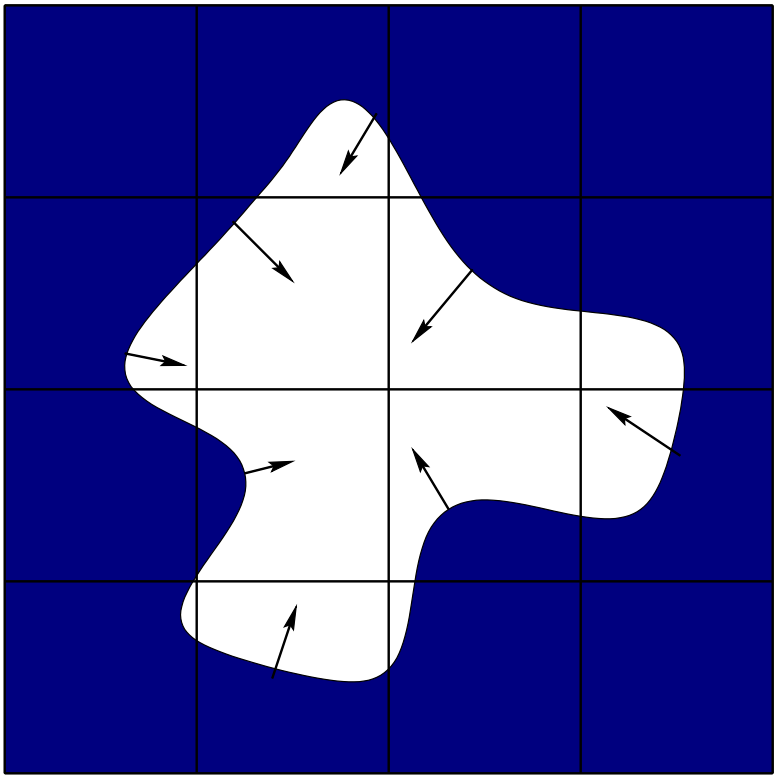
Training Report
Modeling Phase Transformations with the eXtended Finite Element Method

Renaud Merle *,
Duke University, Durham, North Carolina

August 26, 2000



*Ecole Normale Superieure de Cachan, France

Abstract

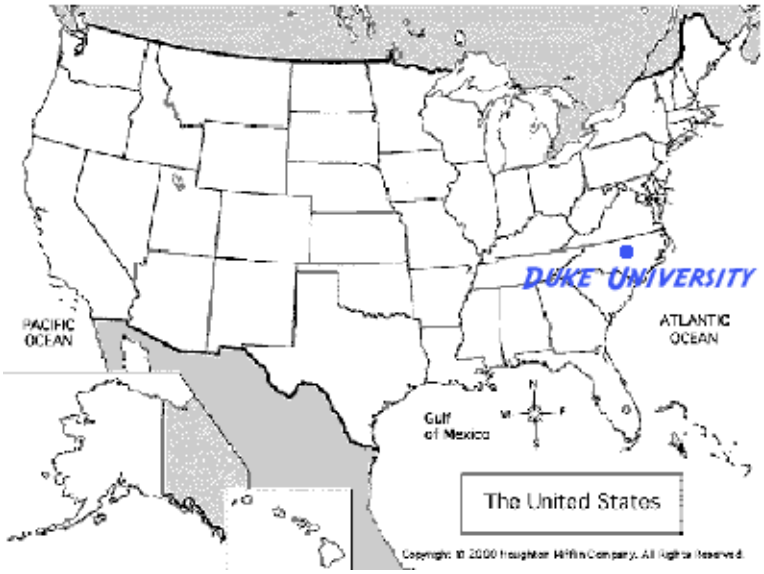
A new method for solving parabolic problems with the finite element method is presented. A standard approximation id enriched with local functionsin order to capture highly variable, moving solutions. To this end, a projection technique is developed for transferring solution in formation between global and local approximations. The method is particularly well suited for studying problems with moving materials interfaces, in particular phase change problems. Several one dimensional benchmark problem and two-dimensional applications are presented to illustrate the overall accuracy and utility of the new method.

Contents

List of Figures

1 PRESENTATION OF DUKE UNIVERSITY

I have made my summer work at Duke University, which is located in Durham, in the forest of North Carolina, USA. More precisely, it is 300 miles to the south of Washington, DC.



The location of Duke University.

Duke University was founded in 1924 by James B. Duke, industrialist from tobacco. He transformed the Trinity College into a new gothic campus which opened in 1930.

Duke University is one of the best universities in the US, especially for its medical studies. The hospital of Durham is basicly ranked 3^{rd} in the US. But all the major subjects are studied in Duke University: Arts, Languages, Medicine, Mathematics, Physics, Biology, Chemistry and Electrical, Mechanical, Civil and Environmental Engineering. The Campus welcomes 20 000 students from all around the world: certainly from the US, but also from China, Turkey, France, Germany ...

I spent the summer (my training was from May, 1^{st} until July, 31^{st}in the Department of Civil and Environmental Engineering which is a part of the Pratt School of Engineering. In this department, the main themes of research are: engineering mechanics, structural engineering mechanics and water resources (*http://www-cee.egr.duke.edu/*). I worked for Dr John Dolbow, Ph.D. from Northwestern University, Assistant Professor of Civil and Environmental Engineering.

Dr. Dolbow is a full time assistant professor at the Department of Civil and Environmental Engineering of Duke University. His research concerns the development of computational methods for nonlinear problems in solid mechanics. In particular, he is interested in modeling quasi-static and dynamic fracture of structural components and the evolution of interfaces with nonlinear constitutive laws.

The work was to adapt the method used for cracks growth (see Daux, Moës, Dolbow, Sukumar, and Belytschko (2000)) to the heat transfer problem, especially with phase transformations. The main objective was to implement the phase transformation algorithm in a two dimensional finite element code built by Nicolas Moes, N.Sukumar,

Hudson Hall, the location of the Pratt School of Engineering.

J.Dolbow and T.Belytchko (see *http://www.tam.nwu.edu/X-FEM/* and *http://ceelab4.egr.duke.edu/ dolbow/xfem.html*)

The computer room in the Department of CEE was equiped by about thirty Unix stations and about ten windows machines.

2 INTRODUCTION

The problem of phase transformations arises in a number of physical problems of interest to the engineering community. These include solidification problems in metal casting, crystallization problems, as well as the behavior of shape-memory alloys. While the specific physical characteristics of these systems vary, they all share a common characteristic: a moving interface across which several fields may be discontinuous. For example, solidification problems can be described by the motion of the freezing (or melting) front, separating the solid and liquid phases. The behavior of shape memory alloys is directly related to the motion of interfaces separating austenite and martensitic variants at the grain scale. The more general problem of a moving interface arises in other problems such as the burning of rocket fuel or the growth of oxide on a silicon wafer.

With regards to the finite element modeling of phase transformations, the foremost concern is "capturing" the solution in the vicinity of the moving interface, where the solution exhibits sharp gradients. Some pioneering work was developed by Lynch and O'Neill (1981), who wrote the shape functions as functions of time as well as the spatial coordinates. The nodes on the freezing front thus moved with the interface, such that the discontinuity in temperature gradient could always be captured. Twenty years later, this basic technique continues to be adopted for a wide range of problems. For directional solidification problems, the technique has been considerably refined by Zabaras and his coworkers (see Sampath and Zabaras (1999)).

A number of problems exist with the moving-mesh approach. Firstly, it is best applied to *directional* solidification problems, wherein the front essentially moves unidirectionally. The main reason is mesh distortion. It is easy to visualize that for freezing fronts which are geometrically complex, the moving mesh approach eventually leads to mesh entanglement. Secondly, for thermo-mechanical problems, we are concerned with the evolution of stress and strain in the object. A moving mesh approach requires projection techniques, and crucially in the vicinity of sharp gradients, leading to a significant loss in accuracy. In part, these concerns motivated the ALE formulation developed by Ghosh and Moorthy (1993). The work proposed herein shares several features with this approach, though it is suggested to be much simpler and more flexible.

The main philosophy of the eXtended Finite Element Method (X-FEM) developed by Dolbow (1999) and Moës, Dolbow, and Belytschko (1999), is to model moving features (cracks, voids, interfaces), with moving *enrichment* strategies. The basic idea is to augment a subset of the nodal basis functions with products of those basis functions and prescribed enrichment functions. It therefore belongs to the general class of partition-of-unity methods as described by Melenk and Babuška (1996). When the prescribed functions contain discontinuities, the result is a method capable of representing cracks whose morphology is independent of the finite element boundaries. As a result, no remeshing is required to simulate crack growth; one merely changes the subset of enriched nodes and updates (if necessary) the enrichment functions. Likewise, its application to phase transformations offers the possibility of using a fixed mesh.

For phase transformation problems, we seek to enrich with functions whose *derivatives* are discontinuous across the interface. While the precise form of the functions which provide the greatest accuracy has yet to be determined, the basic concept is appealing.

With the X-FEM the mesh and interface descriptions are independent, and yet we can still maintain accuracy at the interface, and allow it to move arbitrarily. Thus the method is applicable to freezing front motions beyond those of directional solidification. Moreover, with regard to the thermo-mechanical problem, the projections are postulated to be much more accurate.

This report is organized as follows. In the next section, we describe the governing equations which describe the heat transfer problem. We then discuss the weak form, and the discretization with the X-FEM. We after provide some one-dimensional experiments of the general method, and the last section describes the phase transformation problem and shows some results in two dimension.

3 PROBLEM FORMULATION

In this section, we briefly describe the governing equations for heat transfer problems. Consider the domain Ω shown in Fig. **??**. The governing equations are :

$$c\frac{\partial T}{\partial t} = div\ \vec{q} + f \quad \text{in } \Omega \tag{1a}$$

$$\vec{q} = -\kappa\overrightarrow{\nabla T} \quad \text{in } \Omega \tag{1b}$$

$$T_{|\Gamma_T} = T_g(x,t) \quad \text{imposed on } \Gamma_T \tag{1c}$$

$$\vec{q}_{|\Gamma_q} = \vec{q_h} \quad \text{imposed on } \Gamma_q \tag{1d}$$

where T is the temperature, c the volumetric capacitance, $\vec{q}$ the heat flux vector and κ the thermal conductivity tensor.

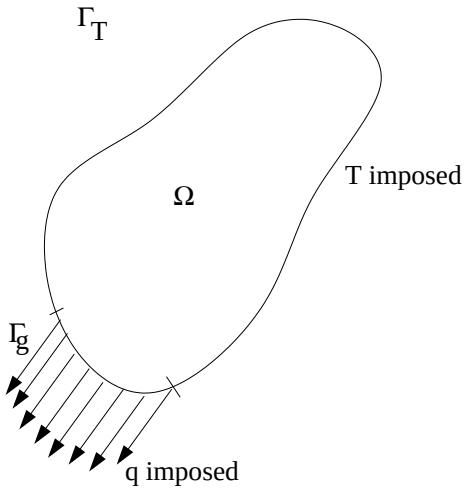


Figure 1: The considered domain.

