A finite-element toolbox for the simulation of solid-liquid phase-change systems with natural convection

Aina Rakotondrandisa, Georges Sadaka, Ionut Danaila

A finite-element toolbox for the simulation of solid-liquid phase-change systems with natural convection

Aina Rakotondrandisa[a], Georges Sadaka[a], Ionut Danaila[a,*]

[a] *Laboratoire de Mathématiques Raphaël Salem, Université de Rouen Normandie, CNRS UMR 6085, Avenue de l'Université, BP 12, F-76801 Saint-Étienne-du-Rouvray, France*

Abstract

We present and distribute a new numerical system using classical finite elements with mesh adaptivity for computing two-dimensional liquid-solid phase-change systems involving natural convection. The programs are written as a toolbox for FreeFem++ (www.freefem.org), a free finite-element software available for all existing operating systems. The code implements a single domain approach. The same set of equations is solved in both liquid and solid phases: the incompressible Navier-Stokes equations with Boussinesq approximation for thermal effects. This model describes naturally the evolution of the liquid flow which is dominated by convection effects. To make it valid also in the solid phase, a Carman-Kozeny-type penalty term is added to the momentum equations. The penalty term brings progressively (through an artificial mushy region) the velocity to zero into the solid. The energy equation is also modified to be valid in both phases using an enthalpy (temperature-transform) model introducing a regularized latent-heat term. Model equations are discretized using Galerkin triangular finite elements. Piecewise quadratic (P2) finite-elements are used for the velocity and piecewise linear (P1) for the pressure. For the temperature both P2 or P1 discretizations are possible. The coupled system of equations is integrated in time using a second-order Gear scheme. Non-linearities are treated implicitly and the resulting discrete equations are solved using a Newton algorithm. An efficient mesh adaptivity algorithm using metrics control is used to adapt the mesh every time step. This allows us to accurately capture multiple solid-liquid interfaces present in the domain, the boundary-layer structure at the walls and the unsteady convection cells in the liquid. We present several validations of the toolbox, by simulating benchmark cases of increasing difficulty: natural convection of air, natural convection of water, melting of a phase-change material, a melting-solidification cycle, and, finally, a water freezing case. Other similar cases could be easily simulated with this toolbox, since the code structure is extremely versatile and the syntax very close to the mathematical formulation of the model.

Keywords: melting, solidification, PCM, Newton method, finite element, mesh adaptivity, Navier-Stokes, Boussinesq, FreeFem++ .

*Corresponding author. Tel.: (+33) 2 32 95 52 50

Email addresses: aina.rakotondrandisa@etu.univ-rouen.fr (Aina Rakotondrandisa), georges.sadaka@univ-rouen.fr (Georges Sadaka), ionut.danaila@univ-rouen.fr (Ionut Danaila)

Programm summary
Program Title: PCM-ToolBox
Catalogue identifier:
Program summary URL:
Program obtainable from:
Licensing provisions: Apache License, 2.0 (Apache-2.0)
No. of lines in distributed program, including test data, etc.: 3077
No. of bytes in distributed program, including test data, etc.: 2.5 Mo
Distribution format: .zip
Programming language: FreeFem++ (free software, www3.freefem.org)
Computer: PC, Mac, Super-computer.
Operating system: Windows, Mac OS, Linux.
Classification:
Nature of problem: The software computes 2D configurations of liquid-solid phase-change problems with convection in the liquid phase. Natural convection, melting and solidification processes are illustrated in the paper. The software can be easily modified to take into account different related physical models.
Solution method: We use a single domain approach, solving the incompressible Navier-Stokes equations with Boussinesq approximation in both liquid and solid phases. A Carman-Kozeny-type penalty term is added to the momentum equations to bring the velocity to zero into the solid phase. An enthalpy model is used in the energy equation to take into account the phase change. Discontinuous variables (latent heat, material properties) are regularized through an intermediate (mushy) region. Space discretization is based on Galerkin triangular finite elements. Piecewise quadratic (P2) finite-elements are used for the velocity and piecewise linear (P1) for the pressure. For the temperature both P2 or P1 discretizations are possible. A second order Gear scheme is used for the time integration of the coupled system of equations. Non-linear terms are treated implicitly and the resulting discrete equations are solved using a Newton algorithm. A mesh adaptivity algorithm is implemented to reduce the computational time and increase the local space accuracy when (multiple) interfaces are present.
Running time:
From minutes for natural convection case to hours for melting or solidification (on a personal laptop). The water freezing case may require several days of computational time.

1. Introduction

Solid-liquid phase-change problems are encountered in numerous practical applications, ranging from metal casting and thermal energy storage (phase-change materials) to food freezing and cryosurgery. Melting and solidification are also fundamental processes in geophysical problems, such as Earth's mantle formation, lava lakes or magma chambers. In many of these problems, temperature gradients induce buoyancy forces in the liquid (melted) phase generating a significant convective flow. Consequently, the appropriate mathematical description of the liquid phase is the usual model for the natural convection flow: the incompressible Navier-Stokes system of equations with Boussinesq approximation for thermal (buoyancy) effects. In this model, the energy conservation equation is written as a convection-diffusion equation for the temperature. In the solid phase, conduction is the main phenomenon and the appropriate model is the classical heat equation. The main modelling difficulty is to link these two models by taking into account the separation of the two phases by a sharp interface, across which thermodynamic properties are discontinuous. We offer below a short description of the two main approaches suggested in the literature to deal with this problem. For a comprehensive review of models for phase-change problems with convection, see Kowalewski and Gobin (2004). Note that a different category of models was recently suggested in the literature, based on the Lattice Boltzmann Method (Luo et al., 2015; Gong et al., 2015). Such methods based on non-deterministic models are not discussed in this introduction.

A first modelling approach, usually referred to as the multi-domain (or deforming-grid) method, is based on the classical Stefan two-phase model. Solid and liquid domains are separated and the corresponding conservation equations are solved in each domain. Boundary conditions at the interface between domains are obtained by imposing the Stefan condition (balance of heat fluxes at the interface). The position of the solid-liquid interface is tracked and moved explicitly using either *front tracking* or *front fixing* methods. The former method uses deforming grids to reconstruct the interface, while the latter is based on a time-depending coordinate transform, mapping the physical domain into a fixed computational domain. For a detailed description of these methods, see *e. g.* Sparrow et al. (1977); Unverdi and Tryggvason (1992); Gupta (2000); Tenchev et al. (2005). The main drawback of deforming-grid methods is their algorithmic complexity, which makes difficult to accurately capture solid-liquid interfaces of complicated shape or structure (*e. g.* with mushy regions between solid and liquid phases). Configurations with multiple interacting interfaces (see the solidification of a phase-change material presented in this paper) are also difficult to simulate with these methods (see also Stella and Giangi (2004)).

The second modelling approach avoids to impose explicitly the Stefan condition at the solid-liquid interface and therefore uses a single-domain (or fixed-grid) model. The same system of equations is solved in both liquid and solid phases. The energy balance at the interface is implicitly taken into account by the model. Consequently, the position of the interface is obtained a posteriori by post-processing the computed temperature field. Phase-field methods (Fabbri and Voller, 1997) and enthalpy methods (Voller and Prakash, 1987; Cao et al., 1989) are the most commonly used single-domain models. In phase-field methods, a supplementary partial differential equation for the evolution of the order parameter (a continuous variable taking the value 0 in the solid and 1 in the liquid) has to be solved, coupled with the conservation laws (Shyy et al., 1996). This new equation

is model dependent and its numerical solution could lead to diffuse solid-liquid interfaces. For recent contributions in this area, see Boettinger et al. (2002); Singer-Loginova and Singer (2008); Favier et al. (2019). We focus below on enthalpy methods, which are the most widely used single-domain models due to their algorithmic simplicity.

The main idea behind enthalpy models is to formulate the energy conservation law in terms of enthalpy and temperature and include latent heat effects in the definition of the enthalpy. The obtained equation applies to both liquid and solid phases and implicitly takes into account the separation of the phases. Another advantage of enthalpy methods, when compared to previously described models, is to remove the limitation of the phase-change occurring at a fixed temperature. The presence of mushy regions can be easily modelled with these methods. Two types of formulations of enthalpy methods exist in the literature, depending on the main variable used to solve the energy equation: enthalpy or temperature-based methods. In enthalpy-based formulations the main variable is the enthalpy (Eyres et al., 1946; Rose, 1960; Bhattacharya et al., 2014). Temperature is computed from the temperature-enthalpy coupling model. An iterative loop is necessary to solve the energy equation, formulated with both enthalpy and temperature variables. For a review of different iterative techniques to solve the energy equation, see Voller (1996). A second variety of enthalpy-based formulations consists in rewriting the energy equation with enthalpy terms only (Rady and Mohanty, 1996; Hannoun et al., 2003). In temperature-based formulations, the energy equation is formulated in terms of temperature only. The latent heat is treated either by deriving an apparent heat capacity coefficient to define the total enthalpy (Morgan et al., 1978; Chiesa and Guthrie, 1974; Gau and Viskanta, 1984) or by introducing a source term in the energy equation (Voller, 1996; Swaminathan and Voller, 1997). Advantages and drawbacks of each approach are discussed in detail in König-Haagen et al. (2017).

Single-domain methods are very appealing for numerical implementations. The same system of equations is solved in the entire computational domain, making possible algorithmic or computer-architecture optimisations. If enthalpy models offer an elegant solution to deal with the same energy conservation equation in both phases, a last modelling problem has to be solved. It concerns the extension of the Navier-Stokes-Boussinesq equations from the liquid to the solid phase. Different techniques to bring the velocity to zero in the solid region were suggested. The most straightforward is the switch-off technique, which decouples solid and liquid computational points and overwrites the value of the velocity by setting it to zero in the solid region. Different implementations of this technique with finite-volume methods are presented in Ma and Zhang (2006); Wang et al. (2010a). In variable viscosity techniques (Gartling, 1980; Voller et al., 1987; Cao and Faghri, 1990), the fluid viscosity depends on the temperature and is artificially increased to huge values in the solid regions through a regularisation or mushy zone. To avoid blow-up or numerical inconsistencies, the large gradients of viscosity must be correctly resolved in the mushy region. This is naturally achieved in finite-element methods with dynamical mesh adaptivity (Danaila et al., 2014), while in finite-volume methods with fixed grids, the time step has to be adapted to the space resolution (Ma and Zhang, 2006). Versions of the variable viscosity approach suggested in Danaila et al. (2014) were further studied by Aldbaissy et al. (2018); Woodfield et al. (2019) and implemented in a different finite-element framework (FEniCS) by Zimmerman and Kowalski (2018).

A third technique used to ensure a zero velocity field in the solid phase is the so-called enthalpy-porosity model (Brent et al., 1988). A penalisation source term is introduced in

the momentum equation to allow the switch from the full Navier-Stokes equations in the liquid phase to a Darcy equation for porous media. The mushy region is thus regarded as a very dense porous medium that sharply brings the velocity to zero in the solid region. The expression of the penalisation source term generally follows the Carman-Kozeny model for the permeability of a porous medium (Hannoun et al., 2003, 2005; Belhamadia et al., 2012), but other mathematically equivalent expressions were suggested (Angot et al., 1999; Favier et al., 2019). Different formulations and implementations of the enthalpy-porosity model are presented in Kowalewski and Rebow (1999); Giangi et al. (2000); Stella and Giangi (2004).

Concerning the space discretization of these models, finite difference (FD) or finite volume (FV) methods are generally used in the literature. When single-mesh models are used, the general strategy to capture the solid-liquid interface 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). Different FE approaches were suggested, from enthalpy-type methods (*e. g.* Elliott (1987)) to front-tracking methods (*e. g.* Li (1983)). Recently, adaptive FE methods were proposed for classical two-phase Stefan problem (Belhamadia et al., 2004) and phase-change systems with convection (Belhamadia et al., 2012; Danaila et al., 2014).

However, to the best of our knowledge, no adaptive finite-element programs exist in the CPC Program Library for phase-change problems. The purpose of this paper is therefore to distribute a finite-element toolbox to solve two-dimensional solid-liquid phase-change problems.

The present toolbox is based on a single-domain enthalpy-porosity model for solid-liquid phase change problems with convection. For the energy conservation equation, a temperature-based formulation takes into account the latent heat by introducing a discontinuous source term. For the mass and momentum conservation equations, we solve in the entire domain the incompressible Navier-Stokes equations with Boussinesq approximation for buoyancy effects. To bring the velocity to zero in the solid phase, we introduce in the momentum equation a penalty term following the Carman-Kozeny model. The coupled system of momentum and energy equations is integrated in time using a second-order Gear scheme. All the terms are treated implicitly and the resulting discretized equations are solved using a Newton method (Danaila et al., 2014). The advantage of this formulation is to permit a straightforward implementation of different types of non-linearities. For the space discretization we use Taylor-Hood triangular finite elements, *i. e.* P_2 for the velocity and temperature and P_1 for the pressure. Temperature is discretized using P_2 or P_1 finite elements. Discontinuous variables (latent heat, thermal diffusivity, etc) at the solid-liquid interface are regularized through an intermediate artificial mushy region. Single domain methods require a refined mesh near the interface, where large enthalpy gradients have to be correctly resolved. An optimized dynamical mesh adaptivity algorithm allows us to adapt the mesh every time step and thus accurately capture the evolution of the interface. Mesh adaptivity feature of the toolbox offers the possibility to deal with complicated phase-change cases, involving multiple solid-liquid interfaces. There are two main novelties in the present numerical approach, when compared to Danaila et al. (2014): *(i)* we use the Carman-Kozeny model to bring the velocity to zero inside the solid phase, instead of a viscosity penalty method (imposing a large value of the

viscosity in the solid); *(ii)* we increase the time accuracy of the algorithm by replacing the first-order Euler scheme with the second-order Gear (BDF2) scheme (see also Belhamadia et al. (2012)).

The programs were built and organized as a toolbox for FreeFem++ (Hecht, 2012; Hecht et al., 2007), which is a free software (under LGPL license). FreeFem++ [1]offers a large variety of triangular finite elements (linear and quadratic Lagrangian elements, discontinuous P_1, Raviart-Thomas elements, etc.) to solve partial differential equations. It is an integrated product with its own high level programming language and a syntax close to mathematical formulations, making the implementation of numerical algorithms very easy. Among the features making FreeFem++ an easy-to-use and highly adaptive software we recall the advanced automatic mesh generator, mesh adaptation, problem description by its variational formulation, automatic interpolation of data, colour display on line, postscript printouts, etc. The FreeFem++ programming framework offers the advantage to hide all technical issues related to the implementation of the finite element method. It becomes then easy to use the present toolbox to code new numerical algorithms for similar problems with phase-change.

The paper is organized as follows. Section 2 introduces the governing equations and the regularization technique used in the enthaphy-porosity model. Section 3 presents the Newton algorithm for the Navier-Stokes-Boussinesq system of equations. The mesh adaptivity technique is also discussed in this section. A description of the programs is given in Section 4. The accuracy of the numerical method is tested in Section 5 using manufactured solutions. Finally, Section 6 is devoted to extensive numerical validations of the method. The robustness of the algorithm is demonstrated by comparing our results with well defined classical benchmarks. The capabilities of the toolbox to deal with complex geometries is also illustrated. The main features of the software and possible extensions are summarised in Section 7.

