Available online at www.sciencedirect.com

ScienceDirect

International Journal of Thermal Sciences 47 (2008) 388–398

International Journal of Thermal Sciences

www.elsevier.com/locate/ijts

A leapfrog algorithm for coupled conductive and radiative transient heat transfer in participating media

Paul Henshall *, Phil Palmer

Surrey Space Centre, University of Surrey, Guildford, Surrey, GU2 7XH, UK

Received 24 April 2006; received in revised form 12 March 2007; accepted 13 March 2007

Available online 10 May 2007

Abstract

This paper describes the first applications of a leapfrog algorithm applicable to coupled conductive and radiative transient heat transfer problems within participating media. The algorithm separates the effects of conduction and radiation within the media into two separate energy balance equations. Both of these new equations are initially considered individually. The solutions of the two separate energy equations are iterated in time through a leapfrog algorithm to obtain the transient temperature response of the media being modelled. For simple geometries and in certain cases, these equations can also be solved analytically. For cases in which analytic solutions can be employed, the algorithm is particularly accurate and stable, and especially for large time-steps. In this paper, the leapfrog algorithm is employed to solve the coupled conductive and radiative transient heat transfer problem in a planar layer for both one-dimensional and two-dimensional cases, for the purposes of validating the algorithm.

Keywords: Coupled conduction radiation; Semi-transparent; Radiative transmission; Leapfrog; Verlet

1. Introduction

Coupled conductive and radiative heat transfer within participating media is an important area of study for many industrial applications. Porous, fibrous and semi-transparent materials are examples of media that can exhibit thermal behaviour in which radiative heat transfer plays an important role. Consequently numerous cases of this type of problem for varying mediums and geometries have been considered in the literature. In modelling such problems, a non-linear integrodifferential energy equation is formulated, which is typically a complicated matter to solve. Some of the earliest references discussing these problems date back to the 1960s. For example, Viskanta and Grosh [1] apply a numerical integration and iteration technique to obtain the temperature distribution of a plane layer of grey gas between infinite, black, parallel plates. Since then, a variety of other techniques have been employed to solve a myriad of problems that consider different material properties and different geometries. For example, Tan et al. [2] employs a nodal analysis based on Hottel's zonal method to treat transient coupled radiative and conductive heat transfer within non-grey, semi-transparent materials. Tan et al. [9] also employs a ray-tracing method and Hottel and Sarofim's zonal method to assess one-dimensional, transient, coupled radiative and conductive heat transfer in a multi-layer absorbing and isotropic scattering composite. Wu et al. [3] uses a discrete ordinate method to investigate heat transfer in a two-dimensional, cylindrical, scattering medium with Fresnel boundaries.

This paper presents the first applications of a technique that employs an iterative leapfrog algorithm to solve coupled conductive and radiative heat transfer problems within participating media. This algorithm was developed for the study of fibre optic cables transmitting intense solar radiation in order to assess the thermal behaviour of these fibres. The motivation behind the algorithm development was to minimise computation time while retaining acceptable accuracies, over several hours of simulation time. Thus, an attempt was made to incorporate analytical heat transfer expressions into a numerical algorithm to combat stability and accuracy issues resulting from large simulation time-steps. An approach making use of a leapfrog algorithm was found to be suitable.

* Corresponding author.
E-mail address: p.henshall@surrey.ac.uk (P. Henshall).

1290-0729/$ – see front matter © 2007 Elsevier Masson SAS. All rights reserved.
doi:10.1016/j.ijthermalsci.2007.03.008

Nomenclature

κ	Absorptivity	$\mathrm{m^{-1}}$
c_p	Specific heat	$\mathrm{J\,kg^{-1}\,K^{-1}}$
D, X, Y	Medium dimensions	m
E	Exponential integral function	
q'''	Internal heating	$\mathrm{W\,m^{-3}}$
i	Directional intensity	$\mathrm{W\,m^{-2}\,sr^{-1}}$
S	Radiative source function	$\mathrm{W\,m^{-2}\,sr^{-1}}$
k	Thermal conductivity	$\mathrm{W\,m^{-1}\,K^{-1}}$
m	Time parameter	
n, p	Expansion integers	
N	Radiation–conduction parameter	
$\vec{q}$	Heat flux	$\mathrm{W\,m^{-2}}$
t	Time	s
T	Temperature	K
x, y	Cartesian coordinates	m
$\mathbf{r}$	Position vector	

Greek symbols

Δ	Discretization	
τ	Optical depth	
λ, μ	Eigenvalues	
θ, φ	Angles	radians
ν	Spectrally dependent	
ρ	Density	$\mathrm{kg\,m^{-3}}$
ω	Scattering albedo	
σ	Stefan–Boltzmann constant	
Ω	Solid angle	sr

Subscripts

1, 2, 3, 4	Referring to boundaries 1, 2, 3, 4
B	Boundaries
B	Black body
Cond	Related to the conductive solution
Rad	Related to the radiative solution
r	Radiation
R	Reference
0	Dummy variable

Superscripts

$*$	Dummy variable
$'$	Non-dimensional
f	Final
i	Initial, incident
1D	One-dimensional
2D	Two-dimensional

The Verlet algorithm was devised by L. Verlet in the early days of dynamical molecular simulation [4]. The leapfrog algorithm is a modified version of the Verlet algorithm. These types of algorithm are well known and are usually applied to control and dynamical problems. A typical control application of a leapfrog algorithm is to determine optimal linear quadradic regulator and Kalman filtering gains. Typically, the algorithm takes two related variables, such as position and velocity, and computes them at alternate half time-step intervals via a third-order Taylor expansion. The value of one variable at a particular time acts as the initial condition for the other variable for the next time-step. Second-order accuracies can be attained even for simple solution techniques to the equations of the related variables. Leapfrog algorithms have the further advantages of being time reversal invariant and symplectic [5].

The mathematical formulation of a leapfrog algorithm for a combined conduction and radiation heat transfer problem results in two auxiliary energy balance equations. Each equation separately considers a single form of energy transfer, namely, conduction and radiation. The solutions to these auxiliary equations are easier to obtain than the solution to the original energy balance equation. For problems involving simple geometries, analytic solutions can be found to the auxiliary energy balance equations. These solutions are then employed iteratively in the leapfrog algorithm to obtain the solution to the original energy balance equation. Effectively, conduction and radiation are considered separately and their effects are interlaced over a single time-step via the leapfrog algorithm. The use of analytic solutions in the technique leads to informative implementation and less computational demand for a given required accuracy. Even though the technique can employ analytic solutions, it is inherently numerical in nature and has with it associated stability and accuracy limits.