4 WEAK FORM AND TIME STEPPING

The usual approach to implement a time-integration scheme with finite elements is to discretize in time *after* developing the Galerkin approximation. In our case, the approximation evolves in time along with a local feature of interest. So an initial point of departure is to reverse the order of these steps as follows.

We consider the solution on the time interval $[0, t_f]$, partitioned into time steps as $[t^n, t^{n+1}]$. Considering the thermal conductivities to be isotropic and constant, we write (1) at time step t^{n+1} as:

$$c\frac{\partial T^{n+1}}{\partial t} - \kappa \Delta T^{n+1} = f^{n+1} \quad \text{on } \Omega \tag{2}$$

and we consider a classical time-stepping scheme, namely generalized Trapezoidal rule :

$$T^{n+1} = T^n + \Delta t \left[\alpha T_t^{n+1} + (1-\alpha) T_t^n\right] \tag{3}$$

Substituting this equation into the above yields, eliminating $T_{,t}^{n+1}$ and after rearranging some terms:

$$T^{n+1} - \frac{\kappa \alpha \Delta t}{c} \Delta T^{n+1} = T^n + (1-\alpha)\Delta t T_{,t}^n + \frac{f^{n+1}}{c}$$

We now multiply by a kinematically admissible weight function δT^{n+1} , and integrate over the domain at time step t^{n+1}. This gives:

$$\begin{aligned} \int_\Omega \delta T^{n+1}\ T^{n+1}\ d\Omega - \frac{\alpha\kappa\Delta t}{c} \int_\Omega \delta T^{n+1}\ \Delta T^{n+1}\ d\Omega = \\ \int_\Omega \delta T^{n+1}\ \left[T^n + (1-\alpha)\Delta t T_{,t}^n\right]\ d\Omega + \frac{1}{c}\int_\Omega \delta T^{n+1}\ f^{n+1}\ d\Omega \end{aligned} \tag{4}$$

We now follow the standard step and use integration by parts to shift a derivative from the trial function to the weight function. This yield to the following expression:

$$\begin{aligned} \int_{\Omega^{n+1}} \delta T^{n+1}\ T^{n+1}\ d\Omega + \frac{\alpha\kappa\Delta t}{c}\int_\Omega \nabla \delta T^{n+1}\ \nabla T^{n+1}\ d\Omega = \\ \int_{\Omega_1^{n+1}} \delta T^{n+1}\ \left(T^n + \left[1-\alpha\right)\Delta t T_{,t}^n\right]\ d\Omega \\ + \frac{1}{c}\int_\Omega \delta T^{n+1}\ f^{n+1}\ d\Omega \\ + \frac{\alpha\Delta t}{c}\int_{\Gamma_q} \delta T^{n+1} \vec{q}_h \cdot \vec{n}\ d\Gamma \end{aligned} \tag{5}$$

After solving this system, we know the temperature T at instant t^{n+1} and we would like to know its time derivative $T_{,t}^{n+1}$. Basicly $\{T^{n+1}, T_{,t}^{n+1}\} \subset span\{\Phi_i\}^{n+1}$ and $\{T^n, T_{,t}^n\} \in span\{\Phi_i\}^n$ where $span\{\Phi_i\}^n$ and $span\{\Phi_i\}^{n+1}$ are the sets of the basis functions $\{\Phi_i\}$ at time steps t^n and t^{n+1}. The equation which links these variables in the classical time-stepping scheme is:

$$T_{,t}^{n+1} = \frac{T^{n+1} - T^n - (1-\alpha)\delta t T_{,t}^n}{\alpha \Delta t} \tag{6}$$

The way to calculate $T_{,t}^{n+1}$ is the same way used in space, that means that we make an L^2 projection of the equation (6) onto the set $span\{\Phi\}^{n+1}$. We multiply the above expression by $\delta T^{n+1} \in \{\Phi\}^{n+1}$ and integrate on Ω. More precisely:

$$\int_\Omega \delta T^{n+1}\ T_{,t}^{n+1}\ d\Omega = \frac{1}{\alpha \Delta t} \int_\Omega \delta T^{n+1}\ T^{n+1}\ d\Omega - \int_\Omega \delta T^{n+1}\ \frac{T^n + (1-\alpha)\Delta t T_{,t}^n}{\alpha \Delta t}\ d\Omega \qquad (7)$$

After substituting δT and T bye the chosen weight functions and approximations and solving this system, we know the field $T_{,t}^{n+1}$ and we can continue to march in time.

5 DISCRETIZATION WITH THE X-FEM

We now describe how to discretize this BVP with the X-FEM.The standard finite element approximation to the temperature field takes the form

$$T^h(\vec{x}) = \sum_I N_I(\vec{x}) T_I \tag{8}$$

where N_I are the classical nodal shape functions and T_I are the corresponding nodal coefficients. The sum is taken over all nodes in the mesh.

In the X-FEM, we extend the above using the construction

$$T^h(\vec{x}) = \underbrace{\sum_I N_I(\vec{x}) T_I}_{classical\ approximation} + \underbrace{\sum_J N_J(\vec{x}) \cdot g(\vec{x},t)\ a_J}_{enrichment} \tag{9}$$

where $g(\vec{x},t)$ is the enrichment function, a_J are enriched degrees of freedom and $J \subset I$ is a subset of the nodes in the mesh. For example, we may take J to consist of all nodes whose shape function supports contain the location of the zone of local behavior ω:

$$J = \{I | \exists\ \vec{x} \in \omega / N_I(\vec{x}) \neq 0\} \tag{10}$$

This approach allows the approximation to follow a moving feature of interest, as both the set J and the function g can evolve with ω. However we note that this approach also requires a-prior knowledge of the motion of the local zone ω.

In the case of the heat equation (**??**), the mass and stiffness matrices are given by:

$$M_{i,j} = \int_\Omega \phi_i \Phi_j\ d\Omega$$
$$K_{i,j} = \int_\Omega \vec{\nabla}\phi_i \cdot \vec{\nabla}\Phi_j\ d\Omega$$

where ϕ_i are the weight functions and Φ_j can be the classical shape functions or the function $g\ N_j$.

6 ONE-DIMENSIONAL BENCHMARK PROBLEMS

6.1 First Problem: A localized, moving heat source

6.1.1 Description of the problem

The objective was to use the X-FEM in a simple case of the heat equation where we knew the exact solution. The particular case I have chosen is to calculate the temperature in a rod composed of one phase, where the exact solution is actually the sum of two terms, both depending on time : one which can be represented by a classical Finite Element Method, the second constituing of a moving term with a very local behavior to illustrate the effectiveness of the X-FEM for this kind of problem:

$$T(x,t) = \underbrace{T_1 \cdot \sin(\frac{\Pi \cdot x}{l}) \cdot e^{-t/\tau}}_{classical} + \underbrace{T_2 \cdot e^{-[x-x_{front}(t)]^2} \cdot e^{-t/\tau}}_{for\ the\ X-FEM} \tag{12}$$

where x and t are the abscissa and the time, and x_{front} the position of the front. The function x_{front} is given by:

$$x_{front} = x_{front}(0) + V_{front} \cdot t \tag{13}$$

The appearance of this solution is the sum of a dissipative term on time (the sinus), and a moving term (the spike) as shown on Fig 2:

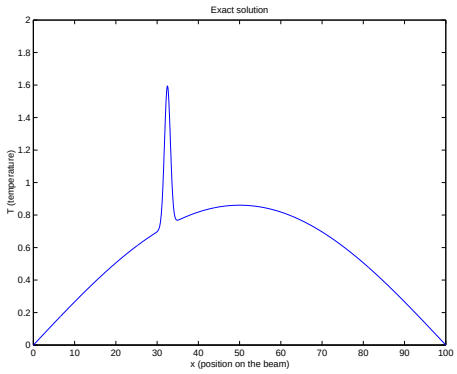


Figure 2: The exact solution.

In this case the forcing term on the right hand side of (2) is given by :

$$f(x,t) = \left(\frac{\Pi}{l}\right)^2 - 2 + 4(x - x_{front}(t))^2 - \frac{\Pi^2}{l}(2V_{front}(x - x_{front}(t))) \cdot e^{-[x-x_{front}(t)]^2} \cdot e^{-t/\tau} \tag{14}$$

I use the X-FEM with the classical piecewise shape functions N_i and the enrichment g is given by (see also Fig. 3 and Fig. 4) :

$$g(x,t) = e^{-[x-x_{front}(t)]^2} \tag{15}$$

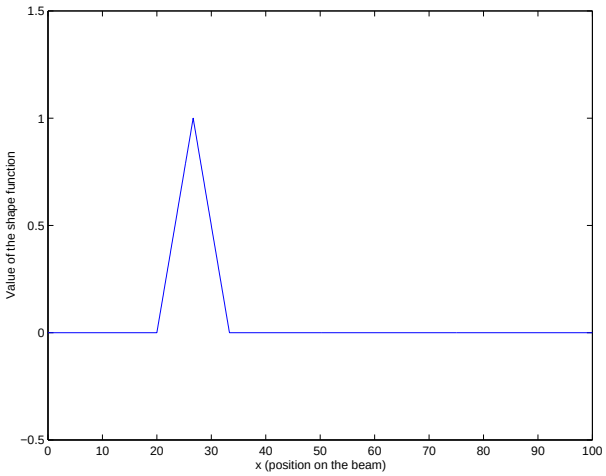


Figure 3: The classical shape function.

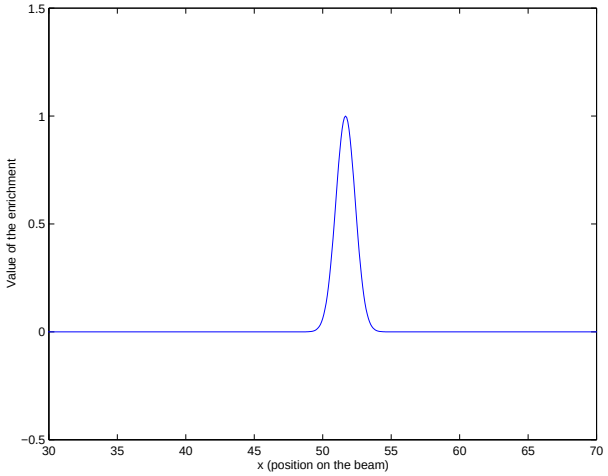


Figure 4: The enrichment function g.

We note that, using this method, the mass and stiffness matrices need to be rebuilt at every time step. But basicly, it is just some terms which are changing. In our case, we enriched only two nodes: the one which is immediately before the front, and the one which is immediately after the front (see Fig. **??**).

