Available online at www.sciencedirect.com
SCIENCE DIRECT®
Journal of Computational Physics 198 (2004) 243–264
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

Numerical simulation of melting in two-dimensional cavity using adaptive grid

Jure Mencinger *

Nova Gorica Polytechnic, 5000 Nova Gorica, Slovenia

Received 6 July 2003; received in revised form 7 January 2004; accepted 12 January 2004
Available online 27 February 2004

Abstract

This paper presents a numerical simulation of melting of chemically pure material in two-dimensional square cavity. A single-domain model is used which does not require interface tracking and allows the use of a fixed grid in order to solve governing transport equations. However, a fully conservative control volume method, using *r*-adaptive moving grid, is implemented to enhance the precision. Simple, user-defined function is introduced to control the density of the grid generated with a robust elliptic generator. The simulation is tested extensively on a one-dimensional phase change problem which enables the comparison of the results with a existing analytical solution. Finally, the method is assessed on cases corresponding to melting of tin and octadecane driven by natural convection.

Keywords: Melting; Grid generation; Adaptive moving grid

1. Introduction

An accurate prediction of the time evolution of a solid–liquid phase change system is not a trivial task, even though the mechanisms of solid–liquid phase change are well known. For example, comparison exercise [1] for the case of melting of a chemically pure substance in a two-dimensional (2D) square cavity, driven by natural convection, shows substantial differences among contributed solutions. Moreover, only 6 out of 13 contributors solved the problem for all four proposed sets of relevant dimensionless numbers, illustrating the need for further development of accurate and robust methods. This paper presents a proposal and an investigation of such a method which is tested on the previously cited problem.

The physical models of the phase change systems on the continuum scale can be divided in two categories, depending on the approach: multi-domain and single-domain. Within the multi-domain approach the transport equations describing the conservation of energy momentum, and other relevant quantities are

* Present address: Laboratory for Fluid Dynamics and Thermodynamics, Faculty of Mechanical Engineering, University of Ljubljana, Aškerčeva 6, 1000 Ljubljana, Slovenia. Tel.: +386-1-47-71-153.
E-mail address: jure.mencinger@fs.uni-lj.si (J. Mencinger).

0021-9991/$ - see front matter
doi:10.1016/j.jcp.2004.01.006

written for each phase separately. Generally, the phase interface moves, therefore the definition domain for each partial differential equation (PDE) changes. Thus, the numerical solution of the relevant PDEs requires interface-tracking. The case of melting in a cavity was investigated with this approach by some of the earliest authors [2–4], who employed the coordinate transformations to map the irregular physical domain to rectangular computational space. They calculated the new interface position and the corresponding coordinate transformation after achieving stationary state. However, this quasi-stationary assumption is limited to cases with relatively slow interface movement. Also, stationary state may not exist even with fixed interface position. More recently, an interface-tracking control-volume finite-element method using a moving non-structured grid was implemented for melting and solidification problems by Wintruff et al. [5]. In their method the grid adapts to the solid–liquid interface, which is calculated from the heat flux balance and the grid quality is controlled with a local grid refinement method. This method becomes troublesome when the topology of the interface changes (merging or breaking) and is rather complicated to be implemented for three-dimensional (3D) problems. Also, the interface-tracking is not suitable for the cases where a phase change region appears (i.e. a solidification of a mixture) instead of a smooth interface.

The need for the interface-tracking and the grid adaptation can be evaded by the single-domain approach, within which the relevant physical quantities are continuous and their corresponding conservation PDEs are valid throughout the entire physical system, regardless of the phase. As the definition domain for the PDEs is not moving, a fixed grid can be used to obtain numerical solutions. Single-domain approach is utilized by the introduction of a phase fraction function $f_k(\mathbf{r})$ so that: $f_k = 1$ if $\mathbf{r}$ inside kth phase, $f_k = 0$ if $\mathbf{r}$ outside kth phase and $f_k \in]0,1[$ in interphase region; more formal definition of f_k can be found elsewhere (e.g. volume averaging method in [6]). A typical representative of single-domain approach is the enthalpy method [7], where the heat flow at the interface is comprised by specifying a volume source term in a phase change region. The common practice [8] to simultaneously calculate the velocity field in liquid phase and deal with fixed velocity condition in solid phase is to treat the solid phase in interphase region as a porous medium with the permeability prescribed as a function of the local phase volume fraction. Bennon and Incropera [9] established this methodology in a more formal manner in their continuum model, which was developed for the treatment of multicomponent phase change systems. Its main feature is the treatment of solid and liquid phase as an interpenetrating solid–liquid mixture. On engineering scale, the interphase region (where $f_k \in]0,1[$) well represents the so-called mushy zone appearing in cases such as solidification of binary mixture. For isothermal phase change, on the other hand, the existence of interphase region is unrealistic. However, the solutions obtained with the described approach can be considered realistic if the region is sufficiently thin, i.e. when 'spreading' of the interface to the phase change region is not too excessive. As the thinness of the interphase region is limited with grid cell size, thickness can be reduced by increasing grid density at the region. This can be done by using an adaptive grid technique.

Lacroix and Voller [10] compared the single-domain fixed grid method with the multi-domain transformed grid method. They recognized that the need for a grid generation and complexity of the implementation were the major drawbacks of the multi-domain approach. On the other hand, they specified the smearing of the solid–liquid interface and ambiguity in defining the permeability coefficient as important disadvantages of the single-domain approach. Because of the the solid–liquid interface smearing, the single-domain approach is generally considered less accurate for isothermal phase change problems. However, it is more robust as it is not sensitive to the changes of the interface topology. Also, it is more general as it can be used for cases with the phase change emerging in a temperature interval. Lacroix and Voller suggested a hybrid method using one-domain approach and adaptive moving grid to ensure larger grid density in the vicinity of the interface and to increase the accuracy of the solution. Such a moving grid method, using finite elements, has been developed and tested recently by Mackenzie and Robertson [11] for one-dimensional (1D) phase change problems and later by Beckett et al. [12] for two-dimensional problems. To generate grid, they utilized a coupled set of parabolic PDEs; the grid density was monitored with the function depending on the distance from the calculated numerical approximation of the phase front.

It is quite obvious that the precision of the solutions, using single-domain approach, can be increased with the use of the adaptive grid. To claim effectiveness of this approach, the time for the grid adaptation process should not exceed the time necessary to solve transport PDEs. This paper tries to establish robust elliptic grid generator, well suited for the described problem. Grid density control is based on Anderson's work [13], i.e. with a simple user-defined function. The aim of the paper is also to evaluate the precision of such an approach. This is done with the comparison of the computed and existing analytical solution for one-dimensional case of melting with conduction. Finally, the effectiveness of the presented method is assessed with simulations of the above mentioned test cases [1].

The outline of the paper is the following: Section 2 introduces the problem of melting in a square cavity and relevant governing equations. The proposed solution method is described in Section 3 with emphasis on adaptive grid generation. Section 4 presents the method's test on melting with conduction, whereas in Section 5 the results of the application to the problem of melting driven by natural convection are shown. Section 6 is reserved for conclusions.

2. Definition of the problem

The problem [1] considers a two-dimensional square cavity, initially filled with a solid chemically pure substance, uniformly at the melting temperature T_M. The walls of the cavity are impermeable and no-slip. Horizontal walls are adiabatic, whereas the left and the right wall are set to the fixed temperatures $T_l > T_M$ and $T_r = T_M$, respectively. As melting starts near the left wall, natural convection takes place in liquid phase because of the existing temperature gradient. The problem is shown schematically in Fig. 1; it is simplified with the following assumptions: (1) the fluid is incompressible, (2) densities of both phases are equal, (3) Boussinesq approximation [14] can be used, (4) liquid viscosity is constant, (5) specific heats of both phases are constant and equal, (7) both thermal conductivities are constant and equal and (8) the solid phase is fixed to the cavity walls. Those assumptions enable the characterization of the problem by three dimensionless parameters: Prandtl, Rayleigh and Stefan number, denoted as *Pr*, *Ra* and *Ste*, respectively.

As stated in Section 1, melting is modelled with single-domain continuum model [9]. Applying it to the above defined problem results in the following governing transport equations:

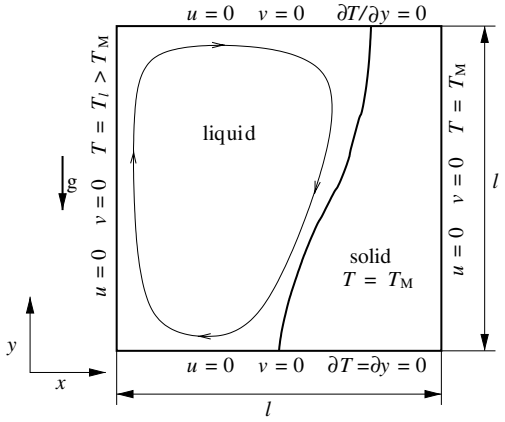

Fig. 1. Schematics of the considered problem.

$$\frac{\partial H}{\partial Fo} + \mathbf{V}\cdot\nabla H = \nabla^2\Theta + \frac{1}{Ste}\mathbf{V}\cdot\nabla f_l, \tag{1}$$

$$\frac{\partial \mathbf{V}}{\partial Fo} + \mathbf{V}\cdot\nabla\mathbf{V} = -\nabla P + Pr\nabla^2\mathbf{V} + Pr\,Ra\Theta\mathbf{j} - \mathbf{V}\cdot\nabla(\mathbf{V}_l - \mathbf{V}) - Pr\,\Pi\frac{(1-f_l)^2}{f_l^3+\delta_c}\mathbf{V}, \tag{2}$$

where H, Θ, $\mathbf{V}$, P and Fo are dimensionless specific enthalpy, temperature, velocity, pressure and time (i.e. Fourier number), respectively. In the used model, the solid–liquid mixture is considered incompressible, therefore $\nabla\cdot\mathbf{V} = 0$. The second term on the right-hand side (RHS) in (1) is the correction of the substantial derivative due to the relative phase movement and has nonzero value in the interphase region ($0 < g_l < 1$) only. The same applies for the fourth term on the RHS of (2). The fifth term on the RHS of (2) presents Darcian damping force insuring zero velocity of solid phase; Π presents dimensionless reciprocal permeability constant, whereas δ_c is small number used solely to prevent division by zero in calculations. This term vanishes in liquid phase only regions ($f_l = 1$), where (2) becomes standard (dimensionless) Navier–Stokes equation.

The specific enthalpy–temperature relation reads

$$H = \Theta + \frac{f_l}{Ste}. \tag{3}$$

By the definition of dimensionless temperature $\Theta = (T - T_M)/(T_l - T_M)$, solid and liquid phase are characterized by negative and positive value of Θ, respectively. However, in order to solve (1) numerically, the approximation of the phase change within the temperature interval $[-\Delta\Theta, \Delta\Theta]$ ($\Delta\Theta > 0$) is introduced. Thus, f_l equals

$$f_l = \min\left(1, \max\left(0, \frac{\Theta - \Delta\Theta}{2\Delta\Theta}\right)\right). \tag{4}$$

Even though the model was developed for the cases where phase change appears within temperature interval, it can also be used for the isothermal phase change problems. Hypothetically existing exact solution of (1) and (2) would exactly match the hypothetically existing solution of the corresponding equations obtained with the multi-domain formulation. In other words, by using the continuum model no error is introduced to the solution of the problem.