2. Mathematical formulation

Consider a planar grey, partially transparent solid that emits, absorbs and scatters radiation. For such a case, convection, viscous dissipation and volume expansion can be omitted. The energy equation associated with this system is given by [6]:

$$\rho c_p \frac{\partial T}{\partial t} = k\nabla^2 T - \nabla \cdot \vec{q} + q''' \tag{1}$$

The capacity term on the left-hand side of Eq. (1) describes the storage of energy within the medium. The first term on the right-hand side of Eq. (1) describes the conduction within the medium and q''' allows for any internal heating of the medium. The divergence of the radiative heat flux vector ($\nabla \cdot \vec{q}$) describes the net radiative energy supplied per unit volume and is given by [6]:

$$\nabla \cdot \vec{q} = 4\pi \frac{\kappa}{\omega}\left[\frac{\sigma T^4}{\pi} - S(\tau, \Omega)\right] \tag{2a}$$

Here the source function (S) describes the intensity distribution throughout the medium from both emission and incoming scattering of radiation. In this case, the source function considers the medium to scatter isotropically and is given by the source function equation [6]:

$$S_\nu(\tau_\nu, \Omega) = (1 - \omega_\nu) i_B(\tau_\nu)$$

$$+\frac{\omega_\nu}{4\pi}\int_{\Omega_i=0}^{4\pi}\left[i_b(\tau_{\nu b},\Omega_i)\mathrm{e}^{-\tau_\nu} + \int_0^{\tau_\nu} S_\nu(\tau_\nu^*,\Omega_i)\mathrm{e}^{-(\tau_\nu-\tau_\nu^*)}\,\mathrm{d}\tau_\nu^*\right]\mathrm{d}\Omega_i \tag{2b}$$

For the leapfrog algorithm, heat transfer via conduction and radiation are, initially, considered separately. To do this two auxiliary energy equations are formed in place of Eq. (1):

$$\rho c_p\frac{\partial T_{\mathrm{rad}}}{\partial t} = 4\pi\frac{\kappa}{\omega}\left[S - \frac{\sigma T_{\mathrm{rad}}^4}{\pi}\right] + q''' \tag{3}$$

$$\rho c_p\frac{\partial T_{\mathrm{cond}}}{\partial t} = k\nabla^2 T_{\mathrm{cond}} \tag{4}$$

Eq. (3) contains only radiative terms and Eq. (4) contains only conductive terms. The parameters T_{rad} and T_{cond} represent the temperature within the medium when radiation or conduction dominates, respectively. Thus, the capacity terms on the left-hand side of Eqs. (3) and (4) describe the temperature change of the medium due to radiation and conduction. This is not to say that the true temperature of the medium is the sum of T_{rad} and T_{cond}. The true temperature is estimated by iteratively coupling the solutions to Eqs. (3) and (4) in the leapfrog algorithm.

It is now possible to find analytic solutions to Eqs. (3) and (4). For Eq. (3), we assume that S is constant over a single iteration. Numerical integration is required to determine S from the temperature profile of the medium, and therefore S is updated at each new time-step. The method for determining the solution to Eq. (4) is geometry dependent. The iterative procedure by which the solutions to the auxiliary equations are interlaced is represented graphically in Fig. 1.

In Fig. 1, it is shown that the solution to Eq. (3) is updated twice for each full time-step. This assists with the interlacing of the two heat transfer processes and reduces any error due to the assumption that S is constant for every half time-step. Boundary conditions and initial profiles are matched and transferred as indicated by the arrows in Fig. 1. In other words, the final profile and boundary values resulting from the solution to the auxiliary radiative equation after one time-step will be the initial and boundary conditions for the solution to the auxiliary conduction equation for the next time-step. Eq. (3) is solved by direct integration via rearranging it into the following form:

$$\int \mathrm{d}t = \int \frac{1}{(A - BT_{\mathrm{rad}}^4)}\,\mathrm{d}T_{\mathrm{rad}} \tag{5a}$$

where

$$A = \frac{q'''\omega + 4\pi\kappa S}{\omega\rho c_p} \tag{5b}$$

$$B = \frac{4\kappa\sigma}{\omega\rho c_p} \tag{5c}$$

This can be done here as S is considered constant and not a function of T_{rad}. Eq. (5a) can be simplified via partial fractions to give:

$$\int \mathrm{d}t = \frac{B^{1/2}}{2A^{1/2}}\int \frac{1}{(\frac{A^{1/2}}{B^{1/2}} + T_{\mathrm{rad}}^2)}\,\mathrm{d}T_{\mathrm{rad}} + \frac{B^{1/2}}{2A^{1/2}}\int \frac{1}{(\frac{A^{1/2}}{B^{1/2}} - T_{\mathrm{rad}}^2)}\,\mathrm{d}T_{\mathrm{rad}} \tag{5d}$$

Standard integrals [7] are then used to evaluate both terms on the right-hand side of Eq. (5d), resulting in the following relationship:

$$2A^{3/4}B^{1/4}(t + K) = \tanh^{-1}\left(T_{\mathrm{rad}}\left(\frac{B}{A}\right)^{1/4}\right) + \tan^{-1}\left(T_{\mathrm{rad}}\left(\frac{B}{A}\right)^{1/4}\right) \tag{5e}$$

K is a constant of integration and can be found from the initial conditions for each time-step. If T_{r0} is the initial T_{rad} distribution then K is:

$$K = \frac{1}{2A^{3/4}B^{1/4}}\tanh^{-1}\left(T_{r0}\left(\frac{B}{A}\right)^{1/4}\right) + \frac{1}{2A^{3/4}B^{1/4}}\tan^{-1}\left(T_{r0}\left(\frac{B}{A}\right)^{1/4}\right) \tag{5f}$$

T_{rad} is not readily obtainable from Eq. (5f), however, it can be found very accurately with an iterative method procedure such as the Newton–Raphson algorithm. It is a more complex matter to analytically solve Eq. (4). For simple geometries, such as planar or cylindrical coordinates, it is possible to find analytic

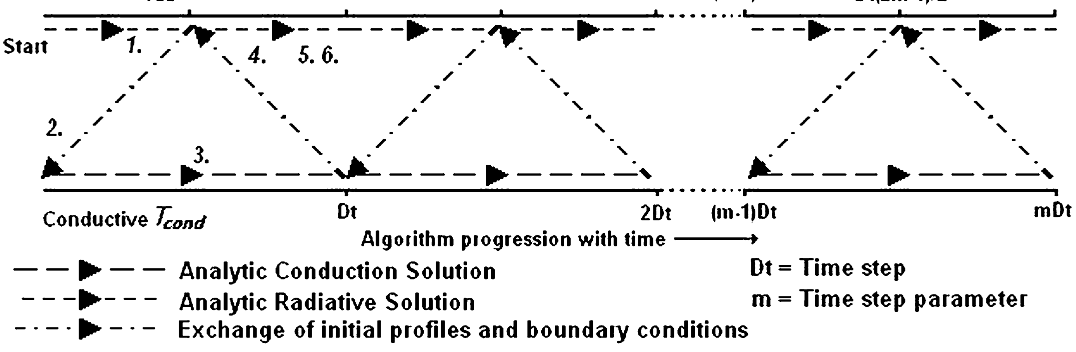


Fig. 1. Leapfrog iterative procedure.

solutions in multi-dimensional cases, although, this is more difficult with the introduction of non-homogeneous boundary conditions. Thus, for more complex conduction problems, a numerical technique should be adopted. In general, the solution to Eq. (4) will require boundary conditions to be specified and evaluated correctly. For prescribed inhomogeneous and homogeneous boundary value problems the boundary conditions will be set by the problem definition and will not change during the course of the algorithm. Then the algorithm is only required to pass initial temperature profiles to the sections of the algorithm devoted to solving the auxiliary conduction and radiative equation for particular time-steps. From Fig. 1, the initial temperature profile can be seen to be the final temperature profile resulting from the solution to the auxiliary radiative equation over the previous time-step, thus:

$$T^i_{\mathrm{cond}}(\mathbf{r}, t_{(m-1)}) = T^f_{\mathrm{rad}}(\mathbf{r}, t_{(2m-1)/2}) \tag{6a}$$

where $\mathbf{r}$ is a position vector that refers to locations not on the boundaries, whose temperatures have already been specified. Similarly the initial temperature profile required for the solution to the auxiliary radiative equation is the temperature profile resulting from the solution of the auxiliary conductive equation from the previous time-step, thus:

$$T^i_{\mathrm{rad}}(\mathbf{r}, t_{(2m-1)/2}) = T^f_{\mathrm{cond}}(\mathbf{r}, t_m) \tag{6b}$$