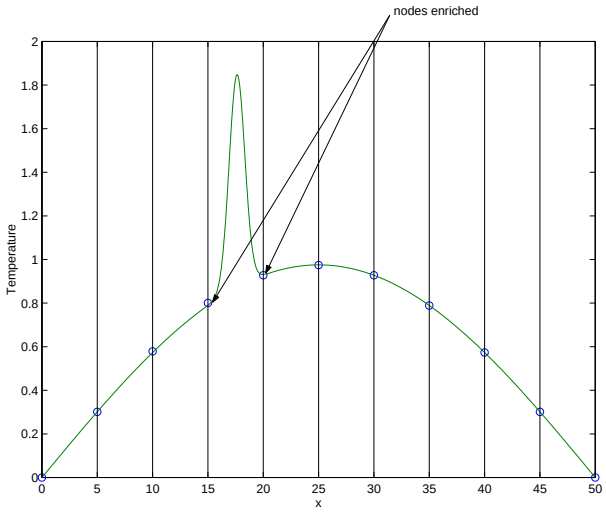


Figure 5: The enriched nodes.

The code corresponding to this case was coded with MATLAB. First I had to learn MATLAB and its environment. The structure of the code is very easy, but there is quite a bit of syntax to learn.

6.1.2 Numerical Results

The code was running with 20 elements. The appearance of the results is shown bellow:

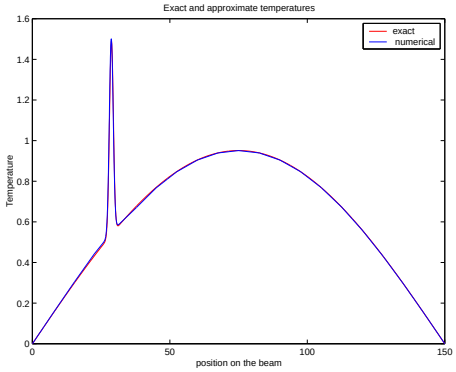


Figure 6: The results of the code.

We can note that the local behavior of the spike is very well computed. The H^1 and L^2 norm are used to measure the errors between the exact solution and the calculated solution:

$$L^2(T - T^h) = \sqrt{\int_l \left[T(x) - T^h(x)\right]^2 \, dl} \tag{16a}$$

$$H^1(T - T^h) = \sqrt{\int_l \left[T(x) - T^h(x)\right]^2 + \left[T_{,x}(x) - T^h_{,x}(x)\right]^2 \, dl} \tag{16b}$$

6.1.3 The research of the best α

I have studied the influence of α on the L^2 and H^1 norms of the error between the exact solution and the calculated solution with the X-FEM. The computation gave the evolution of the L^2 and H^1 norms as a function of $\alpha \in [1/2, 1]$ and the number of time iterations. The computation for $\alpha < 1/2$ was not made, because the method is only conditionaly stable for these values of α.The result is given Fig. 7 and Fig. 8.

We note that the best result corresponds to the case $\alpha \simeq .55$.

6.1.4 Comparison with the classical FEM

To prove the advantage of the X-FEM, we have computed some results to compare with the classical finite element method.

The stationary mesh:

The stationary mesh method consists of a classical finite element approximation with a fixed mesh over time. For the case here of one-dimension, the mesh is composed by linearly equally spaced points. The grid does not depend on the position of the front during time. Each element is a segment, its length is $l_{element} = \frac{l}{n_{elements}}$ where l represents the length of the domain and $n_{elements}$ the number of elements. This method does not need to rebuild the mass M and stiffness K matrices every time step.

The moving mesh:

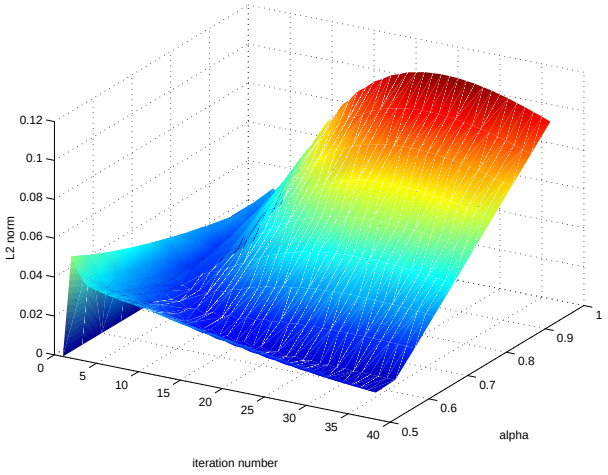


Figure 7: The influence of α on the L^2 norm.

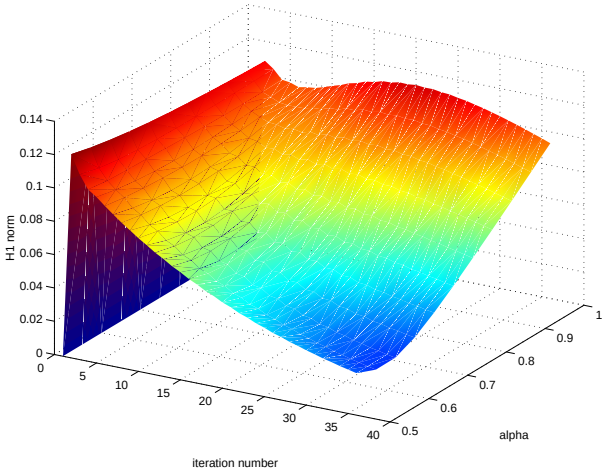


Figure 8: The influence of α on the H^1 norm.

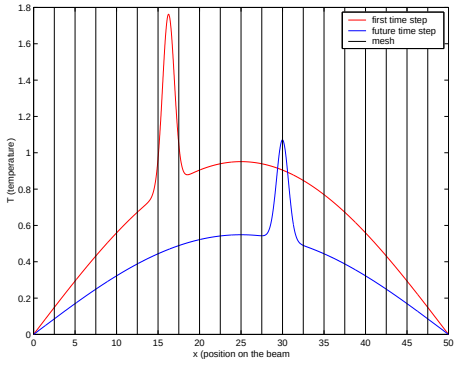


Figure 9: The unmoving mesh.

The moving mesh method consists of remeshing the domain at each time step depending upon the position of the front during time. The mesh is composed by a coarse background mesh and a fine mesh in the vicinity of the front. In our case, the coarse mesh is composed by 20 elements, the region where the mesh is thiner is $[x_{front} - \frac{l_{region}}{2}, x_{front} + \frac{l_{region}}{2}]$.

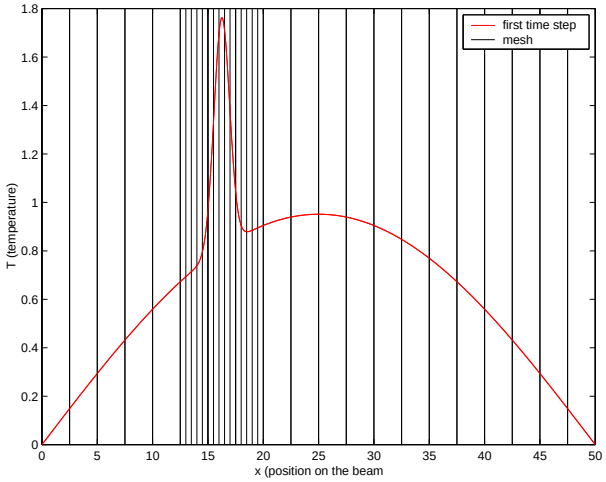


Figure 10: The moving mesh at an initial time step.

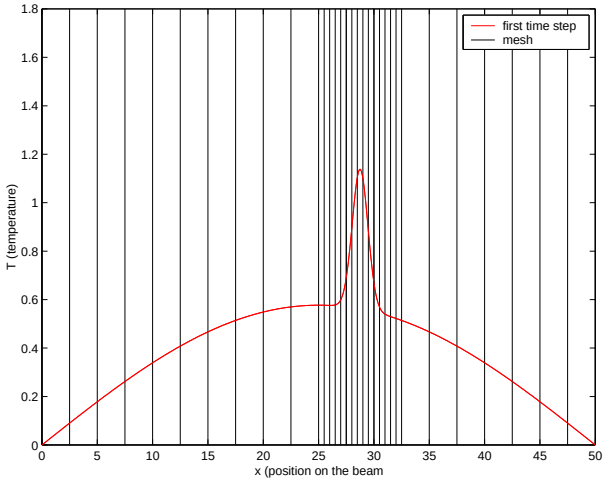


Figure 11: The moving mesh at a subsequent time step.

Results: the comparison between the methods:

The results have been computed for 36 time iterations. We have used the L^2 and H^1 norm to measure the error.

ΣL^2 and ΣH^1 represents the sum of the L^2 and H^1 error over the 36 time steps.

Number of elements	X-FEM		FEM, stationnary mesh	
	ΣL^2	ΣH^1	ΣL^2	ΣH^1
20	0.65	1.73	18.34	27.79
100			17.82	25.43
200			6.69	19.33
400			2.64	18.34

In the case of the classical FEM, with 10 elements and 11 nodes, this method exhibit poor accuracy in computing the solution near the front location, which is not surprising as the local feature is contained within a single element (see Fig. 12). It is only with a dramatic increase in the number of element that the computed solution begin to satisfy (see Fig. 13).

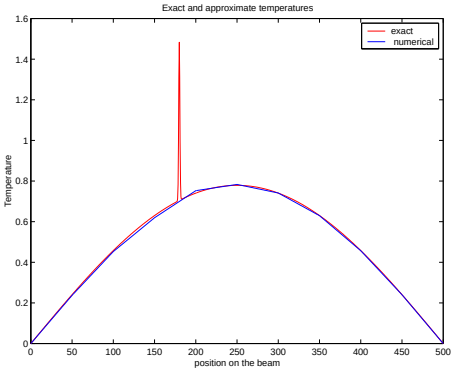


Figure 12: The finite element (10 elements) and exact solution.

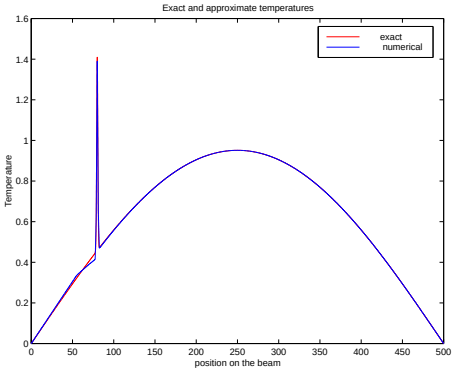


Figure 13: The finite element (400 elements) and exact solution.

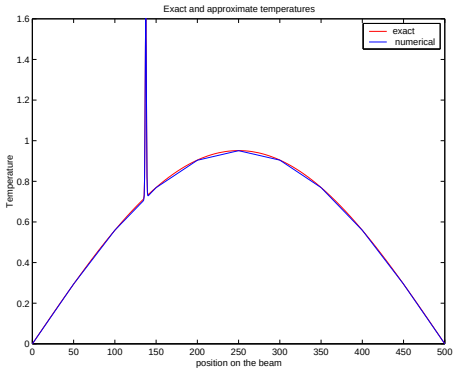


Figure 14: The X-FEM (10 elements) and exact solution.

