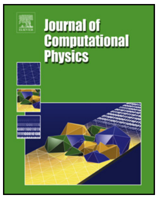
Journal of Computational Physics 274 (2014) 826–840

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp



A Newton method with adaptive finite elements for solving phase-change problems with natural convection

Ionut Danaila [a,*], Raluca Moglan [a,b], Frédéric Hecht [c], Stéphane Le Masson [b]

[a] *Université de Rouen, Laboratoire de Mathématiques Raphaël Salem, CNRS, UMR 6085, Avenue de l'Université, BP 12, F-76801 Saint-Étienne-du-Rouvray, France*
[b] *France Telecom, 2 Avenue Pierre Marzin, BP 40, 22307 Lannion Cedex, France*
[c] *UPMC Univ Paris 06, CNRS, UMR 7598, Laboratoire Jacques-Louis Lions, 4 Place Jussieu, F-75005 Paris, France*

ARTICLE INFO

Article history:
Received 27 May 2013
Received in revised form 4 June 2014
Accepted 19 June 2014
Available online 26 June 2014

Keywords:
Melting
Solidification
Water freezing
Newton method
Finite element
Mesh adaptivity
Boussinesq
Navier–Stokes
PCM
FreeFem++

ABSTRACT

We present a new numerical system using finite elements with mesh adaptivity for the simulation of solid–liquid phase change systems. In the liquid phase, the natural convection flow is simulated by solving the incompressible Navier–Stokes equations with Boussinesq approximation. A variable viscosity model allows the velocity to progressively vanish in the solid phase, through an intermediate mushy region. The phase change is modeled by introducing an implicit enthalpy source term in the heat equation. The final system of equations describing the liquid–solid system by a single domain approach is solved using a Newton iterative algorithm. The space discretization is based on a P2–P1 Taylor–Hood finite elements and mesh adaptivity by metric control is used to accurately track the solid–liquid interface or the density inversion interface for water flows.

The numerical method is validated against classical benchmarks that progressively add strong non-linearities in the system of equations: natural convection of air, natural convection of water, melting of a phase-change material and water freezing. Very good agreement with experimental data is obtained for each test case, proving the capability of the method to deal with both melting and solidification problems with convection. The presented numerical method is easy to implement using FreeFem++ software using a syntax close to the mathematical formulation.

1. Introduction

Solid–liquid phase change systems involving melting or freezing processes are encountered in numerous practical applications, ranging from metal casting and thermal energy storage to food freezing. Most of the models consider the conduction as the principal mechanism in describing the heat transfer during melting or solidification (Stefan problem). Recent models include several other important physical phenomena, such as gravity effects, convection in the liquid phase, the presence of a *mushy* region (containing both solid and liquid particles) at the interface between the two phases, etc. For a comprehensive review of such models, see [1].

In particular, it was found that the natural convection in the liquid plays an important role in the heat transfer between phases and the propagation of the melting/solidification front [2–7]. This is specifically the case in recent practical

[*] Corresponding author. Tel.: +33 2 32 95 52 50; fax: +33 2 32 95 52 86.
E-mail addresses: ionut.danaila@univ-rouen.fr (I. Danaila), raluca.moglan@yahoo.com (R. Moglan), hecht@ann.jussieu.fr (F. Hecht), stephane.lemasson@orange.com (S. Le Masson).

http://dx.doi.org/10.1016/j.jcp.2014.06.036
0021-9991/© 2014 Elsevier Inc. All rights reserved.

applications using phase-change materials (PCM) to store heat energy or to diminish temporary peak temperatures in different technologies (*e.g.* thermal regulation of buildings, passive cooling of electronic devices). Another example of convection dominated phase-change systems is the water freezing, appearing in many environmental applications, such as water transportation, food processing, weather prediction, etc.

In this paper we use a single domain approach to simulate phase-change systems with convection. In the liquid phase, the natural convection flow is simulated by solving the full incompressible Navier–Stokes equations with Boussinesq approximation. The same system of equations is solved in the solid phase by introducing a variable viscosity coefficient taking very large values in the solid (*e.g.* [8]). This model allows the velocity to progressively vanish in the solid through an intermediate *mushy* region, defined accordingly to classical enthalpy methods (*e.g.* [3,9,10]). In enthalpy methods, the phase change is modeled by introducing an enthalpy source term in the heat equation. The phase-change is supposed to occur over a temperature interval setting the width of the mushy region. This temperature interval is also used to regularize discontinuous functions representing the variation of material constants (conductivity, specific heat, latent heat) across the solid–liquid interface.

The main advantage of the single domain approach is that the same system of equations is solved in both liquid and solid phase. In exchange, the numerical method has to tackle two important challenges: properly resolve the convection cells in the fluid region and accurately capture the solid–liquid interface. The former issue is related to the non-linearity in the momentum Navier–Stokes equations, while the latter comes mainly from the very sharp variation of the coefficients of the equations (viscosity, latent heat, etc.) in a small region around the solid–liquid interface. Since most of the numerical methods presented in the numerical heat transfer community use finite difference (FD) or finite volume (FV) methods on a fixed mesh, the general strategy to address these issues is to dramatically increase the mesh resolution in the whole domain. This results is a considerable increase of the computational time, even for two dimensional cases. Finite element (FE) methods offer the possibility to dynamically refine the mesh only in the regions of the domain where sharp phenomena take place (*e.g.* solid–liquid interface, recirculation zones).

FE methods were used in late 1980s to derive mathematically sound numerical algorithms for the Stefan problem (involving Laplace operators). Different modeling approaches were analyzed, from enthalpy-type methods (*e.g.* [11]) to front-tracking methods (*e.g.* [12]). Only recently, adaptive FE methods were proposed for the phase-change problem. A moving mesh technique was developed in [13] and used to simulate melting and solidification problems. The method was based on solving a modified set of equations, including two supplementary partial differential equations modeling the mesh movement between two time steps. An anisotropic mesh adaptation algorithm based on an approximation of a hierarchical error estimator was proposed in [14] for classical two-phase Stefan problem (without convection). A different mesh adaptivity strategy, based on the definition of edge length from a solution dependent metric, was used to deal with the same Stephan problems in three-dimensional simulations. This last adaptivity method, based on solution dependent metrics, was also tested for phase-change systems with convection in [15]. The use of locally adaptive meshes with strong anisotropy proved very effective in reducing the number of computational nodes for phase-change systems.

As a first contribution of the present paper, we introduce an FE method with time-dependent mesh adaptivity by metric control that is effective for a large range of phase-change systems with convection, from melting to solidification. The proposed mesh refinement strategy has the capacity to take into account different metrics and thus the ability to refine the mesh in different regions of interest in the computational domain. In particular, we show that the method is able to simultaneously track several interfaces in the domain, a feature that was not present in previous mesh refinement algorithms. Similar algorithms based on FreeFem++ [16,17] were successfully used for solving different systems of equations with locally sharp variation of the solution, such as Gross–Pitaevskii equation [18,19] or Laplace equations with nonlinear source terms [20].

The second contribution of this paper is the derivation of a Newton algorithm for solving the nonlinear system of equations for the single domain approach of the phase-change system with convection. Newton linearization has been successfully used for fluid dynamics and heat transfer equations (*e.g.* [21]) with the advantage to accelerate computations due to its rapid quadratic convergence. Effective classical or high-order Newton methods for the steady Navier–Stokes equations were proposed in [22,23] using finite-difference methods on non-staggered grids. For the Navier–Stokes–Boussinesq system, a Newton method with explicit treatment of the temperature was briefly introduced in [15]. We derive below a fully-implicit Newton method for the phase-change system based on a finite-element formulation of the Navier–Stokes equations. The advantage of this formulation is to permit a straightforward implementation of different types of non-linearities in the system of equations. For the sake of modularity, we derive and test the Newton algorithm by progressively treating the following non-linearities:

(i) the convective nonlinear term in the Navier–Stokes–Boussinesq equations (test case: natural convection of air in a differentially heated cavity),
(ii) problem *(i)* plus a nonlinear buoyancy term (test case: natural convection of water in a differentially heated cavity),
(iii) problem *(i)* plus the enthalpy nonlinear source term and nonlinear variable viscosity (test case: melting of phase-change material),
(iv) problem *(iii)* plus a nonlinear buoyancy term and nonlinear variation of thermodynamic properties (test case: water freezing).

This graduate presentation of the method offers the possibility to implement the algorithm in a flexible way and to easily adapt it to different problems, ranging from natural convection to melting or solidification. The FreeFem++ syntax, close to the mathematical formulation, greatly facilitates this implementation task. For each test case mentioned above, we use well defined classical benchmarks and estimate the precision of our method by quantitatively comparing the obtained results with experimental and previously published numerical data.

The paper is organized as follows. Section 2 introduces the governing equations and the proposed regularization technique. Section 3 presents the Newton algorithm for the Navier–Stokes–Boussinesq system and for the phase-change system with convection. The mesh adaptivity technique is also discussed in this section. The final Section 4 is devoted to extensive numerical validation of the new method using the four test cases mentioned above.