3. Method of solution

The above transport equations are solved numerically with the control volume (CV) method [15] on structured adaptive moving grid. The specifics of the implementation can be summarized as follows:

- Field values at CV faces are expressed with linear interpolation, depending on the distance of points belonging to neighboring CVs and the middle of their common face. To enhance the stability of the solution algorithm, the deferred correction [16] is utilized: the upwind interpolation is used implicitly but it is corrected with the difference between the linear and the upwind interpolation from the previous iteration.
- The values at corners of CV needed to express fluxes on nonorthogonal grid are expressed in terms of values of the surrounding points with the bilinear interpolation.
- Fluxes resulting from the grid movement are calculated as proposed by Demirdžić and Perić [17] obeying the so-called space conservation law [18].
- Pressure–velocity coupling is implemented via the popular SIMPLE algorithm [19].

- The grid points for the calculation of the velocity components and the pressure are collocated; well known checkerboard pressure problem is avoided as suggested by Rhie and Chow [20].
- Fully implicit temporal differentiation is employed to march solution forward in time with time step ΔFo.
- The discretization of transport equations on a 2D nonorthogonal structured grid results in the algebraic equation which can be written in standard form $a_P\phi_P + \sum_{nb} a_{nb}\phi_{nb} = b_P$ with the coefficients a and the non-homogeneous term b; the index nb runs over the neighbor CVs forming a nine-point computational molecule in 2D. The system of equations obtained by the discretization on all CVs is solved iteratively with the Modified Strongly Implicit Procedure (MSIP) [21]. For the pressure correction equation, multigrid version of MSIP is used.

Further details of the discretization are omitted for briefness and can be obtained from author upon request. The solution algorithm can be summarized with the following scheme:
(1) initialization $Fo = 0.0$, $\mathbf{V} = 0.0$, $H = H_0$;
(2) calculation of a grid shape control function Λ;
(3) grid generation;
(4) calculation of geometrical coefficients and grid velocity;
(5) beginning of the time step $Fo := Fo + \Delta Fo$;
(6) calculation of temperature Θ and liquid phase fraction f_l;
(7) solution of the equation for the specific enthalpy H field;
(8) solution of the equation for the velocity $\mathbf{V}$ field (SIMPLE algorithm);
(9) *if* convergence is reached *then goto* step 2, *else goto* step 5.
The convergence inside the time step is recognized when the norms of the residuum of all calculated fields (H and $\mathbf{V}$) are reduced by a factor ε compared to those from the beginning of the time step.

In the presented algorithm the grid shape for the $(n+1)$th time step is controlled with the grid control function Λ which depends on the calculated fields from the previous, i.e. nth time step. This strategy assumes that Λ changes relatively slowly in time, thus the solution can be presented well on the grid which depends on the solution from the previous time step. If this was not true, then the grid would have to be generated after each iteration inside the time step. This would however increase the total computing time required for the grid generation. In the rest of this section, the grid adaptation method is described.

3.1. Grid generation

A new grid is generated after each time step using elliptic grid generation with grid density control [13], i.e. by solving a system of elliptic PDEs

$$\nabla \cdot (\Lambda^{-1}\nabla\xi) = 0 \quad \text{and} \quad \nabla \cdot (\Lambda^{-1}\nabla\eta) = 0 \tag{5}$$

for the computational coordinates ξ and η running in S–N and W–E directions, respectively. The density of the coordinate lines ($\xi = \text{const.}$ and $\eta = \text{const.}$) is proportional to user-defined scalar function Λ. To construct computational grid, physical coordinates must be expressed in terms of computational coordinates [22]. This is done by interchanging the role of dependent and independent variables in (5); after some mathematical manipulation Anderson [13] obtains

$$\alpha\varphi_{\xi\xi} - 2\beta\varphi_{\xi\eta} + \gamma\varphi_{\eta\eta} = -\Lambda^{-1}\left([\alpha\varphi_\xi - \beta\varphi_\eta]\Lambda_\xi + [\gamma\varphi_\eta - \beta\varphi_\xi]\Lambda_\eta\right), \tag{6}$$

where α, β and γ are the components of covariant metric tensor and φ is either of the Cartesian coordinates X or Y. It should be noted that Anderson used function D which is inversely proportional to grid density. Here, $\Lambda \equiv D^{-1}$ is used simply because of easier interpretation. By using the product rule for differentiation (6) is changed to

$$\alpha\varphi_{\xi\xi} - 2\beta\varphi_{\xi\eta} + \gamma\varphi_{\eta\eta} = \frac{1}{2}\Lambda\Big(\alpha[\Lambda^{-1}\varphi_{\xi}]_{\xi} - \beta[\Lambda^{-1}\varphi_{\eta}]_{\xi} - \beta[\Lambda^{-1}\varphi_{\xi}]_{\eta} + \gamma[\Lambda^{-1}\varphi_{\eta}]_{\eta}\Big). \tag{7}$$

The above form is, in authors view, more suitable for discretization. Namely, the situation is analogous when, for example, the difussive flux $[k\phi_x]_x$ is discretized directly or as $k_x\phi_x + k\phi_{xx}$; only the use of proper relation between the value of k at grid points and the value at intermediate points leads to the equivalent result. Eq. (7) is discretized with the finite differences method; the coefficients of resulting algebraic equation $a_{\mathrm{P}}\varphi_{\mathrm{P}} + \sum_{nb} a_{\mathrm{nb}}\varphi_{\mathrm{nb}} = 0$ are:

$$\begin{aligned}
a_{\mathrm{E}} &= \alpha\left(1 - \frac{1}{2}\frac{\Lambda_{\mathrm{P}}}{\Lambda_{\mathrm{e}}}\right) + \beta\frac{\Lambda_{\mathrm{P}}}{8}(\Lambda_n^{-1} - \Lambda_{\mathrm{s}}^{-1}),\\
a_{\mathrm{W}} &= \alpha\left(1 - \frac{1}{2}\frac{\Lambda_{\mathrm{P}}}{\Lambda_{\mathrm{w}}}\right) - \beta\frac{\Lambda_{\mathrm{P}}}{8}(\Lambda_n^{-1} - \Lambda_{\mathrm{s}}^{-1}),\\
a_{\mathrm{N}} &= \alpha\left(1 - \frac{1}{2}\frac{\Lambda_{\mathrm{P}}}{\Lambda_n}\right) + \beta\frac{\Lambda_{\mathrm{P}}}{8}(\Lambda_{\mathrm{e}}^{-1} - \Lambda_{\mathrm{w}}^{-1}),\\
a_{\mathrm{S}} &= \alpha\left(1 - \frac{1}{2}\frac{\Lambda_{\mathrm{P}}}{\Lambda_{\mathrm{w}}}\right) - \beta\frac{\Lambda_{\mathrm{P}}}{8}(\Lambda_{\mathrm{e}}^{-1} - \Lambda_{\mathrm{w}}^{-1}),\\
a_{\mathrm{NE}} &= -\frac{1}{2}\beta\left(1 - \frac{\Lambda_{\mathrm{P}}}{4}(\Lambda_n^{-1} + \Lambda_{\mathrm{e}}^{-1})\right),\\
a_{\mathrm{NW}} &= \frac{1}{2}\beta\left(1 - \frac{\Lambda_{\mathrm{P}}}{4}(\Lambda_n^{-1} + \Lambda_{\mathrm{w}}^{-1})\right),\\
a_{\mathrm{SE}} &= \frac{1}{2}\beta\left(1 - \frac{\Lambda_{\mathrm{P}}}{4}(\Lambda_{\mathrm{s}}^{-1} + \Lambda_{\mathrm{e}}^{-1})\right),\\
a_{\mathrm{SW}} &= -\frac{1}{2}\beta\left(1 - \frac{\Lambda_{\mathrm{P}}}{4}(\Lambda_{\mathrm{s}}^{-1} + \Lambda_{\mathrm{w}}^{-1})\right),\\
a_{\mathrm{P}} &= a_{\mathrm{E}} + a_{\mathrm{W}} + a_{\mathrm{N}} + a_{\mathrm{S}},
\end{aligned} \tag{8}$$

where capital subscripts, in accordance with commonly used compass notation [15], denote points of nine point computational molecule, whereas small subscripts denote intermediate points. When calculating coefficients (8), Λ is first interpolated to positions ne, nw, se and sw. Then, Λ_{P} is calculated as $\Lambda_{\mathrm{P}} = \frac{1}{4}(\Lambda_{\mathrm{ne}} + \Lambda_{\mathrm{nw}} + \Lambda_{\mathrm{se}} + \Lambda_{\mathrm{sw}})$ and Λ_{e}, for example, as $\Lambda_{\mathrm{e}} = \frac{1}{2}(\Lambda_{\mathrm{ne}} + \Lambda_{\mathrm{se}})$.

For the considered cavity, the following boundary conditions are employed:

$$X(\xi_{\min}) = 0, \quad X(\xi_{\max}) = 1, \quad X_{\eta}(\eta_{\min}) = 0, \quad X_{\eta}(\eta_{\max}) = 0, \tag{9}$$

$$Y(\eta_{\min}) = 0, \quad Y(\eta_{\max}) = 1, \quad Y_{\xi}(\xi_{\min}) = 0, \quad Y_{\xi}(\xi_{\max}) = 0, \tag{10}$$

where $\xi_{\min}$, $\xi_{\max}$, $\eta_{\min}$ and $\eta_{\max}$ are values of ξ and η at left, right, bottom and top wall of the cavity, respectively. Boundary conditions (9) and (10) ensure grid orthogonality at the boundaries and thus the boundary conditions for (1) and (2) are simpler to implement.

To generate grid (7) is solved with MSIP interchangeably for X and Y, updating the coefficients (8) after every new $X - Y$ solution, until the convergence is reached. The grid from previous time step is used as the starting approximation.

3.2. Grid density control

In presented work the control function Λ is calculated as follows. First, function Λ^0 is defined as a linear combination

$$\Lambda^0 = 1 + \sum_n C_n \|f_n\|, \quad \|f_n\| = f_n \frac{\int \mathrm{d}V}{\int |f_n| \, \mathrm{d}V}, \tag{11}$$

where $\|f_n\|$ is a normalized function of calculated fields and C_n is a weighting coefficient. The normalization is used to maintain the relative influence of f_n in Λ^0, determined by C_n. As Λ is user defined function, unnormalized functions can also be used when appropriate. However, when f_n is a gradient of a discontinuous function the normalization is necessary, as shown later in this section.

Λ is obtained by smoothing Λ^0 with Laplace operator, i.e. by time integration of the diffusion equation:

$$\frac{\partial \Lambda}{\partial \tau} = \nabla^2 \Lambda, \quad \Lambda(\tau = 0) = \Lambda^0 \tag{12}$$

in time interval $[0, \tau_D]$, where τ is used to avoid confusing with the physical time t of the process. Therefore, Λ^0 is the initial state of Λ and τ_D presents the smoothing factor. However, the diffusion length $L_D = 2\sqrt{\tau_D}$ is used instead of τ_D because of its easier physical interpretation. On boundaries, Neumann's conditions $\partial \Lambda / \partial n = 0$ are set. Eq. (12) is solved with CV method using the same grid as for transport equations (1) and (2), i.e. the grid Λ^0 was calculated on. Fully implicit time discretization is implemented; the resulting system of linear equations is solved with MSIP. The size of time step $\Delta\tau = \tau_D / n_D$, where n_D is number of steps, influences the solution's accuracy, however it does not seem essential for the accuracy of the overall solution. In this work n_D was set to 5.