For inhomogeneous transient boundary conditions, such as radiative boundary conditions, the algorithm should pass both boundary values and initial conditions between solutions to the auxiliary energy equations. For the case of radiative inhomogeneous boundary conditions, the solution to the radiative auxiliary equation also defines the radiative boundary conditions. Thus, for the solution to the radiative auxiliary equation, the boundary values are transient and their values at the end of a time-step are passed to the solution of the auxiliary conduction equation, where they are considered prescribed, for the next time-steps:

$$T_{\mathrm{cond}}(\mathbf{r}_b, t_{(m-1)}) = T^f_{\mathrm{rad}}(\mathbf{r}_b, t_{(2m-1)/2}) \tag{7a}$$

where $\mathbf{r}_\mathrm{b}$ is a position vector that refers to locations on the boundaries. Conversely for conductive inhomogeneous boundary conditions, such as convective boundary conditions, the solution to the auxiliary conductive equation defines the boundary conditions, which are passed to the solution to the auxiliary radiative equation, where they are considered prescribed.

$$T_{\mathrm{rad}}(\mathbf{r}_b, t_{(2m-1)/2}) = T^f_{\mathrm{cond}}(\mathbf{r}_b, t_{(m-1)-m}) \tag{7b}$$

3. Accuracy and stability

The leapfrog algorithm is inherently numerical, and therefore, its accuracy is affected by changing the time and spatial steps of the algorithm. For the cases in which analytic solutions can be employed the accuracy of the leapfrog algorithm can be excellent even for large time-steps. In employing an analytic solution to Eq. (4), potentially any time-steps can be used, as the solution is not subject to any stability criteria. This would not be the case when employing a numerical technique, such as explicit finite difference, and the necessary stability criteria should be observed. In the next section, the algorithm is validated via a simple example, and its accuracy is assessed.

4. Algorithm validation

The leapfrog algorithm is applied to a plane-layer geometry heat transfer problem where the participating medium is grey and scattering has been ignored. The grey medium has a thermal conductivity of k and an absorption coefficient of κ, both of which are constant. The layer is between parallel black plates at constant uniform temperatures T_1 and T_2 which are spaced a distance D apart (see Fig. 2). This problem was previously considered in Ref. [1].

The problem is considered, one-dimensional in x, dimensionless, with normalised units such that $D = 1$ (therefore x varies from 0 to 1) and the reference temperature $T_R = T_1$. For this geometry the source function is (prime indicates dimensionless terms) [6]:

$$S' = \frac{x\kappa}{2}\Bigg[T_1'^4 E_2^{1\mathrm{D}}(\kappa x) + T_2'^4 E_2^{1\mathrm{D}}\big[\kappa(D-x)\big] + \kappa x\Bigg\{\int_0^D T'^4(x^*) E_1^{1\mathrm{D}}\big[\kappa(x-x^*)\big]\,\mathrm{d}x^* + \int_x^D T'^4(x^*) E_1^{1\mathrm{D}}\big[\kappa(x^*-x)\big]\,\mathrm{d}x^*\Bigg\}\Bigg] \tag{8}$$

where $E_1^{1\mathrm{D}}$ and $E_2^{1\mathrm{D}}$ are one-dimensional exponential, integral functions of first- and second-order respectively [6], integrated from $\varphi = 0$ to $\varphi = \pi$. For this particular problem, $T_2' = T_2/T_1 = 1/10$ and initially the entire medium is at a temperature of $T_1' = T_1/T_1 = 1$. The transient temperature of the participating medium is desired. This problem is solved via the leapfrog algorithm and is compared to the solution found via an Euler finite difference algorithm as described in Ref. [8]. The source function (Eq. (8)) is calculated via the procedure indicated in Ref. [6]. Due to the simplicity of the geometry, analytic solutions are employed in the leapfrog algorithm. Eqs. (3) and (4) in dimensionless form are:

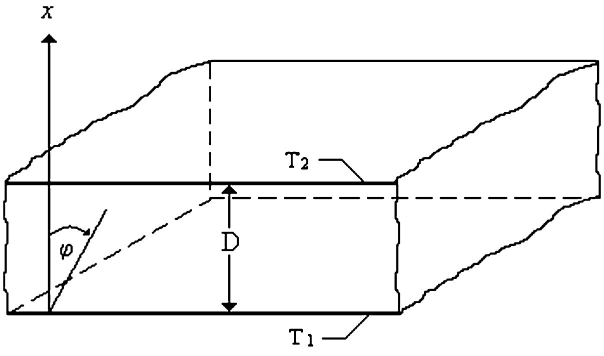


Fig. 2. Plane-layer geometry [6].

$$\frac{\partial T'_{\mathrm{rad}}}{\partial t'} = S' + (q''')' - T'^{4}_{\mathrm{rad}} \tag{9}$$

$$\frac{\partial T'_{\mathrm{cond}}}{\partial t'} = N\nabla^2 T'_{\mathrm{cond}} \tag{10}$$

where

$$T'_{\mathrm{rad}} = \frac{T_{\mathrm{rad}}}{T_R}, \qquad T'_{\mathrm{cond}} = \frac{T_{\mathrm{cond}}}{T_R}, \qquad t' = \frac{4\kappa\sigma T_R^3}{\omega\rho c_p} t$$

$$(q''')' = \frac{q'''\omega}{4\kappa\sigma T_R^3}, \qquad S' = \frac{\pi S}{\sigma T_R^3}, \qquad N = \frac{k\omega}{4\kappa\sigma T_R^3} \tag{11a–f}$$

The parameter N is the conduction-to-radiation parameter and its value denotes the relative effects of conduction and radiation. Eq. (9) is solved by direct integration with the solution:

$$2\big(S' + (q''')'\big)^{3/4} t' + K' = \tanh^{-1}\left(\frac{T'_{\mathrm{rad}}}{(S' + (q''')')^{1/4}}\right) + \tan^{-1}\left(\frac{T'_{\mathrm{rad}}}{(S' + (q''')')^{1/4}}\right) \tag{12}$$

For a 1D planar geometry Eq. (10) reduces to:

$$\frac{\partial T'_{\mathrm{cond}}}{\partial t'} = N\kappa \frac{\partial^2 T'_{\mathrm{cond}}}{\partial \tau^2} \tag{13}$$

where τ is the optical depth ($\tau = x\kappa$). The boundary conditions to be applied to Eqs. (12) and (13) are prescribed by the boundary temperatures (T'_1 and T'_2). However, for the case when $N = 0$, the boundary conditions become transient with the solution to the auxiliary radiative equation, but still no boundary conditions are passed as conduction now plays no role in the algorithm. Initial conditions are passed between solutions to the auxiliary equations as indicated in Eqs. (6a) and (6b), such that:

$$T'_{\mathrm{cond}}(\tau, t'_{(m-1)-m}) = T'_{\mathrm{rad}}(\tau, t'_{(m-1)-(2m-1)/2}) \tag{14a}$$

$$T'_{\mathrm{rad}}(\tau, t'_{(2m-1)/2-m}) = T'_{\mathrm{cond}}(\tau, t'_{(m-1)-m}) \tag{14b}$$