2. Governing equations

We consider in this paper a solid–liquid system placed in a two-dimensional cavity of width L and height H; in the following, subscripts s and l will refer to the solid and liquid phase, respectively. The vertical walls of the cavity are isothermal, of temperatures T_h (hot) and T_c (cold). The fusion temperature will be denoted by T_f. Using a lengthscale $L_{ref} = H$ and a liquid reference state $(\rho_{ref}, V_{ref}, T_{ref})$, we can define the following scaling for the space, velocity, temperature and time variables:

$$\mathbf{x} = \frac{\mathbf{X}}{L_{ref}}, \qquad \mathbf{u} = \frac{\mathbf{U}}{V_{ref}}, \qquad \theta = \frac{T - T_{ref}}{T_h - T_c}, \qquad t = \frac{\tau}{t_{ref}}, \qquad t_{ref} = L_{ref}/V_{ref}. \tag{1}$$

In this setting, the incompressible Navier–Stokes equations with the Boussinesq approximation can be written under the form:

$$\begin{aligned} &\nabla \cdot \mathbf{u} = 0, \\ &\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} + \nabla p - \frac{1}{\mathcal{R}e} \nabla \cdot \big(2\mu_{sl}(\theta)\, \mathbf{D}(\mathbf{u})\big) - f_B(\theta)\, \mathbf{e}_y = 0, \end{aligned} \tag{2}$$

where $\mu_{sl} = \mu_s/\mu_l$ is the non-dimensional viscosity, $f_B = g(\rho_{ref} - \rho)/\rho_{ref}$ the (non-dimensional) buoyancy force and $\mathbf{D}(\mathbf{u}) = (1/2)(\nabla\mathbf{u} + \nabla\mathbf{u}^T)$ the rate of strain (deformation) symmetric tensor with components $D_{ij}(\mathbf{u}) = (1/2)(\partial u_i/\partial x_j + \partial u_j/\partial x_i)$. Note that in the Boussinesq approximation, the viscosity does not vary with the temperature θ; in (2) we use the general form of the viscous stress tensor for incompressible flows, solely because the viscosity μ_{sl} will be varied in the single domain approach to separate the solid ($\mu_{sl} = 10^8$) and liquid ($\mu_{sl} = 1$) zones. Also, in the classical (linear) Boussinesq model, the buoyancy term varies linearly with the temperature:

$$f_B(\theta) = \frac{\mathcal{R}a}{\mathcal{P}r\,\mathcal{R}e^2}\theta. \tag{3}$$

The Reynolds, Prandtl and Rayleigh numbers, are defined as:

$$\mathcal{R}e = \frac{\rho_{ref} V_{ref} L_{ref}}{\mu_l}, \qquad \mathcal{P}r = \frac{\nu_l}{\alpha_l}, \qquad \mathcal{R}a = \frac{g\beta L_{ref}^3 (T_h - T_c)}{\nu_l \alpha_l}, \tag{4}$$

with μ denoting the viscosity, ν the kinematic viscosity, α the thermal diffusivity, β the thermal expansion coefficient and g the gravitational acceleration.

For the energy equation we use an enthalpy transforming model [3,9,10] to obtain the following non-dimensional form:

$$\frac{\partial (C\theta)}{\partial t} + \nabla \cdot (C\theta\mathbf{u}) = \nabla \cdot \left(\frac{K}{\mathcal{P}r}\nabla\theta\right) - \frac{\partial (CS)}{\partial t} - \nabla \cdot (CS\mathbf{u}), \tag{5}$$

where $C = c_s/c_l$ and $K = k_s/k_l$ are non-dimensional specific heat and conductivity, respectively. The non-dimensional source term $S = s/(T_h - T_c)$ takes into account the latent heat of fusion and is theoretically a Heaviside function that is zero in the solid phase and takes large values in the liquid phase. The hypothesis behind the model is that the change of phase occurs over a temperature interval around the fusion point $[\theta_f - \varepsilon_1, \theta_f + \varepsilon_2]$. Different models are proposed in the literature [3,7] for S, using the general form:

$$S = \begin{cases} S_l, & \theta - \theta_f \geq \varepsilon_2 \\ F_S(\theta), & -\varepsilon_1 \leq \theta - \theta_f < \varepsilon_2 \\ S_s, & \theta - \theta_f < -\varepsilon_1 \end{cases} \tag{6}$$

It should be noted that usually the reference temperature is $T_{ref} = T_f$ and consequently $\theta_f = 0$ in the previous formula.

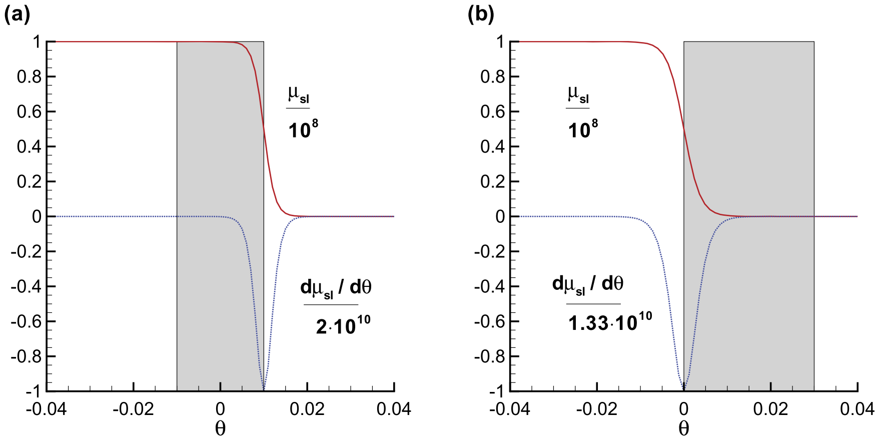


Fig. 1. Illustration of the variation of the viscosity μ_{sl} and the derivative $d\mu_{sl}/d\theta$ for the single domain approach: $\mu_{sl} = 10^8$ in the solid and $\mu_{sl} = 1$ in the liquid. Functions normalized with their maximum value (indicated on the graphs). The (gray) mushy region is defined as the temperature interval $[\theta_f - \varepsilon_1, \theta_f + \varepsilon_2]$, with $\theta_f = 0$ here. Model using the function F defined by Eq. (7) for (a) melting case with $\varepsilon_1 = \varepsilon_2 = 0.01$ and $F(\theta; 2, \varepsilon_1, \varepsilon_1/2)$ (b) water freezing case with $\varepsilon_1 = 0$, $\varepsilon_2 = 0.03$ and $F(\theta; 2, \varepsilon_1, \varepsilon_2/4)$.

In the model (6), the function $F_S(\theta)$ represents a regularization of the enthalpy variation in the mushy region. Linear functions are generally used in the literature with FV (*e.g.* [7]) or FE methods (*e.g.* [24]). In this paper we propose a regularization of the step function by a continuous and differentiable hyperbolic-tangent function with three parameters, defined for all θ:

$$F(\theta; a_s, \theta_s, R_s) = f_l + \frac{f_s - f_l}{2}\left\{1 + \tanh\left(a_s\left(\frac{\theta_s - \theta}{R_s}\right)\right)\right\}, \tag{7}$$

where f_l, f_s are the imposed values in the liquid and solid phases, a_s a smoothing parameter, θ_s the central value (around which we regularize) and R_s the smoothing radius. The same function (7), with different parameters given by physical models, will be used to represent the variation of all material properties (conductivity, thermal capacity).

The variable viscosity allowing the passage from liquid to solid will be also represented by this function. It is important to note that the three parameters a_s, θ_s, R_s are useful to make the regularization corresponding to physical assumptions of the models for the mushy region. In particular, the sharp variation of the derivative of $dF/d\theta$ should be localized near the new phase appearing in the system, liquid for melting and solid for freezing (see also [13]). This principle is illustrated in Fig. 1, showing the variation of the viscosity μ_{sl}, and its derivative $d\mu_{sl}/d\theta$ for the melting and solidification test cases computed in this paper. The computations showed the most important sensitivity to the model for the variable viscosity than for the other material properties (S, K, C).

Since we intend to compare numerical results to experiments, it is important to define the characteristic scales in (1). Several choices are used in the literature:

$$V_{ref}^{(1)} = \frac{\nu_l}{H} \quad\Longrightarrow\quad t_{ref}^{(1)} = \frac{\nu_l}{H^2} \quad\Longrightarrow\quad \mathcal{R}e = 1, \tag{8}$$

$$V_{ref}^{(2)} = \frac{\alpha}{H} \quad\Longrightarrow\quad t_{ref}^{(2)} = t_{ref}^{(1)}\mathcal{P}r \quad\Longrightarrow\quad \mathcal{R}e = 1/\mathcal{P}r, \tag{9}$$

$$V_{ref}^{(3)} = \frac{\nu_l}{H}\sqrt{\frac{\mathcal{R}a}{\mathcal{P}r}} \quad\Longrightarrow\quad t_{ref}^{(3)} = t_{ref}^{(1)}\sqrt{\frac{\mathcal{P}r}{\mathcal{R}a}} \quad\Longrightarrow\quad \mathcal{R}e = \sqrt{\frac{\mathcal{R}a}{\mathcal{P}r}}. \tag{10}$$

The last choice (10) is generally used for classical natural convection problems, while the first two choices are preferred for melting/freezing problems [7,25]. For these latter problems, $\mathcal{P}r > 1$ and $\mathcal{R}a \gg 1$ and consequently $t_{ref}^{(3)} \ll t_{ref}^{(1)} < t_{ref}^{(2)}$. We shall use in the following the first scaling choice (8), setting $\mathcal{R}e = 1$.

