Asociación Argentina
de Mecánica Computacional

Mecánica Computacional Vol XXVII, págs. 1573-1579 (artículo completo)
Alberto Cardona, Mario Storti, Carlos Zuppa. (Eds.)
San Luis, Argentina, 10-13 Noviembre 2008

THE METHOD OF LINES WITH EVENT LOCATION APPLIED TO THE STEFAN PROBLEM

Ulises Lacoa[a] **and Antonio Campo**[b]

[a]*Departamento de Termodinámica y Fenómenos de Transferencia, Universidad Simón Bolívar, Apartado 89000, Caracas 1080-A, Venezuela, ulises@usb.ve, http://www.usb.ve*

[b]*Department of Mechanical Engineering, University of Vermont, Burlington, VT 05405, USA, acampo@uvm.edu, http://www.uvm.edu/programs/mechanical_enginnering*

Keywords: Method of Lines, Event Location, Stefan Problem, Moving Boundary.

Abstract. This paper describes the method of lines with event location applied to one-dimensional Stefan problem. Stefan problems arise in many physical processes, such as, freezing and thawing of foods, solidification of steel and chemical reaction. Mathematically, those problems are special cases of moving boundary problems. The solution of such problems requires solving the transient heat conduction equation in a unknown region which has to be determined as part of the solution. The model experiences structural changes in the definition of the ordinary differential equation. To overcome these difficulties, the model was implemented in MATLAB and the event-function in the Ordinary Differential Equation (ODE) solver was activated. To validate the present method, the results for a test case with known analytical solution are compared. The results for a wide spectrum of Stefan numbers indicate that the method of lines with event location is able to accurately track the moving liquid-solid interface and temperature history.

1 INTRODUCTION

Phase-change problems, also known as Stefan problem, occur naturally in many physical and engineering processes, such as casting, melting and solidification of alloys, production of ice, freezing and thawing of foods and storage of energy. In this class of problems the unknown boundaries need to be tracked or located as part of the solution. Two conditions are required in order to solve phase-change problems, one to determine the unknown boundary and the solution of energy equation. The simplest of such problems is the melting of ice that was first treated by Stefan (Stefan, 1889). The main goal of the present work is to develop a numerical method based on the Method of lines on a fixed grid that is efficient and accurate for phase change problems. This technique and its variations have been implemented successfully in a wide range of conduction-convection problems (Campo et al., 1996; Salazar et al., 2000).

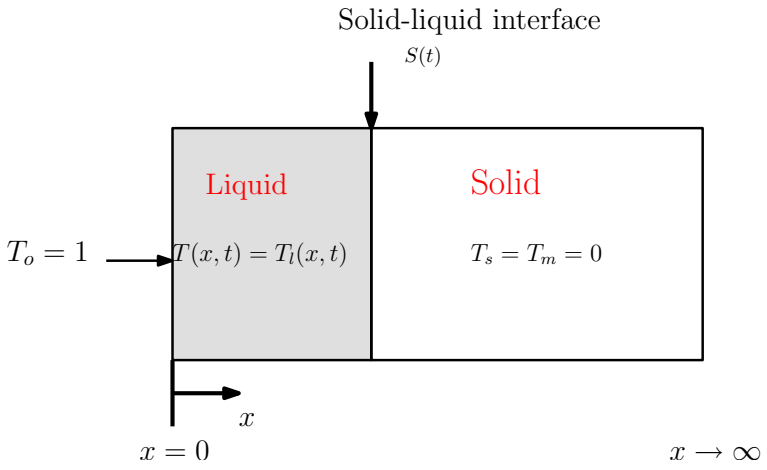

Figure 1: Schematic diagram showing problem description.

2 PROBLEM DESCRIPTION

A semi-infinite system subjected to prescribed temperature was considered. Initially, the full solid was a the phase change temperature T_m , wich was scaled to be zero. The temperature at $x = 0$ was suddenly raised to $T_o = 1$ at time $t = 0$ at maintained at that temperature for all times $t > 0$. The schematic diagram showing boundary condition, coordinates and nomenclature is given in Figure 1. Once the boundary condition was imposed, melting began at the surface $x = 0$ and proceeded into the solid as the discrete liquid-solid interface moved in the positive x direction.