The solution to Eq. (13) can be found via the method of partial solutions [4], and is given by:

$$T'_{\mathrm{cond}}(\tau, t') = \sum_{n}^{\infty} C_n \sin(\lambda_n \tau) \mathrm{e}^{-N\tau\lambda_n t'} + \left(\frac{T'_{\mathrm{cond}}(D, t') - T'_{\mathrm{cond}}(0, t')}{D}\right)\tau + T'_{\mathrm{cond}}(0, t') \tag{15a}$$

where

$$C_n = \frac{2}{D}\int_0^D g(\tau)\sin(\lambda_n\tau)\,\mathrm{d}\tau \tag{15b}$$

and

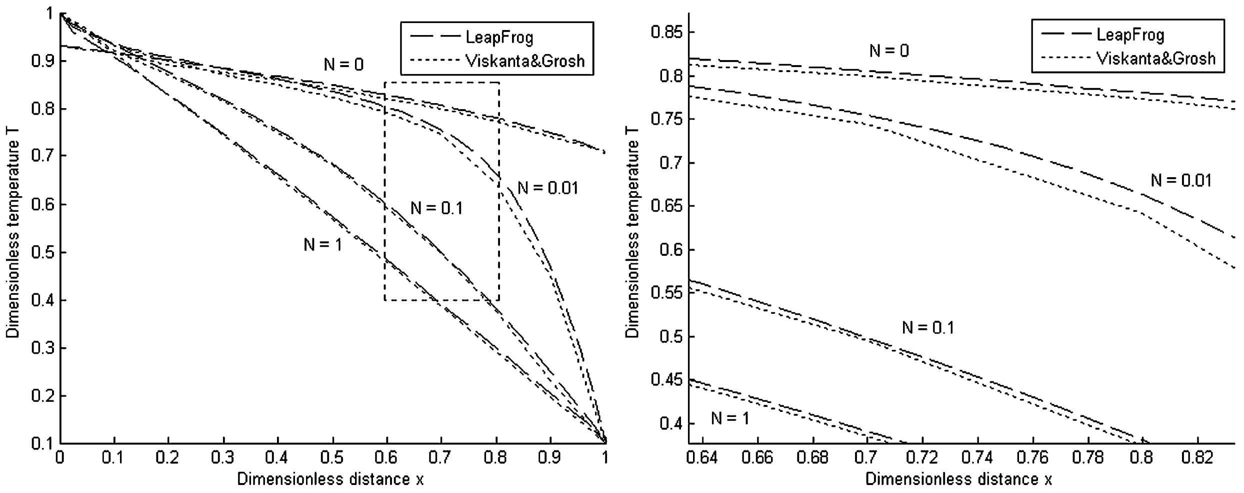
$$\lambda_n = \frac{n\pi}{D} \tag{15c}$$

$$g(t') = T'_{\mathrm{cond}}(\tau, t') - \left(\frac{T'_{\mathrm{cond}}(D, t') - T'_{\mathrm{cond}}(0, t')}{D}\right)\tau + T'_{\mathrm{cond}}(0, t') \tag{15d}$$

The algorithm proceeds via the following procedure as indicated in Fig. 1.

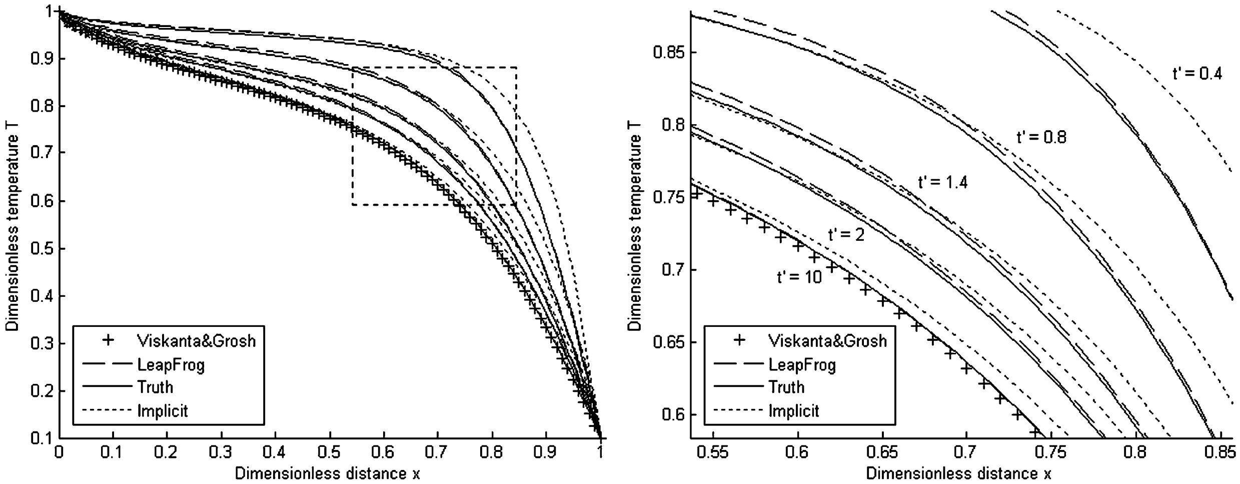
1. The algorithm begins by evaluating S' and evaluating Eq. (12) over the first T'_{rad} half time-step, from the original initial conditions. The boundary temperatures are always fixed at T'_1 and T'_2.
2. The temperature profile resulting from step 1 is now the initial condition for Eq. (13) over the first T'_{cond} full time-step.
3. Eq. (13) is now solved over the first T'_{cond} full time-step.
4. The temperature profile resulting from step 3 is now the initial condition for the evaluation of Eq. (12) over the second T'_{rad} half time-step. S' is evaluated again from the new initial condition.
5. Eq. (12) is now evaluated over the second T'_{rad} half time-step.
6. The temperature profile resulting from step 5 is now the initial condition for the evaluation of Eq. (12) over the third T'_{rad} half time-step. S' is evaluated again from the new initial condition.

Steps 1–6 are repeated until the desired simulation time. In Fig. 3(a), a series of transient temperature profiles, simulated via the leapfrog algorithm, close to steady state, are plotted for varying values of N. Viskanta and Grosh [1] considered the same problem addressed here and developed a numerical iteration procedure to solve for the steady state temperature profile of the plane layer. The results of Viskanta and Grosh are also plotted in Fig. 3(a) for the same values of N. Fig. 3(b) is an enlarged view of the area enclosed by the box seen in Fig. 3(a). Figs. 3(a) and (b) demonstrate that the leapfrog temperature profiles are consistent with the corresponding steady state temperature profile of Viskanta and Grosh. In addition, both temperature profiles representing the $N = 0$ case exhibit the temperature slip at each boundary as is normally observed for such cases [6]. Figs. 4(a) and (b) show the transient behaviour of the leapfrog algorithm, along with an implicit finite difference algorithm and "truth" for the $N = 0.03$ case. In Figs. 4(a) and (b) it can be seen that as simulation time approaches infinity, all the transient algorithms converge on the steady state solution of Viskanta and Grosh, also plotted. The truth values are generated via an implicit finite difference algorithm which progresses with a very small time-step of 0.005 and has a spatial discretization of $\Delta x = 0.005$. The temperature profiles produced by this algorithm are considered the actual transient temperature of the plane layer so that transient errors can be estimated for both the leapfrog and implicit finite difference algorithms. As no exact solution is available for the transient behaviour, it is difficult to determine the true transient error. It was found that a steady state error resulted in each of the transient algorithms considered, and it appeared significantly dependent on the spatial discretization chosen. It was also found that the algorithm time-step had more effect on the transient error than the steady state error. The spatial discretization of the truth algorithm was selected such that it resulted in a steady state error of 0.1% when compared to the results of Viskanta and Grosh. This steady state error is primarily a result of the spatial discretization of the geometry resulting in errors in temperature gradient estimates and the