A further simplification of the energy equation (5) is obtained by supposing that the mushy region is very narrow and acts like a solid. As a consequence, the last (advection) term in (5) could be neglected (see also [7]). The final system of equations that will be solved in the whole domain is the following:

$$\begin{aligned}
&\nabla \cdot \mathbf{u} = 0,\\
&\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}\cdot\nabla)\mathbf{u} + \nabla p - \nabla\cdot\big(2\mu_{sl}(\theta)\,\mathbf{D}(\mathbf{u})\big) - f_B(\theta)\,\mathbf{e}_y = 0,\\
&\frac{\partial (C\theta)}{\partial t} + \nabla\cdot(C\theta\mathbf{u}) - \nabla\cdot\left(\frac{K}{\mathcal{P}r}\nabla\theta\right) + \frac{\partial (CS)}{\partial t} = 0.
\end{aligned} \tag{11}$$

3. Finite element algorithms

Aside from the convective non-linearity of the Navier–Stokes equations, a particular difficulty in solving numerically the phase-change system (11) comes from the strong non-linearity introduced by the source term $\partial(CS)/\partial t$. When FV or FD methods are used with fixed meshes, special semi-implicit methods are necessary to assess the phase change inside a cell [3,26]. Nonetheless, such methods still require a very small cell size to reduce the uncertainty of the localization of the phase-change front. Since the grid is refined in the whole domain, the computation time increases considerably. A supplementary difficulty comes from the technique used in the single domain approach to set to zero the velocity inside the solid phase. In FV or FD methods, continuous correction techniques (*e.g.* switch-off, porosity Darcy source terms) [26,27], employ large numbers that may reduce accuracy and convergence rate and could trigger numerical inconsistencies [8]. As a result, recent techniques in FV methods [7,28] are based on the modification of the numerical algorithm (usually derived from SIMPLE family algorithms) by simply cutting-off the velocity in the solid by a relaxation scheme.

We derive in this section a numerical method that takes advantage from the adaptive capabilities of the finite-element spatial discretization to accurately capture the solid–liquid front and uses for the time advancement a global Newton algorithm to simultaneously treat the non-linearities present in momentum and energy equations. The weak formulation of equation behind the FE method facilitates the use of a continuous correction method based on variable viscosity to bring the velocity to zero in the solid phase. This formulation offers a unified time–space approach that could be easily implemented using generic FE solvers. We use here the free software FreeFem++ [16] using a large variety of triangular finite elements to solve partial differential equations in two or three dimensions. FreeFem++ is an integrated product with its own high level programming language with a syntax close to mathematical formulations; it was recently used to solve different types of partial differential equations, *e.g.* Schrödinger or Gross–Pitaevskii [18,19], incompressible Navier–Stokes equations [29,30], and Laplace equations with nonlinear source terms [20].

3.1. Algorithms for the Navier–Stokes–Boussinesq equations

We first focus on the Navier–Stokes–Boussinesq part of the system (11), *i.e.* only the fluid phase is considered ($\mu_{sl} = \mu_l = C = 1,\ S = 0$). This allows to develop and test the Newton algorithm before applying it to the phase-change system of equations. As a reference algorithm to treat the non-linearity in the momentum equations, we use the characteristic Galerkin method extended to the Navier–Stokes equations in [31]. Both algorithms will be tested in the following section against classical benchmarks of natural convection of air inside a square cavity.

Weak formulations for the Navier–Stokes system are generally based on a separate discretization of the temporal derivative (using finite difference, splitting or characteristics methods) and the generalization of the Stokes problem for the resulting system [32–34]. We consider here a backward implicit Euler scheme for the time advancement and a penalty formulation [32,34] to deal with the diverge free constraint for the velocity field. Note that a second order scheme for the time integration (as the Gear scheme used in [15]) could allow for larger time steps; for the applications considered below, the first order Euler scheme with a relative small time step was preferred to ensure an accurate capture of interfaces.

We obtain the following system of equations that has to be solved at each time level $t_{n+1} = (n+1)\delta t$ in the space domain Ω:

$$\begin{aligned}
&\nabla\cdot\mathbf{u}^{n+1} + \gamma p^{n+1} = 0,\\
&\frac{\mathbf{u}^{n+1}}{\delta t} + \left(\mathbf{u}^{n+1}\cdot\nabla\right)\mathbf{u}^{n+1} + \nabla p^{n+1} - \nabla\cdot\left(2\mu_l\,\mathbf{D}\left(\mathbf{u}^{n+1}\right)\right) - f_B\left(\theta^{n+1}\right)\mathbf{e}_y = \frac{\mathbf{u}^n}{\delta t},\\
&\frac{\theta^{n+1}}{\delta t} + \nabla\cdot\left(\mathbf{u}^{n+1}\theta^{n+1}\right) - \nabla\cdot\left(\frac{K}{\mathcal{P}r}\nabla\theta^{n+1}\right) = \frac{\theta^n}{\delta t},
\end{aligned}\tag{12}$$

with $\gamma > 0$ the penalty parameter. It must be noticed, however, that compared to classical penalty methods, we use here a very low value $\gamma = 10^{-7}$ for the penalty parameter. Since we shall use latter a Taylor–Hood finite element, the formulation (12) will provide only a regularization to ensure a pressure field with zero average and, at the algebraic level, fulfill the diagonal of the pressure term. For the sake of simplicity, we consider homogeneous Dirichlet boundary conditions for the velocity, *i.e.* $\mathbf{u} = 0$ on $\partial\Omega$, and set the classical Hilbert spaces for the velocity and pressure:

$$\mathbf{V} = V \times V,\quad V = H_0^1(\Omega),\qquad Q = \left\{q \in L^2(\Omega)\,\middle|\, \int_\Omega q = 0\right\}\tag{13}$$

Following the generalization of the Stokes problem [32–34], the variational formulation of the system (12) can be written as: find $(\mathbf{u}^{n+1}, p^{n+1}, \theta^{n+1}) \in \mathbf{V}\times Q\times V$, such that

$$\begin{aligned}
&b\left(\mathbf{u}^{n+1}, q\right) - \gamma\left(p^{n+1}, q\right) = 0,\quad \forall q\in Q\\
&\frac{1}{\delta t}\left(\mathbf{u}^{n+1}, \mathbf{v}\right) + c\left(\mathbf{u}^{n+1}; \mathbf{u}^{n+1}, \mathbf{v}\right) + a\left(\mu_l; \mathbf{u}^{n+1}, \mathbf{v}\right) + b\left(\mathbf{v}, p^{n+1}\right) - \left(f_B\left(\theta^{n+1}\right)\mathbf{e}_y, \mathbf{v}\right) = \frac{1}{\delta t}\left(\mathbf{u}^n, \mathbf{v}\right),\quad \forall\mathbf{v}\in\mathbf{V}\\
&\frac{1}{\delta t}\left(\theta^{n+1}, \phi\right) - \left(\mathbf{u}^{n+1}\cdot\nabla\phi, \theta^{n+1}\right) + \left(\frac{K}{\mathcal{P}r}\nabla\theta^{n+1}, \nabla\phi\right) = \frac{1}{\delta t}\left(\theta^n, \phi\right),\quad \forall\phi\in V,
\end{aligned}\tag{14}$$

where $(u, v) = \int_\Omega u \cdot v$ denotes the scalar product in $L^2(\Omega)$ or $(L^2(\Omega))^2$; the bilinear forms a, b and trilinear form c are defined as [33,34]:

$$\begin{aligned}
&a : \mathbf{V} \times \mathbf{V} \to \mathbb{R}, \quad a(\mu; \mathbf{u}, \mathbf{v}) = \int_\Omega 2\mu\, \mathbf{D}(\mathbf{u}) : \mathbf{D}(\mathbf{v}) = \int_\Omega 2\mu \sum_{i,j=1}^{2} D_{ij}(\mathbf{u}) D_{ij}(\mathbf{v}),\\
&b : \mathbf{V} \times Q \to \mathbb{R}, \quad b(\mathbf{u}, q) = -\int_\Omega \nabla \cdot \mathbf{u}\, q = -\sum_{i=1}^{2} \int_\Omega \partial_i u_i \cdot q\\
&c : \mathbf{V} \times \mathbf{V} \times \mathbf{V} \to \mathbb{R}, \quad c(\mathbf{w}; \mathbf{z}, \mathbf{v}) = \int_\Omega \big[(\mathbf{w} \cdot \nabla)\mathbf{z}\big] \cdot \mathbf{v} = \sum_{i,j=1}^{2} \int_\Omega w_j (\partial_j z_i) v_i.
\end{aligned} \tag{15}$$