Another issue has to be addressed in this section. Namely, in presented application, gradient of the discontinuous function f_1 is used to refine the grid near the interface. The calculated value of $\|\nabla f_1\|$ is

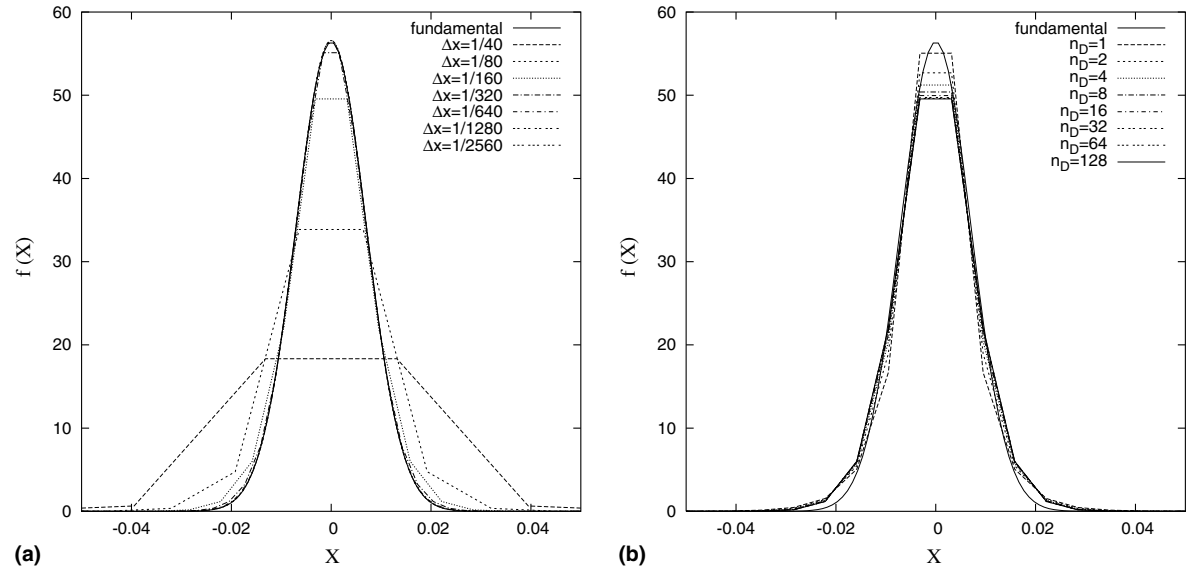

Fig. 2. $\|\nabla H(x)\|$ smoothed with $L_D = 0.01$ (a) on different grids ($n_D = 128$) and (b) using different number of time steps n_D ($\Delta x = 1/160$).

proportional to grid density. As simultaneously, grid density is proportional to $\|\nabla f_1\|$, this would seemingly result in unbounded value of $\|\nabla f_n\|$ and 'infinite' grid density at discontinuity. However, this is prevented with the above described normalization and smoothing of Λ. To demonstrate this, smoothed $\|\nabla H(x)\|$ is calculated on one-dimensional successively refined uniform grid on domain $[-0.5, 0.5]$ where $H(x)$ is Heaviside step-function. As Fig. 2(a) shows, the smoothed $\|\nabla H(x)\|$ converges to diffusion equation's fundamental solution $(\sqrt{\pi}L_D)^{-1}\exp(-(x/L_D)^2)$. This is not surprising as $\|\nabla H(x)\|$ can be interpreted as discretized Dirac δ function, i.e. fundamental solution at $\tau_D = 0.0$. Fig. 2(b) shows the influence of n_D on smoothing precision.

The described grid density control function determination may seem complicated when compared for example with the methodology used by Beckett et al. [12], who used already smooth function $1 + \mu_1/\sqrt{(\mu_2^2|\mathbf{r} - \mathbf{r}_*|^2 + 1)}$ with $\mathbf{r}_*$ the closest point to the phase front. In their method the grid density can easily be controlled with parameters μ_1 and μ_2, on the other hand, $\mathbf{r}_*$ needs to be determined. This can be rather complicated in 3D, as authors recognize, whereas the implementation of the presented method in 3D is straight forward. Also, the presented method is more general as grid density can easily, besides being increased near the interface, be adjusted with respect to other calculated fields.

4. Melting by conduction

When convection is not present, the problem becomes essentially one-dimensional, depending only on the coordinate X. In this case, Neumann's analytical solution [23] describes the temperature field

$$\Theta^{\text{anal.}}(X, Fo) = \begin{cases} 1 - \frac{1}{\operatorname{erf}(\alpha)}\operatorname{erf}\left(\frac{X}{2\sqrt{Fo}}\right), & 0 \leqslant X \leqslant X_i^{\text{anal.}} \\ 0, & X > X_i^{\text{anal.}} \end{cases} \tag{13}$$

where the interface position $X_i^{\text{anal.}}$ is determined by $X_i^{\text{anal.}} = 2\alpha\sqrt{Fo}$. The parameter α depends on Ste via the relation $\sqrt{\pi}\alpha e^{\alpha^2}\operatorname{erf}(\alpha) = Ste$. The above solution was obtained originally for a semi-infinite body but it is also valid for the presented case until phase interface reaches the right wall of the cavity at $Fo = (1/4)\alpha^{-2}$. Neumann's solution is therefore used for testing and assessment of the presented method by comparing the calculated and the analytical values of the interface position X_i and the heat flux at the left wall – Nusselt number Nu. The heat flux at the left wall $Nu^{\text{anal.}}$ is obtained with derivation of (13)

$$Nu^{\text{anal.}} = -\frac{\partial\Theta^{\text{anal.}}}{\partial X}\bigg|_{X=0} = \frac{1}{\operatorname{erf}(\alpha)\sqrt{\pi Fo}}. \tag{14}$$

Presented are the calculations for $Ste = 0.1$ ($\alpha = 0.220016$); the corresponding analytical values $X_i^{\text{anal.}}$ and $Nu^{\text{anal.}}$ are shown in Fig. 3. For 1D case, the system of algebraic equations, resulting from discretization, can be solved directly with the tridiagonal matrix algorithm; so 1D solver (i.e. code) was used to obtain solutions presented in this section. As the 1D solver is faster than 2D solver, the convergence criteria inside the time step can be defined much more rigorously, i.e. by using very small number $\varepsilon = 10^{-9}$. Therefore, only discretization error may be considered in this section. The size of the melting interval is defined by setting $\Delta\Theta = 10^{-5}$; reducing $\Delta\Theta$ does not have noticeable influence on the calculated results.

The interface position in 1D case equals the total liquid fraction in the cavity and is therefore obtained as $X_i = \int_0^1 f_l \, dX = \sum_j f_{lj}\Delta V_j$. Nu is calculated simply as the temperature difference of near boundary points divided by their distance: $Nu = -(T_2 - T_1)/(X_2 - X_1)$, where 1 is index of the left boundary point.

If fixed uniform grid is used to solve the heat transfer equation, the calculated Nu exhibits well-known step-like or wavy pattern (Fig. 4) which was described, for example, by Date [24]. The reason for this

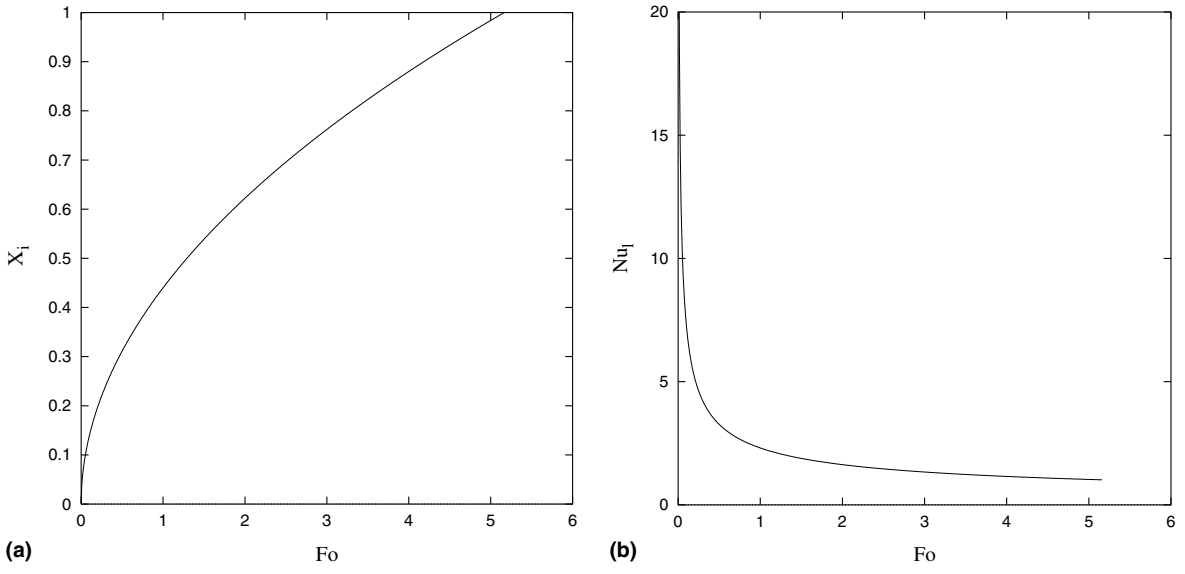

Fig. 3. Analytical values $X_i^{\text{anal.}}$ (a) and $Nu^{\text{anal.}}$ (b) corresponding to $Ste = 0.1$.