3 MATHEMATICAL MODEL

The model is based on the heat conduction equation.

$$\frac{\partial T}{\partial t} = \alpha_l \frac{\partial^2 T}{\partial x^2}, \quad 0 < x < s(t), \quad t > 0 \tag{1}$$

subject to boundary conditions

$$T(x = 0, t) = 1, \quad T(x = s(t), t) = 0 \tag{2}$$

and initial condition

$$T(x, t = 0) = T_m \tag{3}$$

at the interface the energy balance yields:

$$\rho L \frac{ds}{dt} = -k_l \frac{\partial T}{\partial x} \tag{4}$$

where T is the temperature, $s(t)$ is the position of the moving front and L is the latent heat of phase change, α_l is the thermal diffusivity, ρ is the density and k_l is the thermal conductivity.

4 ANALYTICAL SOLUTION

For comparisons purposes we present the analytical solution taken from (Ozisik, 1980). The dimensionless temperature distribution has the form

$$T^* = \frac{T - T_m}{T_o - T_m} = 1 - \frac{erf\left(x^*/2\sqrt{t^*}\right)}{erf(\lambda)} \tag{5}$$

and the dimensionless displacement of the melt interface is given by

$$S^* = 2\lambda\sqrt{t^*} \tag{6}$$

where λ are solutions to the transcendental eq. 7

$$\lambda \exp(\lambda^2) erfc(\lambda) = St/\sqrt{\pi} \tag{7}$$

The Stefan number, St, is defined as

$$St = \rho c(T_o - T_m)/L \tag{8}$$

and the dimensionless x^* variable and time are defined by

$$x^* = \frac{x}{l} \quad , t^* = \frac{\alpha_l t}{l^2} \tag{9}$$

where l is a characteristic length.

5 NUMERICAL METHOD

The partial differential Eq. (1) is solved numerically by the method of lines for the time derivative and finite difference for the space derivative. This hybrid numerical methodology reduces the partial differential equation to a system of first order ordinary differential equations, which is then solved using an ODE solver algorithm. The discretization method is mainly taken from (Chun and Park, 2000; Furenes and Lie, 2006). The spatial domain is divided into $N - 1$ uniform grid cells, Δx. With the interface located between nodes i and $i + 1$, as showed in Fig. 2, the nodal temperatures are given by

$$\frac{dT_k}{dt} = \alpha_l \frac{T_{k+1} - 2T_k + T_{k-1}}{(\Delta x)^2}, \quad for \quad k = 2, ...i - 1 \tag{10}$$

$$\frac{dT_i}{dt} = \alpha_l \frac{T_m - 2T_i + T_{i-1}}{(\Delta x)^2} + \frac{\alpha_l}{k_l} \frac{(1 - \delta)\rho L}{\Delta x} \frac{ds}{dt} \tag{11}$$

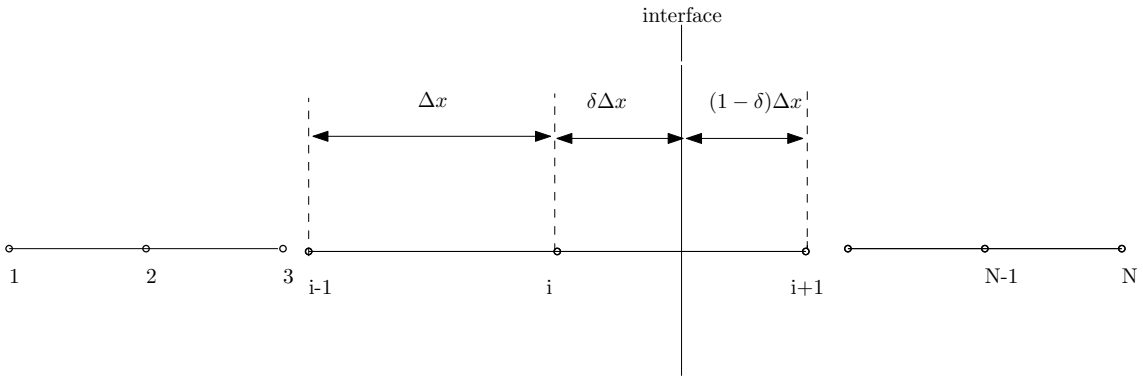

Figure 2: Discretization scheme.