In order to apply the Newton method, the system of equations (14) is regarded as $\mathcal{F}(w) = 0$, with $w = (\mathbf{u}^{n+1}, p^{n+1}, \theta^{n+1}) \in W = \mathbf{V} \times Q \times V$, and $\mathcal{F} : W \to W$ a differentiable mapping. The classical Newton algorithm is then used to advance the solution from time t_n to t_{n+1}: starting from an initial guess $w^0 = (\mathbf{u}^n, p^n, \theta^n)$ (which is the solution at t_n), construct the sequence (w^k) by solving for each inner iteration k:

$$D_w \mathcal{F}\big(w^k\big)\big(w^k - w^{k+1}\big) = \mathcal{F}\big(w^k\big), \tag{16}$$

where $D_w \mathcal{F}$ is the linear operator representing the differential of $\mathcal{F}$. Denoting by $(\mathbf{u}_w, p_w, \theta_w) = w^k - w^{k+1}$ and after differentiating (14), the system of equations (16) can be explicitly written as:

$$\begin{aligned}
&b(\mathbf{u}_w, q) - \gamma(p_w, q) = b\big(\mathbf{u}^k, q\big) - \gamma\big(p^k, q\big),\\
&\frac{1}{\delta t}(\mathbf{u}_w, \mathbf{v}) + c\big(\mathbf{u}_w; \mathbf{u}^k, \mathbf{v}\big) + c\big(\mathbf{u}^k; \mathbf{u}_w, \mathbf{v}\big) + a(\mu_l; \mathbf{u}_w, \mathbf{v}) + b(\mathbf{v}, p_w) - \left(\frac{df_B}{d\theta}\big(\theta^k\big)\theta_w\, \mathbf{e}_y, \mathbf{v}\right)\\
&\quad = \frac{1}{\delta t}\big(\mathbf{u}^k - \mathbf{u}^n, \mathbf{v}\big) + c\big(\mathbf{u}^k; \mathbf{u}^k, \mathbf{v}\big) + a\big(\mu_l; \mathbf{u}^k, \mathbf{v}\big) + b\big(\mathbf{v}, p^k\big) - \big(f_B\big(\theta^k\big)\mathbf{e}_y, \mathbf{v}\big),\\
&\frac{1}{\delta t}(\theta_w, \phi) - \big(\mathbf{u}^k \cdot \nabla\phi, \theta_w\big) - \big(\mathbf{u}_w \cdot \nabla\phi, \theta^k\big) + \left(\frac{K}{\mathcal{P}r}\nabla\theta_w, \nabla\phi\right)\\
&\quad = \frac{1}{\delta t}\big(\theta^k - \theta^n, \phi\big) - \big(\mathbf{u}^k \cdot \nabla\phi, \theta^k\big) + \left(\frac{K}{\mathcal{P}r}\nabla\theta^k, \nabla\phi\right).
\end{aligned} \tag{17}$$

We impose homogeneous Dirichlet boundary conditions $(\mathbf{u}_w, p_w, \theta_w) = 0$ for this system since boundary conditions are time independent. We underline the fact that the Newton loop (following k) has to be iterated until convergence for each time step δt following the algorithm:

```
Navier–Stokes time loop following $n$
set $w^0 = (\mathbf{u}^n, p^n, \theta^n)$
    Newton iterations following $k$
        solve (17) to get $(\mathbf{u}_w, p_w, \theta_w)$
        actualize $w^{k+1} = w^k - (\mathbf{u}_w, p_w, \theta_w)$
    stop when $\|(\mathbf{u}_w, p_w, \theta_w)\| < \xi_N$
actualize $(\mathbf{u}^{n+1}, p^{n+1}, \theta^{n+1}) = w^{k+1}$
```
(18)

It is interesting to note that the previous system of equations depends only on $\mathbf{u}^n$ and θ^n and is independent of p^n, the pressure being in this approach a Lagrange multiplier for the divergence free constraint.

An alternative for the treatment of non-linear terms in the momentum equation is offered by the Galerkin characteristics method [31]. This method is implemented in FreeFem++ through a convection operator computing (for given mesh, convection velocity field and time step) $\Phi \circ \mathbf{X}^n$, with Φ the convected field and $\mathbf{X}^n$ an approximation of the characteristic by an integration back in time from t_{n+1} to t_n for each grid point $\mathbf{x}$. The weak formulation (14) can be easily rewritten using the characteristics method for the convective terms. The resulting system is then solved to advance the solution from t_n to t_{n+1} in one step. The drawback of the method is that it requires small time steps for accurately computing the convection terms. We note that the characteristics method implicitly uses a non-conservative form of the energy equation, while in the Newton method (17) the conservative form was used. The implementation of the Newton method in the non-conservative form led to the same numerical results for the test cases of natural convection of air.

It is important to remind that the characteristics method is slightly diffusive and doesn't need stabilization, since it embeds an up-winding mechanism for the convection terms. We shall compare below the characteristics method (which is a standard function in FreeFem++) with the new Newton method for natural convection flows. This will be also useful to assess on the necessity of stabilization procedures for the convection terms when the Newton method is used. For the cases presented below, the stabilization was not necessary, manifestly because considered flows were laminar (as seen from the scaling procedure presented above); the very refined grids used in computations with P_2 elements proved stable for all simulated flows.

3.2. Algorithms for the phase-change system with convection

We go back now to the system of equations (11). Two new difficulties are to be faced when building an algorithm for this system: the non-linearity introduced by the variation of the viscosity $\mu_{sl}(\theta)$ (from 1 in the fluid to 10^8 in the solid) and, similarly, the non-linearity introduced by the enthalpy function $S(\theta)$. For sake of simplicity, we consider in the heat equation that $K = C = 1$ (the dependence on θ of these quantities was implemented following the same method for the water freezing case). It should be noted that for phase-change material (paraffins), $C = K = 1$ is a physically valid assumption.

We adopt in this paper a full Newton linearization of the equations. A semi-implicit scheme using an explicit treatment of the viscosity $\mu_{sl}(\theta^n)$ in (11) was also tested with similar results (but larger computational times for convergence). The following Newton algorithm proved very robust and effective for the phase-change system (the new or modified terms compared to (17) are underlined):

$$\begin{aligned}
&b(\mathbf{u}_w, q) - \gamma(p_w, q) = b(\mathbf{u}^k, q) - \gamma(p^k, q),\\
&\frac{1}{\delta t}(\mathbf{u}_w, \mathbf{v}) + c(\mathbf{u}_w; \mathbf{u}^k, \mathbf{v}) + c(\mathbf{u}^k; \mathbf{u}_w, \mathbf{v})\\
&\quad + \underline{a(\mu_{sl}(\theta^k);\, \mathbf{u}_w, \mathbf{v})} + \underline{a\left(\frac{d\mu_{sl}}{d\theta}(\theta^k)\,\theta_w;\, \mathbf{u}^k, \mathbf{v}\right)} + b(\mathbf{v}, p_w) - \left(\frac{df_B}{d\theta}(\theta_k)\,\theta_w\,\mathbf{e}_y, \mathbf{v}\right)\\
&\quad = \frac{1}{\delta t}(\mathbf{u}^k - \mathbf{u}^n, \mathbf{v}) + c(\mathbf{u}^k; \mathbf{u}^k, \mathbf{v}) + \underline{a(\mu_{sl}(\theta^k);\, \mathbf{u}^k, \mathbf{v})} + b(\mathbf{v}, p^k) - (f_B(\theta^k)\,\mathbf{e}_y, \mathbf{v}),\\
&\frac{1}{\delta t}(\theta_w, \phi) - (\mathbf{u}^k \cdot \nabla\phi, \theta_w) - (\mathbf{u}_w \cdot \nabla\phi, \theta^k) + \left(\frac{K}{\mathcal{P}r}\nabla\theta_w, \nabla\phi\right) + \underline{\frac{1}{\delta t}\left(\frac{dS}{d\theta}(\theta^k)\,\theta_w,\, \phi\right)}\\
&\quad = \frac{1}{\delta t}(\theta^k - \theta^n, \phi) - (\mathbf{u}^k \cdot \nabla\phi, \theta^k) + \left(\frac{K}{\mathcal{P}r}\nabla\theta^k, \nabla\phi\right) + \underline{\frac{1}{\delta t}\big(S(\theta^k) - S(\theta^n),\, \phi\big)}.
\end{aligned}\tag{19}$$

3.3. Space discretization and mesh adaptivity

For the space discretization of systems (17) or (19) we use standard Taylor–Hood finite elements [35], approximating the velocity with P_2 finite elements (X_h space), and the pressure with the P_1 finite elements (M_h space):

$$\mathbf{X}_h = \left\{\mathbf{v} \in H^1(]0,1[^2)^2 \mid \forall K \in \mathcal{T}_h\ v_{|K} \in P_2\right\} \tag{20}$$

$$M_h = \left\{v \in H^1(]0,1[^2) \mid \forall K \in \mathcal{T}_h\ v_{|K} \in P_1\right\}, \tag{21}$$

where K is an element of the triangulation $\mathcal{T}_h$. The other variables (temperature, enthalpy) are discretized using P_1 finite elements.

Mesh adaptivity by metric control is a standard function offered by FreeFem++ [17]. The mesh generator uses Delaunay-type algorithms developed in [36,37]. The novelty of the present approach is that the adaptivity method is designed to reduce the impact of interpolation on the global accuracy. In order to achieve the mesh convergence, we first conserve all the nodes of the previous mesh if the distance between two mesh points in the metric is larger than $\sqrt{2}/2$ (note that variables are normalized before computing the metrics and the size of the edge of the mesh is close to unity in the metric length) and then continue the classical procedure of the insertion of new mesh points. This reduces the perturbations introduced when the solution is embedded by interpolation from the old mesh to the new one. To interpolate the variables from the old to the new mesh, we use the canonical finite element interpolation operator, depending of the type (P_2 or P_1) of the variable. This procedure could be slightly diffusive when applied every time step, but, at least for the cases simulated in this paper, it proved very stable and robust. Also, to reduce the impact of the interpolation on the global accuracy for time-depending problems, for the same variable used for adaptivity, we consider the metrics computed from actual (at t_{n+1}) and previous (at t_n) values (see also [14]).