6.2 Second Problem: the shock

6.2.1 Description of the problem

This time, I tried to solve a different kind of problem: a problem which includes discontinuous initial data. Consider the first order wave equation :

$$u_{,t} + a u_{,x} = 0$$
$$u_0(x) = \begin{cases} 1 & if\ x < x_{front}(0) \\ 0 & if\ x > x_{front}(0) \end{cases}$$

where a is the speed of the shock and the exact solution is $u_0(x - at)$.

The objective is to investigate the capability of the X-FEM for this kind of problem.

6.2.2 The classical FEM

The Galerkin approximation in this case is a classical approximation:

$$u(x) = \sum_{i=1}^{n} N_i(x) \tag{18}$$

where N_i are the classical piecewise linear shape functions. In this case, we don't need to integrate by parts. The formulation is:

$$\forall i \in 1, \cdots, N, \int_0^l u_{,t}\ N_i\ dx + a \int_0^l u_{,x}\ N_i\ dx = 0 \tag{19}$$

where the time stepping scheme is following:

$$u_{,t} = \frac{u^{n+1} - u^n}{\Delta t} \tag{20}$$

The classical FEM gives oscillations near the shock (see Fig. 15).

We note that the oscillations propagate in time, which was foreseeable, because of the history term on the right hand side.

6.2.3 The X-FEM

Here the chosen enrichment is the Heavyside function $H(x - x_{front})$ with:

$$H(x) = \begin{cases} 1 & if\ x < 0 \\ 0 & if\ x > 0 \end{cases} \tag{21}$$

and $x_{front} = x_{front}(0) + V_{front}\ t$. The formulation using the time stepping scheme (20) is:

$$\forall \delta u \in span\{\Phi_i\}^{n+1}, \int_0^l \delta u\ u^{n+1}\ dl = -a\Delta t \int_0^l \delta u\ u^n_{,x}\ dl + \int_0^l \delta u\ u^n\ dl \tag{22}$$

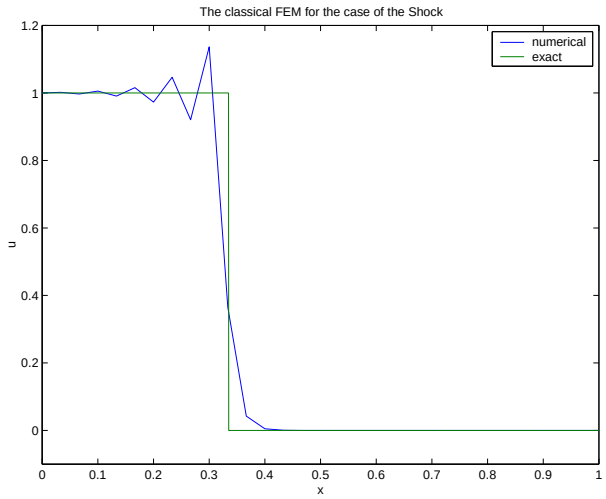


Figure 15: The results for the classical FEM.

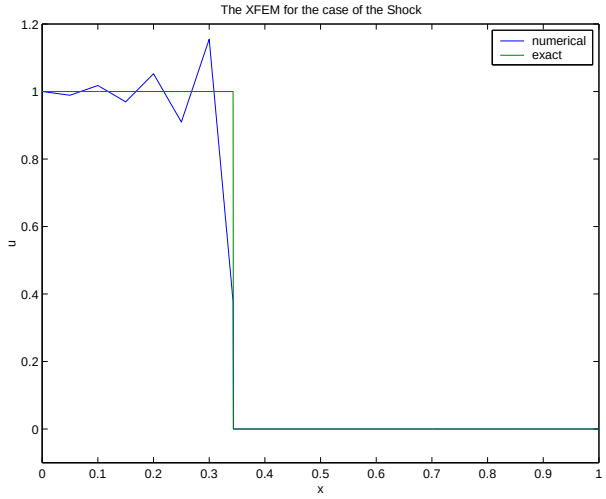


Figure 16: The results for the XFEM, first shot.

in assuming that:

$$H_{,x} = -\delta(x) \tag{23}$$

where $\delta(x)$ is the Dirac function. First, the approximation of $H_{,x}$ was $-\delta(x^n_{front})$.It gives the results shown on Fig.16.

But a better approximation of $H_{,x}$ between the time steps t^n and t^{n+1} was $-\delta(\frac{x^n_{front}+x^{n+1}_{front}}{2})$. This approximation permits to the algorithm to give the exact solution as shown on Fig. 17.

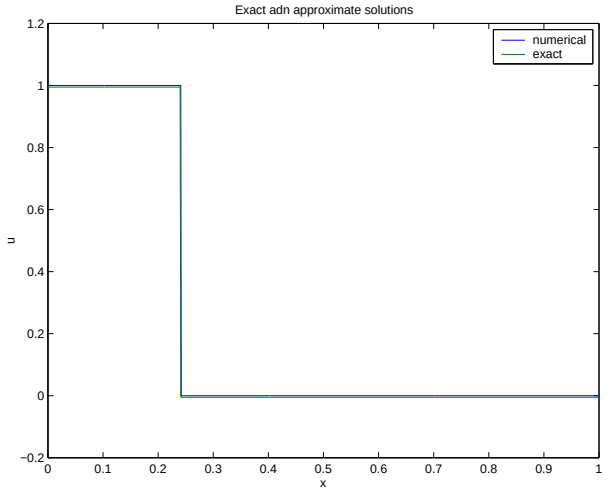


Figure 17: The X-FEM gives the exact solution.

7 APPLICATION TO PROBLEMS WITH PHASE TRANSFORMATIONS

7.1 Problem formulation

Consider the domain Ω which is divided into the regions Ω_1 and Ω_2 by the interface Γ_I as shown in Fig. 18. Knowing the initial temperature in the domain, $T(\vec{x}, 0)$, we seek to

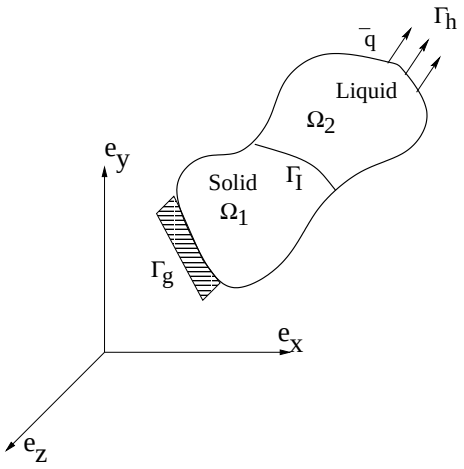


Figure 18: Domain split into Ω_1 and Ω_2 by the surface Γ_I.

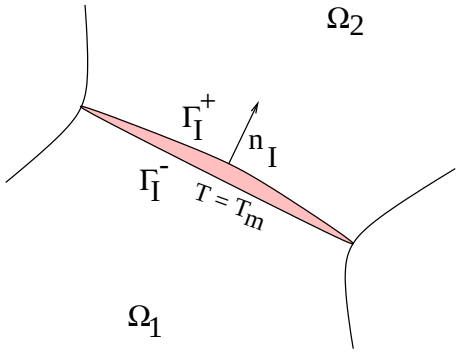


Figure 19: Zoom of the interface Γ_I.

determine the evolution of the function $T(\vec{x}, t)$ which satisfies the governing equations

$$c_1 \frac{\partial T}{\partial t} = \nabla \cdot (\boldsymbol{\kappa}_1 \cdot \overrightarrow{\nabla T}) \quad \text{in } \Omega_1 \text{ (solid phase)} \tag{24a}$$

$$c_2 \frac{\partial T}{\partial t} = \nabla \cdot (\boldsymbol{\kappa}_2 \cdot \overrightarrow{\nabla T}) \quad \text{in } \Omega_2 \text{ (liquid phase)} \tag{24b}$$

where c_i is the volumetric heat capacities of the phases, and $\boldsymbol{\kappa}_i$ are the thermal conductivity tensors.

The above are supplemented with external boundary conditions

$$T = T_g \quad \text{on } \Gamma_g \tag{25a}$$

$$(\boldsymbol{\kappa} \cdot \overrightarrow{\nabla T}) \cdot \overrightarrow{\boldsymbol{n}} = \bar{q} \quad \text{on } \Gamma_h \tag{25b}$$

with T_g and $\bar{q}$ prescribed temperatures and heat fluxes, respectively.

The motion of the surface Γ_I is described by a *Stefan* condition, written as:

$$L \frac{\partial \vec{s}}{\partial t} \cdot \boldsymbol{n}_I = (\boldsymbol{\kappa}_1 \cdot \overrightarrow{\nabla T})|_{\Gamma_I^-} \cdot \vec{\boldsymbol{n}}_I - (\boldsymbol{\kappa}_2 \cdot \overrightarrow{\nabla T})|_{\Gamma_I^+} \cdot \vec{\boldsymbol{n}}_I \tag{26}$$

where L is the volumetric latent heat of fusion. This equation, in turn with the condition that the temperature equal the melting temperature on the interface

$$T = T_m \quad \text{on } \Gamma_I \tag{27}$$

completes the description of the boundary value problem we wish to solve.

7.2 Discretization with the X-FEM

We now describe how to discretize this BVP with the X-FEM. Considering that the motion of the interface will be described by enrichment functions, there are three issues we need to be concerned with:

- the time-stepping algorithm;
- the constraint given by equation (27);
- the choice of enrichment functions.

These are addressed in the following subsections.

7.2.1 Variational Formulation and Time-Stepping

We consider the solution on the time interval $[0, t_f]$, partitioned into time steps as $[t^n, t^{n+1}]$. Considering the thermal conductivities to be isotropic and constant, we proceed as described in section 4 and write (24) at time step t^{n+1} as:

$$c_1 T_{,t}^{n+1} = \kappa_1 \Delta T^{n+1} \quad \text{on } \Omega_1 \tag{28a}$$

$$c_2 T_{,t}^{n+1} = \kappa_2 \Delta T^{n+1} \quad \text{on } \Omega_2 \tag{28b}$$

and we consider the general Trapezoidal rule:

$$T^{n+1} = T^n + \Delta t \left[\alpha T_t^{n+1} + (1-\alpha) T_t^n \right] \tag{29}$$

Substituting this equation into the above yields, and after proceeding to the variational formulation, we have the following expression:

$$\begin{aligned} &\int_{\Omega_1^{n+1}} \delta T^{n+1}\, T^{n+1}\, d\Omega + \frac{\alpha \kappa_1 \Delta t}{c_1} \int_{\Omega_1^{n+1}} \delta T_{,x}^{n+1}\, T_x^{n+1}\, d\Omega = \\ &\qquad \int_{\Omega_1^{n+1}} \delta T^{n+1}\, \left(T^n + (1-\alpha)\Delta t T_{,t}^n\right)\, d\Omega + \frac{\alpha \Delta t}{c_1} \int_{\Gamma_I} \delta T^{n+1}\, \underbrace{\kappa_1 \overrightarrow{\nabla T^{n+1}} \cdot \overrightarrow{n_I}}_{\textit{unknown quantity}}\, d\Gamma \end{aligned} \tag{30a}$$

$$\begin{aligned} &\int_{\Omega_2^{n+1}} \delta T^{n+1}\, T^{n+1}\, d\Omega + \frac{\kappa_2 \alpha \Delta t}{c_2} \int_{\Omega_2^{n+1}} \delta T_{,x}^{n+1}\, T_x^{n+1}\, d\Omega = \\ &\qquad \int_{\Omega_2^{n+1}} \delta T^{n+1}\, \left(T^n + (1-\alpha)\Delta t T_{,t}^n\right)\, d\Omega + \frac{\alpha \Delta t}{c_2} \int_{\Gamma_I} \delta T^{n+1}\, \underbrace{\kappa_2 \overrightarrow{\nabla T^{n+1}} \cdot \overrightarrow{n_I}}_{\textit{unknown quantity}}\, d\Gamma \end{aligned} \tag{30b}$$

where we have assumed that $\Gamma_h = \emptyset$.