where $\delta = (s - i\Delta x)/\Delta x$ is the dimensionless distance from the nodal point i to the interface. The location of i can be found by comparing the nodal temperature with the melting temperature, T_m

$$i = max(T_k \geq T_m), k = 1, ..., N-1. \tag{12}$$

$$\frac{ds}{dt} = -\frac{k_l}{\rho L}\left.\frac{\partial T}{\partial x}\right|_{x=s(t)} \tag{13}$$

To contend with the singularity problem that arises when the interface becomes very close to the grid node, the slope at the interface is calculated from the first two terms of the Taylor series expansion (Verma et al., 2004):

$$\left.\frac{\partial T}{\partial x}\right|_{x=s(t)} = \frac{T_i - T_{i-2}}{2\Delta x} + (1+\delta)\frac{T_{i-2} - 2T_{i-1} + T_i}{\Delta x} \tag{14}$$

Merging equations (13) and (14). The interface velocity is given by

$$\frac{ds}{dt} = -\frac{k_l}{\rho L}\left(\frac{T_i - T_{i-2}}{2\Delta x} + (1+\delta)\frac{T_{i-2} - 2T_{i-1} + T_i}{\Delta x}\right) \tag{15}$$

To summarize, the system of ordinary differential equation that describes the problem is given by equations (10), (11) and (15).

5.1 EVENT LOCATION

However, as Eqs. (10) and (11) show, the model experiences structural changes in the definition of the ordinary differential equation. To overcome these difficulties, the model was implemented in MATLAB (MATLAB, 2004) and the event-function in the ODE solver was activated. The event location strategy was taken from (Furenes and Lie, 2006). The changes are examples of events, and two types exist. Time events are events which occur at a given time, whereas state events are events which ocurr when the subsystem reaches some conditions. Because most ODE solvers are founded on an hypothesis of smoothness in the continuous subsystem, integration along discontinuities without event location may cause severe inefficiency, simulation failures, or incorrect event sequences to be generated. The points we need to find are given when event functions vanish, where $y(t)$ is the state vector at a given time. Finding

these points is called event location. Sometimes, the integration needs to terminate at the time of the event, and possibly restart integrating the states with initial values and state definition depending on the termination time. Most continuous simulation languages have capabilities of locating events. If an ODE solver has the capability of locating events, the step size may be adapted to hit the instants of time when the discontinuities occur. A specification of what kind of events need to be located, and what the solver must do when an event occurs, and the only additional program lines necessary. Numerical results have shown that ODEs with event location has been solved accurately at a cost scarcely greater than when no events occur. During a numerical integration step, it is important to ensure that the model is not allowed to switch form one definition to another. Thus we need the capability of terminating the integration temporarily each time the interface crosses a grid line. The syntax of the ODE solver in MATLAB is

```
[t,y]=solver(@odefun,tspan,y0)
```

Where solver is one of $ode45, ode23, ode113, ode15s, ode23s, ode23t$. The solvers are based on different recurrence algorithms and have different orders of accuracy, odefun is a function that evaluates the right-hand side of the differential equation, `tspan` is a vector specifying the time interval of integration, and $y0$ is the vector of initial conditions. The ODE solver returns a column vector of time points (t), and a solution array (y) in which each arrow corresponds to the solution at the time points in t. In some cases, the ODE solver performance can be improved by overriding these defaults. This can be done by supplying the solvers with one or more property values in an options structure. The available integration properties depend on the ODE solver used. The argument is created with the `odeset` function named `process_events`, The events property is created by

```
options=odeset('Events', @process_events)
```

The ODE solver that solves for the default case while also finding where the events functions of (t, y) are zero, has the syntax

```
[t,y,te,ie]=solver(@solid_process, tspan,y0,options)
```

Where `solid_process` is the name of the ode function. If an events function is specified and events are detected, the ODE solver returns three additional outputs. If an events function is specified and events are detected, the ODE solver returns three additional outputs: a column vector of times at which events occurred `(te)`, solution values corresponding to these times `(ye)`, and the indices of the event that the solver detected `(ie)`. For each events function it must be specified whether the integration is to terminate at a zero and whether the direction of the zero crossing matters.