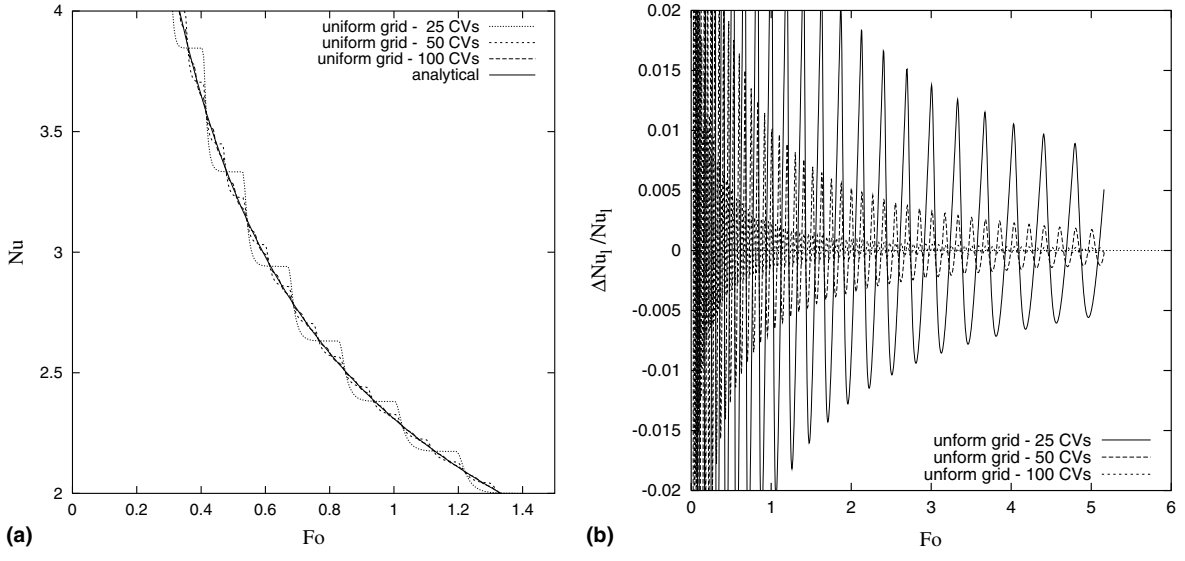

Fig. 4. Nu (a) and relative error $\Delta Nu/Nu^{\text{anal.}}$ (b) calculated on uniform grids with 25, 50 and 100 CVs.

pattern is the following: the temperature of the CV at which melting takes place – 'the melting CV' is within the melting temperature interval, i.e. almost constant and the temperature field almost settles until the melting CV completely 'melts'. The calculated Nu is larger than $Nu^{\text{anal.}}$ until X_i passes the center of the melting CV and smaller afterward. This step-like pattern depends on the *Ste*; smaller *Ste* results in sharper steps. Refining the grid, i.e. reducing the size of the melting CV, results in smaller amplitude and higher frequency of the oscillations.

Table 1
Grid control functions Λ^0 and diffusion lengths L_D defining grid shape

	G_1	G_2	G_3	G_0-uniform
Λ^0	$1+\|\nabla f_1\|$	$1+\|\nabla f_1\|$	$1+2.0\|\nabla f_1\|$	1
L_D	0.05	0.01	0.05	–

The step-like pattern of Nu can be diminished by reducing the size of the melting CV which can be achieved effectively with the adaptive moving grid. On the other hand, the use of adaptive moving grid introduces additional source of the discretization error as the interpolation on moving CV boundaries is needed; the linear interpolation is used. Also upwind interpolation and SMART scheme [25] are tested in this section.

To demonstrate the influence of weighting coefficient in Λ^0 and L_D, three different grid shapes are defined; grids are marked as G_s^n where n is the number of CVs and s is shape number as written in Table 1. Fig. 5 presents grid trajectories of grids G_1^{25}, G_2^{25} and G_3^{25}. The solutions obtained on them cannot be distinguished from the analytical solution on the scale of Fig. 3. Therefore, relative errors of Nusselt number $(Nu - Nu^{\text{anal.}})/Nu^{\text{anal.}} = \Delta Nu/Nu^{\text{anal.}}$ and errors of the interface position $\Delta X_i = X_i - X_i^{\text{anal.}}$ are compared.

Figs. 6(a), (c) and (e) exhibit $\Delta Nu/Nu^{\text{anal.}}$ calculated on grids with 25, 50 and 100 CVs, respectively. Figs. 6(b), (d) and (f) present the corresponding ΔX_i. The comparison of errors on different grid shapes with the same number of CVs shows that the most accurate solution is obtained using the shape G_2. Obviously, the error depends on grid density around the phase interface – the larger the density the smaller the error. Whereas ΔNu is obviously smaller on adaptive grid, the error ΔX_i appears larger on the adaptive grids than on the uniform grid. However, Figs. 6(b), (d) and (f) may be misleading, as $|\Delta X_i|$ is much larger on the uniform grid in the beginning of the process, yet previously described step-like behavior on the uniform grid leads to oscillations of ΔX_i around zero. Namely, ΔX_i is a cumulative quantity; since the amplitude of oscillations of ΔX_i is smaller on adaptive grids than on uniform grid, this means that the solution on former is more accurate. This is especially true for solutions using grid shape G_2 for which ΔX_i in Figs. 6(b), (d) and (f) appears almost constant.

Adaptive moving grid requires small enough ΔFo in order to prevent large movement of CVs (compared to their size) in a single time step. Namely, this increases error and can also cause the divergence of the solution. Time step ΔFo in calculations, referred in Fig. 6 is set to the rather small value 10^{-6}, yet larger value can be used for most cases. The influence of ΔFo on solution's accuracy when using grid G_1^{25}, is shown in Fig. 7(a); the solution is virtually the same when $\Delta Fo = 10^{-4} - 10^{-6}$ and it is still useful for $\Delta Fo = 10^{-2}$, however, the

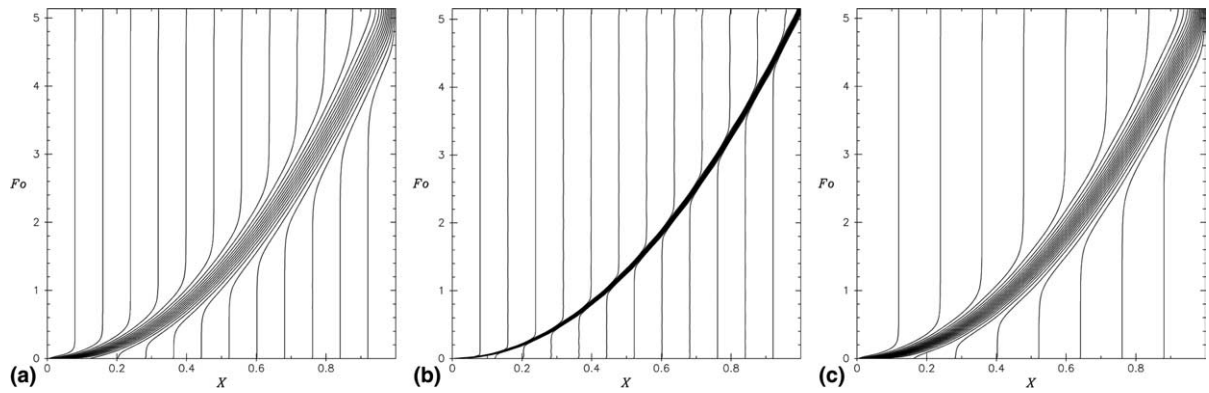

Fig. 5. Grid trajectories for grids G_1^{25} (a), G_2^{25} (b), G_3^{25} (c).

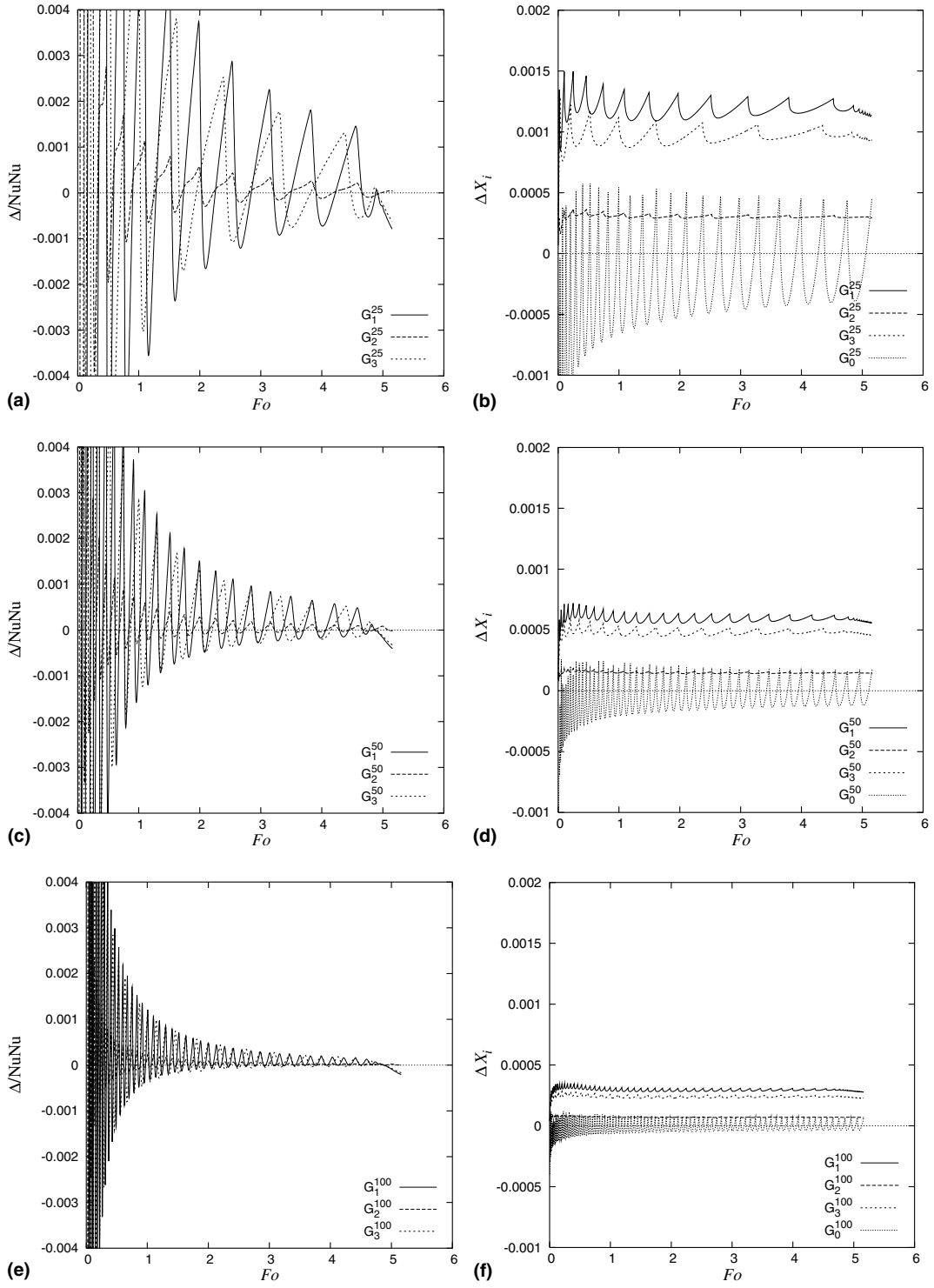

Fig. 6. Relative error $\Delta Nu/Nu^{\text{anal.}}$: (a), (c), (e); error ΔX_i: (b), (d), (f); calculated on various grids.

