



HAL
open science

The eXtreme Mesh deformation approach (X-MESH) for the Stefan phase change model

Nicolas Moes, Jean-François Remacle, Jonathan Lambrechts, Benoît Lé,
Nicolas Chevaugéon

► **To cite this version:**

Nicolas Moes, Jean-François Remacle, Jonathan Lambrechts, Benoît Lé, Nicolas Chevaugéon. The eXtreme Mesh deformation approach (X-MESH) for the Stefan phase change model. *Journal of Computational Physics*, 2023, 477, pp.111878. 10.1016/j.jcp.2022.111878 . hal-04495547

HAL Id: hal-04495547

<https://hal.science/hal-04495547>

Submitted on 8 Mar 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

The eXtreme Mesh deformation approach (X-MESH) for the Stefan phase change model

Nicolas Moës^{a,c}, Jean-François Remacle^b, Jonathan Lambrechts^b, Benoît Lé^a, Nicolas Chevaugeon^a

^a *Nantes Université, Ecole Centrale de Nantes, GeM Institute, UMR CNRS 6183, 1 rue de la Noë, 44321 Nantes, France.*

^b *Université Catholique de Louvain
Institute of Mechanics, Materials and Civil Engineering
1348 Louvain-la-Neuve, Belgium*

^c *Institut Universitaire de France (IUF)*

Abstract

The eXtreme Mesh deformation approach (X-MESH) is a new paradigm to follow sharp interfaces without remeshing and without changing the mesh topology. Even though the mesh does not change its topology, it can follow interfaces that do change their topology (nucleation, coalescence, splitting) and that possibly travel over long distances. To make this possible, the key X-MESH idea is to allow elements to reach zero measure. This permits interface relaying between nodes as well as interface annihilation and seeding in a time continuous manner. The paper targets the Stefan phase change model in which the interface (front) is at a given temperature. Several examples demonstrate the capability of the approach.

Keywords: X-MESH, Front relaying, Sharp interface, Phase change, Stefan model, Solidification

1. Introduction

The Stefan model is a classical phase change model in which motion is neglected: fluid and solid are at rest (expansion due to freezing is neglected). The phase change takes place at a given temperature (273.15K for water-ice)

Email addresses: nicolas.moes@ec-nantes.fr (Nicolas Moës),
jean-francois.remacle@uclouvain.be (Jean-François Remacle),
jonathan.lambrechts@uclouvain.be (Jonathan Lambrechts), benoit.le@ec-nantes.fr
(Benoît Lé), nicolas.chevaugeon@ec-nantes.fr (Nicolas Chevaugeon)

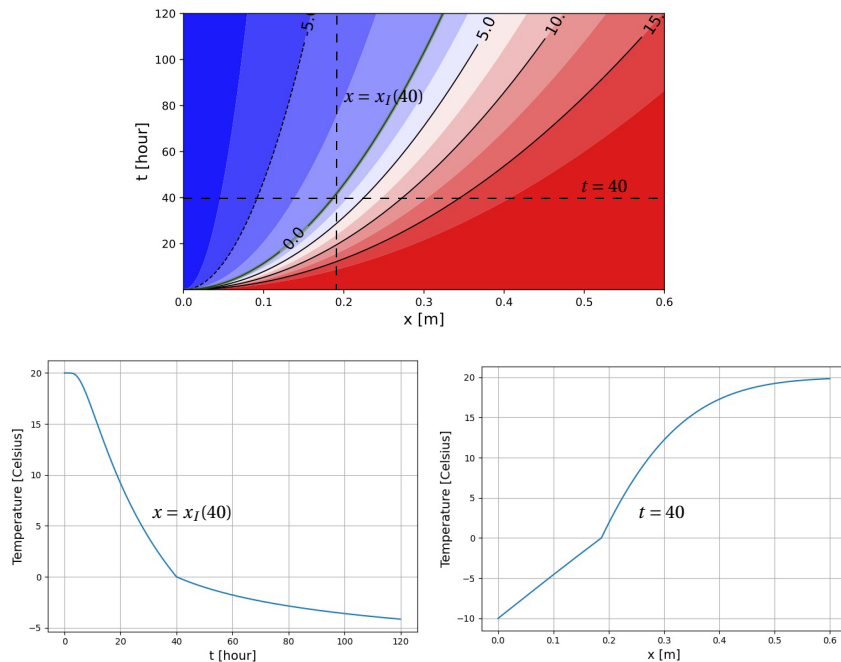


Figure 1: Temperature field for a Stefan problem.

and the free energy is continuous across the front. The internal energy, on the contrary, is discontinuous across the front and its jump defines the latent heat. It is the amount of energy released by the fluid as it becomes solid. The heat flux is also discontinuous across the front. Its jump is equal to the product of the latent heat and the front velocity. An extensive description of the Stefan model may be found in the book by Gupta [1] or the more recent paper [2] targeted towards control. An important seminal work on the mathematical description of the Stefan model can also be found in [3].

The temperature field for a one-dimensional Stefan model of the freezing of water is illustrated on Figure 1. The liquid occupies the region $0 \leq x < \infty$. We look for a temperature field $T(x, t)$ that is initially at a temperature $T(x, 0) = T_i > T_0$ where T_0 is the temperature of phase transition. For all $t > 0$, the left side $x = 0$ of the domain is cooled: $T(0, t) = T_c < T_0$. Solidification begins immediately and the solid phase occupies a region $0 \leq x \leq x_I(t)$ where $x_I(t)$ is the position of the ice front at time t : $T(x_I(t), t) = T_0$. Figure 1 shows the solution $T(x, t)$ for physically sound ice and water parameters ($T_0 = 0^\circ\text{C}$). Even though the temperature $T(x, t)$ is continuous, both its time and space derivatives are discontinuous across the front. The derivative

jumps are linked with the interface normal velocity, v , to ensure temperature continuity. More precisely, we have on the front

$$[[T_{,t}]] + v[[T_{,x}]] = 0,$$

where $T_{,t}$ and $T_{,x}$ indicates the time and space partial derivatives, respectively and $[[\cdot]]$ is the jump symbol.

The Stefan model addresses the stable solidification of a liquid initially at a temperature above the solid-liquid equilibrium temperature. In this model, surface tension and kinetic mobility are neglected [4]. To model unstable processes as dendritic growth, the fixed front temperature must be replaced by the so-called Gibbs-Thomson condition. This relation links the front temperature to its curvature and speed. There also exist anisotropic versions of the Gibbs-Thomson model which take into account the variation of the surface tension and kinetic mobility with respect to the interface orientation.

The Stefan model is known to lead to possible blow-ups in the solution when the initial temperature profile corresponds to an undercooled liquid or a superheated solid [5, 6]. We will not be considering these type of profiles here.

Numerical schemes to address the Stefan model may be classified into tracking and capturing schemes. In the first category, the mesh moves with the interface. This is the case for the Arbitrary Lagrangian Eulerian (ALE) approach which was first developed to follow material interfaces in flows [7–11]. These techniques were then adapted to immaterial interface for phase-change models [12–14].

Unfortunately ALE is not able to handle topological changes of the front and requires remeshing when the mesh is too distorted. Remeshing requires projection of the solution between successive meshes which is highly detrimental to the continuity of the solution in time. Capturing approaches use a fixed mesh. This is the case for the level set approach [15–17] and the extended finite element method [18–20].

This work starts from an observation: none of the methods proposed in the state of the art is entirely satisfactory. We therefore decided to start by listing 4 fundamental properties that would allow us to design a totally satisfactory scheme:

- P1 - Time continuous deforming mesh with fixed topology:** the mesh deforms in time in a continuous manner: nodes trajectories form continuous paths and the mesh adjacencies are fixed.
- P2 - Conforming mesh to sharp physical interfaces.**

P3 - Compliant with topological changes: nucleation, collapse, coalescence and splitting.

P4 - Classical finite element approximation: no enrichment with additional degrees of freedom.

Properties P1, P2 and P4 ensures that the true solution at the nodes evolve smoothly over time which provides benefits for the accuracy of the approximation scheme. Matching the 4 properties simultaneously is currently not possible and very challenging. Up to now, property P3 has always been considered as incompatible with the others: large changes of front topologies have always been concomitant with mesh adaptation or discrete solution enrichment which is incompatible with property P4. After a lot of thought, we came with a striking conclusion. The quest for meshes with nice-only element is counter-productive: bad quality elements or even elements *with a zero measure* opens a new spectrum of possibilities.

We consider in this paper *a new paradigm for mesh movements* allowing elements to reach zero measure at some instant in their evolution. This extreme mesh deformation (X-MESH) allows an interface to be relayed from one node to another in a continuous fashion. It also allows interface annihilation or seeding. Moreover, no remeshing is needed and the mesh topology is kept fixed. Only node movements are needed. Over any time-step, the relation between the field change and mesh movement is written in an arbitrary Lagrangian Eulerian format. The key aspect of the X-MESH is that it keeps the same mesh throughout the simulation whereas traditional ALE simulations do require remeshing for large interface movement and/or topological changes.

Note that for different applications than phase-change, other works in the literature allow extreme/degenerate mesh deformations to account for large mesh deformations. For instance a high-order discontinuous Galerkin method with unstructured space-time meshes for two-dimensional compressible flows on domains with large deformations was considered in [21] and high order direct ALE schemes on moving Voronoi meshes with topology changes were introduced in [22, 23]. Large boundary displacements were also considered with ALE scheme for unsteady inviscid flows in [24]. Finally, it is worth noting the work of Springel in similar directions [25].

The paper is organized as follows. Section 2 describes the equation of the Stefan model both in strong and weak forms. Then it details both time and space discretization schemes as well as the nonlinear iterative algorithm. Section 3 describes the X-MESH front relaying process through which the mesh is always matching the front. In Section 4, a set of 2D simulations are

presented. The straight front semi-infinite two-phase problem is considered as well as the axisymmetric case. Both cases are considered with and without latent heat. Finally, the freezing of a fluid subjected to two rotating heat sinks is considered.

2. Two-phase Stefan model

We consider a domain Ω depicted in Figure 2. Its boundary, $\partial\Omega$, is assumed fixed in time. The domain is composed of the solid and liquid phases denoted Ω_s and Ω_ℓ , respectively. The front separating the phases, Γ , moves with a velocity denoted \mathbf{v} .

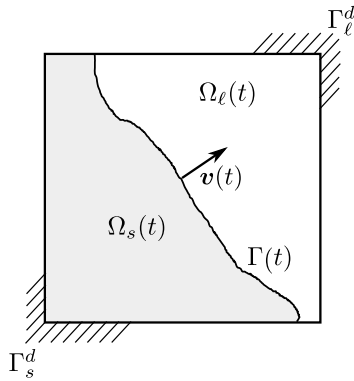


Figure 2: Problem definition.

We seek to determine the evolution of the temperature field $T(\mathbf{x}, t)$ [K] for $(\mathbf{x}, t) \in \Omega \times [0, t_{\max}]$, which satisfies the governing equations

$$\begin{aligned}\rho_s c_s T_{,t} &= \nabla(k_s \nabla T) + \rho_s S \text{ in } \Omega_s, \\ \rho_\ell c_\ell T_{,t} &= \nabla(k_\ell \nabla T) + \rho_\ell S \text{ in } \Omega_\ell,\end{aligned}$$

where ρ_s and ρ_ℓ are the densities [$\text{kg}\cdot\text{m}^{-3}$], c_s and c_ℓ the specific heat capacities [$\text{J}\cdot\text{K}^{-1}\cdot\text{kg}^{-1}$], k_s and k_ℓ the isotropic thermal conductivities [$\text{W}\cdot\text{m}^{-1}\cdot\text{K}^{-1}$] of the phases and S a heat source [$\text{W}\cdot\text{kg}^{-1}$].