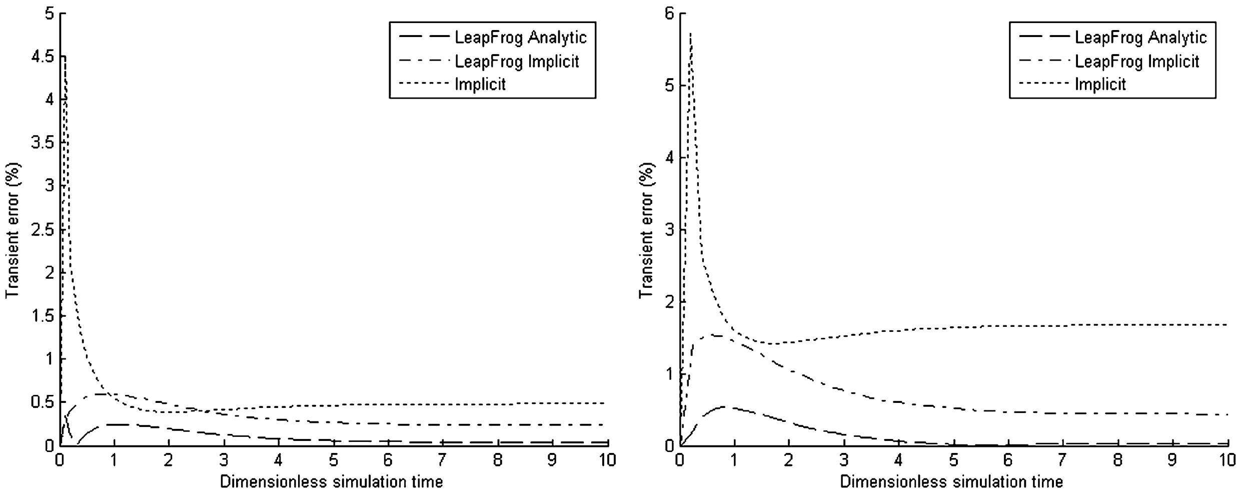
(a) (b)

Fig. 3. (a) Leapfrog and finite difference plane-layer transient temperature profiles for varying values of N. (b) Zoom in of box in (a).



(a) (b)

Fig. 4. Transient plane layer temperature profiles of the leapfrog, finite difference and truth algorithms at varying simulation times with $N = 0.001$. (b) Zoom in of box in (a).



(a) (b)

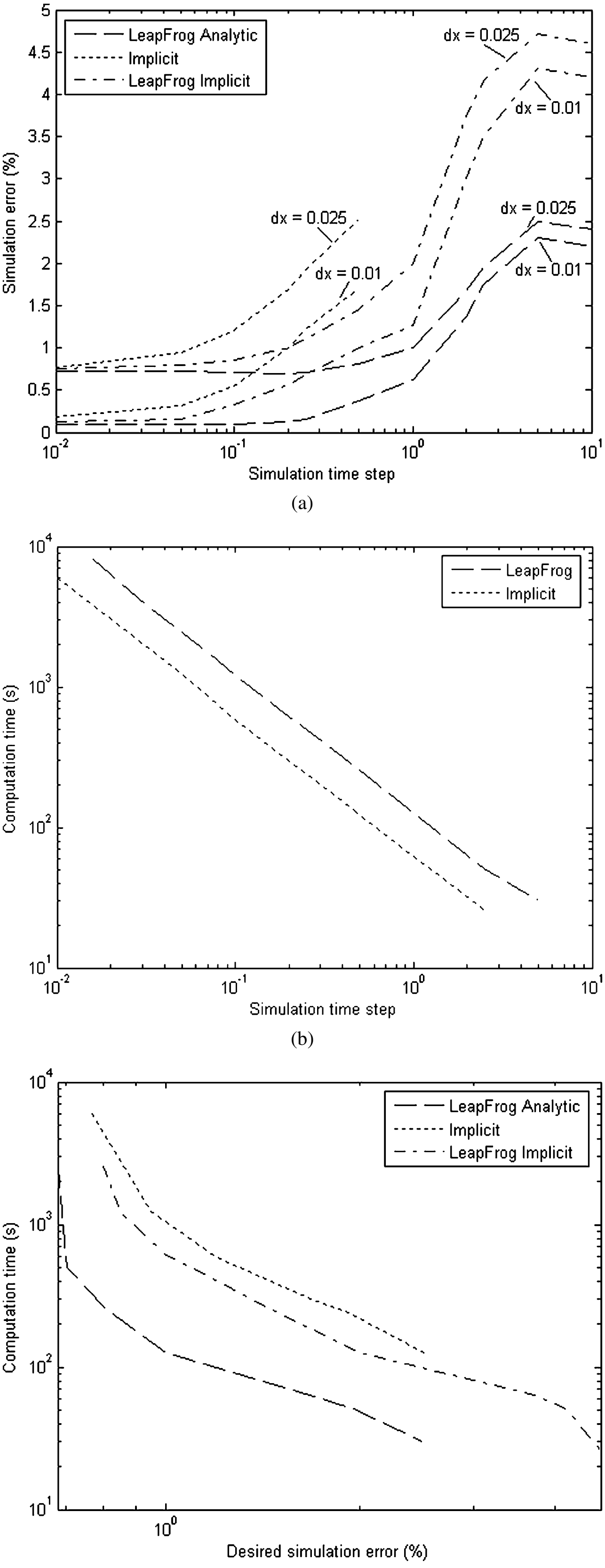
Fig. 5. (a) Transient error for leapfrog and finite difference algorithms with $N = 0.001$ ($\Delta t = 0.1$, $\Delta x = 0.001$). (b) Transient error for leapfrog and finite difference algorithms with $N = 0.03$ ($\Delta t = 0.25$, $\Delta x = 0.025$).

evaluation of the source function. Fig. 4(b) is an enlarged view of the area enclosed in the box seen in Fig. 4(a) and shows the temperature profiles produced by each algorithm at simulation times of 0.4, 0.8, 1.4, 2 and 10 in non-dimensional time units. These times were chosen as they effectively demonstrate the transient temperature behaviour of the plane layer with $N = 0.03$. The maximum simulation time was selected, as by this time transient temperature gradients were on the order of 10^{-6} or lower for all algorithms, at which point the transient algorithm was considered to have converged on its steady state result.

In order to assess the behaviour of each algorithm's accuracy, transient simulations were conducted for a selection of time-steps and spatial discretization values. Three types of algorithm were used for these transient error tests, an implicit finite difference algorithm, a purely analytic leapfrog algorithm and implicit finite difference leapfrog algorithm. The last algorithm employs the implicit finite difference technique to solve the auxiliary conduction equation of the leapfrog algorithm. The implicit finite difference method was chosen for comparison to allow a greater range of time and spatial discretization to be explored. For most of the situations considered, an explicit finite difference method would be unstable. The behaviour of each algorithm's transient error can be seen in Figs. 5(a) and (b), where Fig. 5(a) is the transient error for a simulation with $N = 0.001$ and Fig. 5(b) is the transient error for a simulation with $N = 0.03$. The error is calculated by computing the absolute average difference between each algorithm and the truth algorithm at each time-steps. In both Figs. 5(a) and (b) the transient results of the leapfrog algorithms can be seen to be much more accurate than the implicit finite difference algorithm. Furthermore, the steady state error of the leapfrog algorithms is observed to be more accurate. The steady state error for each algorithm was observed to decrease when the spatial and temporal discretization were reduced.

A further error (simulation error) can be calculated to provide a single figure of merit on how accurate an algorithm was during a simulation by averaging the error plots shown in Figs. 5(a) and (b). This was accomplished for the $N = 0.001$ case and for a range of simulation time-steps with two spatial discretization values (0.01 and 0.025). The results of this analysis are shown in Fig. 6(a). In this figure, it is seen that the leapfrog algorithms perform consistently better than the finite difference algorithm as the time-step employed in the simulation is increased. Furthermore, the finite difference algorithm becomes unstable after a time-step value of 0.5 where as the leapfrog algorithms remain stable and reasonably ($< 5\%$) accurate, even for a single time-step from time zero to time 10. Siegel [10] notes that even implicit numerical solution techniques for coupled conduction and radiation problems tend to become unstable for excessive time-steps.