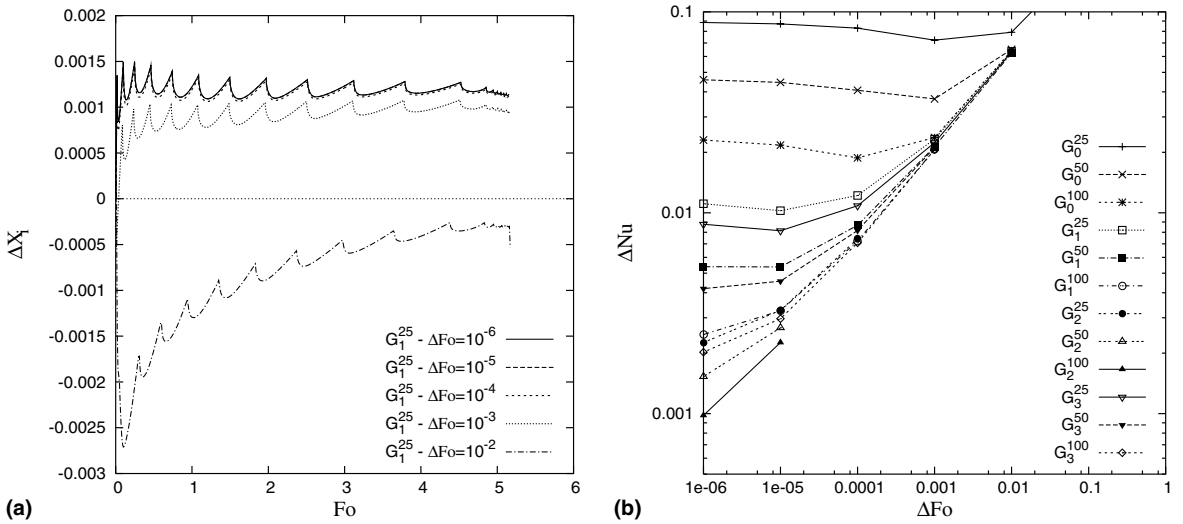

Fig. 7. Error ΔX_i calculated on grid G_1^{25} using different ΔFo (a); average error $\overline{\Delta Nu}$ calculated on various grids as function of ΔFo (b).

Table 2
Maximum ΔFo size for different grids

G_1^{25}	G_2^{25}	G_3^{25}	G_1^{50}	G_2^{50}	G_3^{50}	G_1^{100}	G_2^{100}	G_3^{100}
10^{-2}	10^{-4}	10^{-2}	10^{-2}	10^{-5}	10^{-3}	10^{-3}	10^{-5}	10^{-3}

method fails to converge for $\Delta Fo = 10^{-1}$. Fig. 7(b) displays average error $\overline{\Delta Nu}$ for different grids as function of ΔFo, where in integration $\Delta Nu(Fo = 0)$ was set to 0. Maximal ΔFo, tested in powers of 10, is written in Table 2. Clearly, the use of presented adaptive moving grid introduces additional limitation on the size of ΔFo, which

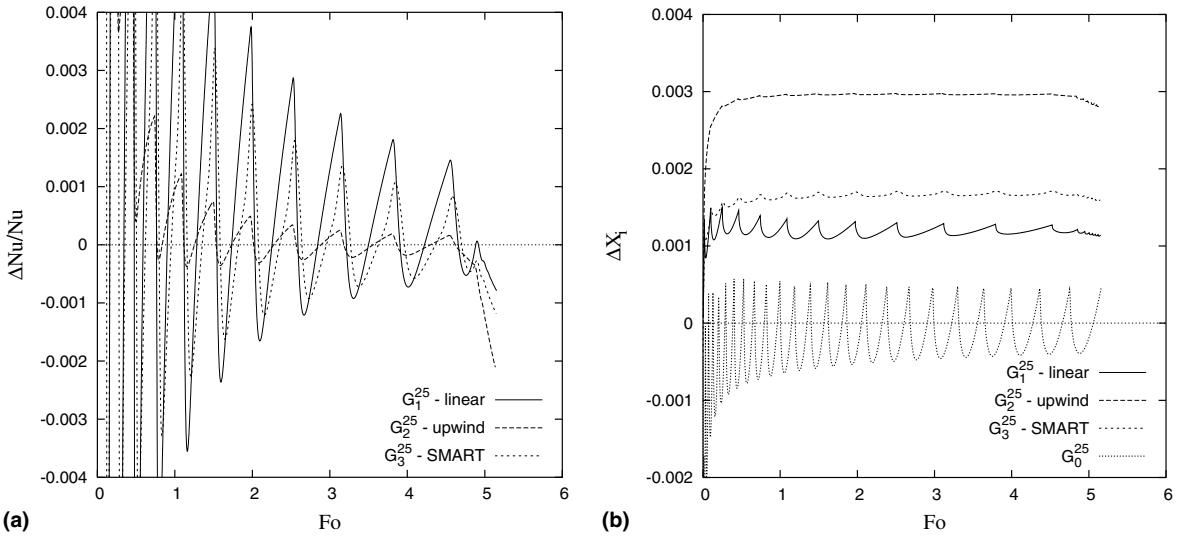

Fig. 8. Errors $\Delta Nu/Nu$ (a) and ΔX_i (b) calculated on grid G_1^{25} using different interpolation schemes.

should be smaller for denser grids. Nevertheless, in practice, this may not present real limitation for cases where the calculation of velocity and other fields requires small time step (stiff problems) for sufficient time resolution. This is also true for the cases and grids presented in the next section.

Besides linear interpolation, the method was also tested with upwind interpolation and SMART scheme [25]. The latter offers a compromise between the precision of QUICK scheme and the demand for boundedness. Comparison of the schemes for grid G_1^{25} can be seen from Fig. 8. The upwinding results in less oscillatory behavior compared with the linear interpolation and also smaller ΔNu on the chosen grids; ΔX_i is larger, yet again less oscillatory. The problem of upwinding is a tendency to smear the interface due to scheme's diffusiveness. This can cause rapid change in Λ if it depends strongly on $\|\nabla f_l\|$. Smooth grid movement is then ensured with some relaxation of Λ, so $\Lambda := \alpha_\Lambda \Lambda + (1-\alpha_\Lambda)\Lambda^*$, where Λ^* is the control function from the previous time step. Setting $\alpha_\Lambda = 0.5$ proved to be appropriate. SMART scheme exhibits similar precision as linear interpolation and due to its boundedness insures physically realistic results, i.e. temperatures within the interval $[-\Delta\Theta, 1.0]$. SMART scheme is therefore good alternative to linear interpolation, even though its implementation is somewhat more complicated.

5. Melting by convection

The presented method is tested on cases proposed by Gobin and Le Quéré [1]. The cases are defined by parameters written in Table 3. Using the assumptions stated in Section 4, cases #1 and #2 present the melting of tin in a 0.1 m high cavity, with the temperature difference $T_l - T_M = 0.3$ and 3 K, respectively. Cases #3 and #4 correspond to the melting of octadecane (paraffin) in the same cavity with a temperature difference of 1 and 10 K, respectively.

As the reference solution does not exist for the considered cases, the accuracy of the obtained numerical solutions can only be estimated. Compared with the previous section, 2D cases are computationally significantly more demanding than 1D cases. Therefore, the time step convergence criteria are less rigorous; ε is set to 10^{-3}. However, solver was tested on conduction case from previous section to produce the same results as 1D solver. In all the calculations presented in this section, $\Delta\Theta$ was set to 10^{-5}.

For briefness and also because of limited computational resources, only one grid type is used. The grid density is controlled with the function

$$\Lambda^0 = 1 + \max\left(\|\nabla f_l\|, \frac{1}{2}\|\nabla H_b\|, \frac{1}{2}\|\nabla U\|, \frac{1}{2}\|\nabla V\|, \frac{1}{2}\|\nabla \Theta\|\right), \tag{15}$$

where H_b denotes the Heaviside step-function with the 'step' at the contact of cavity walls and liquid. This way the grid density is controlled at boundaries of liquid-phase domain. Λ is obtained from Λ^0 by setting $L_D = 0.05$. Grids with 40×40, 80×80 and 128×128 CVs are used for each case. Average Nusselt number at left wall $\overline{Nu}$, total liquid fraction $\overline{f_l}$ and interface positions are compared.

Case #1 is expected to be computationally the least demanding, due to moderate Rayleigh number. The solution, i.e. streamlines, temperature field and corresponding 80×80 grid at different instants is shown in Fig. 9. Figs. 10(a) and (b) show good agreement of $\overline{Nu}$ and $\overline{f_l}$, calculated on differently dense grids. The corresponding Neumann's solution is also drawn to emphasize the relative influence of convective heat

Table 3
Dimensionless parameters defining the four cases

	Case #1	Case #2	Case #3	Case #4
Pr	0.02	0.02	50.0	50.0
Ra	2.5×10^4	2.5×10^5	1.0×10^7	1.0×10^8
Ste	0.01	0.01	0.1	0.1

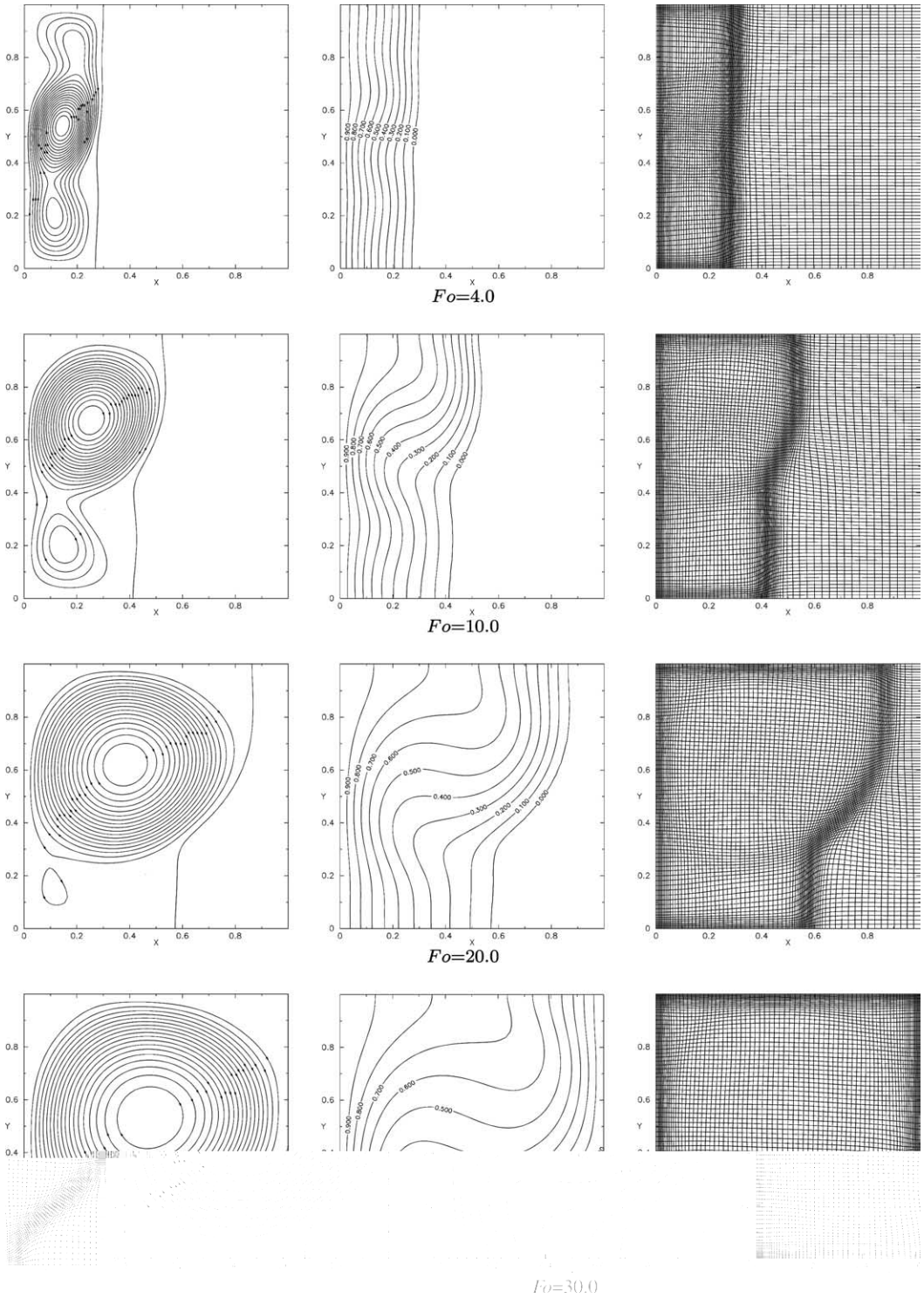

Fig. 9. Streamlines, temperature field and grid for case #1 at $Fo = 4.0$, $Fo = 10.0$, $Fo = 20.0$ and $Fo = 30.0$ calculated using 80×80 CVs.