2. Enthalpy-porosity model

We consider a solid-liquid system placed in a two-dimensional domain Ω. In the following, subscripts s and l will refer to the solid and the liquid phases, respectively. We present in this section the single-domain model, using the same system of equations in both liquid and solid phases.

2.1. Energy equation

The phase change process is modelled using an enthalpy method with a temperature-based formulation (Voller et al., 1987; Cao et al., 1989; Cao and Faghri, 1990). We start with the classical energy equation:

$$\frac{\partial(\rho h)}{\partial t_\varphi} + \nabla \cdot (\rho h \boldsymbol{U}) - \nabla \cdot (k \nabla T) = 0, \tag{1}$$

where, t_φ is the physical time, h the enthalpy, ρ the density, $\boldsymbol{U}$ the velocity vector, T the temperature and k the thermal conductivity. The total enthalpy h is transformed as the

[1]FreeFem++ for different OS can be downloaded from `http://www3.freefem.org/`.

sum of the sensible heat and the latent heat:

$$h = h_{sen} + h_{lat} = c(T + s(T)), \tag{2}$$

with c the local specific heat. The function s is introduced to model the jump of the enthalpy during the solid-liquid transition. For pure materials, s is theoretically a Heaviside step function depending on the temperature: it takes the zero value in the solid region and a large value in the liquid, equal to h_{sl}/c_l, with h_{sl} the latent heat of fusion. If the phase-change is assumed to occur within a mushy zone defined by a small temperature interval $T \in [T_f - T_\varepsilon, T_f + T_\varepsilon]$ around the temperature of fusion T_f, a model for $s(T)$ is necessary. Linear (Voller et al., 1987; Wang et al., 2010a) or more smooth functions (Danaila et al., 2014) can be used to regularize $s(T)$ and thus model the jump of material properties from solid to liquid. In this paper we use a regularization of all step-functions (latent heat source, specific heat, thermal diffusivity or conductivity) by a continuous and differentiable hyperbolic-tangent function suggested by Danaila et al. (2014). We assume moreover that the undercooling phenomenon is negligible during the solidification stage (see also Wang et al. (2010b); Kowalewski and Gobin (2004)).

Equation (1) can be further simplified under the following assumptions: (i) the density difference between solid and liquid phases is negligible, *i. e.* $\rho_l = \rho_s = \rho$ is constant; (ii) the mushy region is narrow and the velocity inside this zone is zero. Consequently, the final temperature transforming model is obtained by combining (2) and (1) and neglecting the convection term $\nabla \cdot (cs\boldsymbol{U})$[2]:

$$\frac{\partial (cT)}{\partial t_\varphi} + \nabla \cdot (cT\boldsymbol{U}) - \nabla \cdot \left(\frac{k}{\rho} \nabla T \right) + \frac{\partial (cs)}{\partial t_\varphi} = 0. \tag{3}$$

2.2. Navier-Stokes equations with Boussinesq approximation

The natural convection in the liquid part of the system is modelled using the incompressible Navier-Stokes equations, with Boussinesq approximation for buoyancy effects. To make this model valid in the entire domain, the momentum equation is modified as following:

$$\frac{\partial \boldsymbol{U}}{\partial t_\varphi} + (\boldsymbol{U} \cdot \nabla)\boldsymbol{U} + \frac{1}{\rho}\nabla p - \frac{\mu_l}{\rho}\nabla^2 \boldsymbol{U} - f_B(T)\boldsymbol{e}_y = A(T)\boldsymbol{U}, \tag{4}$$

where p denotes the pressure, μ_l the dynamic viscosity of the liquid (assumed to be constant) and $f_B(T)$ the Boussinesq force. The Carman-Kozeny penalty term $A(T)\boldsymbol{u}$ is artificially introduced in (4) to extend this equation into the solid phase, where the velocity, pressure, viscosity and Boussinesq force are meaningless. Consequently, $A(T)$ is modelled to vanish in the liquid, where the Navier-Stokes-Boussinesq momentum equation is recovered. A large value of $A(T)$ is imposed in the solid, reducing the momentum equation (4) to $A(T)\boldsymbol{u} = 0$, equivalent to $\boldsymbol{u} = 0$. Exact expressions for f_B and A will be given in the next section.

Finally, the conservation of mass in the liquid phase is expressed by the continuity equation:

$$\nabla \cdot \boldsymbol{U} = 0. \tag{5}$$

[2]In the liquid phase, $\nabla \cdot (cs\boldsymbol{U}) = h_{sl}\nabla \cdot \boldsymbol{U} = 0$; in the solid phase, $s = 0$; in the mushy region, $\boldsymbol{U} = 0$.

2.3. Final system of equations for the single-domain approach

The present numerical code solves a dimensionless form of the previous equations. After choosing a reference length H (usually the height of the cavity when a rectangular domain is considered) and a reference state (ρ, V_{ref}, T_{ref}), we can define the following scaling for the space, velocity, temperature and time variables:

$$\boldsymbol{x} = \frac{\boldsymbol{X}}{H}\,,\ \boldsymbol{u} = \frac{\boldsymbol{U}}{V_{ref}}\,,\ \theta = \frac{T - T_{ref}}{\delta T}\,,\ t = \frac{V_{ref}}{H} t_{\varphi}. \tag{6}$$

T_{ref} is the reference temperature and in most cases $T_{ref} = T_f$ (the temperature of fusion), unless otherwise specified. Consequently, the non-dimensional temperature of fusion is set to $\theta_f = 0$. Temperature difference δT defines a temperature scale, that will be set differently for melting and solidification cases.

The dimensionless system of equations to be solved in both liquid and solid regions can be finally written as:

$$\nabla \cdot \boldsymbol{u} = 0, \tag{7}$$

$$\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} + \nabla p - \frac{1}{Re}\nabla^2 \boldsymbol{u} - f_B(\theta)\, \boldsymbol{e}_y - A(\theta)\boldsymbol{u} = 0, \tag{8}$$

$$\frac{\partial (C\theta)}{\partial t} + \nabla \cdot (C\theta \boldsymbol{u}) - \nabla \cdot \left(\frac{K}{RePr} \nabla \theta \right) + \frac{\partial (CS)}{\partial t} = 0. \tag{9}$$

The linearised Boussinesq buoyancy force (f_B), the Reynolds (Re) and Prandtl (Pr) numbers are defined as:

$$f_B(\theta) = \frac{Ra}{PrRe^2}\theta, \quad Re = \frac{\rho V_{ref} H}{\mu_l} = \frac{V_{ref} H}{\nu_l}, \quad Pr = \frac{\nu_l}{\alpha_l}, \tag{10}$$

with ν the kinematic viscosity and $\alpha = k/(\rho c)$ the thermal diffusivity. In the expression of f_B, the Rayleigh number of the flow is defined as:

$$Ra = \frac{g\beta H^3 \delta T}{\nu_l \alpha_l}, \tag{11}$$

with β the thermal expansion coefficient and g the gravitational acceleration.

If previous non-dimensional numbers are pertinent only in the liquid phase, the non-dimensional conductivity and specific heat are defined in both phases

$$K(\theta) = \frac{k}{k_l} = \begin{cases} 1, & \theta \geq \theta_f, \\ k_s/k_l, & \theta < \theta_f. \end{cases},\ C(\theta) = \frac{c}{c_l} = \begin{cases} 1, & \theta \geq \theta_f, \\ c_s/c_l, & \theta < \theta_f. \end{cases} \tag{12}$$

The non-dimensional function $S = s/s_l$ in the energy equation (9) takes a similar non-dimensional form:

$$S(\theta) = \frac{s}{s_l} = \begin{cases} \dfrac{h_{sl}/c_l}{\delta T} = \dfrac{1}{Ste}, & \theta \geq \theta_f, \\ 0, & \theta < \theta_f, \end{cases} \tag{13}$$

with Ste the Stefan number.

Discontinuous step-functions defined in (12) and (13) are replaced by continuous and differentiable hyperbolic-tangent functions, generically defined for all θ by the formula (Danaila et al., 2014):

$$F(\theta; a_s, \theta_s, R_s) = f_l + \frac{f_s - f_l}{2}\left\{1 + \tanh\left(a_s\left(\frac{\theta_s - \theta}{R_s}\right)\right)\right\}, \tag{14}$$

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. For example, we use for the non-dimensional source term in (9) the following regularisation over the artificial mushy region $\theta \in [-\varepsilon, \varepsilon]$:

$$S(\theta) = \frac{1}{Ste} - \frac{1}{2\,Ste}\left\{1 + \tanh\left(\frac{\theta_r - \theta}{R_s}\right)\right\}, \tag{15}$$

where θ_r is the central value around which we regularize (typically $\theta_r = \theta_f = 0$) and R_r the smoothing radius (typically $R_s = \varepsilon$).

Finally, the penalty term $A(\theta)\boldsymbol{u}$ in momentum equation (8) follows from the Carman-Kozeny model (Voller et al., 1987; Belhamadia et al., 2012; Kheirabadi and Groulx, 2015):

$$A(\theta) = -\frac{C_{\text{CK}}(1 - L_f(\theta))^2}{L_f(\theta)^3 + b}, \tag{16}$$

where $L_f(\theta)$ is the local liquid fraction, which is 1 in the fluid region and 0 in the solid. L_f is regularized inside the artificial mushy-region using a hyperbolic-tangent similar to (15). The Carman-Kozeny constant C_{CK} is set to a large value (as discussed below) and the constant $b = 10^{-6}$ is introduced to avoid divisions by zero.

3. Numerical method

To solve the system of equations (7)-(9) we use a finite-element method that was implemented using the open-source software FreeFem++ (Hecht et al., 2007; Hecht, 2012). Among the numerous numerical tools offered by FreeFem++ , the use of the powerful mesh adaptivity function proved mandatory in this study to obtain accurate results within reasonable computational time. The numerical code was optimized to afford the mesh refinement every time step: the mesh density was increased around the artificial mushy region, offering an optimal resolution of the large gradients of all regularized functions (S, K, L_f), while the mesh was de-refined (larger triangles) in the solid part, where a coarser mesh could be used. A simulation using a globally refined mesh would require a prohibitive computational time for an equivalent accuracy of the melting front resolution. Similar algorithms based on FreeFem++ were successfully used for solving different systems of equations with locally sharp variation of the solution, such as Gross-Pitaevskii equation (Danaila and Hecht, 2010; Vergez et al., 2016) or Laplace equations with nonlinear source terms (Zhang and Danaila, 2013).

The space discretization is based on Taylor-Hood finite elements, approximating the velocity with P_2 Lagrange finite elements (piecewise quadratic), and the pressure with P_1 finite elements (piecewise linear). The temperature and the enthalpy are discretized

using either P_1 or P_2 finite elements. For the time integration, we use a second-order Gear (BDF2) scheme (see also Belhamadia et al. (2012)):

$$\frac{d\phi}{dt} \simeq \frac{3\phi^{n+1} - 4\phi^n + \phi^{n-1}}{2\delta t}, \tag{17}$$

computing the solution ϕ^{n+1} at time $t_{n+1} = (n+1)\delta t$ by using two previous states (ϕ^n, ϕ^{n-1}). We use this scheme to advance in time both velocity ($\phi = \boldsymbol{u}$) and temperature fields ($\phi = \theta$). The other terms in equations (7)-(9) are treated implicitly (*i. e.* taken at time t_{n+1}). The resulting non-linear equations are solved using a Newton algorithm.

3.1. Finite element algorithm

Finite-element methods for solving Navier-Stokes type systems of equations 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 (Temam, 1983; Girault and Raviart, 1986; Quarteroni and Valli, 1994). To simplify the presentation, we consider in the following that $C = 1$ and $K = 1$. For phase-change materials like paraffins this is a physically valid assumption. Nevertheless, we keep in the equations below the variable K, since for the water freezing case it is necessary to take into account the dependence $K(\theta)$. The derivation of supplementary terms associated to $K(\theta)$ is straightforward.

We use the second-order finite-difference scheme (17) and obtain the following implicit semi-discretization in time of the single-domain model (7)-(9):

$$\nabla \cdot \boldsymbol{u}^{n+1} + \gamma p^{n+1} = 0, \tag{18}$$

$$\frac{3}{2}\frac{\boldsymbol{u}^{n+1}}{\delta t} + (\boldsymbol{u}^{n+1} \cdot \nabla)\boldsymbol{u}^{n+1} + \nabla p^{n+1} - \frac{1}{Re}\nabla^2 \boldsymbol{u}^{n+1} - A(\theta^{n+1})\boldsymbol{u}^{n+1} - f_B(\theta^{n+1})\,\boldsymbol{e}_y = 2\frac{\boldsymbol{u}^n}{\delta t} - \frac{\boldsymbol{u}^{n-1}}{2\delta t}, \tag{19}$$

$$\frac{3}{2}\frac{\theta^{n+1} + S(\theta^{n+1})}{\delta t} + \nabla \cdot \left(\boldsymbol{u}^{n+1}\theta^{n+1}\right) - \nabla \cdot \left(\frac{K}{RePr}\nabla\theta^{n+1}\right) = 2\frac{\theta^n + S(\theta^n)}{\delta t} - \frac{\theta^{n-1} + S(\theta^{n-1})}{2\delta t}. \tag{20}$$

This system of non-linear equations is solved at time $t_{n+1} = (n+1)\delta t$, using two previous states: t_n and t_{n-1}. We denote by $f_B(\theta)$ the Boussinesq force that can be non-linear in the general case (*e. g.* natural convection or freezing of water). The penalty parameter γ takes very low values ($\gamma = 10^{-7}$) to ensure a pressure field with zero average and, at the algebraic level, fulfil the diagonal of the pressure term.