The key idea for the mesh adaptivity (see also [36–40]) is to modify the scalar product used in an automatic mesh generator to evaluate distance and volume, in order to construct equilateral elements according to a new adequate metric. The scalar product is based on the evaluation of the Hessian $\mathcal{H}$ of the variables of the problem. Indeed, for a P_1 discretization of a variable χ, the interpolation error is bounded by:

$$\mathcal{E} = |\chi - \Pi_h\chi|_0 \le c \sup_{T\in\mathcal{T}_h}\ \sup_{x,y,z\in T} \left|\mathcal{H}(x)\right|(y-z, y-z) \tag{22}$$

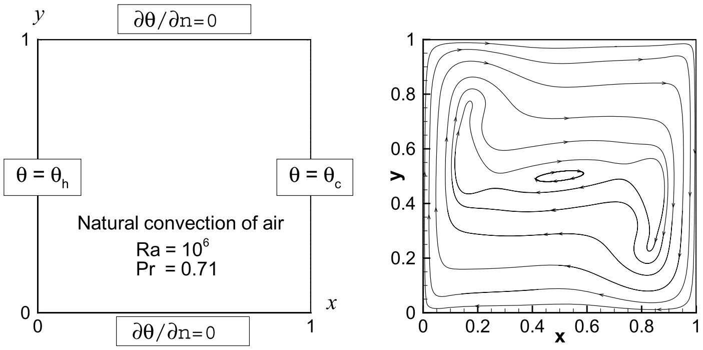


Fig. 2. Natural convection of air in a differentially heated cavity. Problem definition and streamlines of the steady flow.

where $\Pi_h \chi$ is the P_1 interpolate of χ, $|\mathcal{H}(x)|$ is the Hessian of χ at point x after being made positive definite. We can infer that, if we generate, using a Delaunay procedure (*e.g.* [36]), a mesh with edges close to the unit length in the metric $\mathcal{M} = \frac{|\mathcal{H}|}{(c\mathcal{E})}$, the interpolation error $\mathcal{E}$ will be equally distributed over the edges a_i of the mesh. More precisely, we have

$$\frac{1}{c\mathcal{E}} a_i^T \mathcal{M} a_i \le 1. \tag{23}$$

The previous approach could be generalized for a vector variable $\chi = [\chi_1, \chi_2]$. After computing the metrics $\mathcal{M}_1$ and $\mathcal{M}_2$ for each variable, we define a metric intersection $\mathcal{M} = \mathcal{M}_1 \cap \mathcal{M}_2$, such that the unit ball of $\mathcal{M}$ is included in the intersection of the two unit balls of metrics $\mathcal{M}_2$ and $\mathcal{M}_1$ (for details, see the procedure defined in [37]). Since the error is related to the Hessian matrix, when P_2 variables are used for the mesh adaptation, they are first linearly interpolated before proceeding to the metrics calculation (see also [41]). This renders the adaptive mesh not optimal for P_2 variables and theoretically impacts the overall accuracy of the discretization in the refined zone. For practical calculations, the use of fine grids near interfaces makes these effects negligible. Starting from this observation, the use of the so-called mini element (see [42,43]), offering a global linear convergence, could be an interesting alternative for the space discretization, instead of the Taylor–Hood element, to optimize the computational time. This possibility will be explored in future studies, since the mini-element can be programmed in FreeFem++.

For the cases considered in the following, we used up to four metrics intersections to adapt the mesh (the details are indicated for each case). The anisotropy of the mesh is a parameter of the algorithm and it was set to values close to 1. This is an inevitable limitation since we also impose the minimum edge-length of triangles to avoid too large meshes.

The flexibility of the mesh adaptivity algorithm will be used to simultaneously take into account several metrics computed for different variables monitoring the evolution of the phase-change system. For natural convection system, mesh will be refined using the velocity components and temperature as reference variables. For phase-change systems, to accurately track the solid–liquid interface we add the variation of the enthalpy source term as adaptivity criterion. For water, we also add an extra function tracking the anomalous change of density around 4 °C. The capabilities of the mesh adaptivity algorithm are illustrated in the next section.

4. Numerical validation

In this section we test the new numerical system by considering well defined benchmarks used to validate numerical codes for phase-change problems. The difficulty of the computed cases is increased progressively by considering the following physical systems: natural convection of air, natural convection of water, melting of a phase-change system and, finally, the freezing of pure water. For each test case we quantitatively compare our results with experimental and previously published numerical data, showing the benefits of the new method.

4.1. Natural convection of air

We start by testing the Newton algorithm (17) for the Navier–Stokes–Boussinesq equations presented in Section 3.1. We consider a classical benchmark that has been thoroughly studied in literature: the natural convection of air in a differentially heated square cavity. The configuration is sketched in Fig. 2.

For air, the Boussinesq term $f_B(\theta)$ is linear and takes the form (3). To assess for the accuracy of our method by comparing to previous studies [7,44], we chose the flow parameters as follows: $Ra = 10^6$, $Pr = 0.71$, $\theta_h = 0.5$ and $\theta_c = -0.5$. For this value of Ra, a steady state with a single convection cell is reached. The computed streamlines showing the flow structure at steady state are represented in Fig. 2. All computations in this section are performed with a fixed triangular mesh, generated by the Delaunay algorithm starting with $M = 80$ points on each side of the square.

A classical validation for this computation benchmark is to extract the horizontal velocity profile $u(y)$ at mid-domain $(x = 0.5)$ and compare the maximum value $u_{\max}$ and its location Y to values obtained by spectral-accurate simulations by

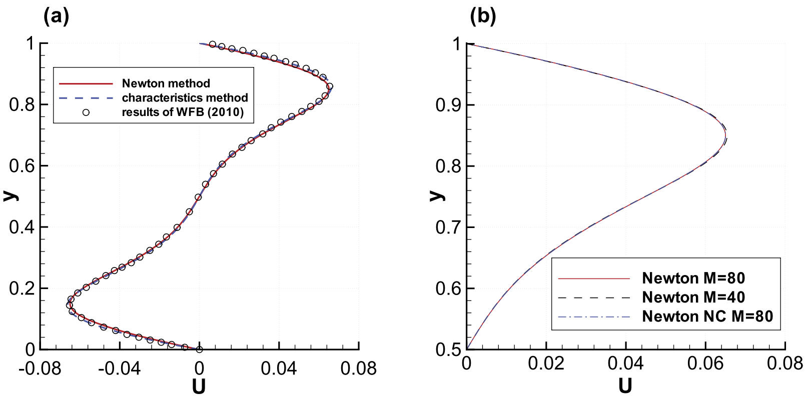


Fig. 3. Natural convection of air in a differentially heated cavity. Longitudinal velocity profile along the vertical symmetry line (a) results obtained using the present Newton method (conservative formulation of the convective term in the heat equation), with mesh resolution $M = 80$; comparison with the characteristics-Galerkin method and the results of Wang, Faghri and Bergman [7] (b) results with different mesh resolutions or a non-conservative (NC) formulation.

Table 1
Natural convection of air in a temperature driven cavity. Maximum value $u_{\max}$ of the horizontal velocity profile at mid-domain ($x = 0.5$) and location Y of this maximum. Comparison to reference values by Le Quéré [44].

Run		$u_{\max}$ at $x = 0.5$ (error)	Y (error)
Reference values	spectral	0.0648344	0.850
Newton	$M = 40$	0.0654152 (0.89%)	0.849812 (0.02%)
Newton	$M = 80$	0.0650082 (0.26%)	0.849906 (0.01%)
Char-Galerkin	$M = 40$	0.0667870 (3.01%)	0.857322 (0.86%)
Char-Galerkin	$M = 80$	0.0662229 (2.14%)	0.856160 (0.72%)

Le Quéré [44]. Fig. 3(a) shows that the horizontal velocity profile from present computations is in good agreement with data extracted from [7]. Also, the Newton method behaves similarly to the characteristics-Galerkin method. Fig. 3(b) displays a detail of the velocity profile, showing the mesh convergence of the results ($M = 40$ and $M = 80$). Since the characteristics method uses a non-conservative form of the energy equation, we have also implemented this form for the Newton method. Fig. 3(b) also shows that the results using either a conservative or non-conservative form of non-linear terms in the energy equation are identical for $M = 80$.

Table 1 offers a quantitative assessment of the accuracy of the present Newton method. The values of $u_{\max}$ and its location Y are compared to reference values from [44]. The Newton method gives results quasi-identical to reference values. The characteristics-Galerkin method is less accurate, but still offers reasonable agreement with reference values, within 2% relative error. We also recall that the characteristics method needs a very small time step for refined meshes ($\delta t = 8 \cdot 10^{-5}$ for $M = 80$), while the Newton method allows larger time steps ($\delta t = 10^{-3}$ for $M = 80$) and, consequently, converges faster to a steady state.

We can conclude from this section that the proposed Newton method offers an efficient way to solve the Navier–Stokes–Boussinesq system of equations for natural convection. After this validation, we test in the following section the capability of the method to deal with nonlinear buoyancy terms.

4.2. Natural convection of water

Pure water exhibits a nonlinear density variation for $T < 10.2\,^\circ\mathrm{C}$ with a maximum at $T_m = 4.0293\,^\circ\mathrm{C}$. We use below the following density–temperature relationship proposed in [45]:

$$\rho(T) = \rho_m\big(1 - w|T - T_m|^q\big), \tag{24}$$

with $\rho_m = 999.972$ [kg/m^3], $w = 9.2793 \cdot 10^{-6}$ [(°C)$^{-q}$], and $q = 1.894816$. Choosing the fusion temperature $T_f = 0\,^\circ\mathrm{C}$ as reference, the buoyancy term $f_B = g(\rho_{ref} - \rho)/\rho_{ref}$ appearing in (11) becomes after scaling:

$$f_B(\theta) = \frac{\mathcal{R}a}{\mathcal{P}r\,\mathcal{R}e^2}\,\frac{1}{\beta(T_h - T_c)}\,\frac{\rho(\theta_f) - \rho(\theta)}{\rho(\theta_f)}, \tag{25}$$

where $\beta = (1/\rho_m)(d\rho/dT)$ is the thermal expansion coefficient with the value [46] $\beta = 6.91 \cdot 10^{-5}$ [(K)$^{-1}$]. Note that (25) compared to the classical linear form (3) not only introduces a new nonlinear term, but also the coefficient in front of this term is very large, since it is proportional to $\mathcal{R}a/(\beta(T_h - T_c))$.