In Fig. 6(a), it can be seen that as the time-step decreases the simulation errors of algorithms with the same spatial discretization converge. This indicates that for a given spatial discretization there is a limit to the simulation accuracy that can be achieved. Yang and Gu [11] demonstrate, for implicit numerical solution methods to parabolic partial differential equations



(a)

(b)

(c)

Fig. 6. (a) Simulation error for leapfrog and finite difference algorithms with $N = 0.001$ vs. simulation time-step. (b) Computation time of leapfrog and finite difference simulations vs. simulation time-step. (c) Required computation time of leapfrog and finite difference simulations vs. desired simulation accuracy.

that for a certain level of spatial discretization, a minimum time-step exists for which the maximum simulation accuracy will occur. Yang and Gu also note that, in some cases, once the numerical simulation error reaches a minimum, it increases slightly if the simulation time-step is further reduced. This was attributed to the increased number of calculation steps for a reduced time-step. This phenomenon identified by Yang and Gu [11] is observed to occur in Fig. 6(a).

In Fig. 6(b), the computation times required for each algorithm are plotted against time-step and it can be seen that the leapfrog algorithms consistently take approximately twice as long as the finite difference algorithm to complete a simulation for a given time-step. These figures demonstrate that the leapfrog algorithms are useful for simulations employing large time-steps in order to achieve a reasonably accurate estimate of the plane layer temperature profile in a short period of time. This is further emphasised in Fig. 6(c), where the computation time required for a simulation is plotted against desired simulation accuracy for all algorithms. In Fig. 6(c), it is clear that the leapfrog algorithms take less time to yield a desired simulation accuracy when compared to the implicit finite difference algorithm.

5. Extension to multiple dimensions

The leapfrog algorithm can easily be extended to multidimensional problems. All that is required is a suitable solution technique to the auxiliary conduction energy equation in the geometry under study. Again, for simple geometries and circumstances, this can be an analytic solution. For the auxiliary radiation energy equation, care should be taken with the source function to ensure the appropriate exponential integral functions are employed for the given geometry [3]. To demonstrate a two-dimensional leapfrog algorithm, a two-dimensional planar coordinate system (in x and y, see Fig. 7) is selected.

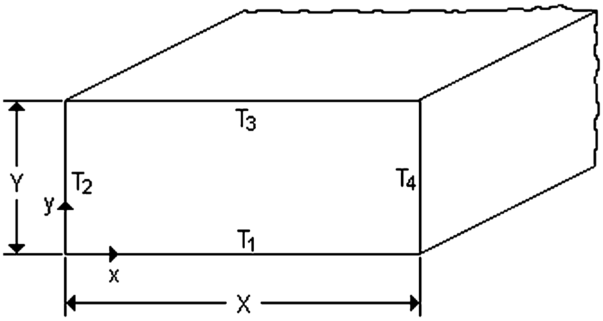


Fig. 7. 2D planar geometry [12].

The grey medium has a thermal conductivity of k, an absorption coefficient of κ, a density of ρ and specific heat of c_p, all of which are constant. The problem is considered normalised, dimensionless with $X = 1$, $Y = 1$ and the initial temperature of the plane layer (T_1) is the reference temperature (T_r) having a value of unity. As in the one-dimensional case, scattering is neglected. For this geometry, Eq. (12) remains unchanged apart from the calculation of S' which is given by [6]:

$$\begin{aligned} S' = {} & (Y-y)\int_{x_0=0}^{X} T_1^4(x_0,Y)\frac{E_2^{2\mathrm{D}}[\kappa\delta_0(x_0,Y)]}{\delta_0^2(x_0,Y)}\,\mathrm{d}x_0 \\ & + (X-x)\int_{y_0=0}^{Y} T_2^4(X,y_0)\frac{E_2^{2\mathrm{D}}[\kappa\delta_0(X,y_0)]}{\delta_0^2(X,y_0)}\,\mathrm{d}y_0 \\ & + y\int_{x_0=0}^{X} T_3^4(x_0,0)\frac{E_2^{2\mathrm{D}}[\kappa\delta_0(x_0,0)]}{\delta_0^2(x_0,0)}\,\mathrm{d}x_0 \\ & + x\int_{y_0=0}^{Y} T_4^4(0,y_0)\frac{E_2^{2\mathrm{D}}[\kappa\delta_0(0,y_0)]}{\delta_0^2(0,y_0)}\,\mathrm{d}y_0 \\ & + \kappa\int_{x^*=0}^{X}\int_{y^*=0}^{Y} T^4(x^*,y^*)\frac{E_1^{2\mathrm{D}}[\kappa\delta^*]}{\delta^*}\,\mathrm{d}x^*\,\mathrm{d}y^* \end{aligned} \tag{16a}$$

where

$$\delta^* = \left[(x-x^*)^2 + (y-y^*)^2\right]^{1/2} \tag{16b}$$

$$\delta(x_0,y_0) = \left[(x-x_0)^2 + (y-y_0)^2\right]^{1/2} \tag{16c}$$

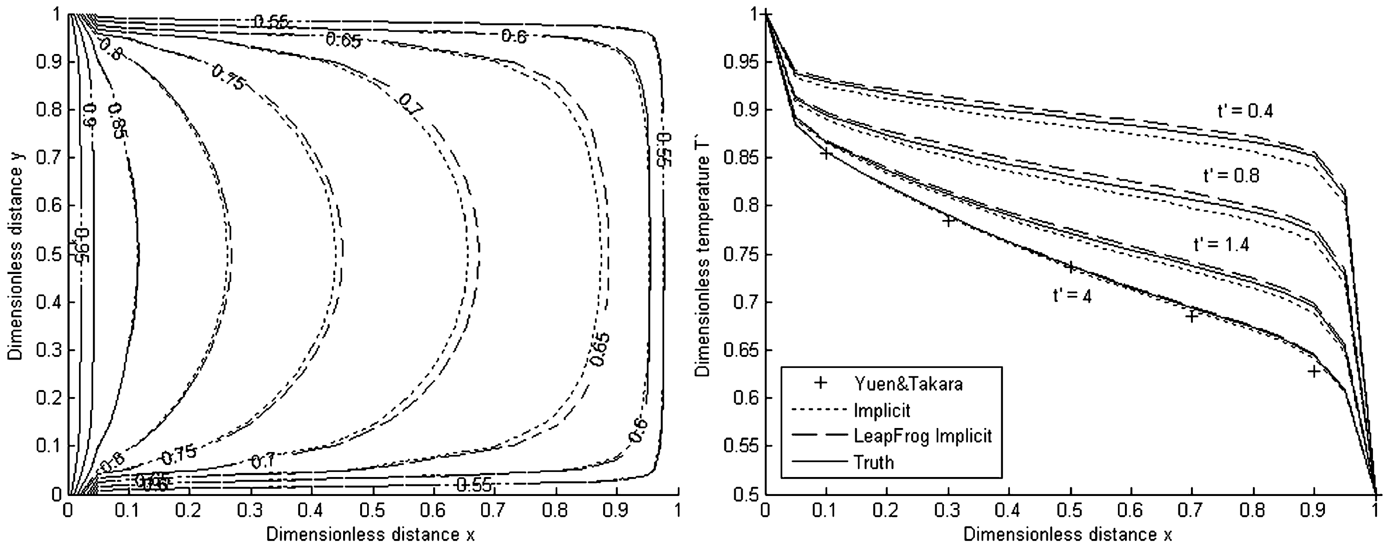
Here $E_1^{2\mathrm{D}}$ and $E_2^{2\mathrm{D}}$ are two-dimensional exponential integral functions of first- and second-order respectively [6]. For this geometry Eq. (10) reduces to:

$$\frac{\partial T'_{\mathrm{cond}}}{\partial t'} = N\left[\frac{\partial^2 T'_{\mathrm{cond}}}{\partial x^2} + \frac{\partial^2 T'_{\mathrm{cond}}}{\partial y^2}\right] \tag{17}$$

Initially, the entire two-dimensional plane layer is at a temperature of $T_i = 1$. Then, the temperatures along boundaries 1, 3 and 4 are instantaneously dropped to $T_1 = T_3 = T_4 = 0.5$. Thus, the boundary conditions for the simulation are the prescribed boundary values given by $T_1 = T_3 = T_4 = 0.5$ and $T_2 = 1$. Three algorithms are employed to assess the behaviour of each algorithm's error, which is accomplished in the same manner as for the one-dimensional case. Two of the algorithms are finite difference methods; one being an implicit algorithm and the other being an explicit algorithm, and the final algorithm is a leapfrog algorithm, which makes use of an implicit solver for the auxiliary conduction equation. Yuen and Takara [12] considered the steady state two-dimensional plane layer problem and solved for the steady state temperature profile of the plane layer.