The space discretization of variables over the domain $\Omega \subset \mathbb{R}^2$ uses a finite-element method based on a weak formulation of the system (18)-(20). We consider homogeneous Dirichlet boundary conditions for the velocity, *i. e.* $\boldsymbol{u} = 0$ on $\partial\Omega$, and set the classical Hilbert spaces for the velocity and pressure:

$$\boldsymbol{V} = V \times V,\ V = H_0^1(\Omega), \quad Q = \left\{ q \in L^2(\Omega) \,\middle|\, \int_\Omega q = 0 \right\} \tag{21}$$

Following the generalization of the Stokes problem (Temam, 1983; Girault and Raviart, 1986; Quarteroni and Valli, 1994), the variational formulation of the system (18)-(20) can be written as: find $(\boldsymbol{u}^{n+1}, p^{n+1}, \theta^{n+1}) \in \boldsymbol{V} \times Q \times V$, such that:

$$
\begin{aligned}
b\left(\boldsymbol{u}^{n+1}, q\right) - \gamma(p^{n+1}, q) = 0, \quad \forall\, q \in Q \quad (22)\\
\frac{3}{2\delta t}\left(\boldsymbol{u}^{n+1}, \boldsymbol{v}\right) + c\left(\boldsymbol{u}^{n+1}; \boldsymbol{u}^{n+1}, \boldsymbol{v}\right) + \frac{1}{Re} a\left(\boldsymbol{u}^{n+1}, \boldsymbol{v}\right)\\
-(A(\theta^{n+1})\,\boldsymbol{u}^{n+1}, \boldsymbol{v}) + b\left(\boldsymbol{v}, p^{n+1}\right) - \left(f_B(\theta^{n+1})\,\boldsymbol{e}_y, \boldsymbol{v}\right)\\
= \frac{2}{\delta t}\left(\boldsymbol{u}^{n}, \boldsymbol{v}\right) - \frac{1}{2\delta t}\left(\boldsymbol{u}^{n-1}, \boldsymbol{v}\right), \quad \forall\, \boldsymbol{v} \in \boldsymbol{V}\\
\frac{3}{2\delta t}\left(\theta^{n+1} + S(\theta^{n+1}), \phi\right) + \left(\boldsymbol{u}^{n+1} \cdot \nabla\theta^{n+1}, \phi\right) + \left(\frac{K}{RePr}\nabla\theta^{n+1}, \nabla\phi\right)\\
= \frac{2}{\delta t}\left(\theta^{n} + S(\theta^{n}), \phi\right) - \frac{1}{2\delta t}\left(\theta^{n-1} + S(\theta^{n-1}), \phi\right), \quad \forall\, \phi \in V,
\end{aligned}
$$

where $(u, v) = \int_\Omega u \cdot v$ denotes the scalar product in $L^2(\Omega)$ or $\left(L^2(\Omega)\right)^2$; the bilinear forms a, b and trilinear form c are defined as (Girault and Raviart, 1986; Quarteroni and Valli, 1994):

$$
\begin{aligned}
a : \boldsymbol{V} \times \boldsymbol{V} \to \mathbb{R}, \qquad & a(\boldsymbol{u}, \boldsymbol{v}) = \int_\Omega \boldsymbol{\nabla}^{\boldsymbol{t}}\boldsymbol{u} : \boldsymbol{\nabla}\boldsymbol{v} = \sum_{i,j=1}^{2} \int_\Omega \partial_j u_j \cdot \partial_j v_i,\\
b : \boldsymbol{V} \times Q \to \mathbb{R}, \qquad & b(\boldsymbol{u}, q) = -\int_\Omega \nabla \cdot \boldsymbol{u}\, q = -\sum_{i=1}^{2} \int_\Omega \partial_i u_i \cdot q,\\
c : \boldsymbol{V} \times \boldsymbol{V} \times \boldsymbol{V} \to \mathbb{R}, \qquad & c(\boldsymbol{w}; \boldsymbol{z}, \boldsymbol{v}) = \int_\Omega \left[(\boldsymbol{w} \cdot \nabla)\,\boldsymbol{z}\right] \cdot \boldsymbol{v} = \sum_{i,j=1}^{2} \int_\Omega w_j (\partial_j z_i) v_i.
\end{aligned}
$$

The system of non-linear equations (22) is solved using a Newton method. To advance the solution from time t_n to t_{n+1}, we start from an initial guess $w_0 = (\boldsymbol{u}^n, p^n, \theta^n)$ (which is the solution at t_n), and construct the Newton sequence $w_k = (u_k, p_k, \theta_k)$ by solving for each inner iteration k:

$$
\begin{aligned}
b\left(\boldsymbol{u}_{k+1}, q\right) - \gamma(p_{k+1}, q) &= 0, \quad (23)\\
\frac{3}{2\delta t}\left(\boldsymbol{u}_{k+1}, \boldsymbol{v}\right) + c\left(\boldsymbol{u}_{k+1}; \boldsymbol{u}_k, \boldsymbol{v}\right) &+ c\left(\boldsymbol{u}_k; \boldsymbol{u}_{k+1}, \boldsymbol{v}\right)\\
+\frac{1}{Re} a\left(\boldsymbol{u}_{k+1}, \boldsymbol{v}\right) - \left(\frac{dA}{d\theta}(\theta_k)\,\theta_{k+1}\,\boldsymbol{u}_k, \boldsymbol{v}\right) &- (A(\theta_k)\,\boldsymbol{u}_{k+1}, \boldsymbol{v}) + b\left(\boldsymbol{v}, p_{k+1}\right)\\
-\left(\frac{df_B}{d\theta}(\theta_k)\,\theta_{k+1}\,\boldsymbol{e}_y, \boldsymbol{v}\right) &= \frac{1}{\delta t}\left(2\boldsymbol{u}^n - \frac{1}{2}\boldsymbol{u}^{n-1}, \boldsymbol{v}\right)\\
+c\left(\boldsymbol{u}_k; \boldsymbol{u}_k, \boldsymbol{v}\right) - \left(\frac{dA}{d\theta}(\theta_k)\,\theta_k\,\boldsymbol{u}_k, \boldsymbol{v}\right) &- \left(\left(\frac{df_B}{d\theta}(\theta_k)\,\theta_k - f_B(\theta_k)\right)\boldsymbol{e}_y, \boldsymbol{v}\right), \quad (24)\\
\frac{3}{2\delta t}\left(\theta_{k+1} + \frac{dS}{d\theta}(\theta_k)\,\theta_{k+1}, \phi\right) + \left(\boldsymbol{u}_k \cdot \nabla\theta_{k+1}, \phi\right) &+ \left(\boldsymbol{u}_{k+1} \cdot \nabla\theta_k, \phi\right)\\
+\left(\frac{K}{RePr}\nabla\theta_{k+1}, \nabla\phi\right) &= \frac{2}{\delta t}\left(\theta^n + S(\theta^n), \phi\right)\\
+\left(\boldsymbol{u}_k \cdot \nabla\theta_k, \phi\right) + \frac{3}{2\delta t}\left(\frac{dS}{d\theta}(\theta_k)\,\theta_k - S(\theta_k),\, \phi\right) &- \frac{1}{2\delta t}\left(\theta^{n-1} + S(\theta^{n-1}),\, \phi\right). \quad (25)
\end{aligned}
$$

Note that the last term of Eq. (24) cancels in the case of a linear Boussinesq force f_B (see Eq. (10)); this is not the case when non-linear variations of the density of the liquid are considered (convection or solidification of water). Note also that the previous system of equations (23)-(25) depends only on $\boldsymbol{u}^n$, $\boldsymbol{u}^{n-1}$, θ^n and θ^{n-1} and is independent of p^n, the pressure being in this approach a Lagrange multiplier for the divergence free constraint.

The Newton loop (following k) has to be iterated until convergence for each time step δt following the algorithm:

$$\begin{array}{l} \text{Navier-Stokes time loop following } n \\ \text{set } w_0 = (\boldsymbol{u}^n, p^n, \theta^n) \\ \quad \text{Newton iterations following } k \\ \quad\quad \text{solve (23) to get } \boldsymbol{w}_{k+1} \\ \quad \text{stop when } \|\boldsymbol{w}_{k+1} - \boldsymbol{w}_k\| < \xi_N \\ \text{actualize } (\boldsymbol{u}^{n+1}, p^{n+1}, \theta^{n+1}) = w_{k+1} \end{array} \tag{26}$$

The FreeFem++ syntax to implement the Newton algorithm is very close to the mathematical formulation given above. After defining a vectorial finite-element space `fespace Wh(Th,[P2,P2,P1,P1]);`, associated to the mesh `Th`, we define the velocity, pressure and temperature variables in a compact manner by `Wh [u1,u2,p,T];`. Corresponding test functions are defined similarly. It is then very easy to define a `problem` formulation in FreeFem++ and include all the terms of the algorithm (23)-(25). This makes the lecture of the programs very intuitive by comparing each term to its mathematical expression. New terms could be added to the variational formulation expressed in the `problem` structure, without affecting other parts of the program. Consequently, the implementation of new models or numerical methods for this problem is greatly facilitated by this modular structure of programs.

3.2. Mesh adaptivity

We use the standard mesh adaptivity function (`adaptmesh`) offered by FreeFem++ (Hecht, 2012). The key idea implemented in this function (see also Borouchaki et al. (1996); Castro-Diaz et al. (2000); Hecht and Mohammadi (1997); George and Borouchaki (1998); Frey and George (1999); Mohammadi and Pironneau (2000)) is to modify the scalar product used in the automatic mesh generator to evaluate distance and volume. Equilateral elements are thus constructed, accordingly to the new metric. The scalar product is based on the evaluation of the Hessian $\mathcal{H}$ of the variables of the problem. For example, for a P_1 discretization of a variable χ, the interpolation error is bounded by:

$$\mathcal{E} = |\chi - \Pi_h \chi|_0 \leq c \sup_{T \in \mathcal{T}_h} \sup_{x,y,z \in T} |\mathcal{H}(x)|(y-z, y-z), \tag{27}$$

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. Using the Delaunay algorithm (*e. g.* George and Borouchaki (1998)) to generate a trianguler mesh with edges close to the unit length in the metric $\mathcal{M} = \frac{|\mathcal{H}|}{(c\mathcal{E})}$ will result in a equally distributed interpolation error $\mathcal{E}$ over the edges a_i of the mesh. More precisely, we get

$$\frac{1}{c\mathcal{E}} a_i^T \mathcal{M} a_i \leq 1. \tag{28}$$

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 Frey and George (1999)).

The `adaptmesh` function offers the possibility to take into account several metrics computed from different variables monitoring the evolution of the phase-change system. For natural convection system, the mesh will be adapted using the values of the two velocity components and the temperature. For phase-change systems, to accurately track the solid-liquid interface we add the variation of the enthalpy source term in the adaptivity criterion. For water systems (convection or freezing), we also add an extra function tracking the anomalous change of density around 4^oC. To reduce the impact of the interpolation on the global accuracy for time-depending problems, we consider, for each variable used for adaptivity, the metrics computed at actual (t_{n+1}) and previous (t_n) time instants (see also Belhamadia et al. (2004)). 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 capabilities of the mesh adaptivity algorithm are illustrated in §6.

4. Description of the programs

In this section, we first describe the architecture of the programs and the organisation of provided files. Then we focus on the list of input parameters and the structure of output files.

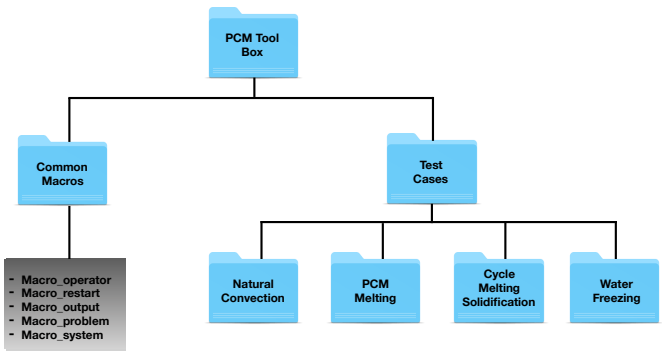

Figure 1: Folder tree structure of the PCM-Toolbox to solve solid-liquid phase-change problems.

4.1. Program architecture

Figure 1 gives a schematic overview of the content of the toolbox. All files are provided in a directory called `PCM-Toolbox`. Many detailed comments are included in the programs, with direct link to the mathematical expressions used in the paper. The used FreeFem syntax was intentionally kept at a low level of technicality and supplemented with detailed comments when specific more technical syntax was used.

This directory is organized as follows:

1. The directory `Common-Macros` contains five files:
   - *Macro_operator.idp* includes macros and functions defining mathematical operators,
   - *Macro_problem.idp*: macros defining the variational formulation of the problem,
   - *Macro_restart.idp*: macros used to start a new simulation from a saved field,
   - *Macro_output.idp*: macros used to save the solution with different formats,
   - *Macro_system.idp*: macros identifying the OS and defining specific OS-commands.
2. The directory `Test-Cases` contains four subdirectories, each of them defining one of the following applications:
   - natural convection of air or water in a differentially heated square cavity,
   - melting of a PCM stored in containers of different shapes,
   - melting followed by solidification of a rectangular PCM,
   - freezing of pure water in a square cavity.

   Each subdirectory contains three files: *NEWTON_$case.edp* is the main FreeFem++ script file, *param_phys.inc* defines the physical parameters and *param_num.inc* the numerical parameters. For example, to run the natural convection case of air in a square cavity, the user can use the following command in a terminal window: `FreeFem++ NEWTON_stat_natconv.edp`.

   The folder structure of each test case is illustrated in Figure 2. The obtained solutions are saved in the folder `OUTPUT/Data`. Depending on the output format selected by the user, data files are generated in specific folders for being visualized with: Tecplot, Paraview, Gnuplot or Medit. We also provide in the folder `Figures` ready-made layouts for these visualisation softwares. The user can thus obtain the figures from this paper using newly generated data. More details about the output structure are given below.

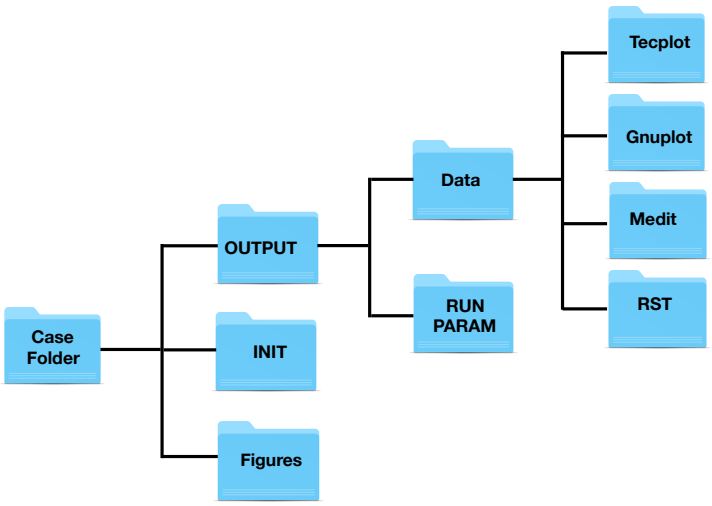

Figure 2: Structure of each Test-case folder.

4.2. Input parameters

Physical parameters and parameters related to the run are separated into two files.