Note that to ensure mass conservation within the Stefan model, densities must be considered equal ($\rho_s = \rho_\ell = \rho$). The temperature field must also satisfy the following boundary and initial conditions:

$$T(\mathbf{x}, t) = T_d, \quad (\mathbf{x}, t) \in \Gamma^d \times [0, t_{\max}], \quad (1a)$$

$$T(\mathbf{x}, 0) = T_i(\mathbf{x}), \quad \mathbf{x} \in \Omega. \quad (1b)$$

Here, T_d is the temperature imposed on the Dirichlet boundary $\Gamma^d = \Gamma_s^d \cup \Gamma_\ell^d$. To simplify the presentation but without loss of generality, the imposed temperature is assumed constant and uniform over Γ^d . The location of Γ^d is also assumed fixed in time. The remainder of the external boundary ($\partial\Omega \setminus \Gamma^d$) is insulated (zero normal temperature gradient).

At any instant t , the location of the phase change front $\Gamma(t)$ is defined by the set of point whose temperature is at the transition temperature

$$\Gamma(t) = \{\mathbf{x} \in \Omega : T(\mathbf{x}, t) = T_0\}. \quad (2)$$

On the phase change front, the heat flux jump is related to the front speed by

$$(k_s \nabla T - k_\ell \nabla T) \cdot \mathbf{n} = \rho l (\mathbf{v} \cdot \mathbf{n}) \text{ on } \Gamma, \quad (3)$$

where \mathbf{n} is the normal to Γ moving at velocity \mathbf{v} and l is the latent heat [$\text{J} \cdot \text{kg}^{-1}$].

A sketch of the temperature profile close to a front in the Stefan model is depicted in Figure 3. In the absence of volumetric heat sources, the profile is concave (convex) in the case of freezing (melting). The direction of the heat flux is always from the fluid to the solid. Its magnitude is lower (higher) in the liquid in the case of freezing (melting). The heat flux jump corresponds to the latent heat released by the fluid in the case of freezing or absorbed by the solid as it becomes a fluid in the case of melting.

2.1. Variational formulation

In order to express the variational format of the equations, it is convenient to introduce the specific internal energy [$\text{J} \cdot \text{kg}^{-1}$]:

$$e(T) = \begin{cases} c_s T, & \text{if } T \leq T_0, \\ c_\ell T + \bar{l}, & \text{else.} \end{cases}$$

It is discontinuous at the transition temperature and the jump is the latent heat l :

$$l = (c_\ell - c_s)T_0 + \bar{l}.$$

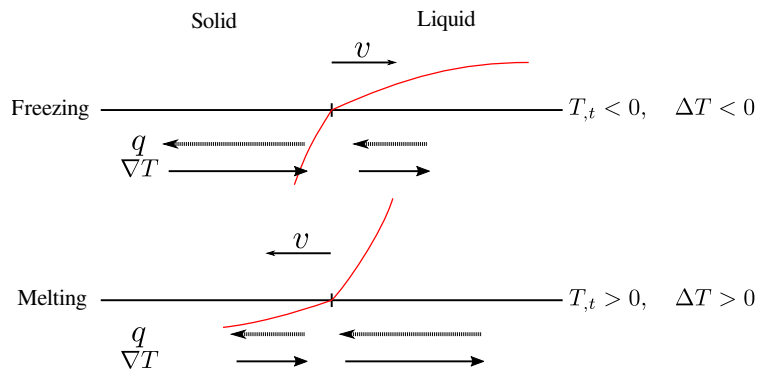


Figure 3: Sketch of the temperature profile across a freezing or melting front in the Stefan model (in the absence of volumetric heat sources). The vector q indicates the heat flux.

Finally, it is also convenient to define the conductivity as a function of the temperature. It is also discontinuous at the transition:

$$k(T) = \begin{cases} k_s, & \text{if } T \leq T_0, \\ k_\ell, & \text{else.} \end{cases}$$

We introduce the Arbitrary Lagrangian Eulerian frame of reference (see for instance [7, 26]). A mapping maps the points \mathbf{X}^0 of a reference domain Ω^0 to the points \mathbf{X} of the domain Ω^t :

$$\mathbf{X} = \mathbf{X}(\mathbf{X}^0, t). \quad (4)$$

The mapping is assumed not to change the shape of the domain. The domain velocity \mathbf{w} , domain deformation gradient \mathbf{F} and Jacobian of the transformation are given by

$$\mathbf{w} = \left. \frac{\partial \mathbf{X}}{\partial t} \right|_{\mathbf{X}^0}, \quad \mathbf{F} = \left. \frac{\partial \mathbf{X}}{\partial \mathbf{X}^0} \right|_t, \quad J = \det \mathbf{F}. \quad (5)$$

The admissible temperature set is defined by

$$\mathcal{U}_d = \{T : \Omega^0 \rightarrow \mathbb{R} : T \in H^1(\Omega^0), T = T_d \text{ on } \Gamma^d\}.$$

At any instant $t \in [0, t_{\max}]$, the temperature field T must belong to the admissible set \mathcal{U}_d and satisfy the following variational principle:

$$\begin{aligned} & \frac{d}{dt} \int_{\Omega^0} \rho e(T) T^* J \, d\mathbf{X}^0 + \int_{\Omega^0} k(T) \mathbf{g}(T) \cdot \mathbf{g}(T^*) J \, d\mathbf{X}^0 + \\ & \int_{\Omega^0} \rho e(T) \mathbf{w} \cdot \mathbf{g}(T^*) J \, d\mathbf{X}^0 = \int_{\Omega^0} \rho S T^* J \, d\mathbf{X}^0, \quad \forall T^* \in \mathcal{U}, \end{aligned} \quad (6)$$

where the test function space is defined by

$$\mathcal{U} = \{T : \Omega^0 \rightarrow \mathbb{R} : T \in H^1(\Omega^0), T = 0 \text{ on } \Gamma^d\}$$

and

$$\mathbf{g}(T) = \mathbf{F}^{-1} \nabla_0 T. \quad (7)$$

For the sake clarity, we also give the above expression integrated on the current domain

$$\begin{aligned} & \frac{d}{dt} \int_{\Omega^t} \rho e(T) T^* \, d\mathbf{X} + \int_{\Omega^t} k(T) \nabla T \cdot \nabla T^* \, d\mathbf{X} + \\ & \int_{\Omega^t} \rho e(T) \mathbf{w} \cdot \nabla T^* \, d\mathbf{X} = \int_{\Omega^t} \rho S T^* \, d\mathbf{X}. \end{aligned}$$

The variational principle (6) gathers all the equations of the Stefan model. Both the internal energy e and conductivity functions are discontinuous at the transition temperature T_0 . This temperature defines the location of the interface $\Gamma(t)$ between the solid and liquid phases. Finally, contrary to classical tracking approaches, we stress the fact that with the X-MESH the domain velocity \mathbf{w} on the front may be different from the physical velocity \mathbf{v} , even in the normal direction.

2.2. Time discretization

We consider a set of discrete instants. Let t^n be such an instant and t^{n+1} the next one. We proceed with a θ -scheme to discretize (6). Given the temperature field T^n at t^n , we seek the temperature field T^{n+1} at t^{n+1} belonging to \mathcal{U}_d and satisfying:

$$\begin{aligned}
r(T^*) &\equiv \int_{\Omega^0} \rho e(T^{n+1}) T^* J^{n+1} \, d\mathbf{X}^0 - \int_{\Omega^0} \rho e(T^n) T^* J^n \, d\mathbf{X}^0 \\
&\quad + \theta \Delta t^{n+1/2} \int_{\Omega^0} k(T^{n+1}) \mathbf{g}^{n+1}(T^{n+1}) \cdot \mathbf{g}^{n+1}(T^*) J^{n+1} \, d\mathbf{X}^0 \\
&\quad + \theta \Delta t^{n+1/2} \int_{\Omega^0} \rho e(T^{n+1}) \mathbf{w}^{n+1/2} \cdot \mathbf{g}^{n+1}(T^*) J^{n+1} \, d\mathbf{X}^0 \\
&\quad - \theta \Delta t^{n+1/2} \int_{\Omega^0} \rho S^{n+1} T^* J^{n+1} \, d\mathbf{X}^0 \\
&\quad + (1 - \theta) \Delta t^{n+1/2} \int_{\Omega^0} k(T^n) \mathbf{g}^n(T^n) \cdot \mathbf{g}^n(T^*) J^n \, d\mathbf{X}^0 \\
&\quad + (1 - \theta) \Delta t^{n+1/2} \int_{\Omega^0} \rho e(T^n) \mathbf{w}^{n+1/2} \cdot \mathbf{g}^n(T^*) J^n \, d\mathbf{X}^0 \\
&\quad - (1 - \theta) \Delta t^{n+1/2} \int_{\Omega^0} \rho S^n T^* J^n \, d\mathbf{X}^0 \\
&= 0, \quad \forall T^* \in \mathcal{U},
\end{aligned} \tag{8}$$

where $\Delta t^{n+1/2} = t^{n+1} - t^n$ and $\mathbf{w}^{n+1/2}$ is the domain velocity discretized as:

$$\mathbf{w}^{n+1/2}(\mathbf{X}^0) = \frac{\mathbf{X}^{n+1}(\mathbf{X}^0) - \mathbf{X}^n(\mathbf{X}^0)}{t^{n+1} - t^n}.$$

2.3. Space discretization

The reference domain Ω^0 is partitioned into a triangular mesh \mathcal{M}^0 . The mesh evolves in time. It is denoted \mathcal{M}^n at some later time t^n . In the X-MESH approach, the mesh topology (number of nodes and adjacencies between mesh entities) is kept fixed. The mesh evolution is only due to node movements.

The set of nodes is denoted \mathcal{N} . The discrete temperature fields at t^n and t^{n+1} are defined as:

$$T^n(\mathbf{X}^0) = \sum_{i \in \mathcal{N}} T_i^n T_i^*(\mathbf{X}^0), \quad T^{n+1}(\mathbf{X}^0) = \sum_{i \in \mathcal{N}} T_i^{n+1} T_i^*(\mathbf{X}^0),$$

where the scalars T_i^n and T_i^{n+1} designate the nodal temperature values at node i while the function T_i^* is the corresponding finite element approximation function (hat function). When entered in (8), the approximation function associated to node i gives a residual denoted r_i :

$$r_i \equiv r(T_i^*).$$

The above residual must be zero for all nodes with a free temperature. These nodes are the ones not belonging to the Dirichlet boundary. The zero residual condition links the mesh and temperature fields at times t_n and t_{n+1} . We express this condition with the R symbol:

$$R((T^{n+1}, \mathbf{X}^{n+1}), (T^n, \mathbf{X}^n)) = 0. \quad (9)$$

Given a mesh at time t^n and a temperature field defined on this mesh, we say that they are compatible if the transition temperature value may only be found at nodes and not in between nodes. This implies that the iso-temperature $T^n(\mathbf{x}) = T_0$ is exactly represented by the mesh: it explicitly appears as mesh edges of the triangular mesh. We denote this mesh/temperature compatibility with the C symbol. Formally, the X-MESH problem to solve over a time step is

$$\begin{aligned} \text{Given } (T^n, \mathbf{X}^n) \in C, \quad \text{find } (T^{n+1}, \mathbf{X}^{n+1}) \in C \\ \text{such that } R((T^{n+1}, \mathbf{X}^{n+1}), (T^n, \mathbf{X}^n)) = 0. \end{aligned} \quad (10)$$

The process is initialized with a compatible couple: $(T^0, \mathbf{X}^0) \in C$. From one time instant to the next, the temperature field evolves and the mesh deforms so that the iso-transition temperature is always part of the mesh. It is important to note that this does not force the same nodes to be part of the front from one time step to the next. During a time step, a node may stay on the front, leave the front or join the front. This is why we stressed the fact in the description of the variational formulation that on the front, the domain (mesh) velocity \mathbf{w} could be different from the physical front velocity \mathbf{v} .