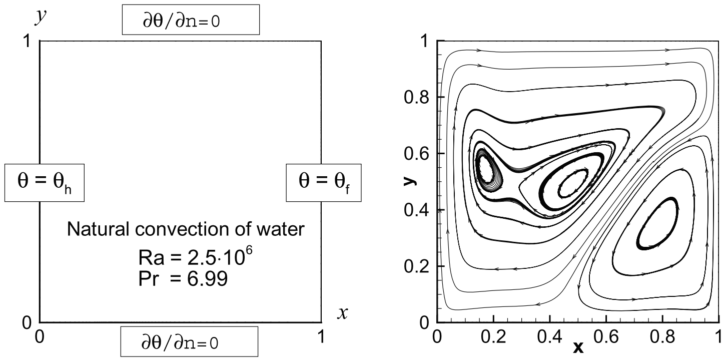


Fig. 4. Natural convection of water in a differentially heated cavity. Problem definition and streamlines of the steady flow.

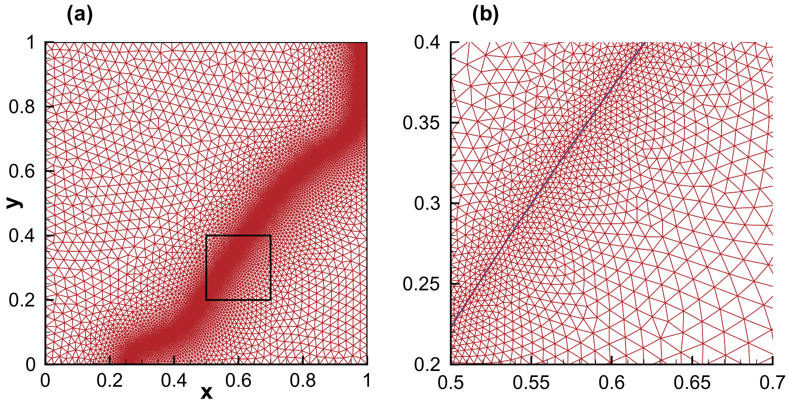


Fig. 5. Natural convection of water in a differentially heated cavity. (a) Mesh corresponding to the final steady state of the computation. The mesh is refined following the line $T = T_m$, with $T_m = 4.0293\,^\circ\mathrm{C}$ corresponding to the maximum of the water density nonlinear variation. (b) Zoom inside the mesh corresponding to the small square in (a) (the level line $T = T_m$ is also displayed).

The Newton algorithm (17), with the nonlinear f_B term given by (25) is solved for the natural convection test case investigated experimentally and numerically in [25,47,48]. It represents (see Fig. 4) a differentially heated cavity filled with pure distilled water. The hot temperature wall is maintained at $T_h = 10\,^\circ\mathrm{C}$, while the cold wall is held at $T_c = T_f = 0\,^\circ\mathrm{C}$. The non-dimensional parameters describing the flow are (see [48] for physical details): $\mathcal{R}a = 2.518084 \cdot 10^6$, $\mathcal{P}r = 6.99$.

The streamlines of the computed steady state are displayed in Fig. 4 and depicts a flow structure very similar to that observed in experiments (to be compared with Fig. 4 from [47]). Due to the anomalous thermal variation of water density, two recirculating zones are formed in the flow: a lower (abnormal) recirculation in the vicinity of the cold wall where $T < T_m$ and an upper (normal) one where the density decreases with temperature ($T > T_m$).

Since the two recirculating zones are separated by the line $T = T_m$ we have used the mesh adaptivity capability of the method to refine the grid along this line. The metrics used for adaptivity are computed from the two components of the velocity and a P_1 function $\phi(T)$ "tracking" the value T_m, defined by the general regularization expression (7) with $f_l = 0$, $f_s = 1$ and $F(\theta; 1, T_m + 0.01, 0.02)$. To reduce the impact of the interpolation on the global accuracy (see also [14]), we use both $\phi(T^n)$ and $\phi(T^{n+1})$ in the adaptivity procedure. The final mesh is displayed in Fig. 5, clearly showing that the mesh is refined along the line $T = T_m$. This allows to accurately capture the structure and the extent of the two recirculating zones, features that are difficult to obtain with fixed meshes (see discrepancies described in [25,47,48]).

The corresponding temperature field of the steady state is presented in Fig. 6. The line of maximum density corresponds to $\theta = \theta_m = 0.4$. The temperature profile $\theta(x)$ along the horizontal symmetry line of the cavity is in good agreement with the numerical results of Michalek and Kowalewski [48] obtained with FV and FD codes (FLUENT and FRECONV3V), commonly used in the heat transfer community. Differences are visible in the vicinity of the maximum density line, region where our mesh is well refined to capture the separation line between the two recirculation zones. It should be noted that the FLUENT simulations in [48] are performed with a fixed uniform grid with 380 × 380 nodes, while our adapted grid (see Fig. 5) has only 8751 vertices (17 257 triangles).

4.3. Melting of phase change material (PCM)

After the validation of the Navier–Stokes–Boussinesq solver, we test in this subsection the numerical system solving the model (19) for a phase-change system with convection. We consider the test case of the melting of an octadecane PCM

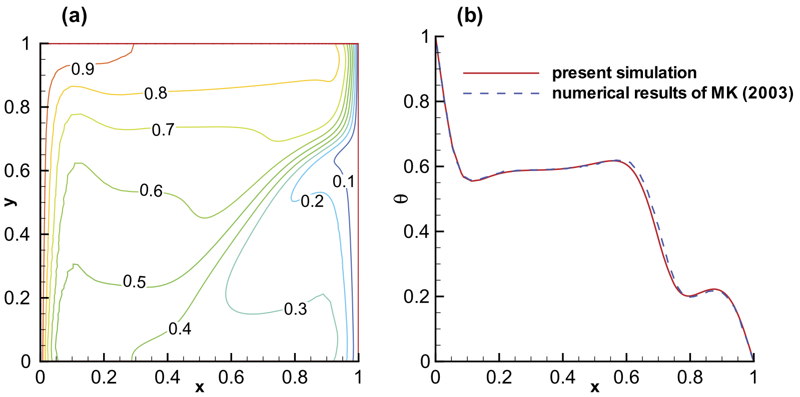


Fig. 6. Natural convection of water in a differentially heated cavity. Non-dimensional temperature θ at steady state. (a) Two-dimensional temperature field. (b) Temperature profile along the horizontal symmetry line. Comparison with the numerical results of Michalek and Kowalewski, 2003 [48].

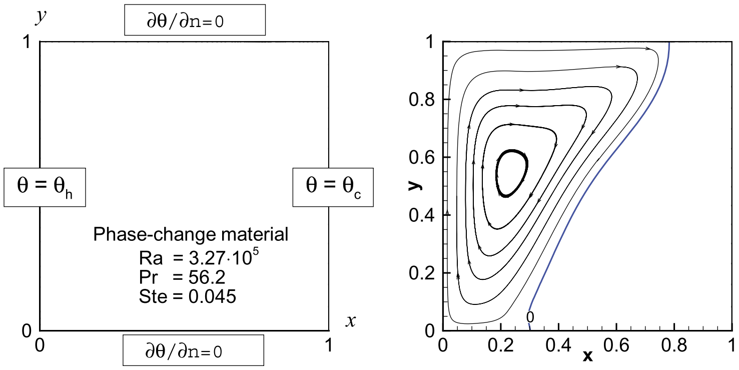


Fig. 7. Melting of a (PCM) phase change material (octadecane). Problem definition and streamlines of the flow developing in the liquid part at $t = 71$. The solid–liquid interface ($\theta = 0$) is also represented (in blue). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

in a square cavity. For this case both experimental and numerical results are available [7,8,49]. The material characteristics (specific heat C and conductivity K) are considered constant, as in the previous numerical studies [7,8]. The physical configuration is sketched in Fig. 7. The material is initially solid ($\theta_0 = -0.01$) and melts progressively starting from the left boundary, maintained at the hot temperature $\theta_h = 1$. The right boundary is also isothermal with cold temperature $\theta_c = -0.01$, while horizontal boundaries are adiabatic. Fig. 7 also offers an image of the system evolution by showing the configuration of the PCM at $t = 78.7$, with the streamlines of the flow developing in the liquid phase and the liquid–solid interface (isoline $\theta = 0$).

Two new non-linearities are introduced in the system (19): the variable viscosity μ_{sl} and the enthalpy source term S. The former is regularized using the function (7), as shown in Fig. 1(a), following a definition of the mushy region $[\theta_f - \varepsilon_1, \theta_f + \varepsilon_2]$ according to [7]: $\theta_f = 0$ and $\varepsilon_1 = \varepsilon_2 = \varepsilon = 0.01$. The S term is regularized using the same function (7), with $S_s = 0$, $S_l = 1/\mathcal{S}te$, where $\mathcal{S}te$ denotes the Stefan number. The physical parameters describing the test case are: $\mathcal{R}a = 3.27 \cdot 10^5$, $\mathcal{P}r = 56.2$ and $\mathcal{S}te = 0.045$.