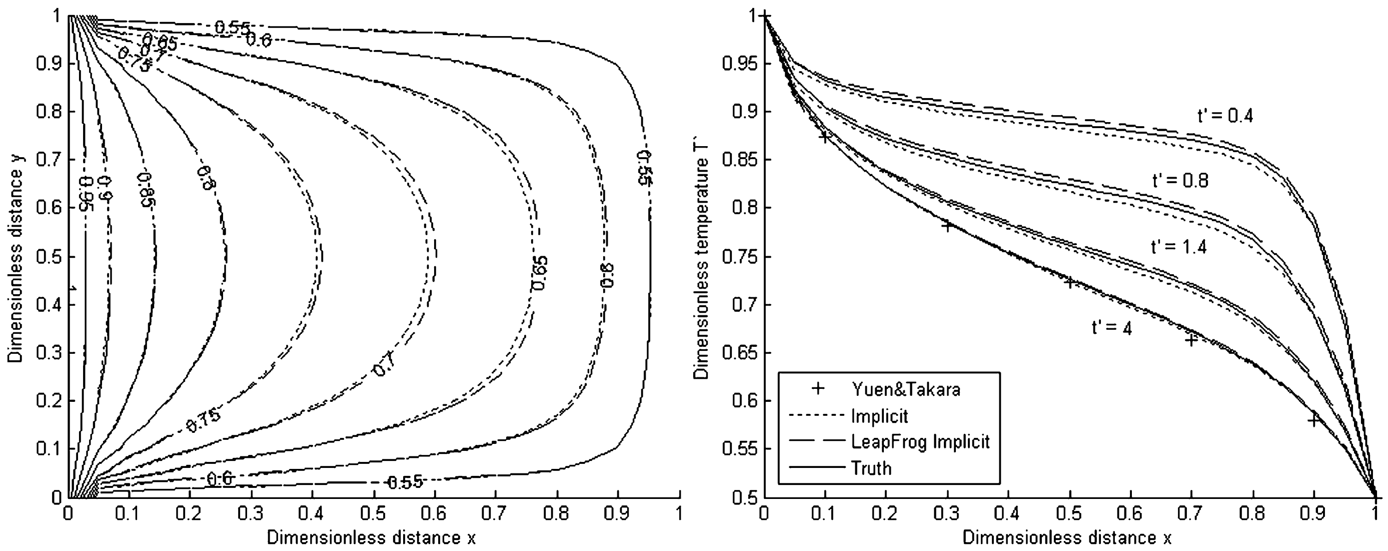
In Fig. 8(a), contour plots of the two-dimensional temperature distribution for the $N = 0.001$ case can be seen. These plots were produced by the ordinary implicit and the implicit leapfrog algorithm and are plotted for comparison. In Fig. 8(b), the transient centre line temperature distributions of the two-dimensional plane layer, produced by the implicit, implicit leapfrog and truth algorithms, are plotted. Also plotted in Fig. 8(b), for comparison, are the results reported by Yuen and Takara for the $N = 0.001$ steady state centre line temperature distribution.

In Fig. 9(a), contour plots of the two-dimensional temperature distribution for the $N = 0.01$ case can be seen. These



(a) (b)

Fig. 8. (a) Euler leapfrog and Euler finite difference plane-layer transient temperature contour plots for $N = 0.001$. (b) Centre line temperature distributions.



(a) (b)

Fig. 9. (a) Pure implicit leapfrog and pure implicit finite difference plane-layer transient temperature contour plots for $N = 0.01$. (b) Centre line temperature distributions.

plots were produced by the ordinary implicit and the implicit leapfrog algorithm and are plotted for comparison. In Fig. 9(b), the transient centre line temperature distributions of the two-dimensional plane layer produced by the implicit, implicit leapfrog and truth algorithms are plotted. Also plotted in Fig. 9(b), for comparison, are the results reported by Yuen and Takara for the $N = 0.01$ steady state centre line temperature distribution.

In Fig. 10, the transient errors of the ordinary implicit, explicit and the implicit leapfrog algorithms are plotted for the $N = 0.001$ case with a time-step of 0.1. In Fig. 10, it is clear that the transient errors of the implicit leapfrog algorithms exhibit the same behaviour as seen in the one-dimensional planar case. A total dimensionless simulation time of 4 was employed as it was found to allow all algorithms to converge within at least 95% of the their steady state temperature values.

Performing a similar analysis as for the one-dimensional plane layer, different simulation time-steps are used in simulations of the two-dimensional plane layer problem and the corresponding simulation errors is calculated. In Fig. 11, the simulation error for each algorithm is plotted against simulation time-step for the $N = 0.001$ case. The implicit leapfrog algorithm is observed to become more accurate than the ordinary finite difference algorithms as time-step increases. Furthermore, both finite difference algorithms were observed to become unstable and inaccurate (>10%) for time-steps greater than 1 whereas the implicit leapfrog algorithm remained stable even for a single step to steady state. Nevertheless a limit

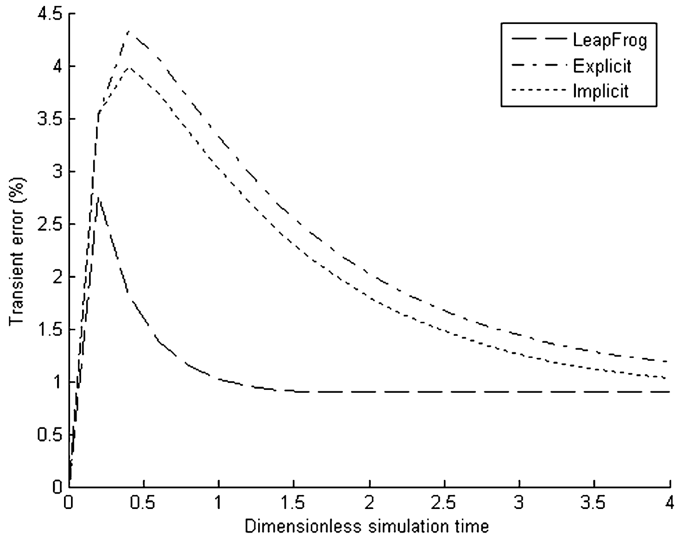


Fig. 10. Transient error vs. simulation time for $N = 0.001$ case ($\Delta t = 0.2$, $\Delta x = 0.1$).