(1) The file *param_phys.inc* contains the physical descriptions of the problem:

- **typeT**: is the finite-element type for the temperature, with possible values `P2` or `P1`,
- **Torder**: is the accuracy order of the time integration scheme, with possible values 1 (Euler scheme) or 2 (Gear scheme),
- **scalAdim**: defines the characteristic scales of the problem, see (6). Possible values 1, 2 or 3 correspond to the following choice of the characteristic scales (Danaila et al., 2014):

$$(1) \quad : \quad V_{ref}^{(1)} = \frac{\nu_l}{H} \Longrightarrow t_{ref}^{(1)} = \frac{H^2}{\nu_l} \Longrightarrow \mathcal{R}e = 1, \tag{29}$$

$$(2) \quad : \quad V_{ref}^{(2)} = \frac{\alpha}{H} \Longrightarrow t_{ref}^{(2)} = t_{ref}^{(1)}\mathcal{P}r \Longrightarrow \mathcal{R}e = 1/\mathcal{P}r, \tag{30}$$

$$(3) \quad : \quad V_{ref}^{(3)} = \frac{\nu_l}{H}\sqrt{\frac{\mathcal{R}a}{\mathcal{P}r}} \Longrightarrow t_{ref}^{(3)} = t_{ref}^{(1)}\sqrt{\frac{\mathcal{P}r}{\mathcal{R}a}} \Longrightarrow \mathcal{R}e = \sqrt{\frac{\mathcal{R}a}{\mathcal{P}r}}, \tag{31}$$

- $\mathbf{x}_l$, $\mathbf{x}_r$, $\mathbf{y}_l$, $\mathbf{y}_r$: are the values defining the dimensions of the cavity $[x_l, x_r] \times [y_l, y_r]$,
- **Pr, Ra, Ste**: are the Prandtl, Rayleigh and Stefan numbers, see (11) and (10),
- $\mathbf{T}_{hot}$, $\mathbf{T}_{cold}$: are dimensionless temperatures according to (6),
- $\mathbf{bcu_1}$, $\mathbf{bcu_2}$, **bcT**: are macros defining the velocity (u) and the temperature (T) boundary conditions.
- **epsi**: is the half width ε of the mushy region. Default value = 0.01,
- **dt**: is the dimensionless time step,
- $\mathbf{t}_{max}$: is the dimensionless final time,
- **Parameters for regularization functions**:
  The parameters of the hyperbolic-tangent function (14) used to regularize discontinuous functions are set by default as follows:

	f_s	f_l	a_s	θ_s	R_s	C_{CK}	b
Enthalpy	0	1/Ste	1	0.01	0.01	-	-
Carman - Kozeny	0	1	1	0.01	0.01	10^6	10^{-7}
Conductivity (water)	1	2.26/0.578	1	θ_f	0.015	-	-

- **rho(T) and Drho(T)**: (water cases only) define the density and its derivative as functions of the temperature, following the model (Gebhart and Mollendorf, 1977):

$$\rho(T) = \rho_m(1 - \omega|T - T_m|^q),$$

ρ_m [kg/m^3]	ω [^{o}C^{-q}]	q	T_m [^{o}C]
999.972	$9.2793 \cdot 10^{-6}$	1.894816	4.0293

- $\mathbf{f_B(T)}$, $\mathbf{df_B(T)}$: define the buoyancy force and its derivative.

(2) The file *param_num.inc* contains the parameters controlling the run.
Restart parameters:

- **Nsave**: the solution is saved every *Nsave* time steps in the `Data` folder (see Figure 2). The temperature and the velocity fields are saved in `Tecplot` and `Medit` folders, while the liquid fraction, the Nusselt number, and the accumulated heat input are saved in the `Gnuplot` folder.
- **Nrestart**: restart files (mesh and solution) are saved every *Nrestart* time steps. Solutions at current and previous iterations, the CPU time, the accumulated heat input Q_0, and the time step dt are saved in the folder `RST`.
- **Ncondt**: allows the user to stop the run and save the solution properly. The file `OUTPUT/zz.condt` is read every *Ncondt* time steps: if the user replaces the value "0" in this file by "1" the run is stopped. This is a simple solution for a clean stop of the job by the user. Default value = 20.
- **Nremesh**: the mesh is adapted every *Nremesh* iterations. If this parameter is set to "1" the mesh is adapted every time step.
- **IFrestart**: is a boolean controlling the set up of the initial field.
  $IFrestart = 0$, the initial condition is built in the code for each test case. For the PCM melting cases, the PCM is initially motionless at isothermal temperature. To set-up a smooth initial field, a few time steps (with very small δt) are computed by increasing progressively the boundary temperature at the hot wall and the Rayleigh number (by continuation). Outputs are saved in `OUTPUT/Data-RST-0`.
  $IFrestart > 0$, (positive integer values) the solution field previously computed at iteration $IFrestart$ is loaded from the folder `OUTPUT/Data-RST-filenameRST/RST`, with `filenameRST` a variable selecting the restart folder.
  $IFrestart < 0$, (negative integer values), the same principle for loading a solution is used, but from the folder `INIT` (see Figure 2). The solution fields stored in this folder could come from different previous calculations (*e. g.* a steady state solution or, for the water, the natural convection field before freezing).

Newton parameters:

- **epsconv**: is the value of the stopping criterion for steady cases,
- **gamma**: is the penalty parameter in (18). Default value = 10^{-7},
- **tolNewton**: is the Newton tolerance ξ_N (see (26)). Default value = 10^{-6},
- **newtonMax**: limits the maximum number of iterations in the Newton algorithm (26). Default value = 50,

Mesh parameters:

- **nbseg**: is the number of segments for the discretisation along the x and y directions,
- **errh**: is the interpolation error level. Default value = 0.02,
- **hmin, hmax**: are the minimum and maximum edge size, respectively,
- **adaptratio**: is the ratio for a prescribed smoothing of the metric. For a value less than 1.1 no smoothing is done. Default value = 1.5,
- **nbvx**: is the maximum number of vertices allowed in the mesh generator. Default value = 50000.

Output parameters:

- **dircase**: is the name of the output folder,
- **fcase**: is the prefix-name for ouput files.
- **Tecplot, Medit, Gnu**: correspond to the name of the visualisation software to be used; the format of the outputs written in `OUTPUT/Data` (see Figure 2) is accordingly set. The files from the Tecplot folder can be easily read also with Paraview.

4.3. Outputs

When a computation starts, the `OUTPUT` directory is created (see Figure 2)). It contains two folders storing the output data and the echo of the run parameters. The folder `Data` contains four subdirectories with different output format files of the computed solution. File names are created using the prefix defined by the parameter **fcase**, the current iteration and the current dimensionless time t. Solution files can be visualized using either Tecplot or any other CFD Visualization tools (Paraview, Visit, etc.). Moreover, *.gmsh* (mesh) and *.rst* (fields) files are generated in the folder `RST` to enable restarts of the computation. Note that the folder `FFglut` contains FreeFem++ scripts that re-read and visualize the RST-files to facilitate the selection of a restart field. An *.echo* file with a summary of the main parameters, informations on the run and the names of the output files is saved in the folder `RUNPARAM`. This directory additionally contains a copy of the *.inc* parameter files, allowing an easy identification of each case and preparing an eventual rerun of the same case.

5. Numerical tests of the accuracy of the numerical method

We start by presenting tests of the accuracy of our numerical method. We used the technique of manufactured solutions (*e. g.* Roache (1998)) which has the advantage of providing an exact solution to a modified problem, related to the initial one. The general idea is to modify the original system of equations by introducing an extra source term, such that the new system admits an exact solution given by a convenient analytic expression. Even though in most cases exact solutions constructed in this way are not physically realistic, this approach allows one to rigorously verify computations.
We tested the space and time accuracy using manufactured solutions for the system of equations (7)-(9) for a stationary case (Burggraf flow) and a time-dependent one

(Nourgaliev et al., 2016). For both cases, we computed the global error ε for different norms in space:

$$\varepsilon = \|\Phi_e - \phi_h\|, \tag{32}$$

with Φ_e the exact solution and ϕ_h the numerical solution. Computations were performed for the convection of air ($C = K = 1$, $A(\theta) = S(\theta) = 0$), with a Rayleigh number $Ra = 10^4$ and a Prandtl number $Pr = 0.71$.

5.1. Space accuracy: Burggraf stationary flow with thermal effects

The Burggraf manufactured solution is a time-independent recirculating flow inside a square cavity $[0,1] \times [0,1]$. It is similar to the well-known entrained cavity flow, with the difference that the velocity singularity at the top corners of the cavity is avoided. We added to the classical Burggraf flow (Shih et al., 1989; Laizet and Lamballais, 2009) a manufactured solution for the temperature, with constant temperature imposed at the top and the bottom walls. Vertical walls are assumed to be adiabatic. The exact solution of the new flow with thermal effects is:

$$\begin{aligned}
u_1(x,y) &= \sigma g'(x) h'(y), \\
u_2(x,y) &= -\sigma g''(x) h(y), \\
p(x,y) &= \frac{\sigma}{Re}\left(h^{(3)}(y) g(x) + g''(x) h'(y)\right) + \frac{\sigma^2}{2} g'(x)^2 \left(h(y) h''(y) - h'(y)^2\right), \\
T(x,y) &= T_c + (T_h - T_c) y + a(x) b(y),
\end{aligned} \tag{33}$$

with $\sigma > 0$ a scaling parameter and functions

$$\begin{aligned}
g(x) &= \frac{x^5}{5} - \frac{x^4}{2} + \frac{x^3}{3}, \\
h(y) &= y^4 - y^2, \\
a(x) &= \cos(\pi x), \\
b(x) &= y(1-y).
\end{aligned} \tag{34}$$

Note that the velocity at the top border of the cavity is:

$$u_1(0,1) = 2\sigma(x^4 - 2x^3 + x^2), \quad u_2(x,1) = 0, \tag{35}$$

which ensures the continuity of the velocity at the corners ($\boldsymbol{u}(0,1) = \boldsymbol{u}(1,1) = 0$), since non-slip walls are imposed for the other borders: $\boldsymbol{u}(x,0) = \boldsymbol{u}(0,y) = \boldsymbol{u}(1,y) = 0$.

The forcing terms that have to be added to the momentum and energy (temperature) equation are derived by injecting the exact solution (33) into the system (7)-(9):

$$\begin{aligned}
f_{u_1} &= 0, \\
f_{u_2} &= \sigma^2 h(y) h'(y) \left(g''(x)^2 - g'(x) g^{(3)}(x)\right) \\
&+ \frac{\sigma}{Re}\left(g^{(4)}(x) h(y) + 2 g''(x) h''(y) + g(x) h^{(4)}(y)\right) \\
&+ \frac{\sigma^2}{2} g'(x)^2 \left(h(y) h^{(3)}(y) - h'(y) h''(y)\right) - \frac{Ra}{PrRe^2} T(x,y), \\
f_T &= u_1(x,y) a'(x) b(y) + u_2(x,y) \left(T_h - T_c + a(x) b'(y)\right) \\
&- \frac{K}{RePr}\left(a''(x) b(y) + a(x) b''(y)\right).
\end{aligned} \tag{36}$$

We used the Taylor-Hood finite element (P_2 for the velocity and P_1 for the pressure) and tested P_1 or P_2 finite elements for the temperature. Figures 3a and 3b illustrate the streamlines and the temperature field, respectively. Figure 3 plots the discretization error ε as a function of the grid size $h = \delta x = \delta y$ for the temperature. Both L^2 and L^∞ norms are displayed. The expected second order accuracy in L^2-norm is obtained with P_1 finite elements (Figure 3c), while an order exceeding three is observed when P_2 finite elements are used (Figure 3d).

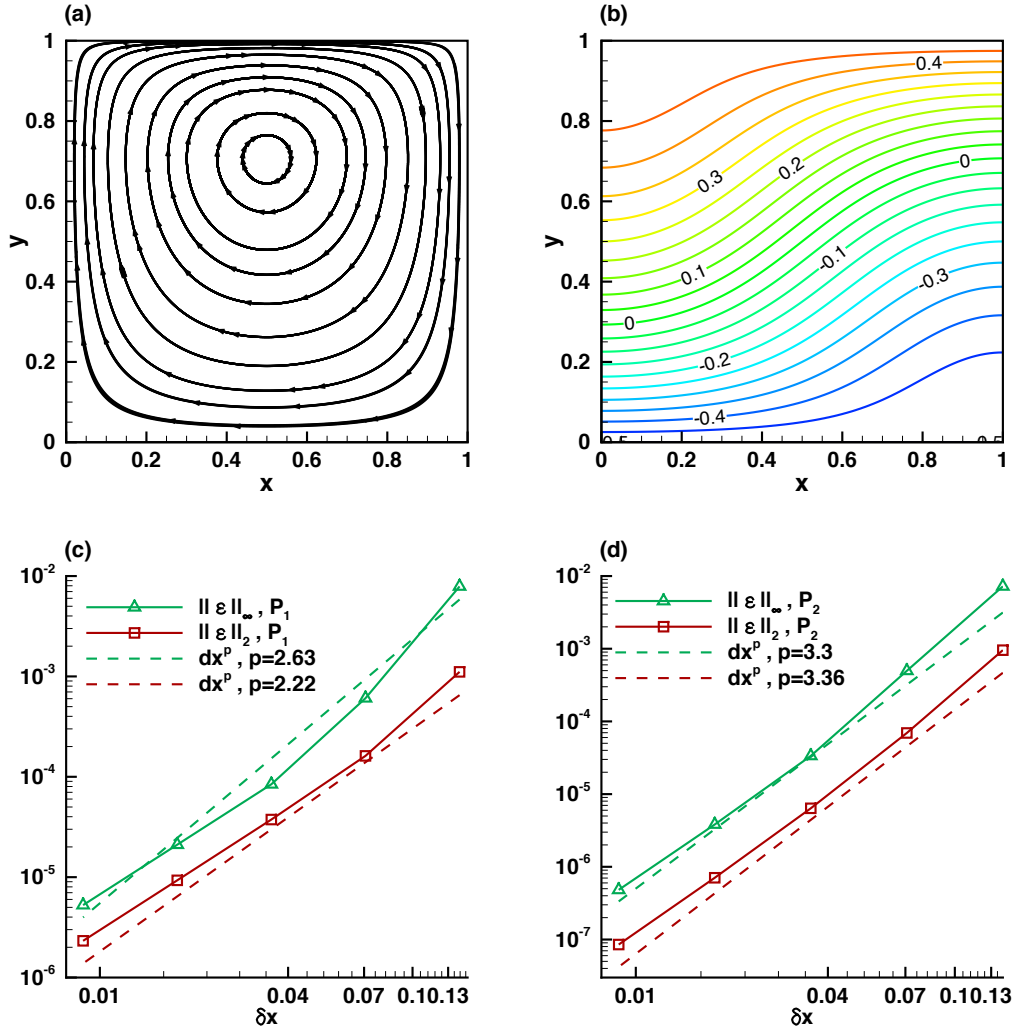

Figure 3: Burggraf stationary flow with thermal effects used to test the space accuracy of the numerical scheme. Streamlines (a) and temperature contours (b) of the flow field. Global error ε (cf. Eq. (32)) for the temperature: (c) P_1 and (d) P_2 finite elements.

5.2. Time accuracy: manufactured unsteady solution

To test the time accuracy of the Gear (BDF2) scheme, we used the manufactured time-dependent solution suggested in Nourgaliev et al. (2016):

$$\begin{aligned} u_1(x,y,t) &= (\delta U_0 + \alpha_u \sin(t)) \cos(x+\gamma_1 t) \sin(y+\gamma_2 t), \\ u_2(x,y,t) &= -(\delta U_0 + \alpha_u \sin(t)) \sin(x+\gamma_1 t) \cos(y+\gamma_2 t), \\ T(x,y,t) &= \bar{T} + (\delta T_0 + \alpha_t \sin(t)) \cos(x+\gamma_1 t) \sin(y+\gamma_2 t), \\ p(x,y,t) &= \bar{P} + (\delta P_0 + \alpha_p \sin(t)) \sin(x+\gamma_1 t) \cos(y+\gamma_2 t), \end{aligned} \tag{37}$$

The values of the constants are reported in Table 1. The corresponding forcing source

γ_1	γ_2	$\bar{P}$	$\bar{T}$	δP_0	δT_0	δU_0	α_p	α_u	α_t
0.1	0.1	0	1.0	0.1	1.0	1.0	0.05	0.4	0.1

Table 1: Parameter for the time-dependent manufactured solution (37).

terms are:

$$\begin{aligned} f_{u_1} &= \alpha_u \cos(t) \cos(a)\sin(b) - U_c\,\gamma_1 \sin(a)\sin(b) + U_c\,\gamma_2 \cos(a)\cos(b) \\ &\quad -U_c\,u_1(x,y,t) \sin(a)\sin(b) + U_c\,u_2(x,y,t) \cos(a)\cos(b) + P_c \cos(a)\cos(b) \\ &\quad +\frac{2}{Re}\,u_1(x,y,t), \\ f_{u_2} &= -\alpha_u \cos(t) \sin(a)\cos(b) - U_c\,\gamma_1 \cos(a)\cos(b) + U_c\,\gamma_2 \sin(a)\sin(b) \\ &\quad -U_c\,u_1(x,y,t) \cos(a)\cos(b) + U_c\,u_2(x,y,t) \sin(a)\sin(b) - P_c \sin(a) \sin(b) \\ &\quad +\frac{2}{Re}\,u_2(x,y,t) - \frac{Ra}{PrRe^2}\,T(x,y,t), \\ f_T &= \alpha_t \cos(t) \cos(a)\sin(b) - T_c\,\gamma_1 \sin(a)\sin(b) + T_c\,\gamma_2 \cos(a)\cos(b) \\ &\quad -T_c\,u_1(x,y,t) \sin(a)\sin(b) + T_c\,u_2(x,y,t) \cos(a)\cos(b) + \frac{2K}{RePr}\,T_c \cos(a)\sin(b), \end{aligned} \tag{38}$$

where $a = (x+\gamma_1 t)$, $b = (y+\gamma_2 t)$ and $U_c = (\delta U_0 + \alpha_u \sin(t))$, $T_c = (\delta T_0 + \alpha_u \sin(t))$, $P_c = (\delta P_0 + \alpha_u \sin(t))$.