6 RESULTS

Numerical calculations are sensitive to the choice of the number of node points, and the value $N = 51$ was chosen after grid sensitivity study. The comparison between analytical and numerical solution for interface liquid-solid position is showed in Fig. 3 for Stefan numbers, $St = 0.1, 1$ and 10. The temperature field at certain fixed point are compared with the analytical solution in Fig. 4 for $St = 0.1$ and Fig. 5 for $St = 10$. The results indicate that the method of lines is able to accurately track the moving liquid-solid interface an temperature history.

7 CONCLUSIONS

A numerical study for the Stefan problem has been developed based on the method of lines. For simplicity, the discretization equation was derived for uniform grid spacing. To validate the present method, one-dimensional half space melting problem were computed. The present

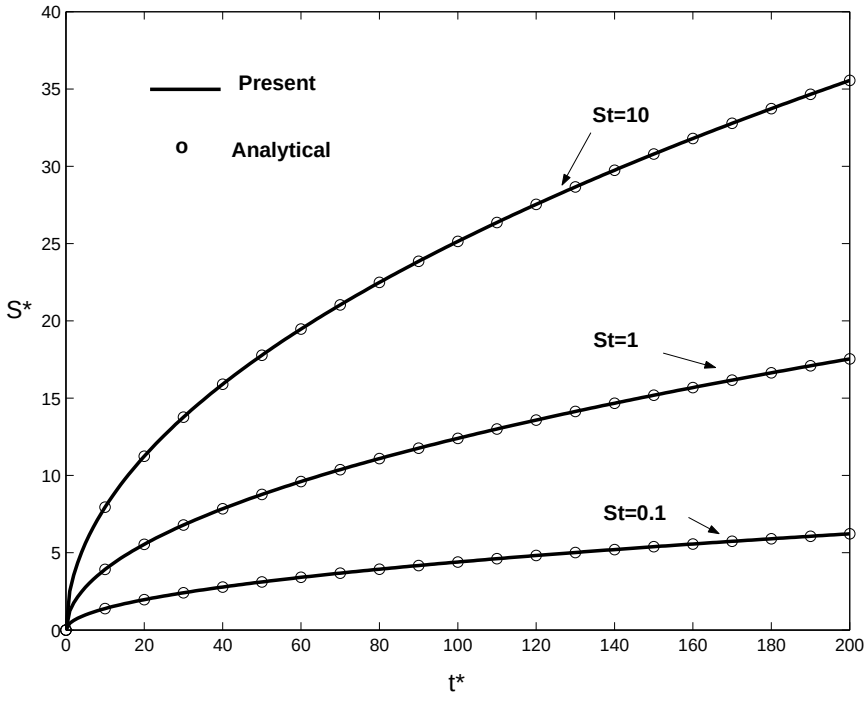

Figure 3: Dimensionless interfacial position as a function of dimensionless time for different Stefan numbers, $St = 0.1, 1$ and 10.