We now focus on the terms in the above which arise on the freezing front Γ_I. These terms consist of a pair of heat fluxes:

$$q^- = \kappa_1 T_{,x} \,|_{\Gamma_I^-} \tag{31a}$$

$$q^+ = \kappa_2 T_{,x} \,|_{\Gamma_I^+} \tag{31b}$$

which are inherently associated with the motion of the interface through (26).

We note that we need to apply a time-step algorithm to the equation describing the motion of the interface. To this purpose, we rewrite equation (26) *at time step* t^n as:

$$\overrightarrow{s^n_{,t}} = \frac{1}{L}\left(\kappa_1 \overrightarrow{\nabla T^n|_{\Gamma_I^-}} - \kappa_2 \overrightarrow{\nabla T^n|_{\Gamma_I^+}}\right) \tag{32}$$

And now let's ignore conventional wisdom, and use a different algorithm , the Forward Euler algorithm (which is the case of the general Trapezoidal rule algorithm when $\alpha = 0$), yields:

$$\overrightarrow{s^n_{,t}} = \frac{\overrightarrow{s^{n+1}} - \overrightarrow{s^n}}{\Delta t} \tag{33}$$

which can be rewritten as:

$$\overrightarrow{s^{n+1}} = \overrightarrow{s^n} + \Delta t \cdot \overrightarrow{s^n_{,t}} \tag{34}$$

7.2.2 Using the LATIN method

The LATIN method Ladevèze (1998) is a general procedure for solving nonlinear problems in structural mechanics. It consists of three key features:

1. A separation of global, linear quantities from local, nonlinear quantities;
2. A two-step iterative procedure which "builds" the solution;
3. An ad-hoc space-time procedure.

The first two elements were adopted in Champaney (1996) to model assembling structures, and in Dolbow et al. (2000) in conjunction with the X-FEM to model crack growth with frictional contact acting on the crack faces.

For the present problem, knowing all quantities at time t^n, we wish to determine the temperatures T^{n+1} and interface fluxes (31) at the new time step such that the weak form (30) is satisfied and the temperature at the interface is equal to the melting temperature T_m. We will refer to the latter constraint as the *constitutive law* on the solid-liquid interface.

We adopt an iterative strategy to resolve this problem. For notational clarity, we will drop the superscript n from all quantities, and assume the iterative procedure is invoked in moving from time step n to $n+1$. We will also use subscripts m and $m+1$, etc., to refer to iteration number.

The iterative procedure adopted by the LATIN method begins by considering two sets of solution spaces. The variables of interest are the temperature T, its value on the interface w:

$$w^+ = T|_{\Gamma_I^+}, \quad w^- = T|_{\Gamma_I^-} \tag{35}$$

the temperature gradient $T_{,x}$ and the heat fluxes on the interface (31). We use the compact notation:

$$v = (T, w, \overrightarrow{\nabla T}, \overrightarrow{q}) \tag{36}$$

to define the solution spaces. Specifically, we let:

$$\boldsymbol{A}_d = \{v \text{ satisfying (30)}\} \tag{37}$$

and

$$\boldsymbol{I} = \{v \text{ satisfying (27)}\} \tag{38}$$

The goal is then to find the element v^{AI} located at the intersection of $\boldsymbol{A}_d$ and $\boldsymbol{I}$, represented geometrically in Fig. 20. We use the superscript A and I to denote an element v in $\boldsymbol{A}_d$ or $\boldsymbol{I}$, respectively. The iterative strategy begins with an initial element v_0^A in $\boldsymbol{A}_d$ and builds a sequence of approximate solutions v_1^A, ..., v_m^A, v_{m+1}^A, ... until convergence. A given iteration $v_m^A \to v_{m+1}^A$ involves two steps: $v_m^A \to v_m^I$ and $v_m^I \to v_{m+1}^A$ as shown in Fig. 20. The iterations stop when the "distance" between v_m^I and v_{m+1}^A (measured with an appropriate norm) is below a specified tolerance.

Two successive approximations are always tied by a given search direction. The direction used when going "down" from the set $\boldsymbol{A}_d$ to the set $\boldsymbol{I}$ is denoted by $\boldsymbol{E}^{AI}$ whereas the direction when going "up" from $\boldsymbol{I}$ to $\boldsymbol{A}_d$ is denoted by $\boldsymbol{E}^{IA}$. The appropriate choice of the search directions in order to achieve (fast) convergence is an important aspect of the LATIN method (see Ladevèze (1998)).

The next two sections describe the two steps in the method:

- Going down from v_m^A to v_m^I in the search direction $\boldsymbol{E}^{AI}$. We will see that this step is local and may be solved independently for each pair of points facing each other on the interface.
- Going up from v_m^I to v_{m+1}^A in the search direction $\boldsymbol{E}^{IA}$. This step involves a global solve.

1. *The local step: update on the interface*

   The local update involves determining a new estimate v_m^I for the fluxes and temperatures on the interface given quantities obtained from a solution $v_m^A \in \boldsymbol{A}_d$. The new approximation is required to satisfy the constitutive law (i.e. $v_m^I \in \boldsymbol{I}$). Additional equations are provided by the search direction.

   To move from an element $v_m^A \in \boldsymbol{A}_d$ to $v_m^I \in \boldsymbol{I}$, the search direction $\boldsymbol{E}^{AI}$ is associated with a linear operator k_0. The search equations are then
   $(v_m^I - v_m^A) \in \boldsymbol{E}^{AI} \Rightarrow$

$$\overrightarrow{q_m^{I+}} - \overrightarrow{q_m^{A+}} = k_0(w_m^{I+} - w_m^{A+}) \quad \text{on } \Gamma_I^+ \tag{39a}$$

$$\overrightarrow{q_m^{I-}} - \overrightarrow{q_m^{A-}} = k_0(w_m^{I-} - w_m^{A-}) \quad \text{on } \Gamma_I^- \tag{39b}$$

   The above can be written more compactly as

$$\overrightarrow{q_m^I} \cdot \overrightarrow{n_I} - \overrightarrow{q_m^A} \cdot \overrightarrow{n_I} = k_0(w_m^I - w_m^A) \quad \text{on } \Gamma_I \tag{40}$$

   with the understanding that the relationship holds on both Γ_I^+ and Γ_I^-, separately. This compact form will be used in the remainder of this section when convenient.

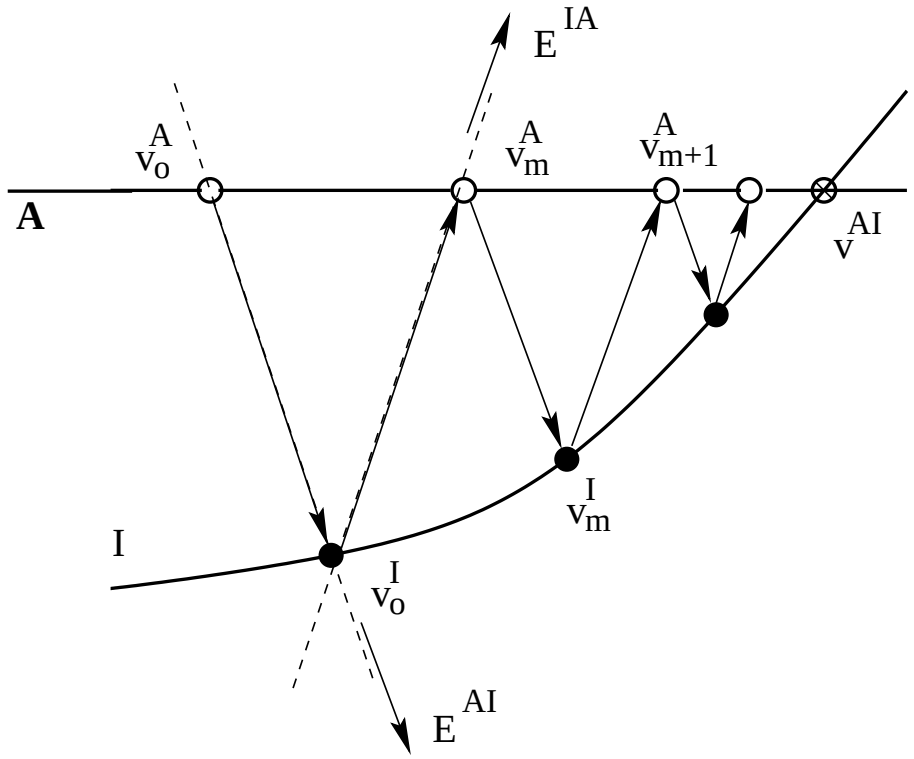


Figure 20: The iterative procedure in the LATIN algorithm.

The process of determining v_m^I given v_m^A then involves solving the above search equations in conjunction with the constitutive law on the interface

$$v_m^I \in \boldsymbol{I} \Rightarrow w_m^+ = w_m^- = T_m \quad \text{on } \Gamma_I \tag{41}$$

The resulting closed form equations for (q_m^I, w_m^I) are very simple in this case. They are given by

$$w_m^{I+} = w_m^{I-} = T_m \tag{42a}$$

$$\overrightarrow{q_m^I} \cdot \overrightarrow{n_I} = \overrightarrow{q_m^A} \cdot \overrightarrow{n_I} + k_0(w_m^I - w_m^A) \tag{42b}$$

2. *The global step*
We wish to move from $v_m^I \in \boldsymbol{I}$ to $v_{m+1}^A \in \boldsymbol{A}_d$. This is accomplished through a global solve in conjunction with additional equations provided by the search direction $\boldsymbol{E}^{IA}$ which is also associated with the k_0 operator on the interface. With a known element v_m^I, the search equations are then
$(v_{m+1}^A - v_m^I) \in \boldsymbol{E}^{IA} \Rightarrow$

$$\overrightarrow{q_{m+1}^A} \cdot \overrightarrow{n_I} - \overrightarrow{q_m^I} \cdot \overrightarrow{n_I} = -k_0(w_{m+1}^A - w_m^I) \text{ on } \Gamma_I \tag{43}$$

This equation is used in conjunction with the weak form of the governing equations (30), by solving for q_{m+1}^A and making the substitution in the surface integrals on Γ_I.