Problem (10) does not have a unique solution (nor shall we try to prove here that the solution exists). The non-uniqueness means that we have some freedom in the mesh evolution. We shall use this freedom to have the mesh return to its initial position progressively after the front has gone away (mesh relaxation).

2.4. A quasi-Newton scheme

We detail an iterative process to solve (10). We consider first the issue of finding the temperature field T^{n+1} using (9) assuming given T^n , \mathbf{X}^n and \mathbf{X}^{n+1} . A direct use of a Newton-Raphson resolution would require to compute the derivative of e and k with respect to the temperature. But those are discontinuous across the interface. Indeed, a Heaviside function H centered at T_0 is involved:

$$\begin{aligned} e(T) &= ((1 - H(T))c_s + H(T)c_\ell) T + H(T)\bar{l}, \\ k(T) &= (1 - H(T))k_s + H(T)k_\ell, \end{aligned}$$

where

$$H(T) = \begin{cases} 0, & \text{if } T \leq T_0, \\ 1, & \text{if } T > T_0. \end{cases}$$

Even if it would be theoretically possible to consider calculating the derivative of the Heaviside as Dirac distributions on the interface, using such an approach poses a problem when an interface nucleates. Indeed, such a scheme would not allow to anticipate the creation of a front.

A more robust approach is to smooth the discontinuous quantities across the interface: such a smoothed approach is widely used in the literature but its main drawback is accuracy. If the interface is smoothed over a thickness ε , the convergence to the exact solution of the Stefan problem cannot be better than $\sqrt{\varepsilon}$ [27]. Here, we propose a quasi-Newton type scheme where the tangent matrix is an approximation of the derivative of the residual while the true sharp residual is kept. The tangent matrix is computed as the derivative of a smoothed residual. For this purpose, we define a smooth step function H^{reg} illustrated on Figure 4:

$$H^{\text{reg}}(T) = \begin{cases} 0, & \text{if } T < T_0 - \delta/2, \\ \frac{T - (T_0 - \delta/2)}{\delta}, & \text{if } T_0 - \delta/2 \leq T \leq T_0 + \delta/2, \\ 1, & \text{if } T > T_0 + \delta/2, \end{cases}$$

where δ [K] is a numerical parameter. We checked that parameter δ has no influence on the converged results but has some slight influence on the convergence rate of the non linear process. From this function, we can define continuous energy and conductivity functions:

$$\begin{aligned} e^{\text{reg}}(T) &= ((1 - H^{\text{reg}}(T))c_s + H^{\text{reg}}(T)c_\ell) T + H^{\text{reg}}(T)\bar{l}, \\ k^{\text{reg}}(T) &= (1 - H^{\text{reg}}(T))k_s + H^{\text{reg}}(T)k_\ell. \end{aligned}$$

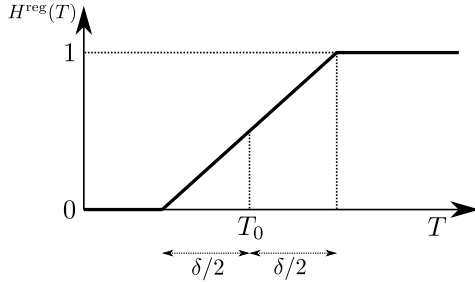


Figure 4: Smooth step function $\alpha(T)$ illustration.

We denote by r_i^{reg} the residual expression in which the sharp functions $e(T)$ and $k(T)$ are replaced by the regularized ones. Quasi-Newton iterations are denoted by the k index. Note that other smoothing functions could be used, see for instance [17].

The staggered solution procedure for a step is detailed in the Algorithm 1. After initialization, the temperature update (quasi-Newton step) alternates with the mesh update (projection step) until convergence. To be precise the projection step also modifies the temperature field but only for the nodes located on the front after the projection.

The initialization step copies the previously converged temperature field and, for nodes not located on the front, relaxes their location with the following formula:

$$\mathbf{X}^{n+1,0} = 0.9\mathbf{X}^n + 0.1\mathbf{X}^0. \quad (11)$$

The initial couple $(T^{n+1,0}, \mathbf{X}^{n+1,0})$ belongs to C . As explained earlier, the Newton step uses a smoothed gradient operator but with the sharp residual. The projection step on C will be detailed in the next section. The idea is to move nodes close to the iso- T_0 on the iso- T_0 and then set the temperature at these nodes to T_0 .

One could think of a simpler scheme than the one described in Algorithm 1 that we shall call fixed point. In this scheme, the temperature at the centroid of each element obtained from $(T^{n+1,k}, \mathbf{X}^{n+1,k})$, and denoted T_{elt} , is used to decide the phase for the whole element. The expression of the internal energy and conductivity are then for a given element

$$\begin{aligned} e^{\text{fix}}(T) &= ((1 - H(T_{\text{elt}}))c_s + H(T_{\text{elt}})c_\ell) T + H(T_{\text{elt}})\bar{l}, \\ k^{\text{fix}} &= (1 - H(T_{\text{elt}}))k_s + H(T_{\text{elt}})k_\ell. \end{aligned}$$

With these expressions, the residual is now a linear expression of T and setting it to zero is equivalent to solving a linear system. With the fixed point

Algorithm 1: The X-MESH solution procedure for a time step. The Einstein summation convention rule holds with indices i and j spanning all nodes except the ones on the Dirichlet boundary. The symbol ϵ [K] denotes the user given tolerance for convergence. The field Δ is a linear interpolation of the Δ_i nodal values (except for the Dirichlet nodes on which a zero value is set).

1 Step Initialization:

$$k = 0 \text{ and } (T^{n+1,0}, \mathbf{X}^{n+1,0}) = \text{relax_in_C}(T^n, \mathbf{X}^n)$$

2 do

3 Quasi-Newton step:

$$A_{ij} = \left. \frac{dr_i^{\text{reg}}}{dT_j^{n+1}} \right|_{(T^{n+1,k}, \mathbf{X}^{n+1,k})}$$

$$A_{ij} \Delta_j = r_j |_{(T^{n+1,k}, \mathbf{X}^{n+1,k})}$$

$$T'_i = T_i^{n+1,k} + \Delta_i$$

4 Projection step:

$$(T^{n+1,k+1}, \mathbf{X}^{n+1,k+1}) = \text{project_on_C}(\mathbf{X}^{n+1,k}, T')$$

5 Stopping criteria:

$$\text{err} = \left(\frac{\int_{\Omega^0} \Delta^2 J^{n+1,k+1} d\mathbf{X}^0}{\int_{\Omega^0} d\mathbf{X}^0} \right)^{1/2}$$

6 if $\text{err} < \epsilon$ then

7 $(T^{n+1}, \mathbf{X}^{n+1}) = (T^{n+1,k+1}, \mathbf{X}^{n+1,k+1})$

8 else

9 $(T^{n+1,k}, \mathbf{X}^{n+1,k}) \leftarrow (T^{n+1,k+1}, \mathbf{X}^{n+1,k+1})$

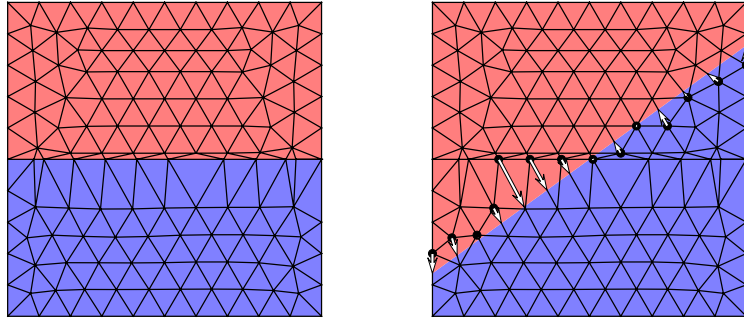
10 $k \leftarrow k + 1$

11 while $\text{err} > \epsilon$;

algorithm, the step iterates between a linear system solve and a projection on C . The strength of the fixed point algorithm is its simplicity. Unfortunately, except for zero latent heat, this algorithm is not robust.

3. Front relaying

(a) Mesh and function at $t = 0$ (b) Active nodes at $t = \pi/5$



(c) Mesh and function at $t = \pi/5$ (d) Relaxation

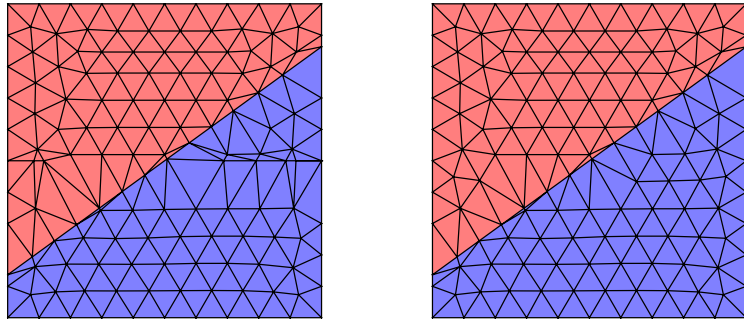


Figure 5: Illustration of the relaying algorithm

A very important ingredient of X-MESH concerns the deformation of the mesh. We want to develop a front tracking method where the connectivity of the mesh remains unchanged but where topological changes of the interfaces are possible. For this, we propose an original method that we call relaying, by analogy with the passage of the baton in a relay race.

To explain relaying, we will use a manufactured example: we consider the function

$$f(x, y, t) = y \cos(t) + x \sin(t)$$

and we ask a mesh to track the iso zero

$$y = \tan(t) x$$

of this function at any time t . The considered domain is a unit square centered at the origin.

At any time t , the mesh is supposed to represent exactly the iso-zero: Figure 5(a) shows the mesh at $t = 0$. This adaptation has been realized with the current algorithm, starting from a standard triangular mesh generated by gmsh [28]. The red color represents the zone where $f(x, y, t) > 0$ while blue color represents the zone where $f < 0$. Figure 5(b) shows function f at time $t = \pi/5$. Nodes colored in black are called active nodes. We define two subsets of the set of nodes of the mesh:

1. Subset S_1 contains both the nodes that were on the front at the previous timestep (here at $t = 0$) and the nodes that have changed of color at current timestep (here $t = \pi/5$) – going from red to blue or vice and versa.
2. Subset S_2 contain all the nodes that belong to the edges that are crossed by the interface at current timestep (here the iso-zero of $f(x, y, \pi/5)$).

The set S of active nodes (in black dots on Figure 5(b)) belong to the intersection $S = S_1 \cap S_2$ of those two sets. Active nodes will move towards the front at current timestep. Choosing active nodes in S_1 ensures that only the nodes upstream of the front move towards it, thus ensuring the relaying. Choosing active nodes in S_2 ensures that every edge crossed by the interface will have at least one of its nodes going towards the front.

Another design choice related to the relaying algorithm is related to moving the active nodes towards the front. We decided here to move the nodes *along the existing edges of the mesh*. Looking at Figure 5(b), we can see that, most of the time, several edges are possible: we choose to use the shortest one, except on the boundaries of the domain where the boundary edge is privileged.