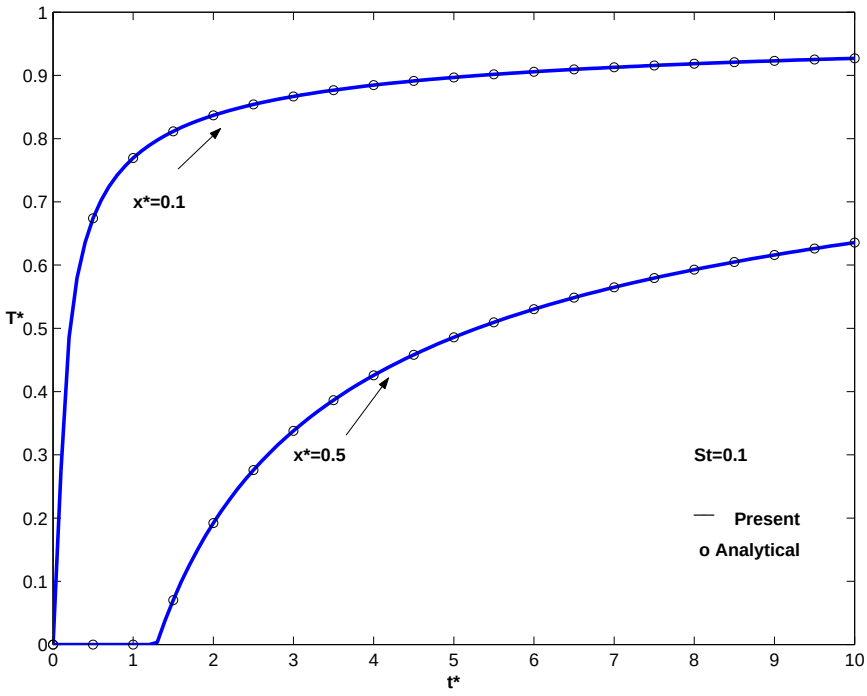

Figure 4: Dimensionless Temperature as a function of dimensionless time for different x* position, $St = 0.1$.

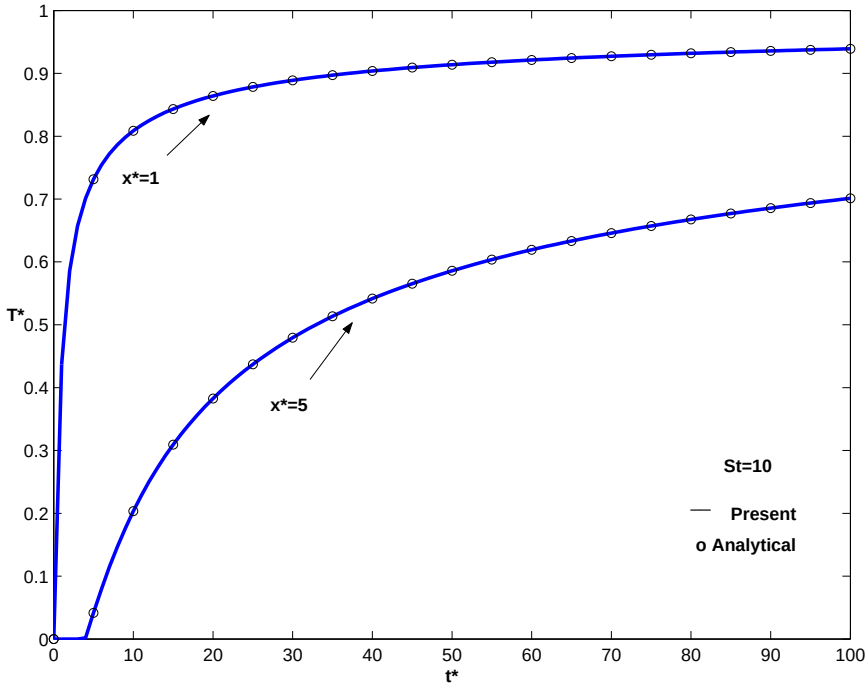

Figure 5: Dimensionless temperature as a function of dimensionless time for different x* position, $St = 10$.

method yields an oscillation-free solution, since the phase boundary is treated as a line rather than a control volume. All numerical results obtained by the present method agree very well with analytical solution across a wide range of Stefan numbers.

REFERENCES

Campo A., Tebeest K., Lacoa U., and Morales J. Application of a finite volume based method of lines to turbulent forced convection in circular tubes. *Numerical Heat Transfer*, 30(5):503–517, 1996.

Chun C. and Park S. A fixed-grid finite-diffrence method for phase-change problems. *Numerical Heat Transfer, Part B.*, 38:59–73, 2000.

Furenes B. and Lie B. Using event location in finite-difference methods for phase-change problems. *Numerical Heat Transfer, Part B.*, 50:143–155, 2006.

MATLAB. *MATLAB 7*. The Mathworks, USA, 2004.

Ozisik M. *Heat Conduction*. Wiley, New York, 1980.

Salazar A., Raydan M., and Campo A. Theoretical analysis of the exponential transversal method of lines for the diffusion equation. *Numerical Methods for Partial Differential Equations*, 16(1):30–41, 2000.

Stefan J. *Sitzber. Wien. Akad. Mat. Naturw.* URSS, 1889.

Verma A., Chandra S., and Dhindaw B. An alternative fixed grid method for solution of the clasical one-phase stefan problem. *Applied Mathematics and Computation*, 158:573–584, 2004.