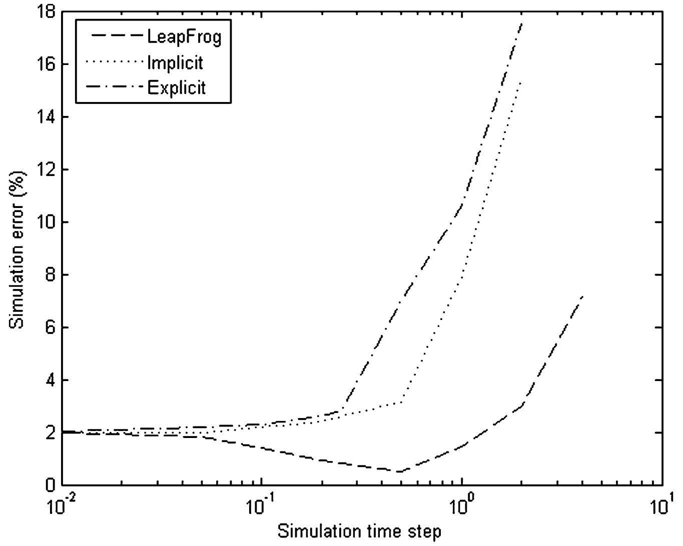


Fig. 11. Simulation error vs. simulation time-step for the $N = 0.001$ case.

in achievable accuracy is again observed. In this case, the phenomenon described by Yang and Gu is clearly visible in the implicit leapfrog algorithm. At a time-step of 0.5, the implicit leapfrog algorithm is observed to be the most accurate of the algorithms considered. With further reduction of the simulation time-step, the accuracy of the implicit leapfrog algorithm is observed to decrease to match that of the other finite difference algorithms. In order to determine if the leapfrog algorithm offers an improvement in performance, the computation time for the simulations is required. The computation times, for these simulations, versus simulation time-step is plotted in Fig. 12(a). In Fig. 12(b), computation time is plotted against desired simulation accuracy for each algorithm. Fig. 12(b) indicates that the implicit leapfrog algorithm is an improvement upon the ordinary finite difference algorithms as for a given desired simulation accuracy the required computation time for the implicit leapfrog method is less than that for the ordinary explicit and implicit finite difference methods. Moreover the implicit leapfrog algorithm is capable of better accuracy than the other finite difference algorithms. This improvement in accuracy is attributed to the analytic calculation of the auxiliary radiative equation in the implicit leapfrog algorithm.

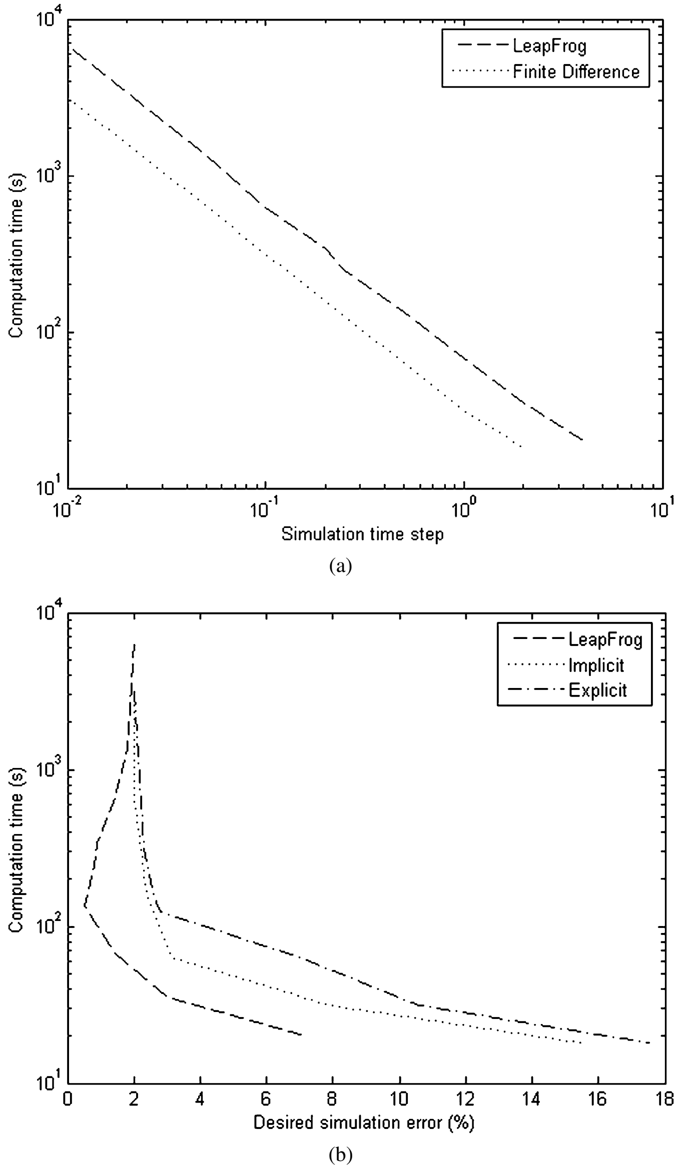


Fig. 12. (a) Computation time of leapfrog pure implicit and finite difference simulations vs. simulation time-step. (b) Required computation time of leapfrog pure implicit and ordinary pure implicit simulations vs. desired simulation error.

6. Conclusion

Employing a leapfrog algorithm to study transient coupled conductive and radiative heat transfer in participating media has been shown to be a valid, accurate and informative numerical approach for solving the energy balance equation in multidimensional geometries. The leapfrog algorithm allows analytic solutions to be employed in a numerical scheme for suitably simple geometries. A one-dimensional plane layer example demonstrated the use of analytical solutions in this capacity and the high levels of accuracy that can be attained even for large time-steps. A two-dimensional plane layer example was also included to demonstrate that the leapfrog algorithm could be employed in multidimensional problems. The main benefit of the leapfrog algorithm was demonstrated to be its ability to provide accurate transient results, performing better than standard explicit and implicit finite difference methods especially for large simulation time-steps.

References

[1] R. Viskanta, R.J. Grosh, Heat transfer by simultaneous conduction and radiation in absorbing medium, J. Heat Transfer 84 (1) (1962) 63–72.

[2] H.-P. Tan, H.-L. Yi, H.-C. Zhang, H.-C. Wang, P.-Y Tong, Coupled radiation–conduction heat transfer in an anisotropically scattering slab with mixed boundaries, J. Quantit. Spectrosc. Radiat. Transfer 83 (3–4) (1 February 2004) 667–698.

[3] C.-Y. Wu, B.-T. Liou, Discrete-ordinate solutions for radiative transfer in a cylindrical enclosure with Fresnel boundaries, Int. J. Heat Mass Transfer 40 (10) (July 1997) 2467–2475.

[4] D.C. Rapaport, The Art of Molecular Dynamics Simulation, Cambridge Univ. Press, Cambridge, UK, 1995.

[5] M.A. López-Marcos, J.M. Sanz-Serna, R.D. Skeel, Explicit symplectic integrators using Hessian-vector products, SIAM J. Sci. Comput. 18 (1) (January 1997) 223–238.

[6] R. Siegel, J.R. Howell, Thermal Radiation Heat Transfer, third ed., Taylor and Francis, London, 1992.

[7] E. Kreyszig, Advanced Engineering Mathematics, eighth ed., John Wiley and Sons Inc., New York, 1999.

[8] G.E. Myers, Analytical Methods in Conduction Heat Transfer, McGraw–Hill, New York, 1971.

[9] H.-P. Tan, J.-F. Luo, X.-L. Xia, Transient coupled radiation and conduction in a three-layer composite with semitransparent specular interfaces and surfaces, ASME 124 (2002).

[10] R. Siegel, Transient effects of radiative transfer in semitransparent materials, Int. J. Engrg. Sci. 36 (1998) 1701–1739.

[11] C. Yang, Y. Gu, Minimum time-step criteria for the Galerkin finite element methods applied to one-dimensional parabolic partial differential equations, Wiley Interscience Periodicals, New York, 2005.

[12] W.W. Yuen, E.E. Takara, Analysis of combined conductive-radiative heat transfer in a two-dimensional rectangular enclosure with a gray medium, in: AIAA Joint Thermophysics and Heat Transfer Conference, AIAA-86-1235 to 86-1367, 1986.