.3), 0.02	3368	1.24	3.74	2.03d-3	633
P-NU-(5.3), 0.08	2877	1.21	3.73	3.20d-3	536
Picard-(4.6), 0.02	1125	10.14	25.51	1.2d-1	896
Picard-(4.6), 0.08	300	11.37	33.10	9.8d-2	248
Picard-(5.3), 0.02	3368	15.14	38.77	1.92d-3	3755
Picard-(5.3), 0.08	2877	17.28	44.34	3.16d-3	3524

5.2 Less nonlinear problems

In the less nonlinear problem, $z_{high}=5$ and $z_{high}=2.5$. From the foregoing statement, we can see that P-NU is better than P-NC, so we only compare Picard with P-NU in this section. Fig. 17 shows a time step convergence study for Picard and P-NU methods with $z_{high}=5$ on 60×60 grid. P-NU achieves the 1st order time convergence. Picard still fails to converge within 20 iterations for the largest $\eta_{\text{target}}(=0.8)$ in Fig. 17. When $\eta_{\text{target}}\leq0.6$, Picard achieves the 1st order time convergence, which cannot be obtained from the more nonlinear problem. Fig. 18 shows the corresponding efficacy plot. From the plot, it is clear that for Picard method, the CPU times corresponding to the largest and the second largest errors are abnormal, and they are larger than the CPU times corresponding to the smaller errors. This is because the nonlinear iterations increase abruptly when the time step is enlarged.

Fig. 19 shows a time step convergence study for Picard and P-NU methods with $z_{high}=2.5$ on 60×60 grid. Both of the two methods achieve the 1st order time convergence. Fig. 20 shows the corresponding efficacy plot. The difference in efficiency between

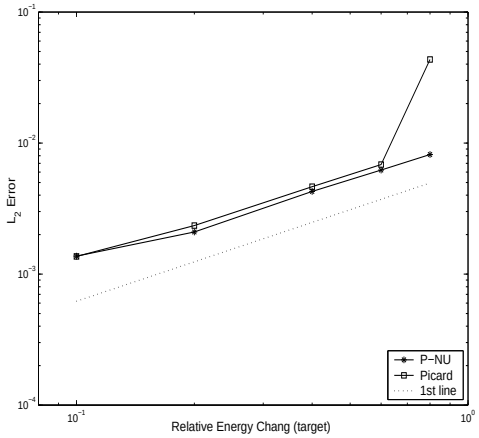

Figure 17: Time step convergence study with $\Delta E/E$ method for $z_{high}=5$.

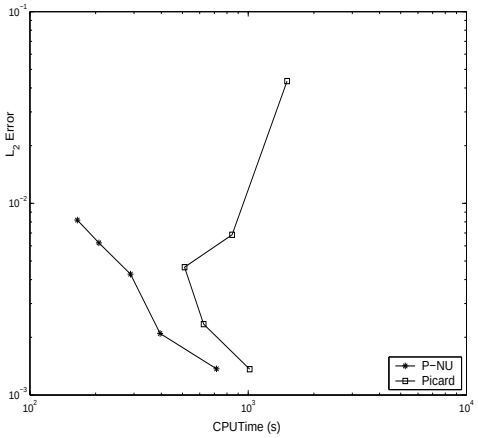

Figure 18: Efficacy with $\Delta E/E$ method for $z_{high}=5$.

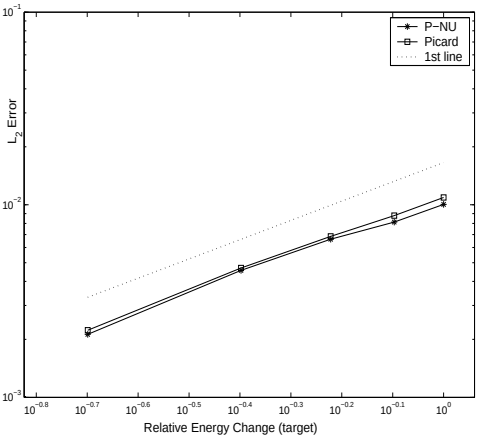

Figure 19: Time step convergence study with $\Delta E/E$ method for $z_{high}=2.5$.

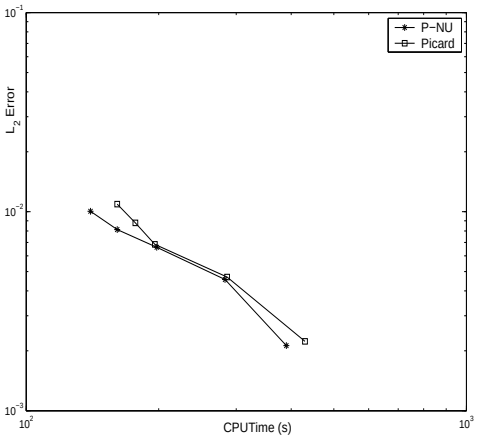

Figure 20: Efficacy with $\Delta E/E$ method for $z_{high}=2.5$.

Picard and P-NU goes down, and Picard method behaves well as the problem is easy to be simulated.