The relaying process is extremely simple and powerful: it handles automatically front nucleation, annihilation and coalescence. In the nucleation case, the active nodes correspond only to nodes for which the sign of f changes since there was no previous nodes on the front. These active nodes will then move to target points corresponding to the nucleation boundary. Annihilation is the inverse process to nucleation in which the front becomes a small closed contour and eventually vanishes. In this case, there are no targets since $f \neq 0$ and thus the front becomes also empty since all active nodes are removed from the front.

Finally, coalescence is also automatic for about the same reasons as annihilation. In the coalescence zone, there are no longer any targets and active nodes are removed from the front.

We preferred to move the nodes along the edges compared to, for example, projecting them on the front. We have one very good reason for doing that. Relaying is very powerful, but it has a price: the necessity of creating arbitrarily small elements, or even *elements of zero measure*.

In [29], authors prove that a sufficient condition for having convergent finite element solution when the mesh is refined is that angles should be bounded from above and thus forbid zero- or obtuse nearly zero-measure elements. Mesh algorithms are thus supposed to try their best to generate *quality meshes* in that sense [30].

Babuška’s conclusion has been mitigated since. In [31], it is shown that there exist meshes that do not respect the angle condition and that converge anyway. Some authors have also wisely noted that the angle condition is only a sufficient condition [32] and that this condition can be largely weakened [33]. In reality, it has been proven that optimally convergent finite element solutions can be obtained with meshes that are *visually unpleasant*. The family of computationally acceptable meshes can thus include patterns of badly shaped elements [34] like isolated caps (a cap is a triangle with an angle close to 180 degrees) but also can consist in more complex structures as bands or clusters. In his paper [33], Pr. Kučera claims that “*one can fabricate very strange triangulations satisfying the assumptions of the theory presented. Such meshes perhaps do not have any value from the practical point of view*”. In our perspective, these “wild” meshes are indeed very useful from the practical point of view and are at the core of what is proposed here.

The presented algorithm does not preclude very small or even inverted element in the course of the iterations. This is quite natural since the front may transition from some nodes to other nodes. It is the matrix coming from the diffusive term which will have the worst conditioning as the element becomes very small because it involves two deformation gradients. To avoid issues with matrix conditioning in the solver, we set a lower threshold to the absolute value of the Jacobian. This value is A_{tol}/A^0 where A_{tol} is an area threshold and A^0 is the area of the element in the initial mesh at $t = 0$. We transform the element Jacobian (see equation (5)) with the following formula

$$J = \max(|J|, A_{\text{tol}}/A^0) \text{sign}(J). \quad (12)$$

which may be viewed as an added volume approach. We found that the sign factor in the above relation was important in practice. In other words, if the

element is inverted, the negative Jacobian sign is kept. We have observed that, keeping the jacobian negative allows Newton’s algorithm to force the mesh to return to a situation where the Jacobian is positive. On the other hand, using $J = \max(|J|, A_{\text{tol}}/A^0)$ makes tangled elements acceptable to the solver and leads to meshes that contain inverted elements at convergence.

Visually unpleasant, yet acceptable patterns in a mesh only harm the conditioning of the finite element matrices [34]. The solution proposed in this paper – to bound the value of the determinant of the Jacobian of the elements from below – is not the most elegant way of controlling this conditioning. We have developed two more rigorous approaches that enable to handle zero measure elements without modifying the Jacobian. This work will be submitted very soon.

Nevertheless, there are some unacceptable mesh patterns: the only really problematic case is the one related to long bands of caps. The relaying algorithm we propose allows to never generate this kind of pattern: moving the nodes along the edges allows us to create only isolated caps, which allows us to ensure the convergence of our computations. On Figure 5(a), we clearly see that most of the small size elements on the front are needles (a needle is a triangular elements having an angle close to 0 degrees) and that only one isolated cap is present.

Figure 5(c) shows the mesh after relocalization of the nodes. The elements downstream of the front have a size that can be up to twice their initial size, which is not ideal. We therefore end our relaying algorithm with a so-called relaxation phase where the nodes that have left the front return to their original position in the initial mesh (see Figure 5(d)).

One would think that any other strategy other than relaying would be the same. Figure 6 shows the evolution of the mesh if, instead of taking the active nodes upstream of the front, we took them downstream. In this case, nodes tend to accumulate downstream of the front, quickly leading to an invalid mesh.

4. Numerical examples

This section begins with a dimensional study of the equations to define useful non-dimensional parameters.

4.1. Non-dimensional form

We will be considering examples of solidification of a liquid domain initially at a temperature T_ℓ . Non-dimensional temperature, coordinates and time

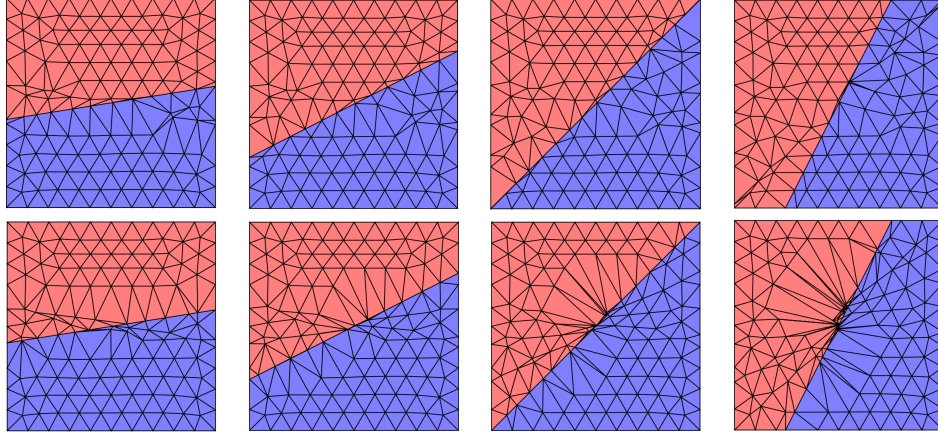


Figure 6: Top figures show evolution of the mesh when active nodes are chosen upstream of the front (relaying) and bottom figure show the same evolution, but when active nodes are chosen downstream of the front.

are denoted by tildas:

$$\tilde{T} = \frac{T - T_0}{T_\ell - T_0}, \quad \tilde{x} = \frac{x}{L_r}, \quad \tilde{t} = \frac{t}{t_r},$$

where L_r and t_r are reference length and time, respectively. We define also the following non-dimensional quantities

$$\kappa = \frac{k_s}{k_\ell}, \quad \alpha = \frac{\alpha_s}{\alpha_\ell}, \quad \gamma = \frac{l}{c_\ell(T_\ell - T_0)}, \quad \tilde{\mathbf{v}} = \frac{t_r}{L_r} \mathbf{v}, \quad \tilde{S} = \frac{\rho S L_r^2}{k_\ell(T_\ell - T_0)}, \quad (13)$$

where we have used the thermal diffusivity of the solid and liquid phase [m^2/s]:

$$\alpha_s = \frac{k_s}{\rho c_s}, \quad \alpha_\ell = \frac{k_\ell}{\rho c_\ell}.$$

Note that $\kappa \leq 1$ and $\alpha \leq 1$. The coefficient γ , usually called the Stefan number, is the ratio between the latent heat and sensible heat. Choosing the following relation between the reference time and length,

$$t_r = \frac{L_r^2}{\alpha_\ell},$$

Property	Units	Symbol	Solid	Liquid	Interface
Density	kg.m ⁻³	ρ	1000	1000	
Specific heat capacity	J.K ⁻¹ .kg ⁻¹	c_s c_ℓ	2090	4185	
Conductivity	W.m ⁻¹ .K ⁻¹	k_s k_ℓ	2.1	0.6	
Latent heat	J.kg ⁻¹	l			3.3×10^5
Phase change temperature	K	T_0			273.15

Table 1: Physical properties for test cases.

Parameter	Units	Symbol	Value
θ -scheme	-	θ	0.5
Regularization temperature	K	δ	8
Minimum allowed area	m ²	A_{tol}	5×10^{-9}
Convergence tolerance	K	ϵ	10^{-5}

Table 2: Numerical parameters for the test cases.

the non-dimensional equations are

$$\begin{aligned} \frac{\tilde{T}_{,\tilde{t}}}{\alpha} &= \tilde{\Delta}\tilde{T} + \frac{\tilde{S}}{\kappa} \quad (\text{solid phase}), \\ \tilde{T}_{,\tilde{t}} &= \tilde{\Delta}\tilde{T} + \tilde{S} \quad (\text{liquid phase}) \end{aligned}$$

and the interface condition is

$$\gamma \tilde{\mathbf{v}} \cdot \mathbf{n} = \left(\kappa \tilde{\nabla}\tilde{T} |_s - \tilde{\nabla}\tilde{T} |_\ell \right) \cdot \mathbf{n}.$$

The X-MESH approach will now be tested on several examples. Unless otherwise noted, the values of the physical properties are taken from table 1 and the numerical parameters from table 2.

4.2. The straight front semi-infinite two-phase problem

The straight front two-phase Stefan problem can be expressed as a heat conduction problem in a semi-infinite domain represented by $\Omega = [0, +\infty[$. There is no volumetric heat source. The domain is initially liquid at temperature $T_\ell = 293$ [K] and the temperature is abruptly lowered at the temperature $T_s = 263$ [K] at the left side of the domain. A solidification front located at $x_f(t)$ moves continuously in the right direction. It is given (see [35]) by a classical square root expression

$$x_f(t) = 2\phi\sqrt{\alpha_s t}. \quad (14)$$

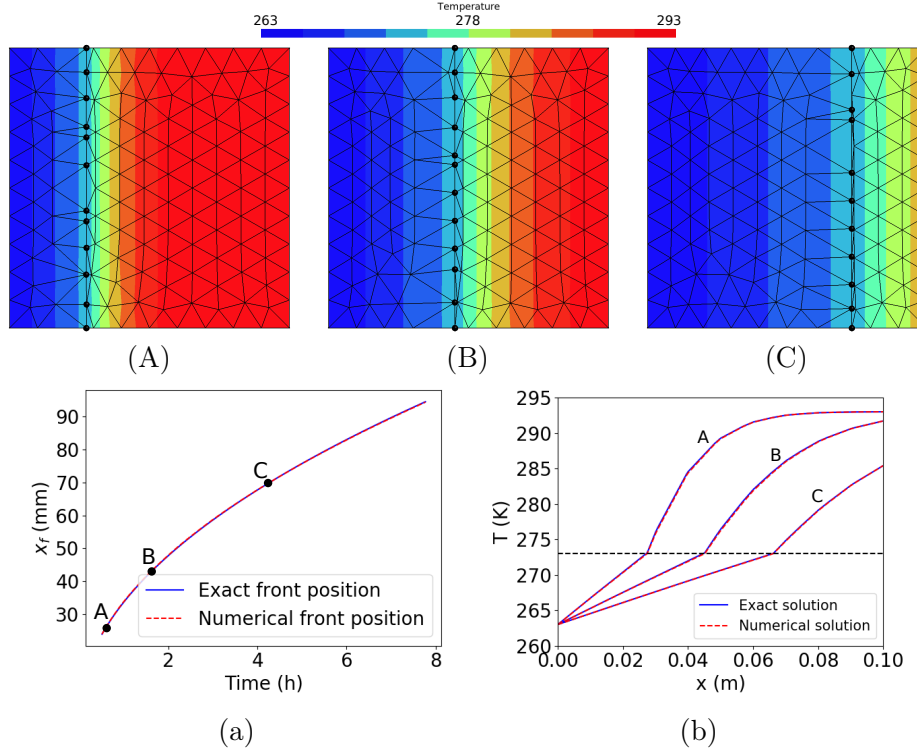


Figure 7: Straight front semi-infinite problem: numerical results for $l = 0$ [J.kg⁻¹] and $h_e = 0.01$ [m]. Temperature fields and fronts (front vertices are represented by big, black dots) at $t = 0.66$ [h] (A), 1.78 [h] (B) and 4.68 [h] (C). Comparison of exact and numerical front positions (a) and temperature profiles along the horizontal direction (obtained by averaging the temperature vertically) (b).