The problem to be solved is then: Find (T^{n+1}_{m+1}, w_{m+1}) which satisfies

$$\begin{aligned}
&\forall \delta T \in \{\Phi\}_{n+1}, \\
&\int_{\Omega^{n+1}} \delta T^{n+1}\, T^{n+1}_{m+1}\, d\Omega + \frac{\kappa \Delta t}{c} \int_{\Omega^{n+1}} \nabla \delta T^{n+1}\, \nabla T^{n+1}_{m+1}\, d\Omega + \frac{\alpha \Delta t}{c} \int_{\Gamma_I} k_0 T_{m+1}|_{\Gamma_I} \cdot \delta T^{n+1}\, dS = \\
&\int_{\Omega^{n+1}} \delta T^{n+1}\, \left[T^n + (1-\alpha)\cdot \Delta t T^{n+1}_t\right]\, d\Omega + \frac{\alpha \Delta t}{c} \int_{\Gamma_I} (q^I_m + k_0 w^I_m)\cdot \delta T^{n+1}\, d\Gamma_I
\end{aligned} \tag{44}$$

where it is understood that

$$\begin{aligned}
\Omega &= \Omega^{n+1}_1 \cup \Omega^{n+1}_2 \\
\kappa &= \kappa_i \quad \text{on } \Omega^{n+1}_i \\
c &= c_i \quad \text{on } \Omega^{n+1}_i
\end{aligned}$$

After solving (44), the local fluxes on the interface at step $m+1$ are given by

$$\overrightarrow{q^A_{m+1}} \cdot \overrightarrow{n_I} = \overrightarrow{q^I_m} \cdot \overrightarrow{n_I} + k_0 (w^I_m - w^A_{m+1}) \tag{45}$$

with

$$w^A_{m+1} = T^A_{m+1}|_{\Gamma_I} \tag{46}$$

Iteration over subsequent steps continues until convergence is met, measured with an appropriate norm.

7.2.3 The Time Projection

After solving this system, we know the temperature T at instant $n+1$ and we would like to know its time derivative $T^{n+1}_{,t}$ in order to apply the equation (30). We proceed as presented in section 4.Basicly $\{T^{n+1}, T^{n+1}_{,t}\} \subset span\{\Phi_i\}_{n+1}$ and $T^n \in span\{\Phi_i\}_n$ where $\{\Phi_i\}_n$ and $\{\Phi_i\}_{n+1}$ are the sets of the basis functions at time steps n and $n+1$. The equation which links these variables is the time-stepping scheme:

$$T^{n+1}_{,t} = \frac{T^{n+1} - T^n - (1-\alpha)\Delta t T^n_{,t}}{\alpha \Delta t} \tag{47}$$

We multiply the last expression by $\delta T^{n+1} \in \{\Phi\}_{n+1}$ and integrate on Ω:

$$\begin{aligned}
\forall \delta T^{n+1} \in \{\Phi_i\}_{n+1}, \int_\Omega \delta T^{n+1}\, T^{n+1}_{,t}\, d\Omega = \\
\frac{1}{\alpha \Delta t} \int_\Omega T^{n+1}\, d\Omega - \int_\Omega \delta T^{n+1}\, \frac{T^n + (1-\alpha)\Delta t T^n_{,t}}{\alpha \Delta t}\, d\Omega
\end{aligned} \tag{48}$$

7.2.4 The X-FEM Approximation

The standard finite element approximation to the temperature field takes the form:

$$T^h(\vec{x}) = \sum_I N_I(\vec{x}) T_I \tag{49}$$

where N_I are nodal shape functions and T_I are the corresponding coefficients. The sum is taken over all nodes in the mesh. When the shape functions are piecewise linear, the above construction has the gradient

$$\nabla T^h(\vec{x}) = \sum_I \nabla N_I(\vec{x}) T_I \tag{50}$$

which can only be discontinuous *across element boundaries*. This is, essentially, one of the motivations for using a moving mesh technique, where the element and interface boundaries always coincide.

In the X-FEM, we extend the above using the construction

$$T^h(\vec{x}) = \underbrace{\sum_I N_I(\vec{x}) T_I}_{\textit{classical approximation}} + \underbrace{\sum_J N_J(\vec{x}) \cdot g(\vec{x})\ a_J}_{\textit{enrichment}} \tag{51}$$

where $g(\vec{x})$ is the enrichment function, a_J are enriched degrees of freedom and $J \subset I$ is a subset of the nodes in the mesh. In the case of phase transformations, we take J to consist of all nodes whose shape function supports contain the *liquid-solid interface*:

$$J = \{I | \exists\ x \in \Gamma_I / N_I(\vec{x}) \neq 0\} \tag{52}$$

where x_f is the location of the freezing front. This construction has a gradient

$$\nabla T^h(\vec{x}) = \sum_I \nabla N_I T_I + \sum_J \left[\nabla N_J \cdot g\ + N_J \cdot \nabla g\right] a_J \tag{53}$$

which can be discontinuous wherever $\vec{\nabla} g$ is discontinuous. This represents the basic idea for this work: the motion of the interface is modeled through a change in the enrichment function g.

7.3 Summary of the algorithm

We suppose that the fields at time step n are known: temperature, position of the interface. The algorithm is as follows:

1. We calculate the velocity of the interface $\overrightarrow{s^{n+1}}$ using (26).
2. We predict the position of the interface using (34).
3. We set up the set of the shape and test functions with this position of the interface.

4. (a) We initialize the values for the LATIN.
   (b) We calculate the values correspond to the local step of the LATIN.
   (c) We build the mass and stiffness matrices and the right hand side from (44)
      - Mass matrix:

$$M_{i,j} = \int_\Omega \Phi_i^{n+1}\Phi_j^{n+1}\ d\Omega + \frac{\alpha\Delta t}{c}\int_{\Gamma_I} k_0\Phi_i^{n+1}\Phi_j^{n+1}\ d\Gamma_I \tag{54}$$

      - Stiffness matrix:

$$K_{i,j} = \frac{\alpha\Delta t}{c}\int_\Omega \kappa\overrightarrow{\nabla\delta T_i^{n+1}}\cdot\overrightarrow{\nabla\delta T_j^{n+1}}\ d\Omega \tag{55}$$

      - Right Hand Side:

$$\begin{aligned} F_i^{n+1} = \int_\Omega \Phi_i^{n+1}\ [T^n + (1-\alpha)\cdot\Delta t T_t^n]\ d\Omega + \\ \frac{\alpha\Delta t}{c}\int_{\Gamma_I}(q_m^I + k_0 w_m^I)\cdot\Phi_i^{n+1}\ d\Gamma_I \end{aligned} \tag{56}$$

      with $T^n = \sum_{I\cup J} T_i^n\cdot\Phi_i(x)$ and $T_t^n = \sum_{I\cup J}(T_t^n)_i\cdot\Phi_i(x)$, where $(T^n)_i$ and $(T_t^n)_i$ are the composante of T^n and T_t^n on Φ_i which is the shape function of time step n.
   (d) We then solve the given system $(M+K)\cdot T^{n+1} = F^{n+1}$.
   (e) We then measure the distance between the both elements (on I and A). If this distance is inferior to the tolerance, we go to the next step, if not, we return to the (b).
5. We make the time projection, that means that we solve the system: $\Pi T_t^{n+1} = \tau^{n+1}$, where :
   - $\Pi_{i,j} = \int_\Omega \Phi_i^{n+1}\Phi_j\ d\Omega$
   - $\tau_i^{n+1} = \frac{1}{\alpha\Delta t}\left[\sum_j T_j^{n+1}\int_\Omega \Phi_j^{n+1}\Phi_i^{n+1} - \left(\sum_j (T_j^n + (1-\alpha)\Delta t(T_{,t}^{n+1})_j)\int_\Omega \Phi_j^n\Phi_i^{n+1}\right)\right]$
6. Then we go to the first step.

7.4 The research of the best enrichment function

To solve this problem, I examined the one-dimensional case, for a two phases problem. The objective is to find the enrichment function g which gives the smallest amount of error *regardless of the position of the front in the element.* I assumed that this function was the same in dynamic case and in the steady-state case.

The problem is basicly the following: find a g function which gives the less amount of error for solving a field where there is a discontinuity in the spatial derivative (see Fig. 21).

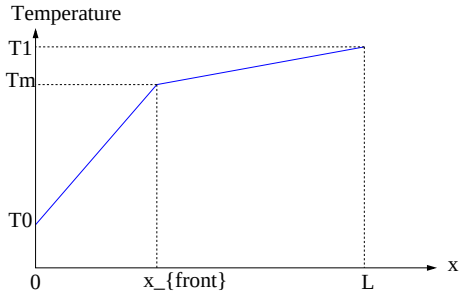


Figure 21: The exact field to solve.

The main idea is that the Gallerkin approximation T^h looks the best like the exact solution T.

To best describe the local behavior of the solution, we have assumed that the enrichment was equal to 0 in the elements non cut by the interface.

$$\forall x \notin Element_{i_0} / x_{front} \in Element_{i_0}, g(x) = 0, \tag{57a}$$

$$\forall x \in Element_{i_0}, T^h(x) = T_{i_0} N_{i_0} + T_{i_0+1} N_{i_0+1} + g(x) \left(a_{i_0} N_{i_0} + a_{i_0+1} N_{i_0+1}\right) \tag{57b}$$

The general constraint are:

- T^h must be continuous.
- $\frac{\partial T^h}{\partial x}$ must be discontinuous.

From the second proposition, we can deduce that the function $g \cdot (a_{i_0} N_{i_0} + a_{i_0+1} N_{i_0+1})$ is continuous, which means:

$$g(x^-_{front}) \left(a_{i_0} N_{i_0}(x_{front}) + a_{i_0+1} N_{i_0+1}(x_{front})\right) = g(x^+_{front}) \left(a_{i_0} N_{i_0}(x_{front}) + a_{i_0+1} N_{i_0+1}(x_{front})\right) \tag{58}$$

This leads to two possibles cases:

- $a_{i_0} N_{i_0}(x_{front}) + a_{i_0+1} N_{i_0+1}(x_{front}) \neq 0$: in this case, g is continuous.
- $a_{i_0} N_{i_0}(x_{front}) + a_{i_0+1} N_{i_0+1}(x_{front}) = 0$: in this case, g can be discontinuous.

7.4.1 First idea: the discontinuous function

The first idea is to use a discontinuous function. I choose the *Heavyside* function which is defined by:

$$H = \begin{cases} -1 & \text{for } x < x_{front} \\ 1 & \text{for } x > x_{front} \end{cases} \tag{59}$$

We can justify this function by saying that the exact solution can be a linear combinaison of $\{N_1, \cdots, N_N\} \bigcup \{HN_{i_0}, HN_{i_0+1}\}$in a one element case. Indeed, we can explain it with some schematics on Fig. 22.