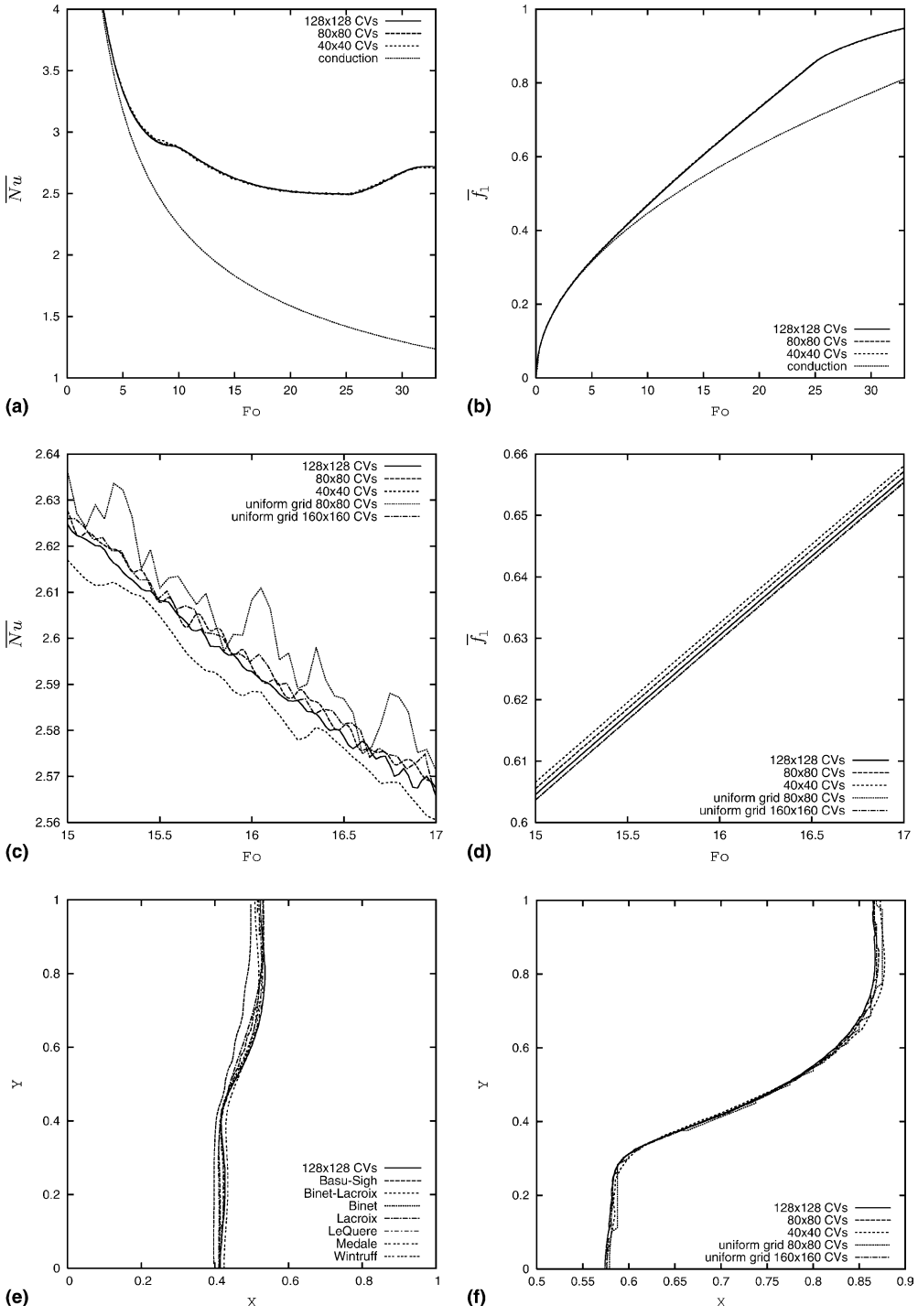

Fig. 10. Results for case #1: $\overline{Nu}$ at left wall (a), total liquid fraction f_l (b) and corresponding magnified pictures (c) and (d), interface positions: compared with [1] at $Fo = 10.0$ (e) and magnified at $Fo = 20.0$ (f).

transfer. Amplified picture of $\overline{Nu}$ in Fig. 10(c) reveals that calculated $\overline{Nu}$ is not a smooth function of Fo. On the other hand, $\overline{f_l}$ appears smooth on the same scale, as Fig. 10(d) shows. The origin of the roughness, i.e. oscillations of $\overline{Nu}$ is the same as described in the previous section. So, larger amplitude and smaller frequency of oscillations appear on coarser grid, which agrees with the results from the previous section. For this case, fixed uniform grids with 80×80 and 160×160 CVs were also used. The results agree with those calculated on adaptive moving grid as shown in Fig. 10(c); the amplitude of oscillations is the largest on the coarser uniform grid. By simple comparison of $\overline{Nu}$, one can estimate that the accuracy of the solution on 80×80 adaptive grid is comparable with the one on 160×160 uniform grid. The same can be assessed from Fig. 10(f) which enables the comparison of the calculated interface position at $Fo = 20.0$. Fig. 10(e) shows the comparison against some of the solutions at $Fo = 10.0$ from [1]; the solutions agree reasonably well.

Case #2 exhibits different physical behavior than case #1 as can be seen in Fig. 11. In the beginning of the process, a multicellular flow structure forms in the melt which agrees with the results of Wintruff et al. [26] and with analysis of Le Quéré and Gobin [27]. Stella and Giangi [28] obtained similar behavior in their simulation of pure gallium melting. Recently, Hannoun et al. [29] confirmed multicellular flow in their comprehensive study. As shown in Fig. 12(a), $\overline{Nu}$ oscillates. This oscillatory behavior agrees with the results of Mohamad and Viskanta [30]. In their simulation of natural convection of low-Pr fluids in square cavity, oscillatory behavior appears if Re is larger than Pr-dependent critical value. In order to capture oscillatory behavior properly, ΔFo should be sufficiently small, as Fig. 12(c) illustrates; the figure presents $\overline{Nu}$, calculated on 80×80 grid with ΔFo was set to 10^{-4}, 5.0×10^{-5}, 2.5×10^{-5} and 10^{-5}. All calculations were started from the same state at $Fo = 3.0$.

All set values of ΔFo can be considered appropriate as essentially the same physical behavior is obtained, i.e. flow instability which leads to merging of two dominating flow cells to one cell. Of course, smaller ΔFo should be used to enhance the precision. Fig. 12(d) shows $\overline{Nu}$ at earlier instability leading to transition of flow pattern with three to two main flow cells; the instability appears from $Fo \sim 1.4$ on 40×40 grid to $Fo \sim 1.85$ on 128×128 grid. The oscillations of $\overline{Nu}$ which originate from the discretization, as already discussed, are not noticeable for case #2. The reason for this is that the calculated 'physical' oscillations have substantially larger amplitude and frequency. In Figs. 12(e) and (f) the calculated interface position at $Fo = 4.0$ and $Fo = 10.0$, respectively, is compared against solutions from [1] with multicellular flow. The solution of Hannoun et al. [29] at $Fo = 4.0$ is also included in Fig. 12.

Case #4 exhibits similar physical behavior as case #3, however with considerably larger heat transfer. Therefore, only case #4 will be presented: streamlines, temperature and grid (80×80 CVs) are displayed in Fig. 13. Compared with cases for tin, the convective heat transfer has a more important role, as Figs. 14(a) and (b) illustrate. The influence of the grid size on the obtained solution is shown in Figs. 14(c)–(f). Again, calculated interface position is compared against some results from [1] at $Fo = 0.06$ and $Fo = 0.1$ in Figs. 14(e) and (f), respectively. The largest difference in the interface position appears at the top. This can be explained by the fact that the grid around the top of the solid–liquid interface is not sufficiently refined in horizontal direction which can be seen in Fig. 13. This also explains why $\overline{Nu}(Fo)$ is still relatively rough, as smoothness of $\overline{Nu}(Fo)$ depends extensively on the grid density around the interface. Even though oscillations appear to be numerical artifact, the question can be asked, if some oscillations could also result from changing geometry due to interface movement.

The time step size ΔFo, average number of iterations per time step $\overline{N}_{\text{iter}}$ and approximate relative CPU time required for grid generation r_{gridgen} are given in Table 4. It shows that the grid generation requires from 20% to 30% of total computing time. This ratio depends on the precision of grid generation; in the presented calculation the convergence is considered when initial residuum is reduced by factor 5×10^{-2}. To run the simulation until 95% of the matter is melted, the total CPU time on 1.6 GHz AMD Athlon (Linux OS & Intel Fortran 7.0 compiler) varies from $\sim$50 h for case #4 on 40×40 grid to $\sim$120 days for case #2 on 128×128 grid. The latter is however ran with relatively small $\Delta Fo = 2.5 \times 10^{-5}$, thus requiring approximately 10^6 time steps.