The dimensionless coefficient ϕ is obtained by solving the following transcendental equation:

$$\frac{c_s(T_0 - T_s)}{l} \frac{e^{-\phi^2}}{\text{erf}(\phi)} - \frac{1}{\gamma\sqrt{\alpha}} \frac{e^{-\alpha\phi^2}}{(1 - \text{erf}(\phi\sqrt{\alpha}))} - \phi\sqrt{\pi} = 0,$$

where $\text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt$ is the error function. The temperature field in the solid phase is given by

$$T(x, t) = T_s + \frac{(T_0 - T_s)}{\text{erf}(\phi)} \text{erf}\left(\frac{x}{2\sqrt{\alpha_s t}}\right), \text{ for } x \leq x_f$$

and in the liquid phase by

$$T(x, t) = T_\ell - \frac{(T_\ell - T_0)}{(1 - \operatorname{erf}(\phi\sqrt{\alpha}))} \left(1 - \operatorname{erf} \left(\frac{x}{2\sqrt{\alpha_\ell t}} \right) \right), \text{ for } x > x_f.$$

Although the geometry of the problem is 1D and semi-infinite, we use a square domain of dimensions 0.1×0.1 [m²]. To simulate the infinite boundary condition, the temperature is imposed on the left (constant equal to T_s) and on the right boundary (varying with time according to the above formula evaluated for $x = 0.1$ [m]). Top and bottom sides are insulated. The domain is meshed with linear triangular elements with a characteristic size of $h_e = 0.01$ [m]. Computations are performed with $l = 0$ [J.kg⁻¹] (Figure 7) and with $l = 3.3 \times 10^5$ [J.kg⁻¹] (Figure 9). We detail now how the time step for these simulations has been chosen. From the analytical solution (14), we can compute the front velocity:

$$\frac{dx_f}{dt} = \phi \sqrt{\frac{\alpha_s}{t}}. \quad (15)$$

It is proportional to the ϕ coefficient. With the ice-water parameters of Table 1, $\phi = 0.282$ for $l = 0$ [J.kg⁻¹] and $\phi = 0.141$ for $l = 3.3 \times 10^5$ [J.kg⁻¹]. The latent heat tends to slow down the freezing front compared to the no-latent heat case but without changing the order of magnitude of the velocity. We observe in (15) that the front velocity is infinite at $t = 0$ [s]. To avoid singularity issues in the numerical simulation we start from a time $t^0 > 0$ and use the analytical solution at that time as initial condition. We also make sure that the mesh represents exactly the initial front (by applying the `project_on_C` operator). Relation (15) may then be used to estimate the numerical front advance Δx over the time step Δt :

$$\Delta x = \phi \sqrt{\frac{\alpha_s}{t}} \Delta t.$$

A variable time step can be chosen to ensure a constant front advance:

$$\Delta t^{n+1/2} = \sqrt{\beta t^n} \quad \Rightarrow \quad \Delta x \simeq \phi \sqrt{\beta \alpha_s}, \quad (16)$$

where β [s] is a parameter.

In the case $l = 0$ [J.kg⁻¹], Figure 7, we choose $t^0 = 500$ [s] and $\beta = 100$ [s] which corresponds to a *small time step* i.e. a time step that only allows the front to advance of a length $\Delta x \simeq h_e/3 = 0.01/3$ [m]. In the simulations, we observe that the front relaying algorithm acts as we had imagined it:

elements of very small sizes are created and nodes of the front pass the relay (or the baton) to other nodes as they move along the edges downstream of the front. One can observe that front nodes are quasi-aligned on a vertical straight line. The numerical position x_f^{num} of the front (Figure 7, bottom left) is computed by taking the average x coordinates of the phase change front nodes. The temperature values plotted along the x direction (Figure 7, bottom right) are obtained by interpolating the temperature fields on a 500 (x -direction) \times 20 (y -direction) uniform grid, and averaging along the y direction. Figure 8 (a) shows for different values of h_e , the relative error on the front position with respect to time, noted $\xi(t)$ [no unit], and computed as

$$\xi(t) = \frac{|x_f^{\text{num}}(t) - x_f(t)|}{x_f(t)},$$

while Figure 8 (b) shows a relative error integrated over the total duration of the simulation:

$$\bar{\xi} = \frac{\int_0^{t_{\max}} |x_f^{\text{num}}(t) - x_f(t)| dt}{\int_0^{t_{\max}} x_f(t) dt},$$

where t_{\max} is the total duration of the simulation. Linear mesh convergence is observed, which makes sense because the front velocity depends on the jump of temperature gradients which converge at best linearly. In their paper [19], Ji, Chopp and Dolbow find the same convergence rate using XFEM. Yet, the error values are significantly lower with X-MESH.

Regarding the case $l = 3.3 \times 10^5$ [J.kg⁻¹], Figure 9, we choose $t^0 = 1000$ [s] and $\beta = 100$ [s]. It leads again to a *small time step* i.e. a time step that only allow the front to advance of a length $\Delta x \simeq h_e/6 = 0.01/6$ [m]. Convergence results are shown in Figure 10. Contrary to the case $l = 0$ [J.kg⁻¹], more oscillations of $\xi(t)$ are observed, and the order of convergence of $\bar{\xi}$ is not exactly 1. The differences may be explained by the solving method used for each problem. While a simple fixed point scheme as described at the end of §2.4 was sufficient to converge the case without latent heat, a more sophisticated quasi-Newton approach is required to converge the case with latent heat as described in Algorithm 1. Indeed, a fixed point scheme where the initial temperature update does not take into account the variation of energy due to latent heat has the effect of predicting a front location that is way too far from its converged position. Then, the front oscillates without converging. The quasi-Newton scheme acts like a damper and allows to converge in less than 10 iterations, without oscillations. More effort is

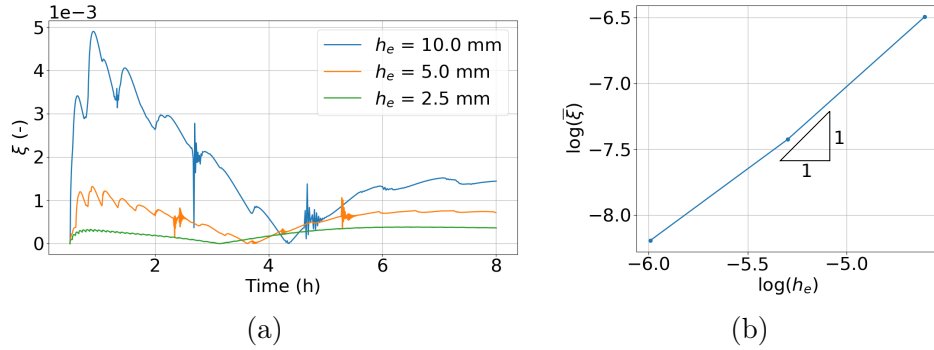


Figure 8: Straight front semi-infinite problem: error comparisons for $l = 0$ [J.kg⁻¹]. Relative front location error evolution with respect to time for different mesh refinements (a). Integrated error convergence with respect to the mesh element size (b).

necessary to solve the case where $l > 0$, which can be due to the fact that the heat flux is not continuous (see equation 3). It may also explain why the error on the front position has more oscillations (see Figure 10), although the error levels are absolutely acceptable.

A tempting question is now: can we do large time steps i.e. time steps that allows the front to advance of more than one element: $\Delta x > h_e$? The answer is surprisingly *yes*. Figure 11 shows three meshes corresponding to three different time steps where $t^0 = 2000$ [s] and $\beta = 20000$ [s] were chosen so that the phase change front crosses the domain in 5 time steps and more than three elements are crossed by the front in one time step. In this case, the nodes of the new front at t^{n+1} are not necessarily connected by edges to the nodes of the front at time t^n . Nevertheless, the front relaying algorithm can be applied as is and it is not necessary to compute the set of relays that allowed the transition from the configuration at time t^n to the one at time t^{n+1} . Moreover, the number of iterations required to converge the quasi-Newton algorithm remains reasonable (under 25 iterations per time step) even in the case of very large time steps. Note that to obtain the above results a value $\theta = 1$ was used (giving a fully implicit resolution). If one chooses $\theta = 0.5$, the front location agreement is even better but some oscillations are observed in the wake of the front as shown in Figure 12. Finally, note that as the front advances quickly, the relaxation has not enough time to take effect and the mesh has not recovered yet in the wake of the front at the final step of propagation. In a general approach, the relaxation coefficients 0.9 and 0.1 in (11) should ideally involve the front speed and element size.

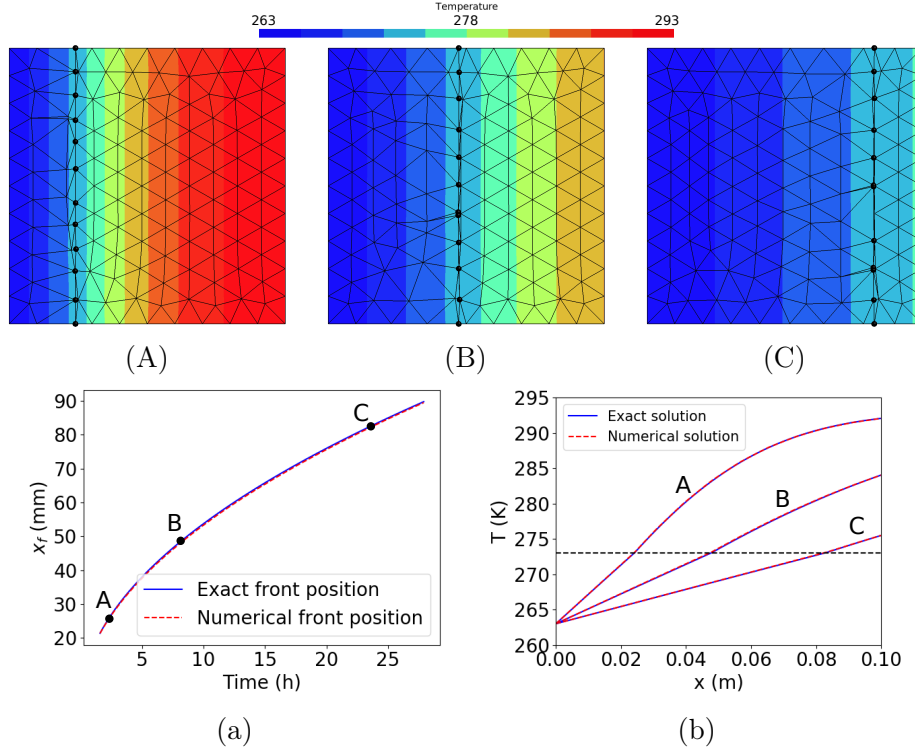


Figure 9: Straight front semi-infinite problem: numerical results for $l = 3.3 \times 10^5$ [J.kg⁻¹] and $h_e = 0.01$ [m]. Temperature fields and fronts (front vertices are represented by big, black dots) at $t = 3.01$ [h] (A), 7.89 [h] (B) and 23.71 [h] (C). Comparison of exact and numerical front position (a). Analytical and numerical temperature solutions along the horizontal direction (b).