Guided by the results obtained in §5.1 for the space accuracy, we fixed the grid size to $h = dx = 0.01$ to ensure small spatial discretization errors. For diminishing values of the time step δt, the solution was evolved in time up to the time instant $t_{max} = \pi$ at which the error (32) was computed. The time convergence is displayed in Figure 4 for the temperature variable. The expected second order convergence in time is obtained for both P_1 (Figure 4a) and P_2 (Figure 4b) discretizations of the temperature.

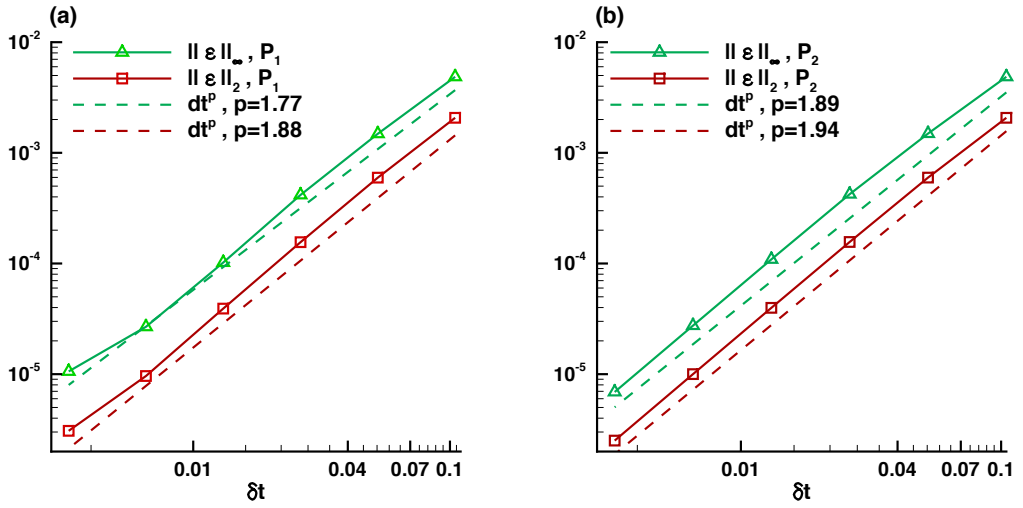

Figure 4: Time accuracy of the numerical scheme tested using the time-dependent manufactured solution of Nourgaliev et al. (2016). Evolution of the global error ε given in (32) for the temperature at $t_{max} = \pi$. Discretizations using: (a) P_1 and (b) P_2 finite elements.

6. Numerical simulations of natural convection and phase-change problems

In this section we test the robustness of the distributed toolbox. We consider well defined benchmarks used validate numerical codes for natural convection and phase-change problems. The difficulty of the computed cases is increased progressively by considering the following physical systems: (i) natural convection of air (§6.1), (ii) melting of a phase-change material (§6.2), (iii) alternate melting and solidification of phase-change system (§6.3) and (iv) the convection and the freezing of pure water (§6.4). This approach allows to test the programs by adding progressively non-linearities in the Newton algorithm. For each test case, we compare our results with experimental and previously published numerical data.

6.1. Natural convection of air

We start by testing the Newton algorithm (23)-(25) for the case of natural convection, *i. e.* $C = K = 1$, $A(\theta) = S(\theta) = 0$. We consider the classical problem of the thermally driven square cavity $[0, 1] \times [0, 1]$, filled with air. The Boussinesq term $f_B(\theta)$ is then linear and takes the form (10). Top and bottom walls are adiabatic, while the temperature is fixed on the left (cold) wall and the right (hot) wall. Natural convection flows are computed for three values of the Rayleigh number: $Ra = 10^4, 10^5, 10^6$. The Prandtl number is set to $Pr = 0.71$. It was shown in Le Queré (1991) that the flow in this configuration becomes unsteady for $Ra = 10^{8.5}$. Therefore, steady states could be computed for the chosen values of the Rayleigh number.

We provide programs for both steady (time-independent) and time-dependent cases. The steady case is performed using a continuation following the Rayleigh number: a smaller value for Ra is set initially and is then smoothly increased until reaching the

wanted value. The time-dependent case is computed until a steady state with a single convection cell is reached. To make the simulation more complicated, we present two configurations: *i)* the classical differentially heated square cavity and *ii)* a differentially heated cavity with an inner heated obstacle.

6.1.1. Classical differentially heated square cavity

The temperature is imposed at the left (cold) wall as $\theta_c = -0.5$ and at the right (hot) wall as $\theta_h = 0.5$. Top and bottom walls are adiabatic. The initial condition models a cavity filled with motionless air ($\boldsymbol{u} = 0$), with a linear distribution of the temperature. Both steady and time-dependent codes converge to the same flow state with a single convection cell. For this final state, horizontal $u(y)$ and vertical $v(x)$ velocity profiles were extracted at mid-domain ($y = 0.5$ and $x = 0.5$, respectively) and plotted in Figure 5. Our results are in very good agreement with reference numerical results obtained by Le Queré (1991) with a spectral code.

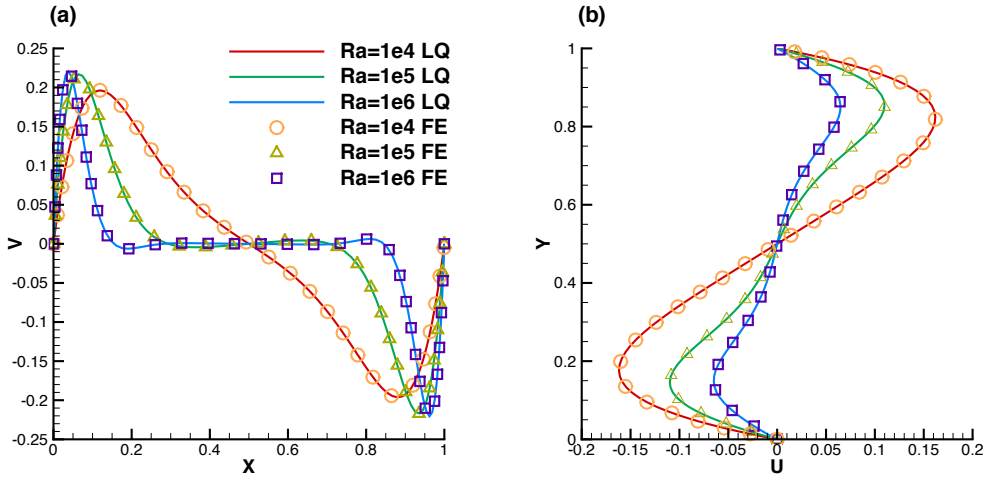

Figure 5: Natural convection of air in a differentially heated cavity for values of Ra ranging from 10^4 to 10^6. (a) Vertical velocity profile $v(x)$ along the horizontal symmetry line ($y = 0.5$). (b) Longitudinal velocity profile $u(y)$ along the vertical symmetry line ($x = 0.5$). Results obtained using the present Newton method (symbols), with a mesh resolution $nbseg = 80$. Comparison with the spectral simulations by Le Queré (1991) (solid lines).

Table 2 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 Le Queré (1991). The Newton method gives results identical to reference values, with a difference less than 0.01%.

Run		u_{max} at x=0.5 (error)	Y (error)
Reference values	spectral	0.0648344	0.850
Newton (Steady)	$nbseg = 80$	0.0648297 (0.007 %)	0.850394(0.05 %)
Newton (Unsteady)	$nbseg = 80$	0.0648296 (0.007 %)	0.850532 (0.06 %)

Table 2: Natural convection of air in a differentially heated 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 Queré (1991).

6.1.2. Differentially heated cavity with an inner heated obstacle

We consider the same differentially heated cavity as previously and add a centred square obstacle. The boundaries of the inner square are non-slip isothermal walls, maintained at a dimensionless hot temperature $\theta_h = 0.8$. The solution computed for $Ra = 10^6$ and $Pr = 0.71$ is compared with the results obtained by Moglan (2013), who used a 6th order finite-difference method with an immersed boundary method to model the obstacle. The temperature distribution in the cavity is shown in Figure 6a. The vertical velocity profile $v(x)$ along the horizontal symmetry line ($y = 0.5$) is displayed in Figure 6b and shows a very good agreement with the numerical results reported by Moglan (2013).

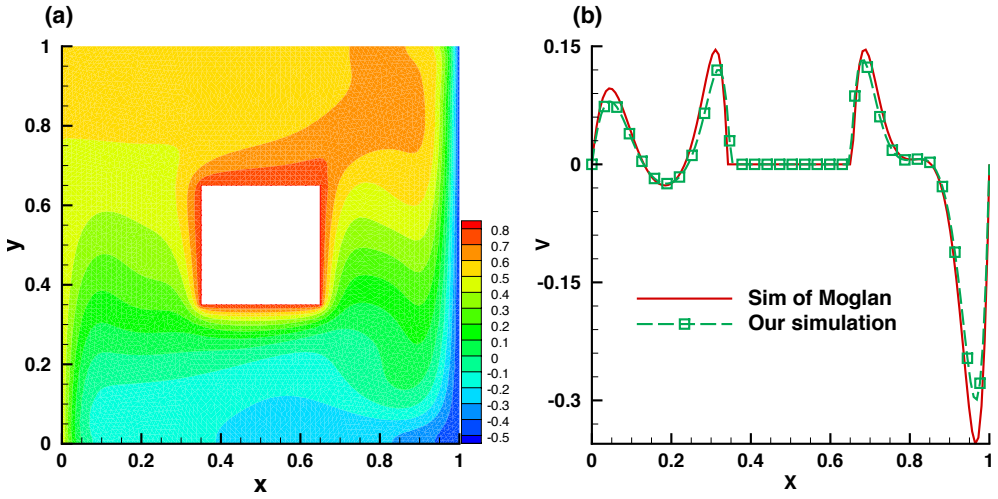

Figure 6: Natural convection of air in a differentially heated cavity with an inner heated square for $Ra = 10^6$. Temperature field (a) and vertical velocity profile along the horizontal symmetry line (b). Results obtained using the present Newton method (red solid line), with mesh resolution $nbseg = 80$. Comparison with the results obtained using a 6th order finite-difference method by Moglan (2013).

6.2. Melting of a phase-change material (PCM)

We continue our validation tests by considering the full system (23)-(25) for the case of the melting of a phase-change material. Two new non-linearities are now present in the system: the Carman-Kozeny penalty term $A(\theta)$ and the enthalpy source term $S(\theta)$. The function S is regularized using (15). We also consider that the material properties in the liquid and solid are the same, *i. e.* $C = K = 1$. This is a frequent assumption (Wang et al., 2010a; Ma and Zhang, 2006). Five cases were computed (the exact values of the defining parameters are summarized in Table 3):

- PCM-Case #1 simulates the experimental study of Okada (1984). It consists of a differentially heated square cavity, filled with octadecane paraffin.
- PCM-Case #2 is extracted from the collective publication by Bertrand et al. (1999), in which the results of different numerical approaches were compared for the simulation of the melting of a PCM.
- PCM-Case #3 simulates the melting of a cylindrical PCM with heated inner tubes, as in Luo et al. (2015).
- PCM-Case #4 simulates the melting of Gallium in a rectangular cavity heated by the side-wall, as in Hannoun et al. (2003).
- PCM-Case #5 reproduces the simulation of Nourgaliev et al. (2016) using highly distorted meshes to simulate a natural convection flow with solid crush formation.

	case #1	case #2	case #3	case #4	case #5
Ra	$3.27 \cdot 10^5$	10^8	$5 \cdot 10^4$	$7 \cdot 10^5$	10^6
Pr	56.2	50	0.2	0.0216	0.1
Ste	0.045	0.1	0.02	0.046	4.854
δt	0.1	10^{-3}	10^{-3}	10^{-5}	10^{-3}
V_{ref}	$\frac{\nu_l}{H}$	$\frac{\nu_l}{H}$	$\frac{\nu_l}{H}$	$\frac{\nu_l}{H}$	$\frac{\nu_l}{H}\sqrt{\frac{Ra}{Pr}}$
CPU time	2876.39	40522.7	2060.42	14621.2	6657.58
Number of triangles	$2,900$	$7,000$	$3,076$	$4,820$	$2,769$

Table 3: Parameters for the cases simulating the melting of a phase-change material.

To guide the user of the toolbox, we also provide in Table 3 the values of the CPU time necessary to run each case (with default parameters) on a personal computer and the typical number of triangles of the generated adaptive mesh.

6.2.1. PCM-Case #1: Melting of an octadecane PCM in a square cavity

Okada (1984) studied experimentally the melting of an octadecane PCM in a square cavity of height $H = 1.5$ cm. His results were often used to validate numerical methods (Okada, 1984; Wang et al., 2010a; Ma and Zhang, 2006; Danaila et al., 2014). The material is initially solid ($\theta_0 = -0.01$) and melts progressively starting from the left boundary, maintained at a hot temperature $\theta_h = 1$. The right boundary is also isothermal, with

cold temperature $\theta_c = -0.01$. Horizontal boundaries are adiabatic. The other parameters of this case are reported in Table 3.

The computation starts from a refined mesh near the hot boundary. Mesh adaptivity is applied at each time step using metrics computed from three variables: the two fluid velocities and the enthalpy source term S. To reduce the impact of the interpolation on the global accuracy, we use two successive fields (S^n) and (S^{n+1}) in the adaptivity procedure. This allows us to refine the mesh in the fluid part of the domain and inside the artificial mushy. Figure 7a gives an illustration of the adapted mesh at dimensionless time $t = 78.7$. In Figure 7b we compare the position of the solid-liquid interface at $t = 39.9$ and $t = 78.7$ with the experimental data of Okada (1984) and previously published numerical results (Wang et al., 2010a; Danaila et al., 2014). The obtained shape and position of the liquid-solid interface is closer to experimental results than numerical results reported in Wang et al. (2010a). This is a direct consequence of the mesh adaptivity capabilities of our method.