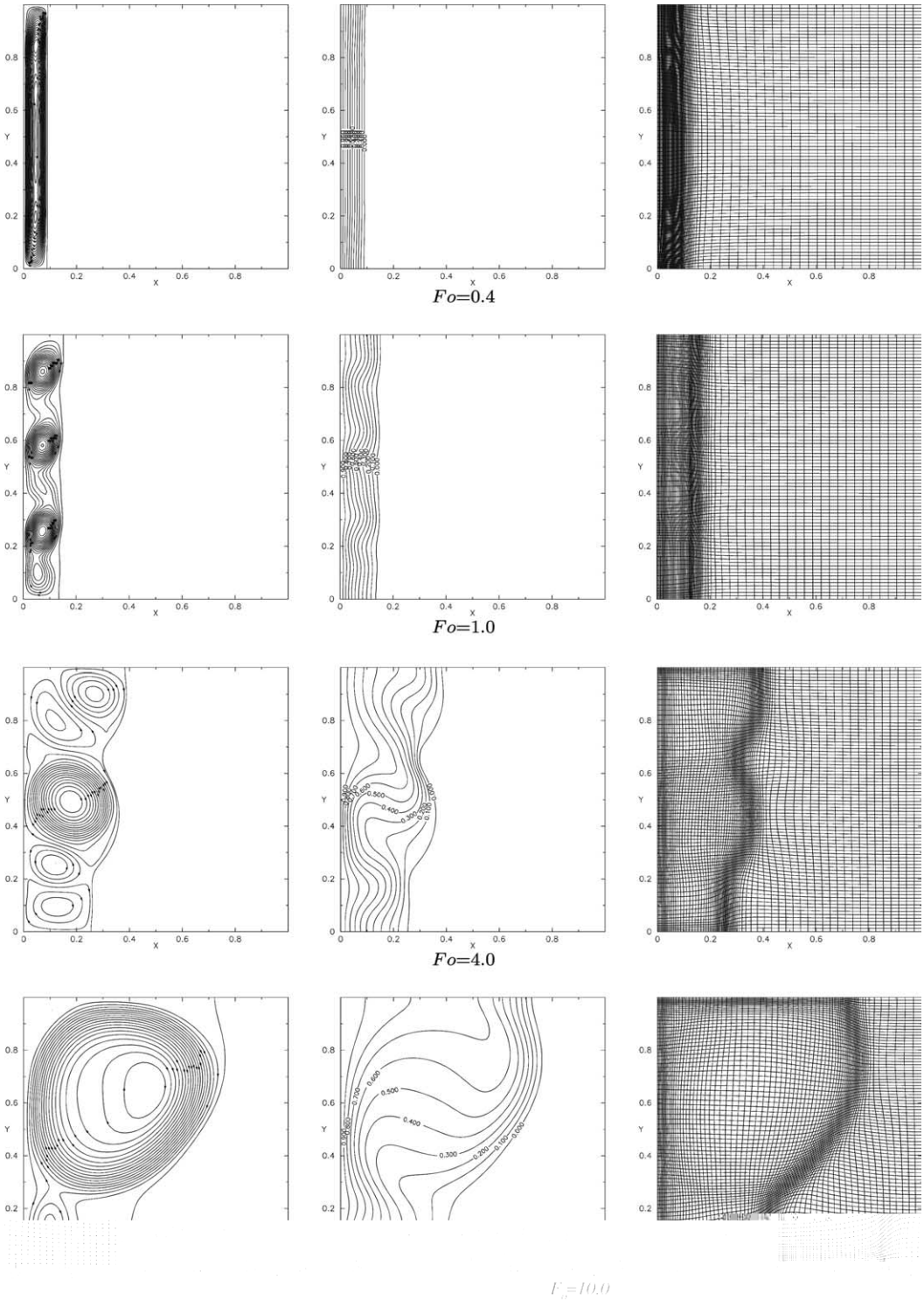

Fig. 11. Streamlines, temperature field and grid for case #2 at $Fo = 0.4$, $Fo = 1.0$, $Fo = 4.0$ and $Fo = 10.0$ calculated using 80×80 CVs.

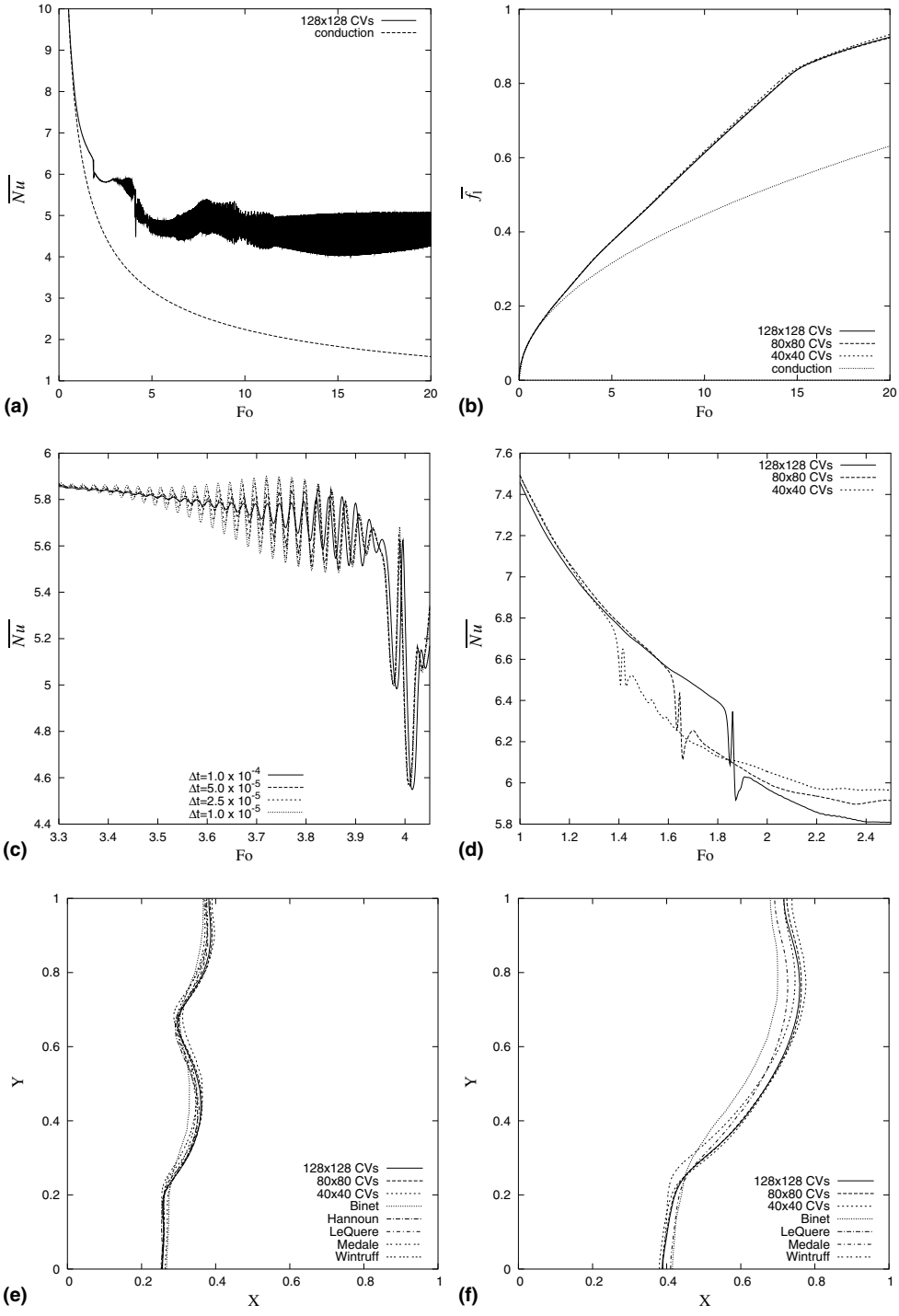

Fig. 12. Results for case #2: $\overline{Nu}$ at left wall (a), total liquid fraction f_l (b), magnified pictures of $\overline{Nu}$ using different Δt (c) and different grid (d), comparison of interface positions with [1,29] at $Fo = 4.0$ (e) and $Fo = 10.0$ (f).

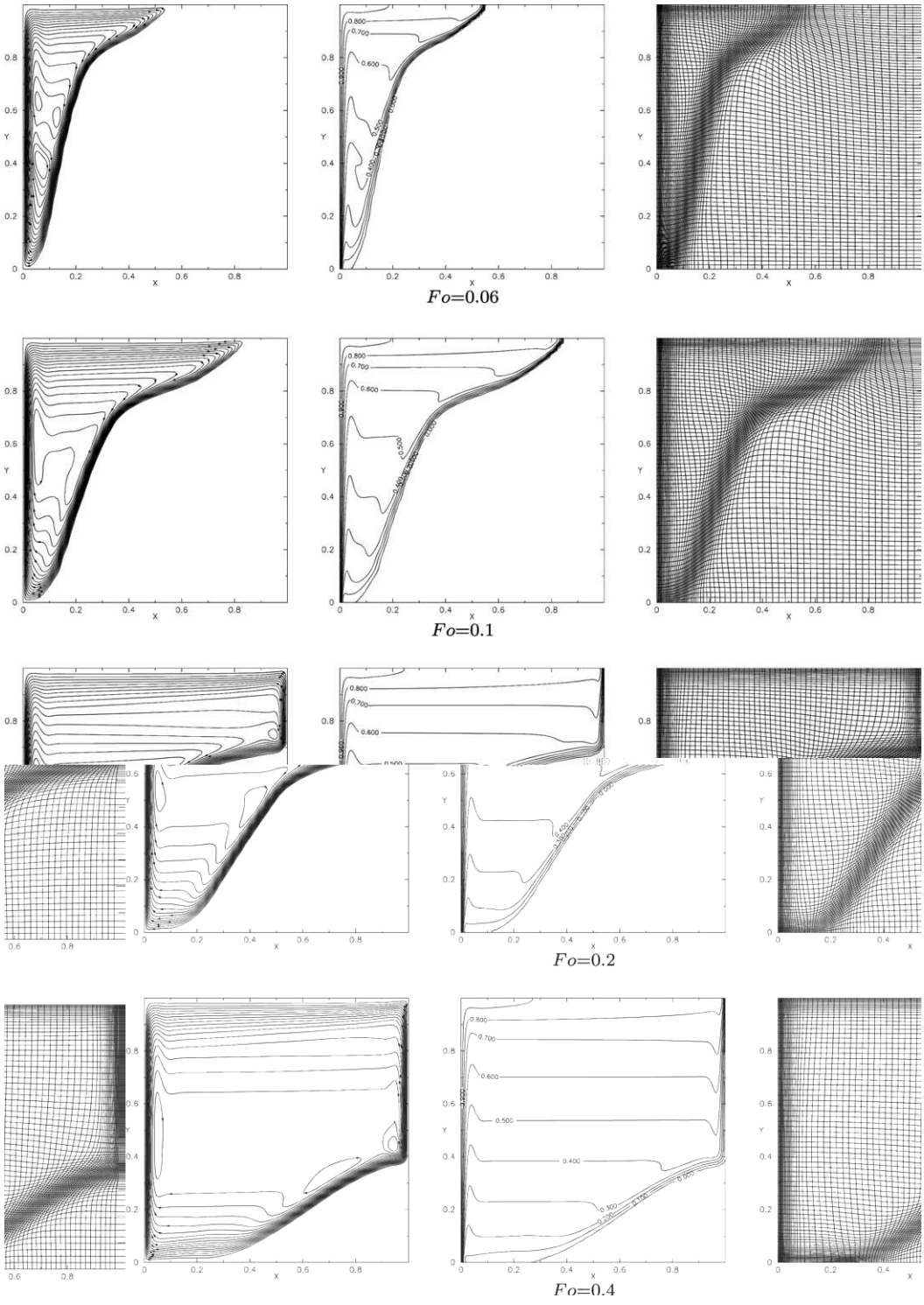

Fig. 13. Streamlines, temperature field and grid for case #4 at $Fo = 0.06$, $Fo = 0.1$, $Fo = 0.2$ and $Fo = 0.4$ calculated using 80×80 CVs.