The computation starts from a refined mesh near the hot boundary and then uses mesh adaptivity at each time step. The mesh is refined using metrics computed from three variables: the two fluid velocities and the enthalpy source term S. Again, to reduce the impact of the interpolation on the global accuracy, we use both two successive fields (S^n) and (S^{n+1}) in the adaptivity procedure. This allows to well refine the fluid part of the domain and the mushy region around the liquid–solid interface. A picture of the mesh and the corresponding temperature field is presented in Fig. 8. We notice the smooth appearance of iso-lines for the temperature at the interface, which is not the case in the previous simulations [7,8] using a fixed mesh and a velocity correction for grid points undergoing phase change.

A more quantitative assessment of the capabilities of our method is offered by comparing the position of the liquid–solid interface at $t = 78.7$. We compare in Fig. 9 our results with the experimental data by Okada [49] and previously published numerical results by Wang, Faghri and Bergman [7]. The obtained shape and position of the liquid–solid interface is closer to experimental results than numerical results reported in [7]. This is a direct consequence of the mesh adaptivity capabilities of our method.

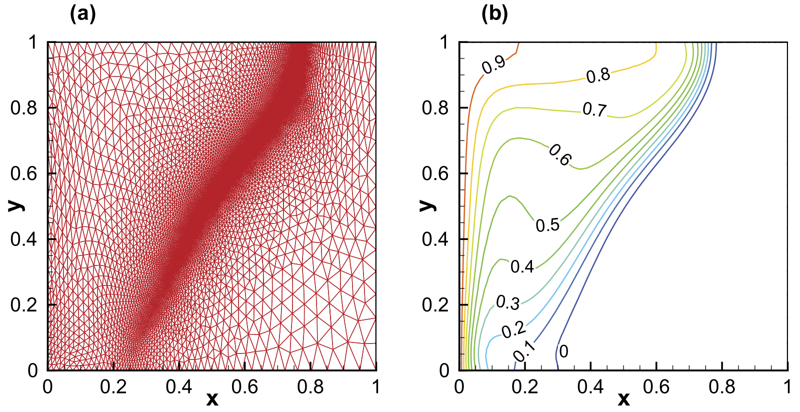


Fig. 8. Melting of a (PCM) phase change material (octadecane). Configuration at $t = 78.7$: (a) adapted finite-element mesh and (b) temperature iso-lines in the liquid phase.

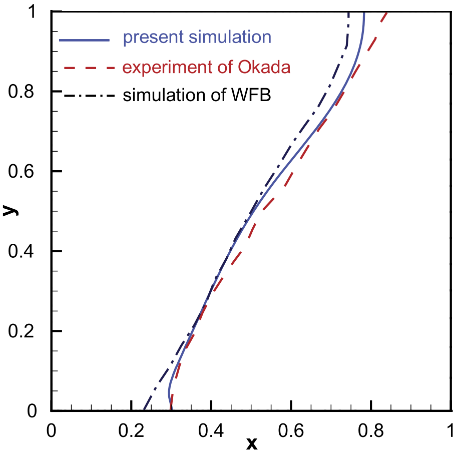


Fig. 9. Melting of a (PCM) phase change material (octadecane). Solid–liquid interface at $t = 78.7$: comparison with numerical results of Wang, Faghri and Bergman [7] and experimental data of Okada [49].

4.4. Water freezing

For the last validation of the numerical system, we consider the difficult case of water freezing in a square cavity. This problem was investigated experimentally and numerically in [25,47,48]. The configuration is the same as described in Section 4.2 for the natural convection of water. After the convection steady pattern is obtained in the cavity, freezing starts by dropping the temperature of the cold wall from $T_c = 0\,°\mathrm{C}$ to $T_c = -10\,°\mathrm{C}$. The physical parameters describing the problem are: $\mathcal{R}a = 2.518084 \cdot 10^6$, $\mathcal{P}r = 6.99$, $\mathcal{S}te = 0.125$.

This case brings together all the non-linearities previously described: nonlinear buoyancy term (25), variable viscosity and enthalpy source term. In the system (19) we took into account the variation of the thermal conductivity K, using the same type of regularization (7). The simulation starts, as in experiment, from the steady state obtained for the natural convection (see Fig. 4). The new boundary condition on the right cold wall is imposed by setting a very thin layer of $\delta x = 0.01$ with constant temperature $\theta = \theta_c$ and zero velocity.

Since in previous numerical contributions [25,47,48], important discrepancies between computed and experimental configurations were obtained, we directly superimpose in Fig. 10 the experimental image from [25] with our numerical results for the same physical time $\tau = 2340$ [s]. Although the experimental image might not be perfectly fitted in the square $[0, 1]^2$, we can nevertheless notice the remarkable agreement for the shape and position of the solid–liquid interface computed with the present numerical system. The flow pattern in the liquid phase also corresponds very well to the experimental image. Since a good agreement with the experiment was sought, the simulation was performed with very small time steps ($\delta t = 10^{-4} \approx 0.15$ [s]), but still reasonable grids (less that 7000 nodes) due to the efficiency of the adaptivity algorithm.

The refined mesh is displayed in Fig. 11(a), clearly showing the anisotropic mesh refinement along the two lines of interest: the solid–liquid interface ($T = 0\,°\mathrm{C}$) and the line of maximum water density ($T = 4\,°\mathrm{C}$). As a consequence, the temperature level lines in Fig. 11(b) are smooth and clearly define the two interfaces in the system: the liquid–solid interface ($\theta = 0$) and the density inversion interface ($\theta = 0.4$) separating the two recirculating liquid regions.

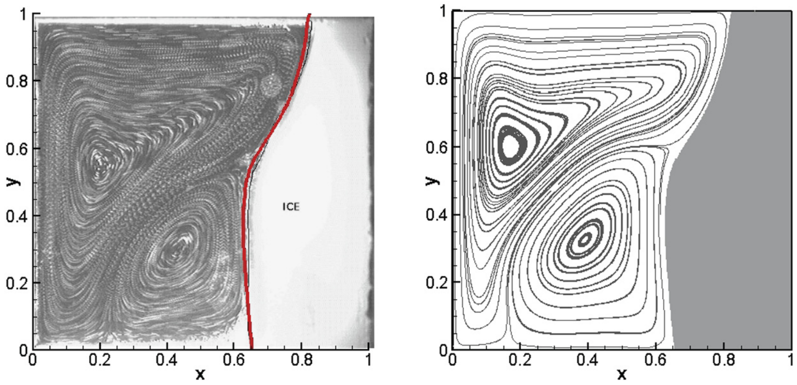


Fig. 10. Freezing of pure water. Configuration at (physical time) $\tau = 2340$ [s]: (a) experimental image from [25]; the thick red line represents the solid–liquid interface computed with the present method (b) computed streamlines showing the two recirculating zones in the fluid phase. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

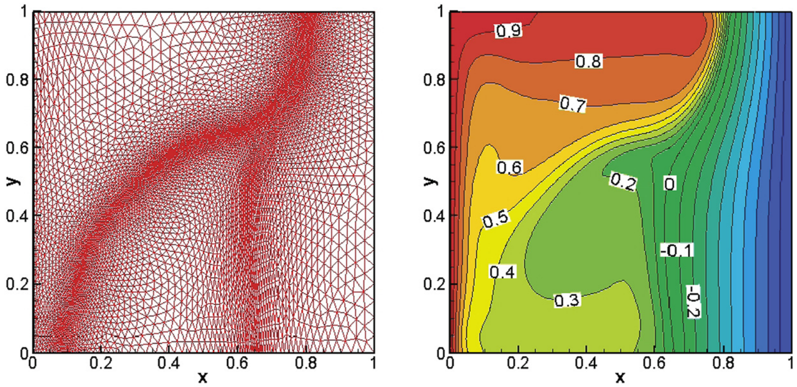


Fig. 11. Freezing of pure water. Computed configuration at (physical time) $\tau = 2340$ [s]: (a) finite-element mesh refined along the solid–liquid interface ($T = 0\,°\mathrm{C}$) and also along the line of maximum water density ($T = 4\,°\mathrm{C}$) (b) temperature iso-lines.

The water freezing case is computationally very challenging and detailed physical models are needed in order to obtain a good agreement with experiments. Some models to take into account realistic boundary conditions were proposed and tested in [25,47,48]. They could be implemented very easily with the present numerical system and a detailed investigation of the water freezing case will be reported in a future contribution.

5. Conclusions

A new single domain numerical approach was derived and validated for phase-change systems. The key ingredients of the method are the use of an adaptive finite element method with a well chosen regularization of the functions representing the variation of thermodynamic properties at the solid–liquid interface, and a fully linearized Newton algorithm for the time integration of the system of equations. The advantage of the adaptivity method by metric control is to allow accurate and simultaneous tracking of different interfaces in the system (the liquid–solid interface but also the density inversion interface for water flows).

The advantage of the Newton global formulation is to easily permit the implementation of different types of non-linearities in the system of equations and, consequently, to explore numerically different physical models. This feature was illustrated by considering four test cases with increasing complexity:

(i) natural convection of air in a differentially heated cavity,
(ii) natural convection of water in a differentially heated cavity,
(iii) melting of a phase-change material,
(iv) water freezing.

For all considered test cases, a very good agreement with experimental data or previously published numerical results is obtained, proving the capability of the method to tackle a large range of problems, from natural convection to melting and solidification.