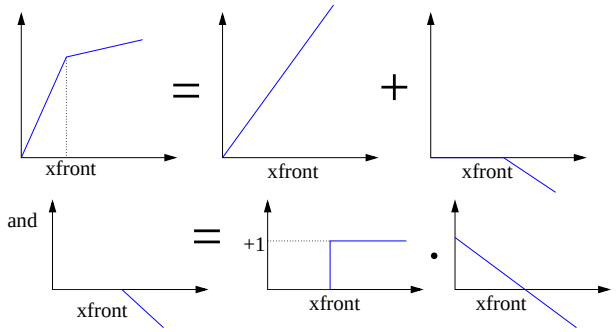


Figure 22: The Heavyside function.

7.4.2 Second idea: an hyperbolic continuous function

This idea is based on the expression:

$$g(x) = \frac{T - T_{classical}}{a_{i_0} N_{i_0} + a_{i_0+1} N_{i_0+1}} \tag{60}$$

In this case, we can then build a enrichment function which resembles the following:

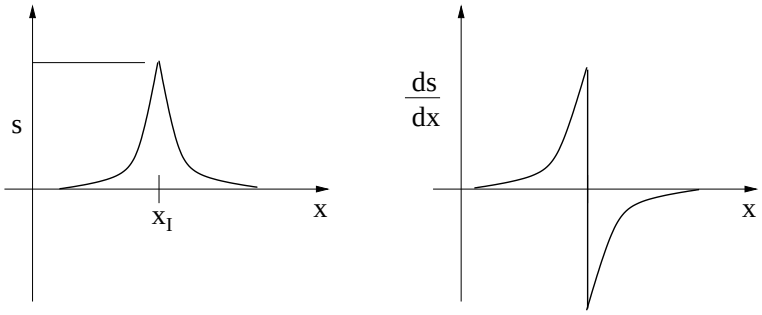


Figure 23: The hyperbolic enrichment.

7.4.3 Third idea: a "classical, continuous shape function"

This idea is based on the expression:

$$\sum_I N_I(x) \cdot g(x) = g(x) \tag{61}$$

So the enrichment function $g(x)$ brings just the particular form in the location of the front (see Fig. 24) which gives an enrichment like a shape function (see Fig. 25).

7.4.4 Numerical results

The numerical experiments were made with a code built with MATLAB. The errors were measured with the classical L^2 andH^1 norms, for 100 different positions of the front linearly spaced in the element.

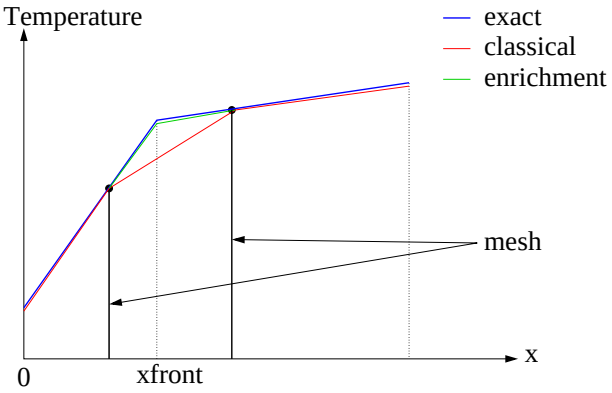


Figure 24: The enrichment shape.

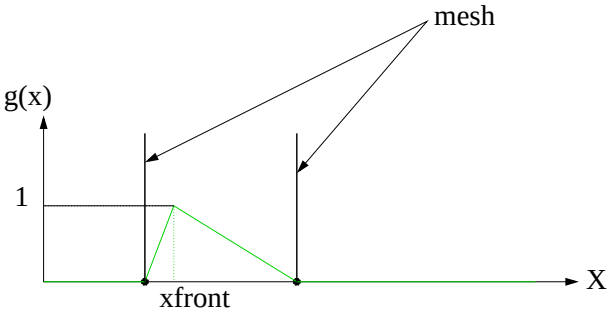


Figure 25: The enrichment shape.

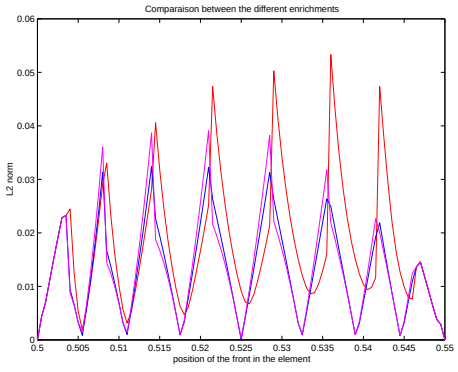


Figure 26: The different enrichment compared with L^2 norm.

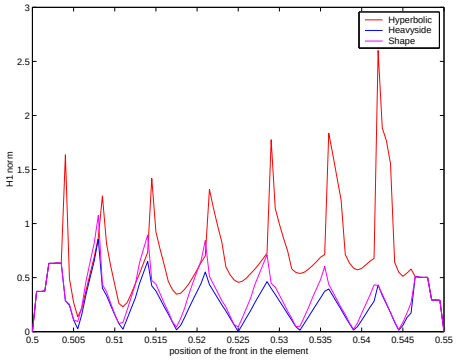


Figure 27: The different enrichment compared with H^1 norm.

We can note that the best results are obtained with the Heavyside function, even if the "classical shape" is very close.

7.5 Numerical examples in two dimensions

The work for the two dimensional code was to implement the presented method in the code my advisor used. The code was written in C++. I had to learn a lot of things on C++, the basis of the object oriented programming (structures, classes, pointers, etc ...) from Budd (1998) and Deitel and Deitel (1998) and a lot concerning the way of programming of this code: The way of working of the code, how to use the big number of different existing routines and how to code new routines. In the 2-D code, I had implemented the enrichment "shape" for accuracy reasons. Indeed, with a continuous enrichment function, we can evaluate the temperature at more points than with a discontinuous enrichment function.

7.5.1 The way to integrate the matrix equations

Given our domain Ω divided into elements, we begin by subdividing the elements which are cut by the interface in subelements. In this case, it is not necessary to divide the elements in subtriangles respecting the geometry of the interface like in the case of crack growth. Indeed the move of the cracks and the move of the solid-liquid interface is different (see Fig. 28 and Fig. 29).

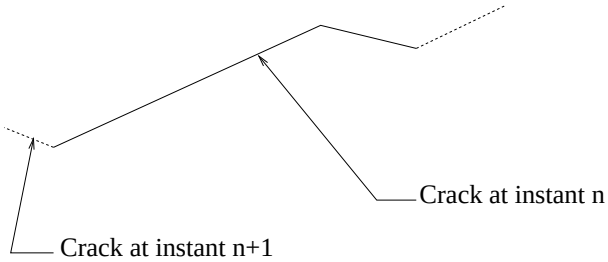


Figure 28: The move of the crack.

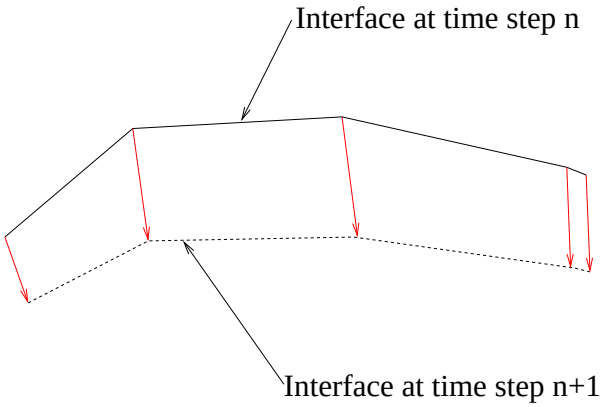


Figure 29: The move of the interface.

The enriched degrees of freedom are the dofs of each subelements cut by the interface. Each element or subelement is composed by 4 Gauss points located regularly in the element.

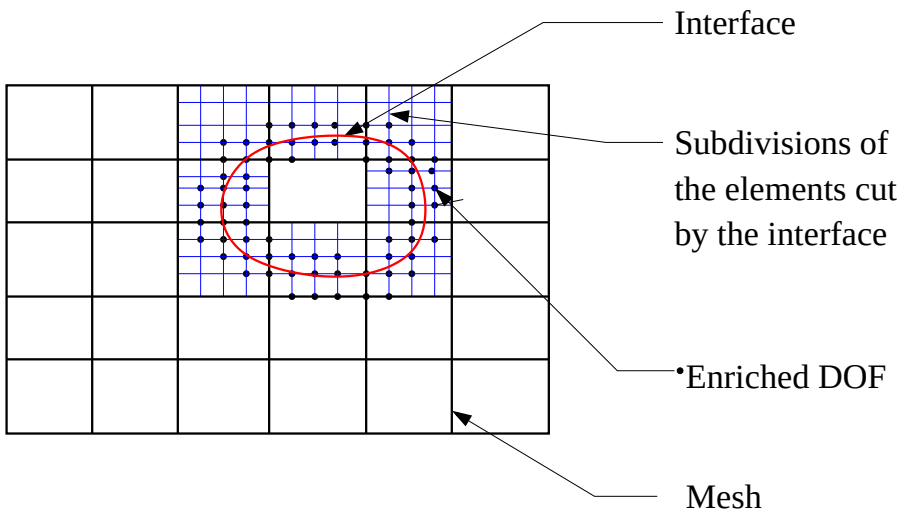


Figure 30: The mesh used in the X-FEM.

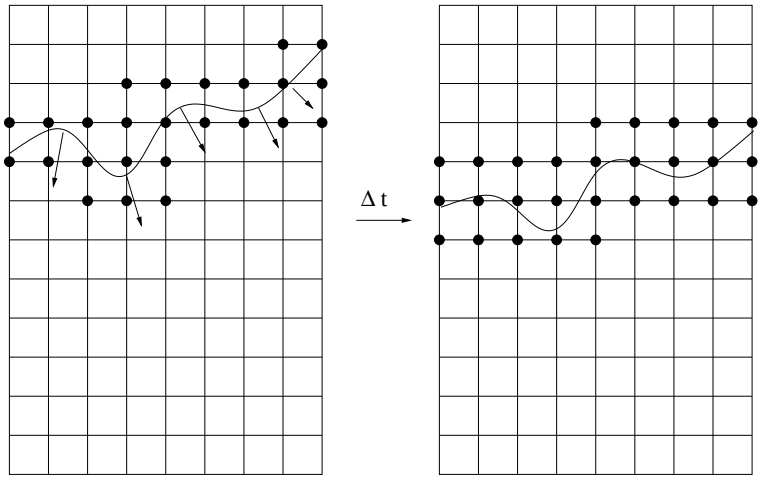


Figure 31: The move of the enriched nodes.