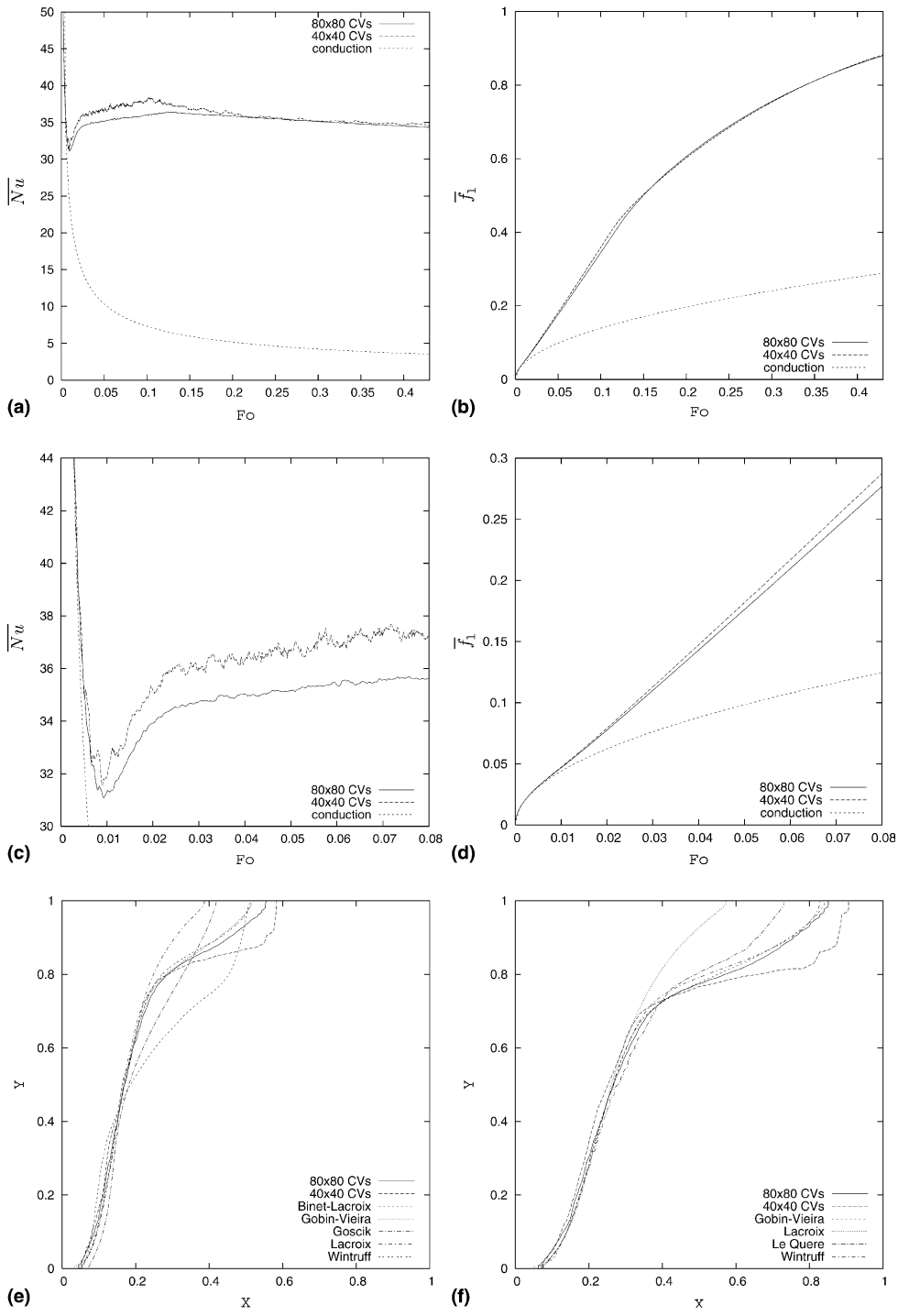

Fig. 14. Results for case #4: $\overline{Nu}$ at left wall (a), total liquid fraction f_l (b) and corresponding magnified pictures (c) and (d), comparison of interface positions at $Fo = 0.06$ (e) and $Fo = 0.1$ (f) with [1].

Table 4
Time step size ΔFo, average number of iterations per time step $\overline{N}_{\text{iter}}$, and relative CPU time required for grid generation r_{gridgen}

Case	Grid	ΔFo	$\overline{N}_{\text{iter}}$	r_{gridgen}
#1	40×40	1.0×10^{-3}	12.8	~0.24
#1	80×80	1.0×10^{-3}	12.3	~0.29
#1	128×128	5.0×10^{-4}	11.4	~0.25
#2	40×40	5.0×10^{-5}	11.7	~0.23
#2	80×80	1.0×10^{-4}	10.6	~0.27
#2	128×128	2.5×10^{-5}	9.1	~0.29
#4	40×40	2.0×10^{-6}	33.2	~0.21
#4	80×80	1.0×10^{-5}	45.2	~0.20

6. Conclusions

The work shows that using the adaptive moving grid in phase change system is effective because the grid generation requires a relatively small fraction of total computing time, at least for the presented examples. For case #1, the solution on 80×80 adaptive grid is comparable with the solution on 160×160 uniform grid. To assess the efficiency more precisely, adaptive and fixed grid solutions should be compared in other cases as well, which would however require additional computational resources. It is reasonable to expect that the method's efficiency is larger for cases where larger gradients appear.

For cases with low Pr, Stella and Gaingi [28] stated that 'only the use of a fine mesh allows the observation of the multicellular flow structure', which is also confirmed by the work of Hannoun et al. [29]. Presented method, however, proves to be useful also with small grids; even on coarse 40×40 grid it captures well the physical behavior of the system such as the flow instabilities in case #2. Presented solutions suggest that the origin of the discrepancies of contributed solutions in comparison exercise [1] could simply be insufficient grid density for the specified cases.

When compared with an interface-tracking, the presented method is more general because it is applicable for temperature interval phase change problems, such as the solidification of binary mixture. Also, its implementation for three-dimensional problems is straight forward. In addition, the reduction of a number of CVs, when compared to uniform grid, would be much larger for 3D problems than for 2D problems.

On the other hand, an interface-tracking method could produce more accurate results for pure substances as the grid can be fitted 'exactly' to the calculated interface. However, the interface would always be 'one step behind' if calculated from the heat flux balance at the previous time step. This is not the case in the presented method where only the grid is 'one step behind' but the interface position is computed implicitly. Nevertheless, the detailed comparison of the two approaches remains to be done.

References

[1] D. Gobin, P. Le Quéré, Melting from an isothermal vertical wall, synthesis of a numerical comparison exercise, Comput. Assisted Mech. Eng. Sci. 7 (2000) 289–306.
[2] M. Okada, Analysis of heat transfer during melting from a vertical wall, Int. J. Heat Mass Transfer 27 (1984) 2057–2066.
[3] C.J. Ho, R. Viskanta, Heat transfer during melting from an isothermal vertical wall, ASME J. Heat Transfer 106 (1984) 12–19.
[4] C. Benard, D. Gobin, A. Zanoli, Moving boundary problem: heat conduction in the solid phase of a phase-change material during melting driven by natural convection in the liquid, Int. J. Heat Mass Transfer 29 (1986) 1669–1681.
[5] I. Wintruff, C. Günther, A.G. Class, An interface tracking control-volume finite-element method for melting and solidification problems – Part I: formulation, Numer. Heat Transfer B 39 (2001) 101–125.
[6] J. Ni, C. Beckermann, A volume-averaged two-phase model for transport phenomena during solidification, Metall. Trans. B 18 (1991) 349–361.
[7] J. Crank, Free and Moving Boundary Problems, Oxford University Press, Oxford, 1984.

[8] V.R. Voller, C. Prakash, A fixed grid numerical modelling methodology for convection–diffusion mushy region phase-change problems, Int. J. Heat Mass Transfer 30 (1987) 1709–1719.
[9] W.D. Bennon, F.P. Incropera, A continuum model for momentum, heat and species transport in binary solid–liquid phase change systems – I. Model formulation, Int. J. Heat Mass Transfer 30 (1987) 2161–2170.
[10] M. Lacroix, V.R. Voller, Finite difference solutions of solidification change problems: transformed versus fixed grids, Numer. Heat Transfer B 17 (1990) 25–41.
[11] J.A. Mackenzie, M.L. Robertson, The numerical solution of one-dimensional phase change problems using an adaptive moving mesh method, J. Comput. Phys. 161 (2000) 537–557.
[12] C. Beckett, J.A. Mackenzie, M.L. Robertson, A moving mesh finite element method for the solution of two-dimensional Stefan problems, J. Comput. Phys. 168 (2001) 500–518.
[13] D.A. Anderson, Grid cell volume control with an adaptive grid generator, Appl. Math. Comput. 35 (1990) 209–217.
[14] J.S. Turner, Buoyancy Effects in Fluids, University Press, Cambridge, 1973.
[15] J.H. Ferziger, M. Perić, Computational Methods for Fluid Dynamics, Springer, Berlin, 1998.
[16] P.K. Khosla, S.G. Rubin, A diagonally dominant second-order accurate implicit scheme, Comput. Fluids 2 (1974) 207–209.
[17] I. Demirdžić, M. Perić, Finite volume method for prediction of fluid flow in arbitrarily shaped domains with moving boundaries, Int. J. Numer. Methods Fluids 10 (1990) 771–790.
[18] I. Demirdžić, M. Perić, Space conservation law in finite volume calculations of fluid flow, Int. J. Numer. Methods Fluids 8 (1988) 1037–1050.
[19] S.V. Patankar, Numerical Heat Transfer and Fluid Flow, Hemisphere, Washington, DC, 1980.
[20] C.M. Rhie, W.L. Chow, Numerical study of the turbulent flow past an airfoil with trailing edge separation, AIAA J. 21 (1983) 1525–1532.
[21] G.E. Schneider, M. Zedan, A modified strongly implicit procedure for the numerical solution of field problems, Numer. Heat Transfer 4 (1981) 1–19.
[22] J.F. Thompson, B.K. Soni, N.P. Weatherill, Handbook of Grid Generation, CRC Press, Boca Raton, 1999.
[23] U. Grigull, H. Sander, Heat Conduction, Hemisphere Publishing Corporation, Washington, DC, 1984.
[24] A.W. Date, A strong enthalpy formulation for the Stefan problem, Int. J. Heat Mass Transfer 34 (1991) 2231–2235.
[25] P.H. Gaskell, A.K.C. Lau, Curvature-compensated convective transport: SMART, a new boundedness-preserving transport algorithm, Int. J. Numer. Methods Fluids 8 (1988) 617–641.
[26] I. Wintruff, C. Günther, A.G. Class, An interface tracking control-volume finite-element method for melting and solidification problems – Part II: verification and application, Numer. Heat Transfer B 39 (2001) 127–149.
[27] P. Le Quéré, D. Gobin, A note on possible flow instabilities in melting from the side, Int. J. Therm. Sci. 38 (1999) 595–600.
[28] F. Stella, M. Giangi, Melting of a pure metal on a vertical wall: numerical simulation, Numer. Heat Transfer A 38 (2000) 193–208.
[29] N. Hannoun, V. Alexiades, T.Z. Mai, Resolving the controversy over tin and gallium melting in a rectangular cavity heated from the side, Numer. Heat Transfer B 44 (2003) 253–276.
[30] A.A. Mohamad, R. Viskanta, Transient natural convection of low-Prandtl-number fluids in a differentially heated cavity, Int. J. Numer. Methods Fluids 13 (1991) 61–81.