4.3. The axisymmetric two-phase problem

The next example is the axisymmetric problem of the solidification of a liquid domain due to a Dirac heat sink Q (100 [W.m⁻¹]) located at $r = 0$ [m]. At $t = 0$ [s], the temperature is uniform: $T_\ell = 293$ [K]. Then, for later times $t \in [0, t_{\max}]$, the position of the solidification front (see [35]) is given by

$$r_f(t) = 2\phi\sqrt{\alpha_s t}.$$

The coefficient ϕ is obtained by solving

$$\frac{Q}{4\pi\rho\alpha_s l}e^{-\phi^2} + \frac{1}{\alpha\gamma} \frac{\exp(-\phi^2\alpha)}{\text{Ei}(-\phi^2\alpha)} - \phi^2 = 0,$$

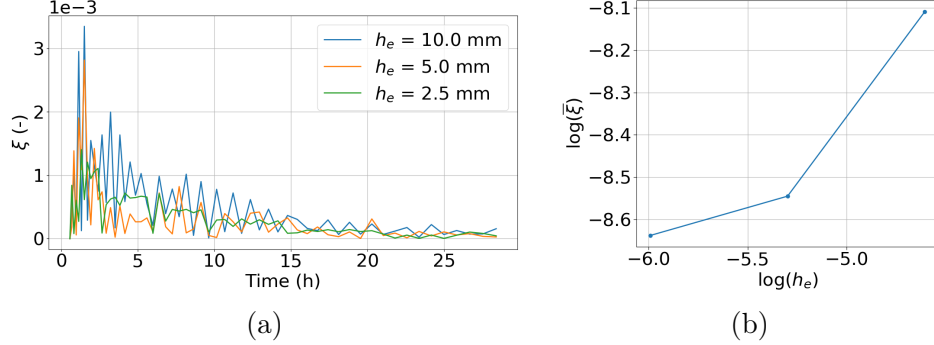


Figure 10: Straight front semi-infinite problem: error comparisons for $l = 3.3 \times 10^5$ [J.kg $^{-1}$]. Relative front location error evolution with respect to time for different mesh refinements (a). Integrated error convergence with respect to the mesh element size (b).

where Ei is the exponential integral function $\text{Ei}(x) = \int_{-\infty}^x \frac{e^t}{t} dt$. We have $\phi = 0.162$ for $l = 0$ [J.kg $^{-1}$] and $\phi = 0.094$ for $l = 3.3 \times 10^5$ [J.kg $^{-1}$]. The temperature field for the solid ($r \leq r_f$) and liquid phase ($r \geq r_f$) is given by

$$T(r, t) = \begin{cases} T_0 + \frac{Q}{4\pi k_s} \left[\text{Ei}\left(-\frac{r^2}{4\alpha_s t}\right) - \text{Ei}(-\phi^2) \right], & \text{for } r \leq r_f, \\ T_\ell - \frac{T_\ell - T_0}{\text{Ei}(-\phi^2 \alpha)} \text{Ei}\left(-\frac{r^2}{4\alpha_\ell t}\right), & \text{for } r \geq r_f. \end{cases} \quad (17)$$

Two versions of this problem are now presented, one that is easy and a harder one.

4.3.1. The easy version

To avoid the infinite initial front speed, we start the simulation at $t^0 = 3600$ [s] with the exact solution as initial temperature. We also consider a ring geometry to avoid the direct effect of the Dirac sink. The exact temperature evolution is imposed to an inner radius $R_{\text{int}} = 0.01$ [m] and an external radius $R_{\text{ext}} = 0.1$ [m]. For this easy case a zero latent heat is considered. Finally, a variable time step $\Delta t^{n+1/2} = \sqrt{25t^n}$ [s] is chosen. The results obtained for a coarse mesh with a finite element size $h_e = 0.005$ [m] are given in Figure 13 while the errors for different mesh sizes are plotted in Figure 14. The numerical value of the front position which is plotted and used to compute the error is obtained by taking the average of the r coordinate of the nodes of the phase change front. Here again, the front is correctly captured by the X-MESH approach.

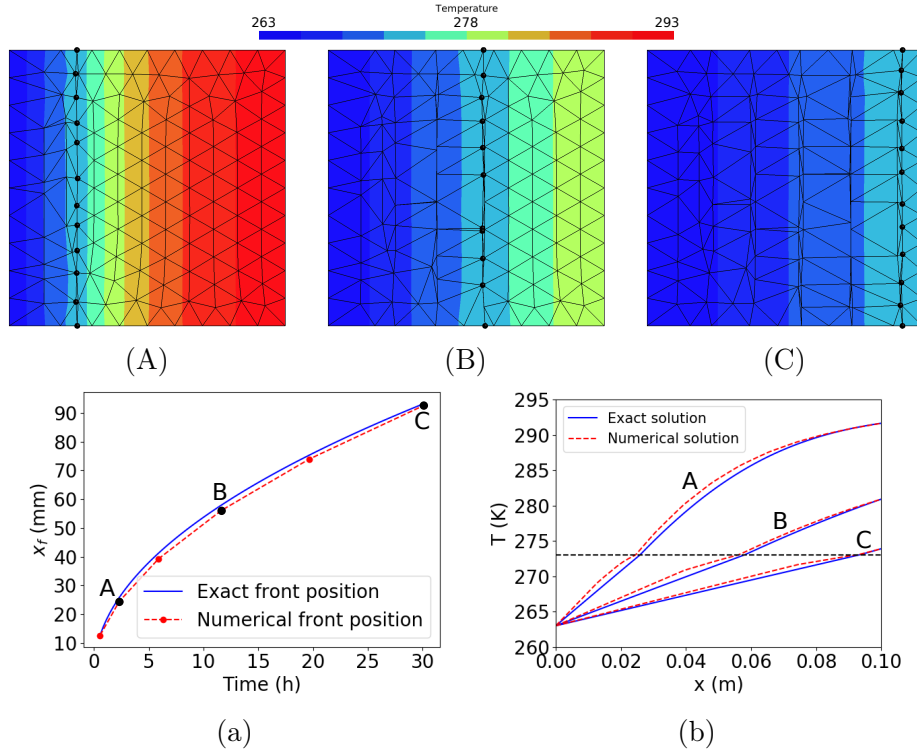


Figure 11: Straight front semi-infinite problem solved with five time steps and $\theta = 1$: numerical results for $l = 3.3 \times 10^5$ [J.kg⁻¹] and $h_e = 0.01$ [m]. Temperature fields and fronts (front vertices are represented by big, black dots) after the first, third and fifth time-step ($t = 2.31$ [h] (A), 11.62 [h] (B) and 30.1 [h] (C)). Comparison of exact and numerical front positions (a). Analytical and numerical solutions along the horizontal direction (b).

4.3.2. The harder version

This test case is there to push further the capabilities of our method. We use again the axisymmetric test case, but with a latent heat of 3.3×10^5 [J.kg⁻¹] and by explicitly modeling the Dirac heat sink in the center of the circle. The Dirac sink (100 [W.m⁻¹]) contributes to the three nodes of the element in which it is located proportionally to its barycentric coordinates. Instead of considering an infinite domain, we consider a square with dimensions 3×3 [m²] and apply the analytical solution to the outer contour. We start the simulation at $t^0 = 0$ [s] to see if the method can handle the nucleation and face the infinite initial velocity of the analytical solution. Figure 15

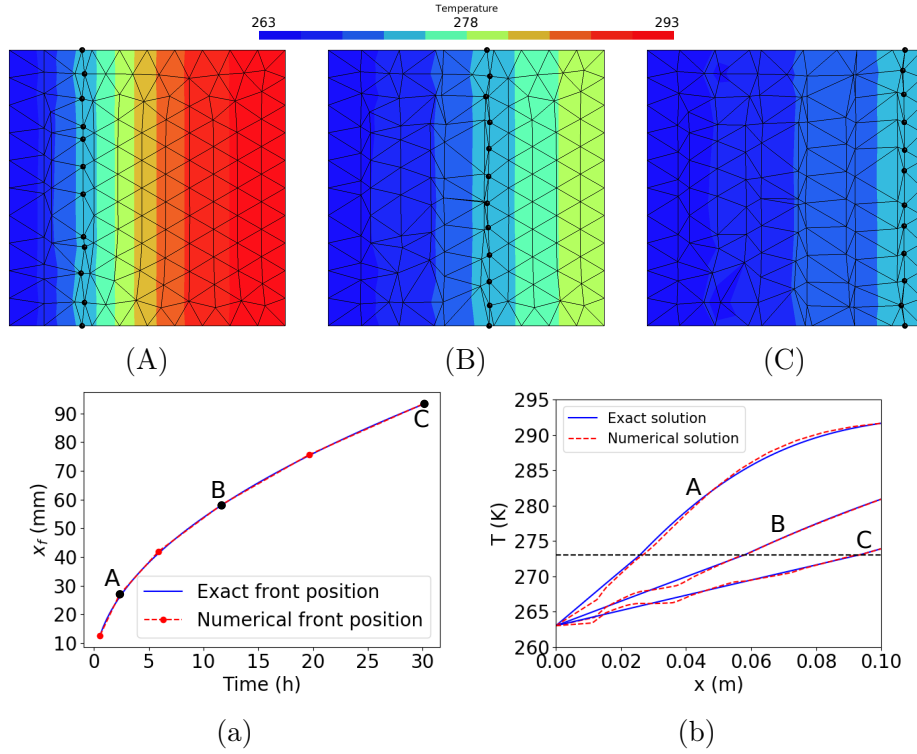


Figure 12: Straight front semi-infinite problem solved with five time steps and $\theta = 0.5$: numerical results for $l = 3.3 \times 10^5$ [J.kg⁻¹] and $h_e = 0.01$ [m]. Temperature fields and fronts (front vertices are represented by big, black dots) after the first, third and fifth time-step ($t = 2.31$ [h] (A), 11.62 [h] (B) and 30.1 [h] (C)). Comparison of exact and numerical front position (a). Analytical and numerical solutions along the horizontal direction (b).

shows solutions at different times as well as a comparison with the analytical solution. An average of 9 quasi-Newton iterations were necessary to reduce the residual by a factor 10^6 . Figures show that our method is able to nucleate a front. At early stages, the shape of the front is not circular due to the insufficient resolution of the mesh. The Stefan model is very stable: when the front grows, it becomes quasi-circular and our numerical solution is very close to the analytical solution. Meshes of Figure 15 show the effect of mesh relaxation: the mesh at the center of the circle that was initially affected by the front relaying algorithm progressively returns to its initial state.

Regarding the time step, we could not rely on formula (16) since $t^0 = 0$.