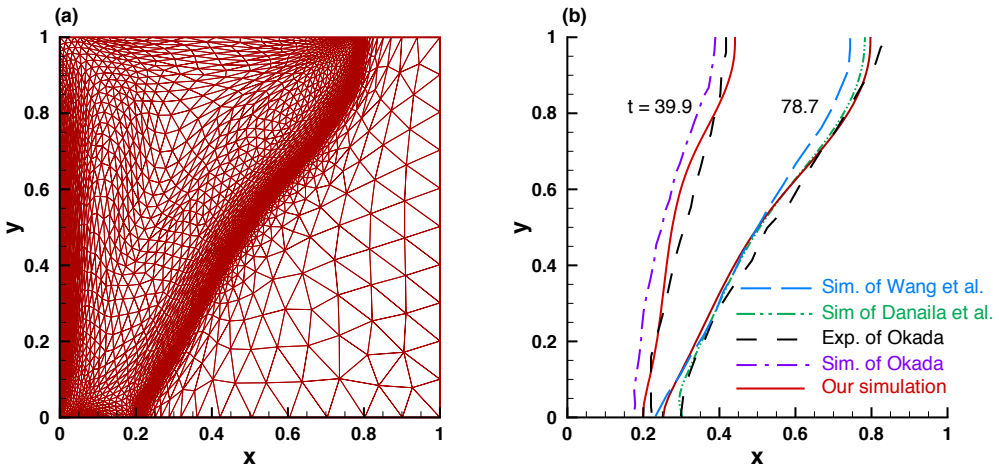

Figure 7: PCM-Case #1. Melting of the PCM. (a) Adapted finite-element mesh at time instant $t = 78.7$. (b) Position of the solid-liquid interface. Comparison with experimental data of Okada (1984) and numerical results of Danaila et al. (2014) and Wang et al. (2010a) for two time instants ($t = 39.9$ and 78.7).

This comparison also allowed us to finely tune the value of the constants used in the model (16). Even though it is generally assumed that a large value for $C_{\rm CK}$ must be set, the exact value of this constant could influence the accuracy of the results (Kheirabadi and Groulx, 2015; Mathura and Krishna, 2017). This choice of the value of this constant is a still open problem. Very good agreement with the experimental result of Okada (1984) is obtained for $C_{\rm CK}$ varying in the range $[10^6, 10^8]$. Imposing a too large value ($C_{\rm CK} = 10^{10}$) results in artificially slowing the propagation of the melting front. Consequently, we set for all subsequent simulations $C_{\rm CK} = 10^6$.

6.2.2. PCM-Case #2: Melting of an octadecane PCM with high Rayleigh number

This case considers the same problem of the melting of a PCM, but with a very high value of the Rayleigh number $Ra = 10^8$ (see Table 3). This case is very challenging since the natural convection becomes important in the fluid flow, and enhances considerably the heat transfer.

Bertrand et al. (1999) compiled results provided by five different authors (Lacroix, Le Quéré, Gobin-Vieira, Delannoy and Binnet-Lacroix). These results will be hereafter referred to as (say) 'Lacroix, from Bertrand et al. (1999)'. They have attempted a first comparison by taking several numerical methods to compute the basic configuration presented in this section. Two investigators among the five failed to predict the process and showed unrealistic behaviours in Figures 8a and 8b: Lacroix and Delannoy seem to be insufficiently converged as shown by Figure 8a, and Binet-Lacroix overestimates the average Nusselt number by more than 30% (Figure 8b). Hence, this collection of results allows us to validate our numerical method and check if realistic results are obtained for complex physical configurations. For comparison purpose, we extract from simulations the position of the melting front and the Nusselt number Nu at the left wall ($x = 0$) for each of the five methods presented by Bertrand et al. (1999). The Nusselt number Nu is defined as follows:

$$Nu = \int_0^1 \left(\frac{\partial \theta}{\partial x} \right)_{x=0} dy. \tag{39}$$

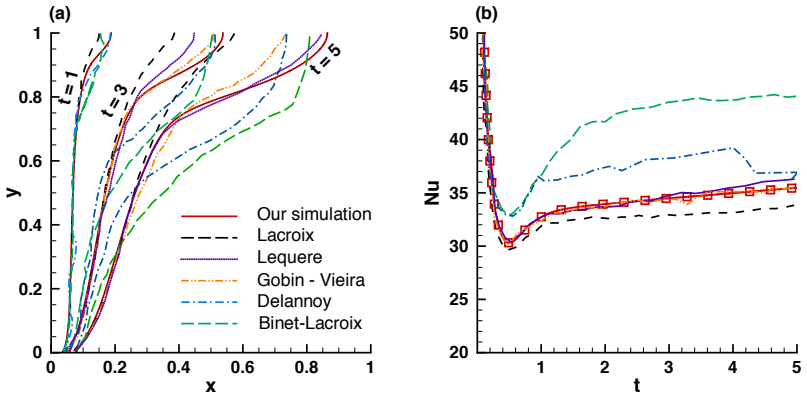

Figure 8: PCM-Case #2. Melting of the PCM with high Rayleigh number ($Ra = 10^8$). Comparison with five sets of results presented in Bertrand et al. (1999). (a) Location of the solid-liquid interface at dimensionless time instants $t = 1$, 3 and 5. (b) Temporal evolution of the Nusselt number.

The position of the melting front for three time instants, $t = 1$, 3 and 5 is reported in Figure 8a. Our results are for each case in fairly good agreement with those of Gobin and Le Quéré. Details of their numerical method are presented in Gobin and Le Quéré (2000). Gobin uses a front-tracking method using a coordinate transformation with a finite volume method in a 62×42 grid and Le Quéré solves a single domain method using a second-order scheme with a finite volume method and a 192×192 grid. The time evolution of the Nusselt number is presented in Figure 8b. A very good agreement is

obtained with the results of Gobin and Le Quéré. A relative difference, less than 2%, is observed for the Nusselt number.

The high Rayleigh number $Ra = 10^8$ is a very demanding numerical test. The high velocity, inducing a very narrow thermal boundary layer, can lead to unrealistic results if under-resolved. This explains why some numerical methods failed in Bertrand et al. (1999). The interest of the mesh adaptation is clearly demonstrated for this case, since we initially used a coarse 40×40 grid.

6.2.3. PCM-Case #3: Melting of cylindrical PCM with inner heated tubes

Previous cases #1 and #2 considered phase change problems evolving in a simple geometry, a square cavity. A more complex geometry, suggested by Luo et al. (2015), is simulated in this section. It consists of a cylindrical PCM of radius $R = 1$ with tube inclusions of different arrangements. The interest in studying this case is not solely the challenge of the complex configuration, but also the possibility to compare our results with those of Luo et al. (2015), obtained using a completely different model based on the Lattice Boltzmann Method. This configuration is also interesting from a practical point of view. Agyenim et al. (2010) pointed out that more than 70% of the PCM containers used for heat storage are using shell-tube systems.

We simulate three configurations with one, four and nine heated tubes. The size of the tubes is adjusted to have the same contact area for all configurations. The radius R_i of the inner tube is $R_i = R/4$ for the case with one tube, $R_i = R/8$ for the four heated tubes case and $R_i = R/12$ for the case with nine tubes. A Dirichlet boundary condition ($\theta = \theta_h$) is applied to the boundary of inner tubes. A Neumann boundary condition ($\partial\theta/\partial n = 0$) is used for the outer boundary. For the velocity, all boundaries are considered as non-slip walls ($\boldsymbol{u} = 0$). Only half of the domain is simulated since all configurations are symmetric with respect to the vertical axis (see Figure 9). The mesh is refined initially around the inner tubes, and is dynamically adapted at each time step around the melting front and the thermal boundary layer area. The same metrics presented in Section 6.2.1 are used for the mesh adaptivity.

Figure 9 shows the temperature field and the position of the solid-liquid interface (black line) for the three configurations for time instants corresponding to the same liquid fraction $L_f = 80\%$. The distribution of the inner tubes in the liquid phase influences directly the fluid motion and the shape of melting front. The more the number of inner tubes, the stronger the natural convection is in the melted PCM. The shape of the solid-liquid interface displays complex patterns, depending on the space arrangement of the inner tubes. The mesh is nicely adapted following the evolution of the melting interface, even after its separation in several distinct fronts touching the outer boundary (see Figures 9b, c).

To estimate the efficiency of each configuration, we plot in Figure 10 the time evolution of the liquid fraction L_f. By including more heated tubes the heat transfer is enhanced, inducing a faster melting time. The nine-tube configurations melts 5 times faster than the reference configuration with one tube. Note also from Figure 10 a good agreement between our results and those reported by Luo et al. (2015) for the evolution of the liquid fraction.

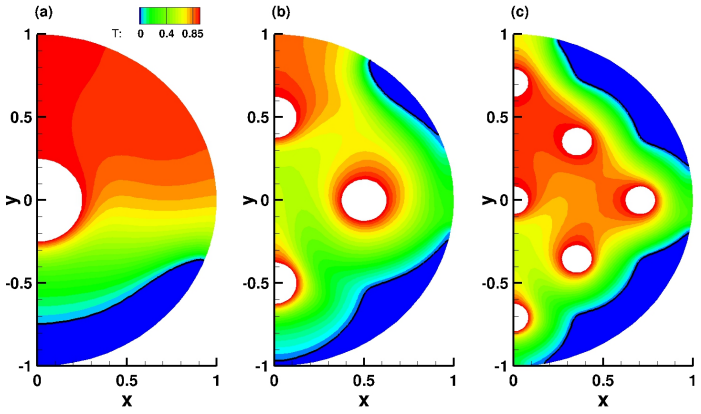

Figure 9: PCM-Case #3. Temperature fields for the melting of a cylindrical PCM with inner heated tubes. Time instants corresponding to the same liquid fraction $L_f = 80\%$. Configurations with (a) one tube ($t = 2.5$), (b) four tunes ($t = 0.99$) and (c) nine tubes ($t = 0.4$). Melting fronts are localized with black lines (only half of the domain is simulated).

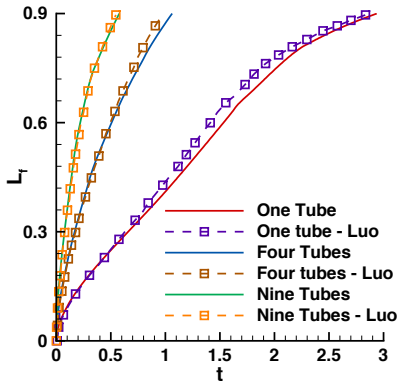

Figure 10: PCM-Case #3. Time evolution of the liquid fraction for configurations with one, four, and nine heated tubes. Comparison with numerical results of Luo et al. (2015).

6.2.4. PCM-Case #4: Melting of Gallium in a rectangular cavity

The melting of the Gallium in a rectangular cavity was a controversial case since Dantzig (1989) raised the question whether the convection in the fluid is mono-cellular or multi-cellular. Experimental results exhibited indeed a mono-cellular structure, while many researchers claimed that this observation was incorrect. Prior to Dantzig (1989), both experimental and numerical studies reported a single convection cell in the fluid phase. Later, simulations provided solutions with multi-cellular flow. Hannoun et al. (2003) concluded that the mono-cellular observation was caused by a problem of convergence of the numerical solution, due to coarse grids or inconsistencies in the mathematical model.

Therefore, this test case simulating the melting of the Gallium is a relevant exercise to test the accuracy of our method. The parameters of this case are reported in Table 3. To capture the very small convection cells during the first step of the melting, Hannoun et al. (2003) used a $800 \times 1,120$ fixed grid in a rectangular domain of dimensions 6.35 cm $\times$ 8.89 cm. With our adaptive method, a maximum of 4820 triangles are necessary to reproduce the numerical result of Hannoun et al. (2003). The grid size is thus reduced with our method by a factor of 100.

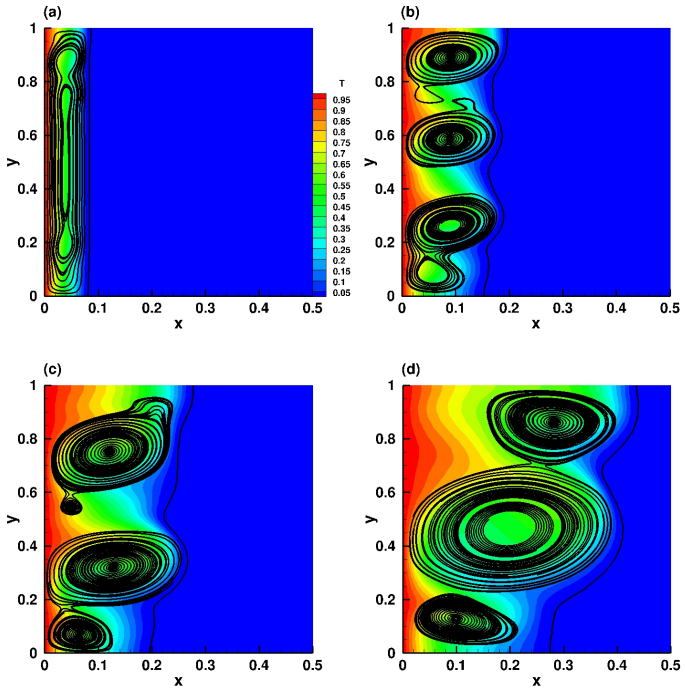

Figure 11: PCM-Case #4. Melting of Gallium: temperature field, streamlines, and melting front for dimensionless time instants (panels a to d): 0.0015, 0.006, 0.01, and 0.019. For a better view of the convection cells, a ratio 2:1 was used for the axis dimensions.

The time evolution of the flow is presented in the Figure 11. Temperature filed, streamlines and position of the melting front are plotted for several time instants: $t =$ 0.0015, 0.006, 0.01, and 0.019. These values were chosen to visualise the merging of convection cells in the fluid flow and correspond to physical times $20s, 85s, 155s, 280s$ in Hannoun et al. (2003). The number of rolls was considered as a validation criterion by several authors (Hannoun et al., 2003; Cerimele et al., 2002; Giangi and Stella, 2000). Three cells are observed at $t = 0.006$ (Figure 11). The number of cells decreases later through a process of roll merging, as it was also reported by Hannoun et al. (2003). Our numerical results are in good agreement with the observations of Hannoun et al. (2003), Cerimele et al. (2002) and Giangi and Stella (2000).

6.2.5. PCM-Case #5: Solid crust formation in a highly distorted mesh

Nourgaliev et al. (2016) used a discontinuous Galerkin finite element method to simulate the solid crust formation inside a highly distorted domain (Figure 12). The fluid is initially motionless with an initial dimensionless temperature $\theta_0 = 2$. The temperature of fusion is set to $\theta_f = 1.4$, according to Nourgaliev et al. (2016) parameters (see Table 3 for the values of all parameters). The left side boundary is maintained at a cold temperature $\theta_c = 1.39$ in the initial stage. The right wall is isothermal, with hot temperature $\theta_h = 2$. A nearly steady-state natural circulation is induced in the early time evolution of the flow. Then, the cold temperature at the left wall is decreased to $\theta_c = 1$, below the temperature of solidification. At this point, the formation of a solid crust layer starts at the left boundary. Figure 12 shows the temperature field and the streamlines for the time instant $t = 30$. Our results are in a very good agreement with those of Nourgaliev et al. (2016).