The method derived in this contribution was implemented using FreeFem++ software, using a programming syntax close to the mathematical formulation. Since FreeFem++ is a free software, the method could be easily implemented and tested by anyone interested in simulating phase-change problems (the general adaptivity procedure is now the standard function *adaptmesh* in FreeFem++). This offers the possibility to address other computational challenges related to different physical or mathematical models in this field, including complex phenomena (*e.g.* magnetohydrodynamics [50]) or inverse problems (*e.g.* reconstruction of material properties [30]). A further development of the method will include the extension of the method for three-dimensional cases. This ongoing work showed that a careful investigation of several algorithmic issues is necessary, as also pointed out in [15]: change from direct to efficient iterative solvers, improve the robustness of the anisotropic mesh adaptivity algorithms, etc. Three-dimensional simulations are beyond the scope of the present contribution and will be reported in a forthcoming paper.

Acknowledgements

We thank the two anonymous reviewers for their valuable comments and suggestions.

References

[1] A. Faghri, Y. Zhang, Transport Phenomena in Multiphase Systems, Elsevier, 2006.

[2] K. Morgan, A numerical analysis of freezing and melting with convection, Comput. Methods Appl. Mech. Eng. 28 (3) (1981) 275–284.

[3] V.R. Voller, C. Prakash, A fixed grid numerical modelling methodology for convection–diffusion mushy region phase-change problems, Int. J. Heat Mass Transf. 30 (8) (1987) 1709–1719.

[4] P. Jany, A. Bejan, Scaling theory of melting with natural convection in an enclosure, Int. J. Heat Mass Transf. 31 (6) (1988) 1221–1235.

[5] K.J. Evans, D. Knoll, M. Pernice, Development of a 2-d algorithm to simulate convection and phase transition efficiently, J. Comput. Phys. 219 (1) (2006) 404–417.

[6] G. Vidalain, L. Gosselin, M. Lacroix, An enhanced thermal conduction model for the prediction of convection dominated solid–liquid phase change, Int. J. Heat Mass Transf. 52 (2009) 1753–1760.

[7] S. Wang, A. Faghri, T.L. Bergman, A comprehensive numerical model for melting with natural convection, Int. J. Heat Mass Transf. 53 (2010) 1986–2000.

[8] Z. Ma, Y. Zhang, Solid velocity correction schemes for a temperature transforming model for convection phase change, Int. J. Numer. Methods Heat Fluid Flow 16 (11) (2006) 204–225.

[9] Y. Cao, A. Faghri, W.S. Chang, A numerical analysis of Stefan problems for generalized multi-dimensional phase-change structures using the enthalpy transforming model, Int. J. Heat Mass Transf. 32 (7) (1989) 1289–1298.

[10] Y. Cao, A. Faghri, A numerical analysis of phase change problem including natural convection, ASME J. Heat Transf. 112 (1990) 812–815.

[11] C. Elliott, Error analysis of the enthalpy method for the Stefan problem, IMA J. Numer. Anal. 7 (1987) 61–71.

[12] C.H. Li, A finite-element front-tracking enthalpy method for Stefan problems, IMA J. Numer. Anal. 3 (1983) 87–107.

[13] R.T. Tenchev, J.A. Mackenzie, T.J. Scanlon, M.T. Stickland, Finite element moving mesh analysis of phase change problems with natural convection, Int. J. Heat Fluid Flow 26 (4) (2005) 597–612.

[14] Y. Belhamadia, A. Fortin, E. Chamberland, Anisotropic mesh adaptation for the solution of the Stefan problem, J. Comput. Phys. 194 (1) (2004) 233–255.

[15] Y. Belhamadia, A.S. Kane, A. Fortin, An enhanced mathematical model for phase change problems with natural convection, Int. J. Numer. Anal. Model. 3 (2) (2012) 192–206.

[16] F. Hecht, O. Pironneau, A.L. Hyaric, K. Ohtsuke, FreeFem++ (manual), http://www.freefem.org, 2007.

[17] F. Hecht, New developments in FreeFem++, J. Numer. Math. 20 (2012) 251–266.

[18] I. Danaila, P. Kazemi, A new Sobolev gradient method for direct minimization of the Gross–Pitaevskii energy with rotation, SIAM J. Sci. Comput. 32 (2010) 2447–2467.

[19] I. Danaila, F. Hecht, A finite element method with mesh adaptivity for computing vortex states in fast-rotating Bose–Einstein condensates, J. Comput. Phys. 229 (2010) 6946–6960.

[20] Y. Zhang, I. Danaila, Existence and numerical modelling of vortex rings with elliptic boundaries, Appl. Math. Model. 37 (2013) 4809–4824.

[21] J.E. Dennir Jr., R.B. Schmabel, Numerical Methods for Unconstrained Optimization and Nonlinear Equations, Prentice Hall, Englewood Cliffs, NJ, 1986.

[22] T.W.H. Sheu, R.K. Lin, Newton linearization of the incompressible Navier–Stokes equations, Int. J. Numer. Methods Fluids 44 (3) (2004) 297–312.

[23] T.W.H. Sheu, R.K. Lin, On a high-order Newton linearization method for solving the incompressible Navier–Stokes equations, Int. J. Numer. Methods Eng. 62 (11) (2005) 1559–1578.

[24] Z. Chen, T. Shih, X. Yue, Numerical methods for Stefan problems with prescribed convection and nonlinear flux, IMA J. Numer. Anal. 20 (2000) 81–98.

[25] T.A. Kowalewski, M. Rebow, Freezing of water in differentially heated cubic cavity, Int. J. Comput. Fluid Dyn. 11 (1999) 193–210.

[26] V.R. Voller, M. Cross, N.C. Markatos, An enthalpy method for convection/diffusion phase change, Int. J. Numer. Methods Eng. 24 (1) (1987) 271–284.

[27] A.D. Brent, V.R. Voller, K.J. Reid, Enthalpy-porosity technique for modeling convection–diffusion phase change: application to the melting of a pure metal, Numer. Heat Transf. 13 (3) (1988) 297–318.

[28] W. Jin, W. Tao, Y. He, Z. Li, Analysis of inconsistency of simple-like algorithms and an entirely consistent update technique—the cut algorithm, Numer. Heat Transf., Part B 53 (2008) 289–312.

[29] Z. Sheng, M. Thiriet, F. Hecht, An efficient numerical method for the equations of steady and unsteady flows of homogeneous incompressible newtonian fluid, J. Comput. Phys. 230 (2011) 551–571.

[30] V. Bukshtynov, B. Protas, Optimal reconstruction of material properties in complex multiphysics phenomena, J. Comput. Phys. 242 (2013) 889–914.

[31] O. Pironneau, J. Liou, T. Tezduyar, Characteristic-Galerkin and Galerkin/least-squares space–time formulations for the advection–diffusion equations with time-dependent domains, Comput. Methods Appl. Mech. Eng. 100 (1992) 117–141.

[32] R. Temam, Navier–Stokes Equations and Nonlinear Functional Analysis, SIAM, Philadelphia, 1983.

[33] V. Girault, P.-A. Raviart, Finite Element Methods for Navier–Stokes Equations, Springer-Verlag, Berlin, 1986.

[34] A. Quarteroni, A. Valli, Numerical Approximation of Partial Differential Equations, Springer-Verlag, Berlin, Heidelberg, 1994.

[35] C. Taylor, P. Hood, A numerical solution of the Navier–Stokes equations using the finite element technique, Comput. Fluids 1 (1973) 73–100.

[36] P.L. George, H. Borouchaki, Delaunay Triangulation and Meshing, Hermès, Paris, 1998.

[37] P.J. Frey, P.L. George, Maillages, Hermès, Paris, 1999.

[38] M. Castro-Diaz, F. Hecht, B. Mohammadi, Anisotropic grid adaptation for inviscid and viscous flows simulations, Int. J. Numer. Methods Fluids 25 (2000) 475–491.

[39] F. Hecht, B. Mohammadi, Mesh adaptation by metric control for multi-scale phenomena and turbulence, in: AIAA Paper 97, 1997, p. 0859.

[40] B. Mohammadi, O. Pironneau, Applied Shape Design for Fluids, Oxford Univ. Press, 2000.

[41] Y. Belhamadia, A. Fortin, E. Chamberland, Three-dimensional anisotropic mesh adaptation for phase change problems, J. Comput. Phys. 201 (2) (2004) 753–770.

[42] D.N. Arnold, F. Brezzi, M. Fortin, A stable finite element for the Stokes equations, Calcolo 21 (1984) 337–344.

[43] F. Brezzi, M. Fortin, Mixed and Hybrid Finite Element Methods, Springer-Verlag, New York, 1991.

[44] P.L. Queré, Accuratte solutions to the square thermally driven cavity at high Rayleigh number, Comput. Fluids 20 (1991) 24–41.

[45] B. Gebhart, J. Mollendorf, A new density relation for pure and saline water, Deep-Sea Res. 24 (1977) 831–848.

[46] T.J. Scanlon, M.T. Stickland, A numerical analysis of buoyancy-driven melting and freezing, Int. J. Heat Mass Transf. 47 (2004) 429–436.

[47] M. Giangi, T.A. Kowalewski, F. Stella, E. Leonardi, Natural convection during ice formation: numerical simulation vs. experimental results, Comput. Assist. Mech. Eng. Sci. 7 (2000) 321–342.

[48] T. Michalek, T.A. Kowalewski, Simulations of the water freezing process — numerical benchmarks, Task Q. 7 (3) (2003) 389–408.

[49] M. Okada, Analysis of heat transfer during melting from a vertical wall, Int. J. Heat Mass Transf. 27 (1984) 2057–2066.

[50] Y. Safa, M. Flueck, J. Rappaz, Numerical simulation of thermal problems coupled with magnetohydrodynamic effects in aluminium cell, Appl. Math. Model. (2009) 1479–1492.