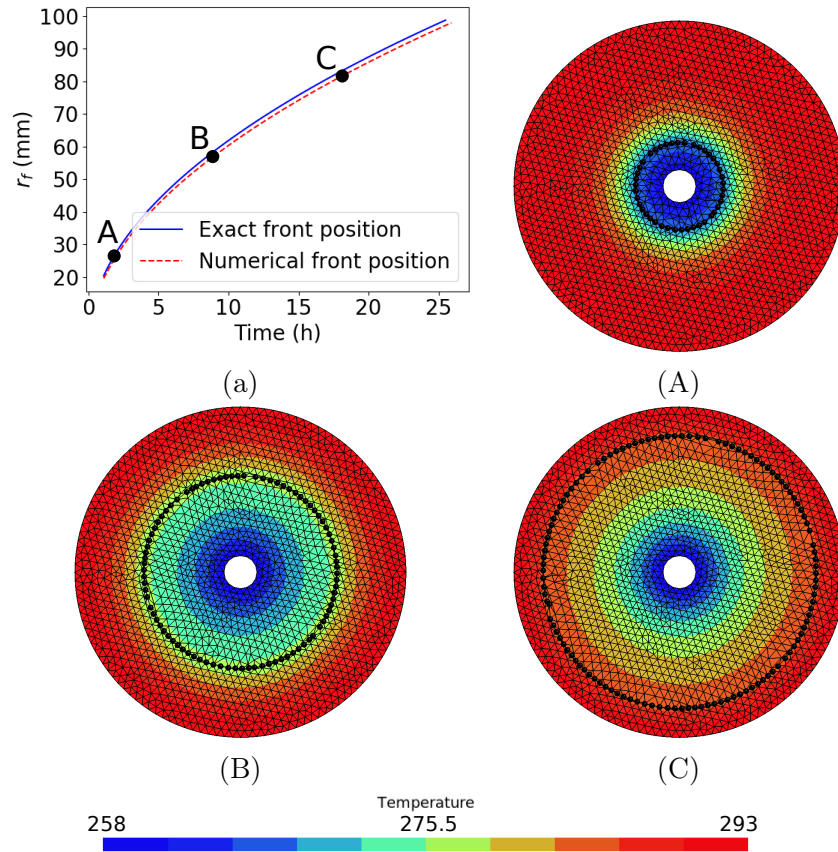


Figure 13: Axisymmetric problem: numerical results for $l = 0$ [$\text{J}\cdot\text{kg}^{-1}$] for $h_e = 0.005$ [m]. Comparison of exact and numerical front positions (a). Temperature fields and fronts at $t = 2.1$ [h] (A), 9.6 [h] (B) and 18.9 [h] (C). The front vertices are represented by big, black dots.

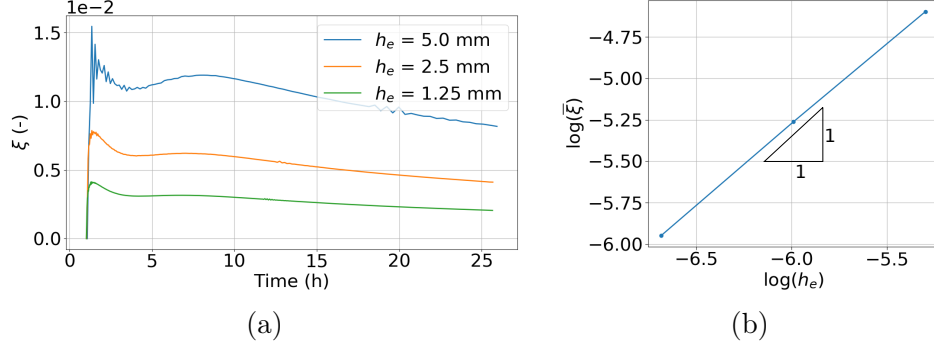


Figure 14: Axisymmetric problem: error comparisons for $l = 0$ [J.kg^{-1}]. Relative front location error evolution with respect to time for different mesh refinements (a). Integrated error convergence with respect to the mesh element size (b).

We used instead the implicit relation

$$\Delta t^{n+1/2} = \sqrt{\Delta t^{1/2} t^{n+1}} \quad (18)$$

in which the β parameter has been replaced by the choice of the first time step $\Delta t^{1/2}$. Noting that $t^{n+1} = t^n + \Delta t^{n+1/2}$, we get the time step value for any instant t^n :

$$\Delta t^{n+1/2} = \Delta t^{1/2} \left(\frac{1 + \sqrt{1 + 4t^n/\Delta t^{1/2}}}{2} \right). \quad (19)$$

In the simulation, the value $\Delta t^{1/2} = 25$ [h] has been used.

4.4. Two rotating heat sinks

In this last example, we start from an initial uniform temperature field $T_\ell = 283$ [K] on a square domain of dimensions 3×3 [m^2]. This corresponds to image (1) on Figure 16. The same T_ℓ temperature is used as boundary condition on the square contour. Two Dirac heat sinks of intensity 500 [W.m^{-1}], initially situated at positions $x = \pm 0.75$ [m] and $y = 0$ [m], rotate on a circle of radius 0.75 [m] with a period of 240 [h]. A constant time step of 4 [h] is used in the simulation.

The problem is solved with $l = 3.3 \times 10^5$ [J.kg^{-1}]. Initially, there is no front. Then, due to the heat sinks, the temperature decreases below T_0 , and two fronts start to nucleate. Afterwards, two icy zones grow until merging with each other. At a later time, the middle of the domain is totally frozen. This example illustrates the X-MESH capability to nucleate, merge and

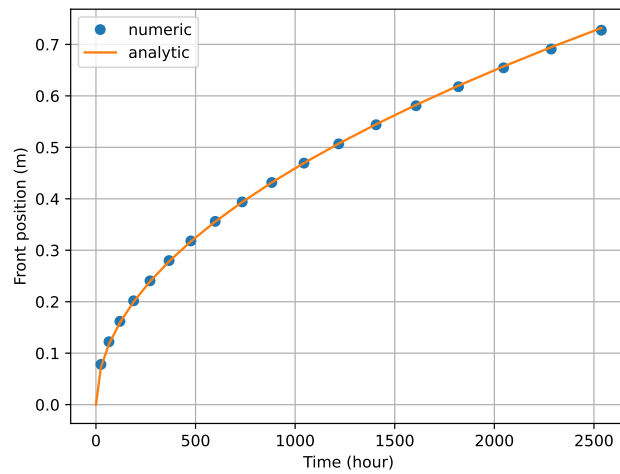
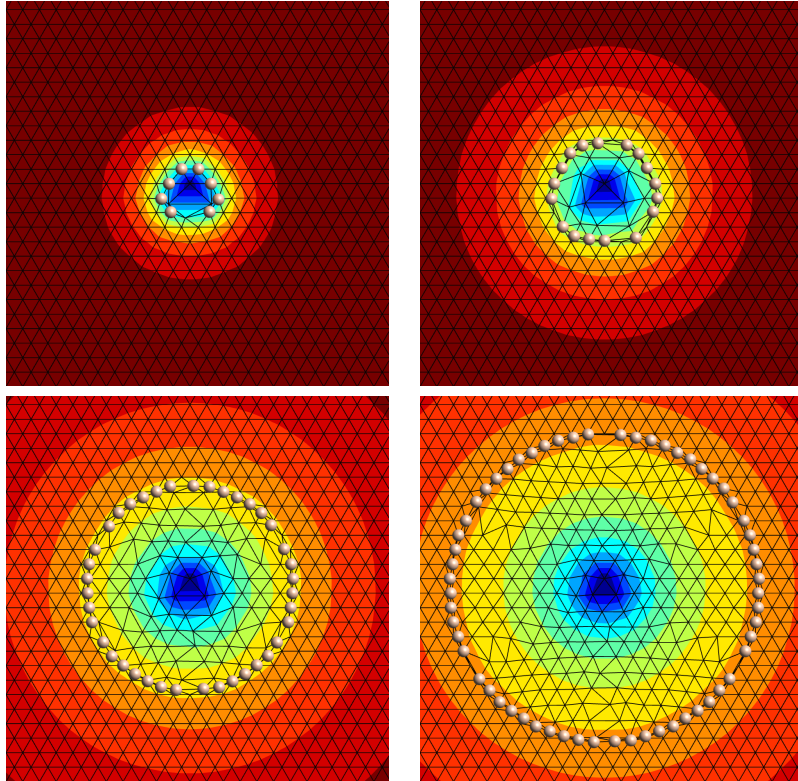


Figure 15: Axisymmetric problem with $l = 3.3 \times 10^5$ [J.kg⁻¹]: the freezing front is nucleated and then propagates (top). The front vertices are represented by big, yellow dots. Comparison of exact and numerical front positions (bottom).

annihilate fronts. The nonlinear solver converged in less than 15 iterations at every time step. It took less than two minutes to complete the simulation with a code that is far from being optimized.

5. Conclusions

In this paper, a new method to address the Stefan problem has been introduced. The extreme mesh deformation approach (X-MESH) tracks the solidification/melting front without remeshing and with a classical finite element formulation. Moreover, the mesh topology is kept fixed. The X-MESH introduces the notion of front relaying allowing to keep an optimal mesh quality, at least for elements which are not directly connected to the front. The elements in contact with the front may have zero size during the relay. To deal with these zero- or close to zero-measure element, we used an added volume approach but we plan to replace it with dualization which will not require any volume approximation. The iterative process used in the solver is based on a quasi-Newton scheme which is quite easy to implement. The tangent operator uses smoothed energy and conductivity functions while the residual uses the true sharp functions.

The numerical experiments on two problems with analytic solutions showed the quality of the obtained X-MESH solution. The final experiments with rotating heat sinks did demonstrate the X-MESH capability to have front nucleation, merge and annihilation in the same simulation without remeshing.

The front relaying methodology proposed here, although quite basic, allows to solve problems that are far from being basic. Even if, at convergence of the quasi-Newton, we found that the obtained meshes are all valid, there is no guarantee that the proposed approach does not produce inverted triangles. The front-relaying scheme could be improved in this sense.

Finally, note that for the Stefan model considered, no history variable are needed. If history variable were present (as cumulative plasticity in plasticity models), these variables would need to be transferred as the mesh moves even though the mesh topology is kept fixed throughout the simulation. This, also, needs to be analyzed in further studies.

References

- [1] S. Gupta, The classical Stefan problem: basic concepts, Elsevier Amsterdam, 2003.