6 Conclusion

A detailed computational cost and efficiency comparison are presented for Picard and P-N methods with time step control to solve multimaterial non-equilibrium radiation diffusion problem. By using two time step control methods, our P-N method can yield high quality numerical results with reduced cost. This is accomplished by running the simulation at a larger time step, while maintaining accuracy through converging the non-

linearities within less iterations. For our P-N, it does not need to introduce more restricted time steps than that used for Picard. The behavior of Picard method becomes ruin as the test problems become more nonlinear, while the good performance of P-NU still remains. The properties of coefficient matrix of linear system from P-NU are better than those from P-NC, so P-NU is the most effective among Picard, P-NC and P-NU.

Acknowledgments

This work was partially supported by the Basic Research Project of National Defence (B1520110011), the Foundation of CAEP (2010A0202010), and the Foundation of National Key Laboratory of Science and Technology on Computational Physics. The authors thank the reviewers for their suggestions which improved the presentation of the paper significantly.

References

[1] R. L. Bowers and J. R. Wilson, Numerical Modeling in Applied Physics and Astrophysics, Jones & Bartlett, Boston, 1991.
[2] P. N. Brown, D. E. Shumaker and C. S. Woodward, Fully implicit solution of large-scale non-equilibrium radiation diffusion with high order time integration, J. Comp. Phys., 204:760–783, 2005.
[3] R. Eymard, T. Gallouët and R. Herbin, Finite Volume Methods, Handbook of Numerical Analysis, P. G. Ciarlet, J. L. Lions (Eds.), Vol. 7, 713–1020, 1997.
[4] D. A. Knoll, W. J. Rider and G. L. Olson, An efficient nonlinear solution method for non-equilibrium radiation diffusion, J. Quant. Spectrosc. Radiat. Transfer, 63:15–29, 1999.
[5] D. A. Knoll, W. J. Rider and G. L. Olson, Nonlinear convergence, accuracy, and time step control in non-equilibrium radiation diffusion, J. Quant. Spectrosc. Radiat. Transfer, 70:25–36, 2001.
[6] V. A. Mousseau and D. A. Knoll, New physics-based preconditioning of implicit methods for non-equilibrium radiation diffusion, J. Comput. Phys., 190:42–51, 2003.
[7] V. A. Mousseau and D. A. Knoll, Temporal accuracy of the non-equilibrium radiation diffusion equations applied to two-dimensional multimaterial simulations, Nuclear Science and Enginerring, 154:174–189, 2006.
[8] V. A. Mousseau, D. A. Knoll and W. J. Rider, Physics-based preconditioning and the Newton-Krylov method for non-equilibrium radiation diffusion, J. Comp. Phys., 160:743–765, 2000.
[9] C. C. Ober and J. N. Shadid, Studies on the accuracy of time-integration methods for the radiation-diffusion equations, J. Comp. Phys., 195:743–772, 2004.
[10] G. L. Olson, Efficient solution of multi-dimensional flux-limited non-equilibrium radiation diffusion coupled to material conduction with second-order time discretization, J. Comp. Phys., 226:1181–1195, 2007.
[11] W. J. Rider, D. A. Knoll and G. L. Olson, A multigrid Newton-Krylov method for multidimensional equilibrium radiation diffusion, J. Comp. Phys., 152:164–191, 1999.
[12] W. J. Rider and D. A. Knoll, Time step size selection for radiation diffusion calculations, J. Comp. Phys., 152:790–795, 1999.

[13] Z. Sheng and G. Yuan, A nine point scheme for the approximation of diffusion operators on distorted quadrilateral meshes, SIAM J. Sci. Comput., 30:1341–1361, 2008.
[14] Z. Sheng, J. Yue and G. Yuan, Monotone finite volume schemes of non-equilibrium radiation diffusion equations on distorted meshes, SIAM J. Sci. Comput., 31:2915–2934, 2009.
[15] A. Shestakov, J. Greenough and L. Howell, Solving the radiation diffusion and energy balance equations using pseudo-transient continuation, J. Quantitative Spectroscopy Radiative Transfer, 90:1–28, 2005.
[16] L. Spitzer and R. Harm, Transport phenomena in a completely ionized gas, Phys. Rev., 89(5):977, 1953.
[17] G. Yuan, Acceleration techniques of iterative solution methods for nonlinear parabolic equations, Annual Report of Laboratory of Computaional Physics, 366–413, 2004.
[18] G. Yuan and X. Hang, Acceleration methods of nonlinear iteration for nonlinear parabolic equations, J. Comput. Math., 24:412–424, 2006.
[19] G. Yuan and Z. Sheng, Analysis of accuracy of a finite volume scheme for diffusion equations on distorted meshes, J. Comp. Phys., 224:1170–1189, 2007.
[20] G. Yuan, X. Hang, Z. Sheng and J. Yue, Progress in numerical methods for radiation diffusion equations, Chinese J. Comp. Phys., 26:475–500, 2009.
[21] J. Yue, G. Yuan and Z. Sheng, Picard-Newton iterative method for multimaterial nonequilibrium radiation diffusion problem on distorted quadrilateral meshes, Lecture Notes in Engineering and Computer Science, The World Congress on Engineering 2009, Vol. 2, 1180–1185.