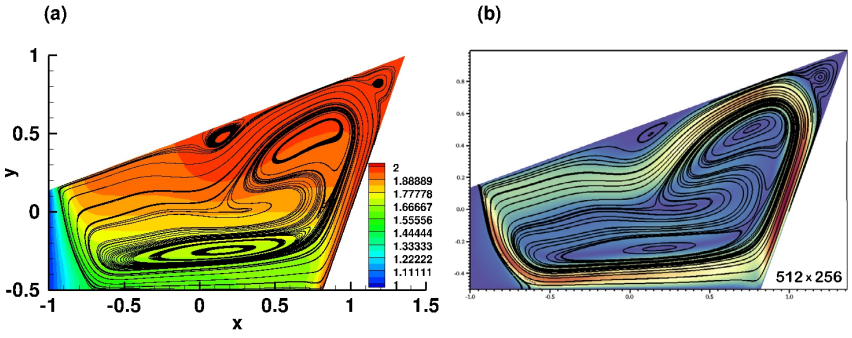

Figure 12: PCM-Case #5. Solid crust formation in a distorted mesh. Temperature field and streamlines at dimensionless time $t = 30$: our simulation (a) and Nourgaliev et al. (2016) (b).

6.3. Melting-solidification cycle of a PCM

We address in this section the challenging problem of simulating a complete melting-solidification cycle of a PCM. The simulation starts from the final state obtained in §6.2.1. At $t = 78.7$, the PCM is partially melted, with the liquid fraction $L_f = 0.5$ (see Figure 13a). The temperature of the left boundary is then dropped to a cold temperature $\theta_c = -0.01 < \theta_f$, identical to that of the right wall. The solidification starts and the solid phase propagates into the cavity from both left and right sides. Panels (a) to (e) in Figure 13 depict the time evolution of the solidification process. The solid phase is represented in blue and corresponds to the region of temperature $\theta \leq \theta_f = 0$, while the solid-liquid interfaces (*i. e.* contour lines of $\theta = 0$) are represented by bold solid lines. The solidification is completely achieved at $t = 4600$. Note that the mesh adaptivity of our toolbox is able to accurately track the two solidification fronts. The adapted mesh shown in Figure 13f illustrates that a finer mesh is well generated along the interface ($\theta = 0$), while a coarser mesh is obtained in the regions of low gradients.

A comprehensive physical description of this case is given in our recent paper Rakotondrandisa et al. (2019). The case of the solidification process starting after a complete melting of the PCM is also presented in this paper. We can conclude from this section that the Newton method is able to deal with either melting or solidification process. The simulations are in good agreement with existing experimental and numerical results for the melting and show consistent behaviour for the solidification.

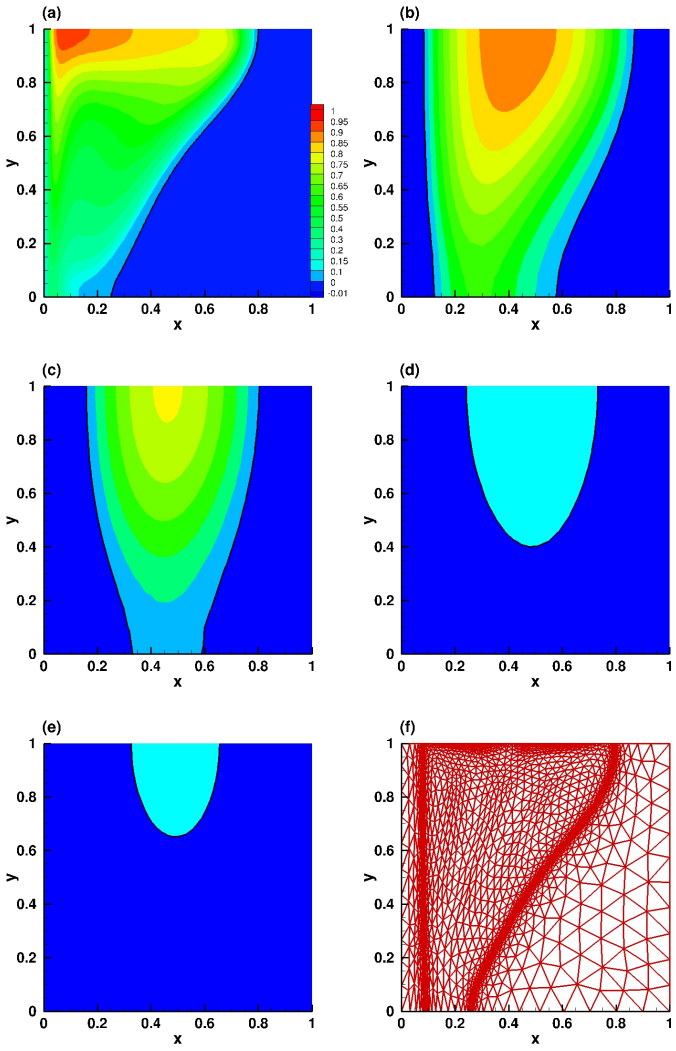

Figure 13: Solidification of the PCM after a partial melting. Temperature contour-lines in the fluid phase. The solid part is represented in blue and corresponds to the region of temperature $\theta \leq \theta_f = 0$. Time instants (panels a to e): $t = 78.92$, $t = 1072$, $t = 2702$, $t = 3902$ and $t = 4501$. The adapted mesh corresponding to panel (b) is shown in panel (f).

6.4. Natural convection of water and water freezing

We consider in this section the natural convection and the solidification of water. With these cases, we test the ability of our numerical system to deal with additional non-linear terms. Since pure water exhibits non-linear density variation for $T < 10.2\,^{\circ}\mathrm{C}$, with a maximum at $T_m = 4.0293\,^{\circ}\mathrm{C}$, the Boussinesq force becomes non-linear. We used the following density-temperature relationship suggested by Gebhart and Mollendorf (1977):

$$\rho(T) = \rho_m \left(1 - w\,|T - T_m|^q\right), \tag{40}$$

with $\rho_m = 999.972$ [kg/m^3], $w = 9.2793 \cdot 10^{-6}$ [$(^{\circ}C)^{-q}$], and $q = 1.894816$.

Hence, the bouyancy term $f_B = g(\rho_{ref} - \rho)/\rho_{ref}$ in (7) is not any more linear and becomes after scaling:

$$f_B(\theta) = \frac{Ra}{Pr\,Re^2}\,\frac{1}{\beta\delta T}\,\frac{\rho(\theta_f) - \rho(\theta)}{\rho(\theta_f)}, \tag{41}$$

where $\beta = (1/\rho_m)\,(d\rho/dT)$ is the thermal expansion coefficient taking the value $\beta = 6.91 \cdot 10^{-5}$ [$(\mathrm{K})^{-1}$] (Scanlon and Stickland, 2004).

We simulate a differentially heated square cavity filled with liquid pure distilled water. This problem was investigated experimentally and numerically by Giangi et al. (2000); Kowalewski and Rebow (1999); Michalek and Kowalewski (2003). The non-dimensional parameters describing the problem are (see Michalek and Kowalewski (2003) for physical details): $\mathcal{R}a = 2.518084 \cdot 10^6$, $\mathcal{P}r = 6.99$ and $\mathcal{S}te = 6.99$.

6.4.1. Natural convection of water

The initial temperature is linearly distributed in the square cavity, with a hot temperature $T_h = 10\,^{\circ}\mathrm{C}$ at the left wall and a cold temperature $T_c = T_f = 0\,^{\circ}\mathrm{C}$ at the right wall. The temperature field and the streamlines of the steady state are presented in Figure 14a. The isoline $\theta = \theta_m$, corresponding to the line of maximum density is represented by a dashed line. 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 $\theta < \theta_m$ and an upper (normal) one where the density decreases with temperature ($\theta > \theta_m$).

A more precise comparison with previously published results is shown in Figure 14b. The obtained temperature profile $\theta(x)$ along the horizontal symmetry line of the cavity ($y = 0.5$) is in good agreement with the numerical results of Michalek and Kowalewski (2003). Their results were obtained with finite-volumes and finite-differences codes (FLUENT and FRECONV3V). 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 Michalek and Kowalewski (2003) are performed with a fixed uniform grid with 380×380 nodes, while our adapted grid has only 1,807 vertices (3,430 triangles).

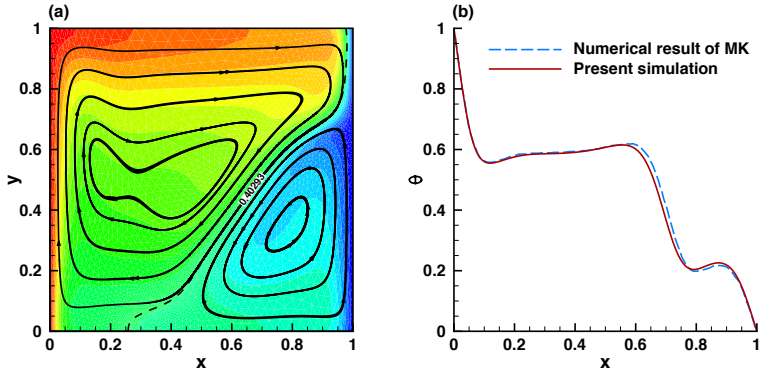

Figure 14: Natural convection of water in a differentially heated cavity. Non-dimensional temperature θ at steady state. (a) Two-dimensional temperature field and streamlines showing the two recirculating zones. (b) Temperature profile along the horizontal symmetry line. Comparison with the numerical results of Michalek and Kowalewski (2003).

6.4.2. Water freezing

We finally consider the difficult case of water freezing in a square cavity. The initial state for this computation is the convection steady pattern in the cavity presented in Figure 14. The freezing starts by dropping smoothly the temperature of the cold right wall from $T_c = 0\,^{\circ}\mathrm{C}$ to $T_c = -10\,^{\circ}\mathrm{C}$.

Figure 15(a) superimposes the experimental image from Kowalewski and Rebow (1999) with our numerical results for the same physical time $t_\varphi = 2340[s]$. The flow pattern in the liquid phase, shown in Figure 15(b) also corresponds very well qualitatively 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 3000 nodes) due to the efficiency of the adaptivity algorithm.

The two recirculating zones being separated by the line $T = T_m$ we used the mesh adaptivity capability of the method to refine the grid along this line. The metrics used for adaptivity were 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:

$$\phi(T) = \frac{1}{2}\left\{1 + \tanh\left(\frac{T_m - T}{R_\phi}\right)\right\}, \tag{42}$$

with $R_\phi = 0.02$. To reduce the impact of the interpolation on the global accuracy (see also Belhamadia et al. (2004)), we used both $\phi(T^n)$ and $\phi(T^{n+1})$ in the adaptivity procedure. The final mesh is displayed in Figure 15c, clearly showing that the mesh is refined along the lines $T = T_m$ and $T = 0\,^{\circ}\mathrm{C}$. This allows to accurately capture the structure and the extent of the two recirculating zones, features that are difficult to obtain with fixed meshes (discrepancies in numerical results are described in Giangi et al. (2000); Kowalewski and Rebow (1999); Michalek and Kowalewski (2003)). As a consequence, the temperature contours lines in Figure 15d 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.

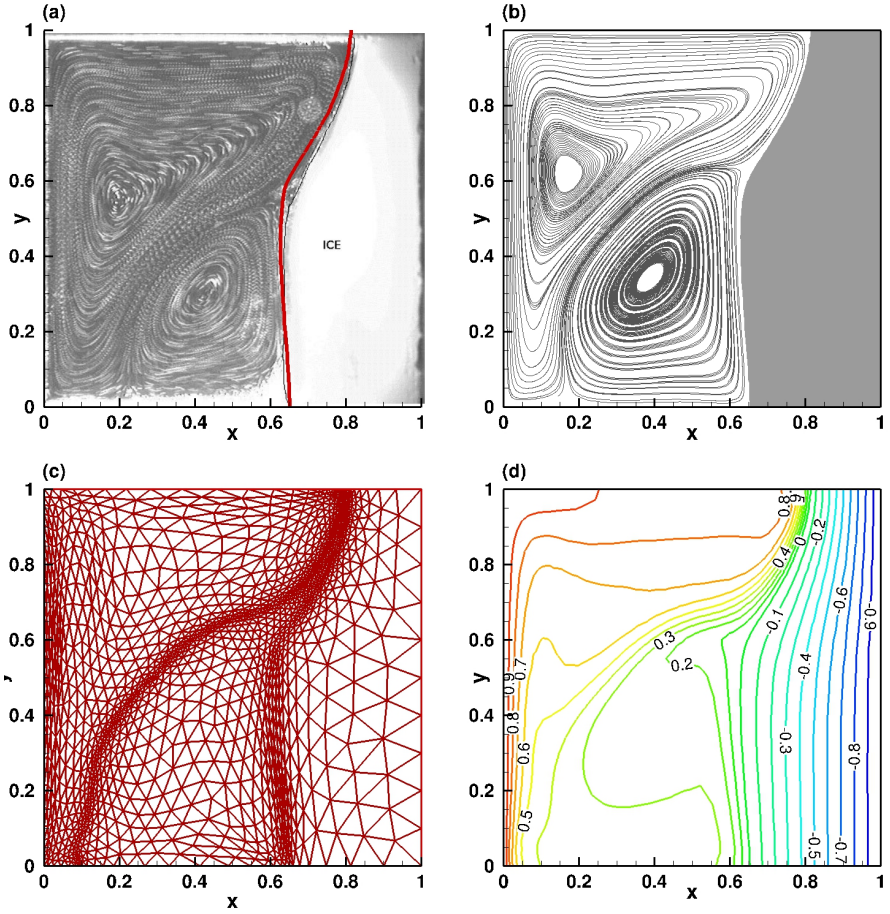

Figure 15: Freezing of pure water. Configuration at (physical time) $t_\varphi = 2340[s]$: (a) experimental image from Kowalewski and Rebow (1999); the thick red line represents the solid-liquid interface computed with the present method (b) computed streamlines showing the two recirculating zones in the fluid phase (c) finite-element mesh refined along the solid-liquid interface ($T = 0\,^\circ$C) and also along the line of maximum water density ($T = 4\,^\circ$C) (d) temperature iso-lines.

7. Summary and conclusions

We provide with this paper an adaptive finite-element toolbox for solving two-dimensional phase-change problems with convection. The programs were written using FreeFem++ , a free software offering a programming syntax close to the mathematical formulation. A single domain numerical approach was first derived. The details of the finite-element formulation were then presented. The key ingredients of the implemented method are: (i) a second order accuracy in space and time; (ii) 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 (iii) a fully implicit discretization with a Newton algorithm for solving the non-linear system of equations.

Four test cases were presented, by adding progressively non-linearities in the system of equations:
(i) natural convection of air in a differentially heated cavity,
(ii) melting of a PCM,
(iii) melting-solidification cycle of a PCM,
(iv) natural convection and freezing of water.
The computations for case *(iii)* were rendered more challenging by considering complex geometries (highly distorted mesh, cylindrical PCM with inner heated tubes) and computationally demanding cases (high Rayleigh numbers). The efficiency of the adaptivity method by metric control was investigated by tracking simultaneously several interfaces (two melting fronts during the solidification cycle and density inversion interface for water flows).