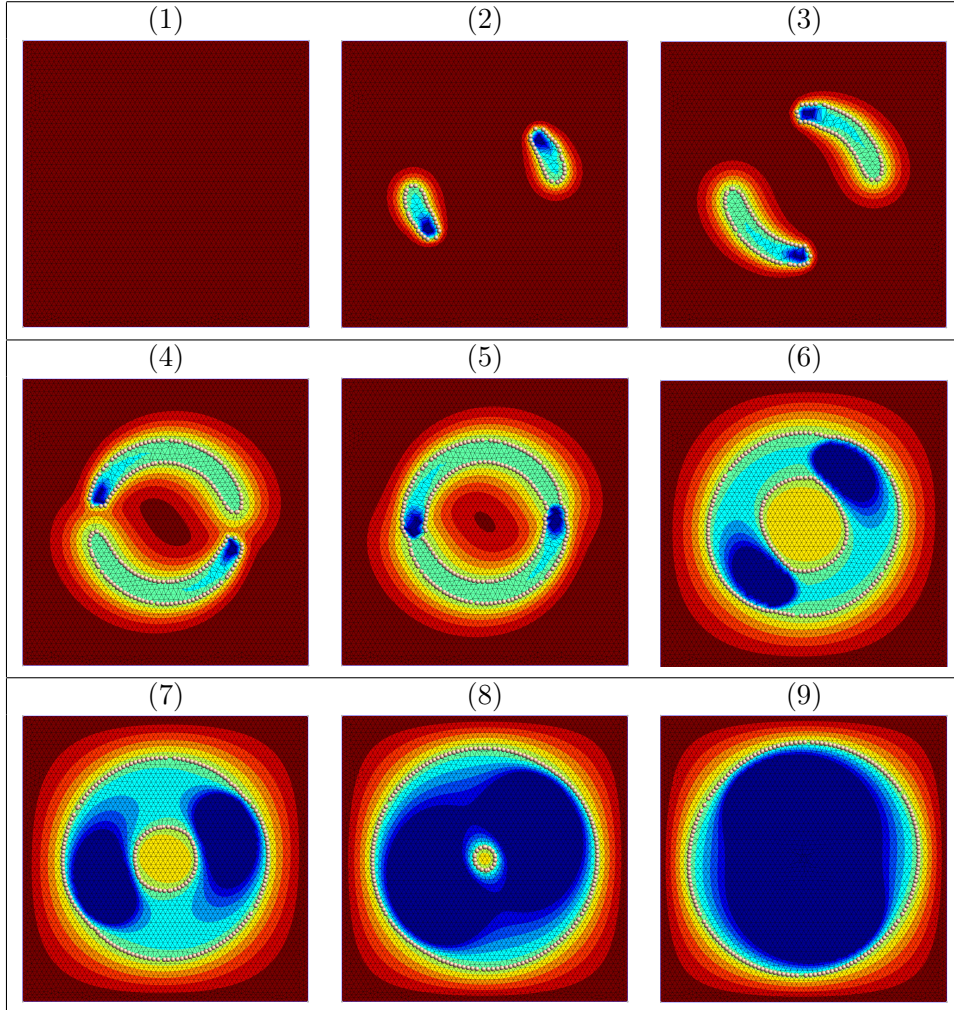


Figure 16: Stefan problem with two rotating heat sinks. Front vertices are represented as yellowy dots. The color scale is saturated from below: any temperature below -10 degrees Celsius is represented in dark blue. At time 0 [s], the temperature is uniform and no front is yet present (Figure (1)). Two solidification fronts rapidly nucleate (Figure (2)), grow (Figures (3) \rightarrow (4)) and eventually merge (Figure (5)), forming two solidification fronts. The internal solidification front progresses towards the center of the domain (Figures (5) \rightarrow (8)) and is finally annihilated, leaving the center of the domain entirely solidified (Figure (9)). (An animated version of this simulation is provided as a supplementary material to this paper).

- [2] M. Koga, S. and Krstic, Phase Change Model: Stefan Problem, Springer International Publishing, Cham, 2020, pp. 1–13. doi:10.1007/978-3-030-58490-0_1.
- [3] A. Friedman, Free boundary problems for parabolic equations I. Melting of solids, *Journal of Mathematics and Mechanics* (1959) 499–517.
- [4] M. Jaafar, D. Rousse, S. Gibout, J.-P. Bédécarrats, A review of dendritic growth during solidification: Mathematical modeling and numerical simulations, *Renewable and Sustainable Energy Reviews* (2017). doi:10.1016/j.rser.2017.02.050.
URL <http://dx.doi.org/10.1016/j.rser.2017.02.050>
- [5] J. Back, S. W. McCue, M. H.-N. Hsieh, T. Moroney, The effect of surface tension and kinetic undercooling on a radially-symmetric melting problem, *Applied Mathematics and Computation* 229 (2014) 41–52.
- [6] J. R. King, J. D. Evans, Regularization by kinetic undercooling of blow-up in the ill-posed stefan problem, <http://dx.doi.org/10.1137/04060528X> 65 (2006) 1677–1707. doi:10.1137/04060528X.
URL <http://www.siam.org/journals/siap/65-5/60528.html>
- [7] J. Donea, A. Huerta, J.-P. Ponthot, A. Rodríguez-Ferran, *Arbitrary Lagrangian–Eulerian Methods*, John Wiley & Sons, Ltd, 2004, Ch. 14, pp. 1–38. doi:<https://doi.org/10.1002/0470091355.ecm009>.
- [8] R. Loubère, P.-H. Maire, M. Shashkov, J. Breil, S. Galera, ReALE: A reconnection-based arbitrary-Lagrangian–Eulerian method, *Journal of Computational Physics* 229 (12) (2010) 4724–4761. doi:<https://doi.org/10.1016/j.jcp.2010.03.011>.
URL <https://www.sciencedirect.com/science/article/pii/S002199911000121X>
- [9] W. Boscheri, M. Dumbser, D. Balsara, High-order ADER-WENO ALE schemes on unstructured triangular meshes—application of several node solvers to hydrodynamics and magnetohydrodynamics, *International Journal for Numerical Methods in Fluids* 76 (10) (2014) 737–778.
- [10] A. Barlow, P. Maire, W. J. Rider, R. N. Rieben, M. J. Shashkov, Arbitrary Lagrangian–Eulerian methods for modeling high-speed compressible multimaterial flows, *Journal of Computational Physics* 322 (2016) 603–665. doi:<https://doi.org/10.1016/j.jcp.2016.07.001>.

URL <https://www.sciencedirect.com/science/article/pii/S0021999116302807>

- [11] D. Burton, N. Morgan, M. Charest, M. Kenamond, J. Fung, Compatible, energy conserving, bounds preserving remap of hydrodynamic fields for an extended ALE scheme, *Journal of Computational Physics* 355 (2018) 492–533. doi:<https://doi.org/10.1016/j.jcp.2017.11.017>.
URL <https://www.sciencedirect.com/science/article/pii/S0021999117308550>
- [12] M. J. Baines, M. E. Hubbard, P. K. Jimack, R. Mahmood, A moving-mesh finite element method and its application to the numerical solution of phase-change problems, *Communications in Computational Physics* 6 (3) (2009) 595–624. doi:[10.4208/cicp.2009.v6.595](https://doi.org/10.4208/cicp.2009.v6.595).
- [13] E. Gros, G. Anjos, J. Thome, Moving mesh method for direct numerical simulation of two-phase flow with phase change, *Applied Mathematics and Computation* 339 (2018) 636–650. doi:<https://doi.org/10.1016/j.amc.2018.07.052>.
URL <https://www.sciencedirect.com/science/article/pii/S0096300318306180>
- [14] Y. Zhang, A. Chandra, F. Yang, E. Shams, O. Sahni, M. Shephard, A. A. Oberai, A locally discontinuous ALE finite element formulation for compressible phase change problems, *Journal of Computational Physics* 393 (2019) 438–464. doi:[10.1016/j.jcp.2019.04.039](https://doi.org/10.1016/j.jcp.2019.04.039).
URL <https://doi.org/10.1016/j.jcp.2019.04.039>
- [15] S. Chen, B. Merriman, S. Osher, P. Smereka, A simple level set method for solving Stefan problems, *Journal of Computational Physics* 135 (1) (1997) 8–29. doi:[10.1006/jcph.1997.5721](https://doi.org/10.1006/jcph.1997.5721).
- [16] J. Shaikh, A. Sharma, R. Bhardwaj, On sharp-interface level-set method for heat and/or mass transfer induced Stefan problem, *International Journal of Heat and Mass Transfer* 96 (2016) 458–473. doi:[10.1016/j.ijheatmasstransfer.2015.12.074](https://doi.org/10.1016/j.ijheatmasstransfer.2015.12.074).
URL <http://dx.doi.org/10.1016/j.ijheatmasstransfer.2015.12.074>
- [17] V. Vasil'ev, M. Vasilyeva, An accurate approximation of the two-phase stefan problem with coefficient smoothing, *Mathematics* 8 (11) (2020) 1–25. doi:[10.3390/math8111924](https://doi.org/10.3390/math8111924).

- [18] R. Merle, J. Dolbow, Solving thermal and phase change problems with the eXtended finite element method, *Computational Mechanics* 28 (5) (2002) 339–350. doi:10.1007/s00466-002-0298-y.
URL <http://www.springerlink.com/index/RNQ9AG3XN8E8QHK5.pdf>
- [19] H. Ji, D. Chopp, J. Dolbow, A hybrid extended finite element/level set method for modeling phase transformations, *International Journal For Numerical Methods in Engineering* 54 (2002) 1209–1233.
URL <http://www3.interscience.wiley.com/journal/93518898/abstract>
- [20] M. He, Q. Yang, N. Li, X. Feng, N. Liu, An Extended Finite Element Method for heat transfer with phase change in frozen soil, *Soil Mechanics and Foundation Engineering* 57 (6) (2021) 497–505. doi:10.1007/s11204-021-09698-z.
- [21] L. Wang, P.-O. Persson, A high-order discontinuous Galerkin method with unstructured space–time meshes for two-dimensional compressible flows on domains with large deformations, *Computers & Fluids* 118 (2015) 53–68. doi:10.1016/J.COMPFLUID.2015.05.026.
- [22] E. Gaburro, W. Boscheri, S. Chiocchetti, C. Klingenberg, V. Springel, M. Dumbser, High order direct Arbitrary-Lagrangian-Eulerian schemes on moving Voronoi meshes with topology changes, *Journal of Computational Physics* 407 (2020) 109167. doi:<https://doi.org/10.1016/j.jcp.2019.109167>.
URL <https://www.sciencedirect.com/science/article/pii/S0021999119308721>
- [23] E. Gaburro, A unified framework for the solution of hyperbolic PDE systems using high order direct Arbitrary-Lagrangian-Eulerian schemes on moving unstructured meshes with topology change, *Archives of Computational Methods in Engineering* 28 (3) (2021) 1249–1321.
- [24] B. Re, C. Dobrzynski, A. Guardone, An interpolation-free ALE scheme for unsteady inviscid flows computations with large boundary displacements over three-dimensional adaptive grids, *Journal of Computational Physics* 340 (2017) 26–54.
- [25] V. Springel, E pur si muove: Galilean-invariant cosmological hydrodynamical simulations on a moving mesh, *Monthly Notices of the Royal Astronomical Society* 401 (2) (2010) 791–851.

- [26] D. Boffi, L. Gastaldi, Stability and geometric conservation laws for ALE formulations, *Computer Methods in Applied Mechanics and Engineering* 193 (42-44) (2004) 4717–4739. doi:[10.1016/j.cma.2004.02.020](https://doi.org/10.1016/j.cma.2004.02.020).
- [27] M. Azaiez, F. Jelassi, M. Mint Brahim, J. Shen, Two-phase Stefan problem with smoothed enthalpy, *Communications in Mathematical Sciences* 14 (6) (2016) 1625–1641.
- [28] C. Geuzaine, J.-F. Remacle, Gmsh: A 3-D finite element mesh generator with built-in pre-and post-processing facilities, *International journal for numerical methods in engineering* 79 (11) (2009) 1309–1331.
- [29] I. Babuska, A. Aziz, On the angle condition in the finite element method, *SIAM Journal on Numerical Analysis* 13 (1976) 214–226. doi:<https://doi.org/10.1137/0713021>.
- [30] J. Shewchuk, What is a good linear finite element? interpolation, conditioning, anisotropy, and quality measures (preprint), University of California at Berkeley 73 (2002) 137.
- [31] A. Hannukainen, S. Korotov, M. Křížek, The maximum angle condition is not necessary for convergence of the finite element method, *Numerische Mathematik* 120 (1-6) (2012) 79–88. doi:[10.1007/s00211-011-0403-2](https://doi.org/10.1007/s00211-011-0403-2).
- [32] A. Hannukainen, S. Korotov, M. Křížek, The maximum angle condition is not necessary for convergence of the finite element method, *Numerische mathematik* 120 (1) (2012) 79–88.
- [33] V. Kučera, On necessary and sufficient conditions for finite element convergence (2016). arXiv:[1601.02942](https://arxiv.org/abs/1601.02942).
URL <http://arxiv.org/abs/1601.02942>
- [34] M. Duprez, V. Lleras, A. Lozinski, Finite element method with local damage of the mesh, *ESAIM: Mathematical modelling and numerical analysis* 53 (2019) 1–10. arXiv:[arXiv:1401.0559v1](https://arxiv.org/abs/1401.0559v1).
- [35] H. S. Carslaw, J. Jaeger, *Conduction of heat in solids*, Clarendon Press, 1959.