For each test case, we provided a separate folder containing all the necessary files (parameters, restart files) necessary to run them directly. We described in the text body of the paper the expected results and their validation. A very good agreement with experimental data or numerical results was obtained for all considered test cases, proving the capability of our method to tackle a large range of problems. Ready-made scripts and layouts are provided with the toolbox to allow the user to generate the figures presented in this paper with newly generated data after running the programs. Validation data sets from experiments or previous publications are included in these layouts.

Since FreeFem++ is a free software, the method could be easily implemented and tested by anyone interested in simulating phase-change problems. All technical issues related to the implementation of the finite element method are hidden, allowing to focus on numerical algorithms and their performance. This offers the possibility to address other computational challenges related to different physical or mathematical models in this field.

The extension of the method for 2D and 3D cases, using domain decomposition methods adapted to parallel computing will be reported in a forthcoming contribution.

Acknowledgements

This project was co-financed by the European Union with the European regional development funds and by the Normandy Regional Council via the M2NUM (ERDF, HN0002081) and M2SiNUM (ERDF, 18P03390/18E01750/18P02733) projects. Part of this work was performed using computing resources of CRIANN (Centre Régional Informatique et d'Applications Numériques de Normandie, Normandy, France).

Bibliography

Agyenim, F., Hewitt, N., Eames, P., Smyth, M., 2010. A review of materials, heat transfer and phase change problem formulation for latent heat thermal energy storage systems (lhtess). Renewable and sustainable energy reviews 14 (2), 615–628.

Aldbaissy, R., Hecht, F., Mansour, G., Sayah, T., 2018. A full discretisation of the time-dependent Boussinesq (buoyancy) model with nonlinear viscosity. Calcolo 55 (4), 44.

Angot, P., Bruneau, C.-H., Fabrie, P., 1999. A penalization method to take into account obstacles in incompressible viscous flows. Numer. Math. 81 (4), 497–520.

Belhamadia, Y., Fortin, A., Chamberland, E., 2004. Anisotropic mesh adaptation for the solution of the Stefan problem. Journal of Computational Physics 194 (1), 233–255.

Belhamadia, Y., Kane, A. S., Fortin, A., 2012. An enhanced mathematical model for phase change problems with natural convection. International Journal of Numerical Analysis and Modeling 3 (2), 192–206.

Bertrand, O., Binet, B., Combeau, H., Couturier, S., Delannoy, Y., Gobin, D., Lacroix, M., Le Quéré, P., Médale, M., Mencinger, J., et al., 1999. Melting driven by natural convection a comparison exercise: first results. International Journal of Thermal Sciences 38 (1), 5–26.

Bhattacharya, A., Kiran, A., Karagadde, S., Dutta, P., 2014. An enthalpy method for modeling eutectic solidification. Journal of Computational Physics 262, 217–230.

Boettinger, W. J., Warren, J. A., Beckermann, C., Karma, A., 2002. Phase-field simulation of solidification. Annual review of materials research 32 (1), 163–194.

Borouchaki, H., Castro-Diaz, M. J., George, P. L., Hecht, F., Mohammadi, B., 1996. Anisotropic adaptive mesh generation in two dimensions for cfd. In: 5th Inter. Conf. on Numerical Grid Generation in Computational Field Simulations. Mississipi State Univ.

Brent, A. D., Voller, V. R., Reid, K. J., 1988. Enthalpy-porosity technique for modeling convection-diffusion phase change: Application to the melting of a pure metal. Numerical Heat Transfer 13, 297–318.

Cao, Y., Faghri, A., 1990. A numerical analysis of phase change problem including natural convection. ASME Journal of Heat Transfer 112, 812–815.

Cao, Y., Faghri, A., Chang, W. S., 1989. A numerical analysis of Stefan problems for generalized multi-dimensional phase-change structures using the enthalpy transforming model. International Journal of Heat and Mass Transfer 32 (7), 1289–1298.

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

Cerimele, M. M., Mansutti, D., Pistella, F., 2002. Numerical modelling of liquid/solid phase transitions: Analysis of a gallium melting test. Computers & fluids 31 (4), 437–451.

Chiesa, F. M., Guthrie, R. I. L., 1974. Natural convective heat transfer rates during the solidification and melting of metals and alloy systems. Journal of Heat Transfer 96 (3), 377–384.

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

Danaila, I., Moglan, R., Hecht, F., Le Masson, S., 2014. A Newton method with adaptive finite elements for solving phase-change problems with natural convection. J. Comput. Physics 274, 826–840.

Dantzig, J. A., 1989. Modelling liquid–solid phase changes with melt convection. International Journal for Numerical Methods in Engineering 28 (8), 1769–1785.

Elliott, C. M., 1987. Error analysis of the enthalpy method for the Stefan problem. IMA journal of numerical analysis 7 (1), 61–71.

Eyres, N., Hartree, D. R., Ingham, J., Sarjant, R. J., Wagstaff, J. B., et al., 1946. The calculation of variable heat flow in solids. Phil. Trans. R. Soc. Lond. A 240 (813), 1–57.

Fabbri, M., Voller, V. R., 1997. The phase-field method in the sharp interface limit: a comparison between model potential. Journal of Computational Physics 130, 256–265.

Favier, B., Purseed, J., Duchemin, L., 2019. Rayleigh-Bénard convection with a melting boundary. Journal of Fluid Mechanics 858, 437?473.

Frey, P. J., George, P. L., 1999. Maillages. Hermès, Paris.

Gartling, D., 1980. Finite element analysis of convective heat transfer problems with change of phase. In: Morgan, K., Taylor, C., Brebbia, C. (Eds.), Computer Methods in Fluids. Pentech, London, pp. 257–284.

Gau, C., Viskanta, R., 1984. Melting and solidification of a metal system in a rectangular cavity. International Journal of Heat and Mass Transfer 27 (1), 113–123.

Gebhart, B., Mollendorf, J., 1977. A new density relation for pure and saline water. Deep Sea Res. 24, 831–848.

George, P. L., Borouchaki, H., 1998. Delaunay triangulation and meshing. Hermès, Paris.

Giangi, F., Stella, M., 2000. Melting of a pure metal on a vertical wall: numerical simulation. Numerical Heat Transfer: Part A: Applications 38 (2), 193–208.

Giangi, M., Kowalewski, T. A., Stella, F., Leonardi, E., 2000. Natural convection during ice formation: numerical simulation vs. experimental results. Computer Assisted Mechanics and Engineering Sciences 7, 321–342.

Girault, V., Raviart, P.-A., 1986. Finite element methods for Navier-Stokes equations. Springer Verlag, Berlin.

Gobin, D., Le Quéré, P., 2000. Melting from an isothermal vertical wall. synthesis of numerical comparison exercise. Computer Assisted Mechanics and Engineering Sciences 7 (3), 289–306.

Gong, W., Johannes, K., Kuznik, F., 2015. Numerical simulation of melting with natural convection based on lattice Boltzmann method and performed with cuda enabled gpu. Communications in Computational Physics 17 (5), 1201–1224.

Gupta, S. C., 2000. A moving grid numerical scheme for multi-dimensional solidification with transition temperature range. Computer methods in applied mechanics and engineering 189 (2), 525–544.

Hannoun, N., Alexiades, V., Mai, T. Z., 2003. Resolving the controversy over tin and gallium melting in a rectangular cavity heated from the side. Numerical Heat Transfer: Part B: Fundamentals 44 (3), 253–276.

Hannoun, N., Alexiades, V., Mai, T. Z., 2005. A reference solution for phase change with convection. International Journal for Numerical Methods in Fluids 48 (11), 1283–1308.

Hecht, F., 2012. New developments in Freefem++. Journal of Numerical Mathematics 20, 251–266.

Hecht, F., Mohammadi, B., 1997. Mesh adaptation by metric control for multi-scale phenomena and turbulence. AIAA paper 97, 0859.

Hecht, F., Pironneau, O., Hyaric, A. L., Ohtsuke, K., 2007. FreeFem++ (manual). www.freefem.org.

Kheirabadi, A. C., Groulx, D., 2015. The effect of the mushy-zone constant on simulated phase change heat transfer. In: Proceedings of CHT-15, ICHMT International Symposium on Advances in Computational Heat Transfer, Ichmt Digital Library Online. Begel House Inc.

König-Haagen, A., Franquet, E., Pernot, E., Brüggemann, D., 2017. A comprehensive benchmark of fixed-grid methods for the modeling of melting. International Journal of Thermal Sciences 118, 69–103.

Kowalewski, A., Gobin, D., 2004. Phase change with convection: modelling and validation. Springer.

Kowalewski, T. A., Rebow, M., 1999. Freezing of water in differentially heated cubic cavity. International Journal of Computational Fluid Dynamics 11, 193–210.

Laizet, S., Lamballais, E., 2009. High–order compact schemes for incompressible flows: A simple and efficient method with quasi–spectral accuracy. Journal of Computational Physics 228 (16), 5989–6015.

Le Queré, P., 1991. Accuratte solutions to the square thermally driven cavity at high rayleigh number. Computational Fluids 20, 24–41.

Li, C., 1983. A finite-element front-tracking enthalpy method for Stefan problems. IMA Journal of Numerical Analysis 3, 87–107.

Luo, K., Yao, F.-J., Yi, H.-L., Tan, H.-P., 2015. Lattice Boltzmann simulation of convection melting in complex heat storage systems filled with phase change materials. Applied Thermal Engineering 86, 238–250.

Ma, Z., Zhang, Y., 2006. Solid velocity correction schemes for a temperature transforming model for convection phase change. International Journal For Numerical Methods Heat Fluid Flow 16 (11), 204–225.

Mathura, K., Krishna, D. J., 2017. Influence of mushy zone constant on thermohydraulics of a PCM. Energy Procedia 109, 314 – 321, international Conference on Recent Advancement in Air Conditioning and Refrigeration, RAAR 2016, 10-12 November 2016, Bhubaneswar, India.

Michalek, T., Kowalewski, T. A., 2003. Simulations of the water freezing process - numerical benchmarks. Task Quarterly 7 (3), 389–408.

Moglan, R., 2013. Modeling and numerical simulation of flow and heat phenomena in a telecommunication outdoor cabinet. Ph.D. thesis, Université de Rouen Normandy.

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

Morgan, K., Lewis, R., Zienkiewicz, O., 1978. An improved algrorithm for heat conduction problems with phase change. International Journal for Numerical Methods in Engineering 12, 1191–1195.

Nourgaliev, R., Luo, H., Weston, B., Anderson, A., Schofield, S., Dunn, T., Delplanque, J.-P., 2016. Fully-implicit orthogonal reconstructed discontinuous galerkin method for fluid dynamics with phase change. Journal of Computational Physics 305, 964–996.

Okada, M., 1984. Analysis of heat transfer during melting from a vertical wall. International Journal of Heat and Mass Transfer 27, 2057–2066.

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

Rady, M. A., Mohanty, A. K., 1996. Natural convection during melting and solidification of pure metals in a cavity. Numerical Heat Transfer, Part A Applications 29 (1), 49–63.

Rakotondrandisa, A., Danaila, I., Danaila, L., 2019. Numerical modelling of a melting-solidification cycle of a phase-change material with complete or partial melting. International Journal of Heat and Fluid Flow 76, 57 – 71.

Roache, P. J., 1998. Verification and Validation in Computational Science and Engineering. Hermosa Publishers.

Rose, M. E., 1960. A method for calculating solutions of parabolic equations with a free boundary.

Mathematics of Computation, 249–256.
Scanlon, T. J., Stickland, M. T., 2004. A numerical analysis of buoyancy-driven melting and freezing. International Journal of Heat and Mass Transfer 47, 429–436.
Shih, T. M., Tan, C. H., Hwang, B. C., 1989. Effect of grid staggering on numerical schemes. International Journal for Numerical Methods in Fluids 9 (2), 193–212.
Shyy, W., Udaykurnar, H., Rao, M. M., Smith, R., 1996. Computational fluid dynamics with moving boundaries. Taylor & Francis.
Singer-Loginova, I., Singer, H. M., 2008. The phase field technique for modeling multiphase materials. Reports on progress in physics 71 (10), 106501.
Sparrow, E. M., Patankar, P. V., Ramadhyani, S., 1977. Analysis of melting in the presence of natural convection in the melt region. Journal of Heat Transfer 99 (4), 520–526.
Stella, F., Giangi, M., 2004. Modelling methodologies for convection-diffusion phase-change problems. In: Kowalewski, A., Gobin, D. (Eds.), Phase change with convection: modelling and validation. Springer, pp. 219–272.
Swaminathan, C. R., Voller, V. R., 1997. Towards a general numerical scheme for solidification systems. International journal of heat and mass transfer 40 (12), 2859–2868.
Temam, R., 1983. Navier-Stokes equations and nonlinear functional analysis. SIAM, Philadelphia.
Tenchev, R. T., Mackenzie, J. A., Scanlon, T. J., Stickland, M. T., 2005. Finite element moving mesh analysis of phase change problems with natural convection. International Journal of Heat and Fluid Flow 26 (4), 597–612.
Unverdi, S. O., Tryggvason, G., 1992. A front-tracking method for viscous incompressible, multifluid flows. Journal of Computational Physics 100, 25–37.
Vergez, G., Danaila, I., Auliac, S., Hecht, F., 2016. A finite-element toolbox for the stationary Gross-Pitaevskii equation with rotation. Comput. Phys. Comm. 209, 144–162.
Voller, V. R., 1996. An overview of numerical methods for solving phase change problems. In: Minkowycz, W. J., Sparrow, E. M. (Eds.), Advances in Numerical Heat Transfer. Taylor & Francis, pp. 341–375.
Voller, V. R., Cross, M., Markatos, N. C., 1987. An enthalpy method for convection/diffusion phase change. Int. J. Numer. Meth. Eng. 24, 271–284.
Voller, V. R., Prakash, C., 1987. A fixed grid numerical modelling methodology for convection-diffusion mushy region phase-change problems. International Journal of Heat and Mass Transfer 30, 1709–1719.
Wang, S., Faghri, A., Bergman, T. L., 2010a. A comprehensive numerical model for melting with natural convection. International Journal of Heat and Mass Transfer 53, 1986–2000.
Wang, S., Faghri, A., Bergman, T. L., 2010b. Numerical modeling of alternate melting and solidification. Numerical Heat Transfer, Part B: Fundamentals 58 (6), 393–418.
Woodfield, J., Alvarez, M., Gamez-Vargas, B., Ruiz-Baier, R., 2019. Stability and finite element approximation of phase change models for natural convection in porous media. Journal of Computational and Applied Mathematics 360, 117 – 137.
Zhang, Y., Danaila, I., 2013. Existence and numerical modelling of vortex rings with elliptic boundaries. Applied Mathematical Modelling 37, 4809–4824.
Zimmerman, A., Kowalski, J., 2018. Monolithic simulation of convection-coupled phase-change: Verification and reproducibility. In: Schäfer, M., Behr, M., Mehl, M., Wohlmuth, B. (Eds.), Recent Advances in Computational Engineering. ICCE 2017. Lecture Notes in Computational Science and Engineering, vol 124. Springer, pp. 177–197.