7.5.2 Two phase Stefan Problems

The problem considered is a two-phase Stefan problem. The domain is a square of length 2 m where the boundary conditions are following:

$$T(x = -1, y) = -10^o C \tag{62a}$$
$$T(x = 1, y) = 4^o C \tag{62b}$$
$$\vec{\nabla T} \cdot \vec{n} = 0 \text{ on } y = -1 \text{ and } y = 1 \tag{62c}$$

The material coefficients are :

Density in liquid domain: $c_1 = 0.62\ cal.K^{-1}.cm^{-3}$
Density in solid domain: $c_2 = 0.49\ cal.K^{-1}.cm^{-3}$
Thermic conductivity in liquid domain: $\kappa_1 = 0.0069\ cal.cm^{-1}.sec^{-1}.K^{-1}$
Thermic conductivity in solid domain: $\kappa_2 = 0.0096\ cal.cm^{-1}.sec^{-1}.K^{-1}$
Latent Heat: $L = 19.2\ cal.cm^{-3}$

This problem was solved using the adaptation of the generalized Trapezoidal rule with $\alpha = 0.6$ and a time step $\Delta t = 2\ s$.The initial solution is composed by two planes, an horizontal one at $T = 4^o C$ and the other one from $T = -10^o C$ to $T = 4^o C$ (see Fig. 32)

The interface is evolving from $x = -0.8$ to $x \simeq 0.6$, case of the steady-state. The figure 33 gives the evolution of the interface with in comparison to the values given by Lynch and O'Neill (1981). The figure 34 gives the evolution of the profile of the temperature during time.

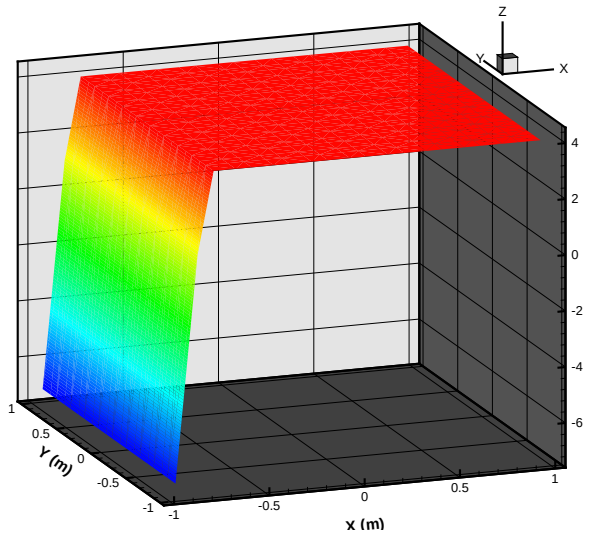


Figure 32: The initial distribution.

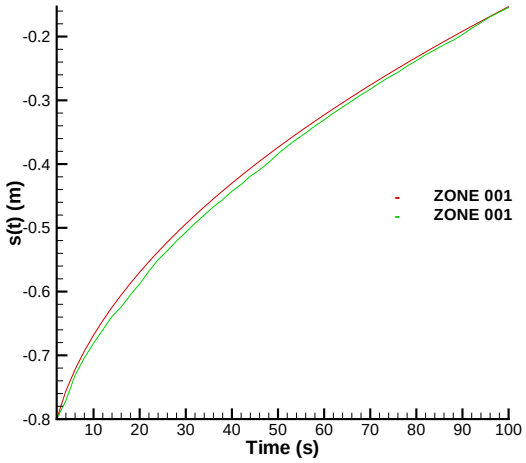


Figure 33: The evolution of the interface.

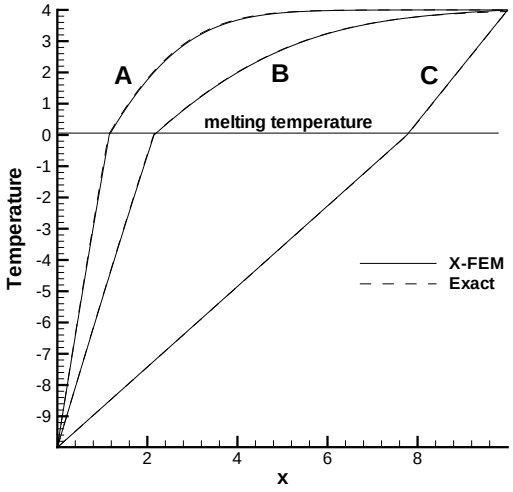


Figure 34: The evolution of the profile for A)t = 180, B) t = 626 and C) steady-state.

7.5.3 Square of liquid Alluminium

The next example is a square of aluminum. The boundary conditions are:

$$T = 550^o C \text{on the square boundary} \tag{64}$$

The figure 35 shows the initial distribution of the temperature. The temperature at the top is $T = 750^o C$. The material coefficients are :

Density in liquid domain: $c_1 = 0.669\ cal.K^{-1}.cm^{-3}$
Density in solid domain: $c_2 = 0.669\ cal.K^{-1}.cm^{-3}$
Thermic conductivity in liquid domain: $\kappa_1 = 0.0548\ cal.cm^{-1}.sec^{-1}.K^{-1}$
Thermic conductivity in solid domain: $\kappa_2 = 0.0548\ cal.cm^{-1}.sec^{-1}.K^{-1}$
Latent Heat: $L = 250\ cal.cm^{-3}$

The computation was made with using the adaptation of the generalized Trapezoidal rule with $\alpha = 0.6$ and a time step $\Delta t = 0.5\ s$.

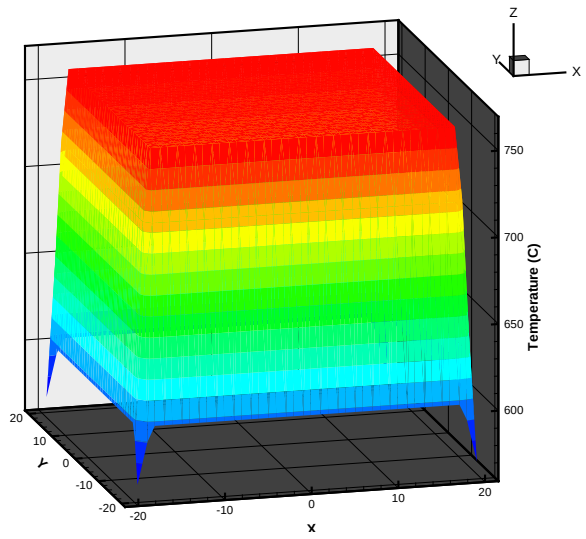


Figure 35: The initial temperature.

The appearance of several next time steps are given on Fig.36, 37,38 and 39. We can note that the liquid region leads the melting temperature very quicly, even if the interface liquid-solid moves slowly.

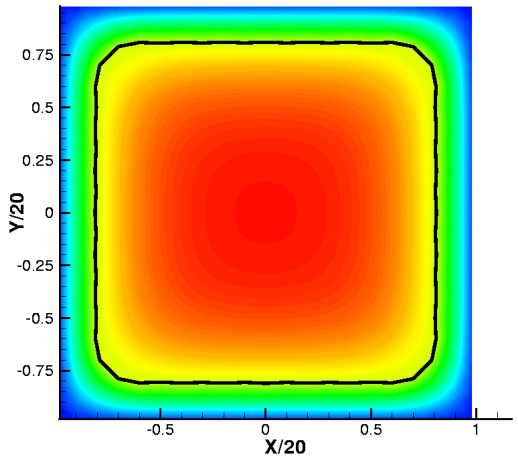


Figure 36: The Temperature after 5 s.

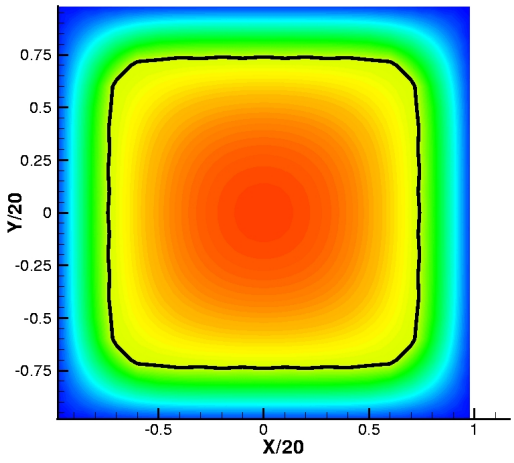


Figure 37: The Temperature after 10 s.

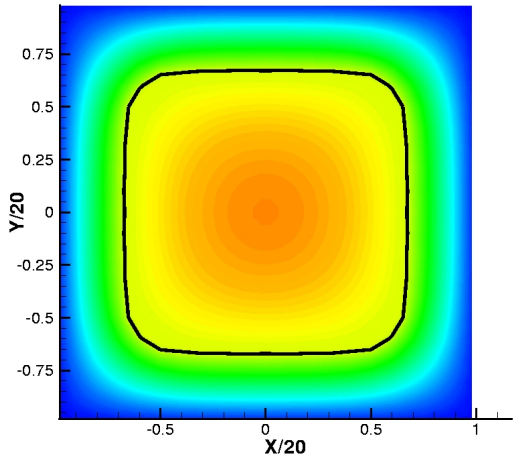


Figure 38: The Temperature after 15 s.

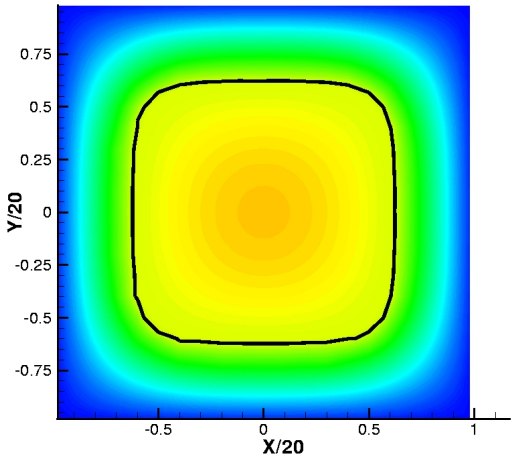


Figure 39: The Temperature after 20 s.

8 CONCLUSION

This stage enriched me a lot. First I learned a lot of things.

- The first enrichment for me was the discovery of a foreign country: its way of life, the language. I discovered a lot of different cultures (Duke was crowded with foreign students): a lot of European people, but also from China and Turkey.

- I learned a lot about the media of the research in the United States. A lot of things are different from what happenses in France. For Example, the students are initialized very early for the research. Just after the Graduation (which is equivalent to the License in France), the students have to do research during their Master's Degree. The other main difference with what I knew from France was the financing fundings of the research, which is more a competition than in France.

- The others things I learned were the scientific knowledge: the Finite Element Method sure, Continuous Mechanics, Physical Behavior of the phase transformations, but also a lot concerning the programming (C++, MATLAB).

Concerning the presented method, the application of the X-FEM for problem including "difficult locations" (discontinuities, high variations, cracks, ...), in other words for problems invloving non-linear constitutive laws. proved to be a very interesting technique. In several phase transformations cases, the presence of terms of interface speed in the equation of temperature on the interface can make avoid the Latin in the presented algorithm, which could be the algorithm faster.

Acknowledgements

I would like to acknowledge all the following persons:

- Christophe Daux who has put me in contact with Duke University.
- my adviser Dr. John Dolbow, first to have me proposed me this work, to have welcomed me in the US and at Duke University, but also fro his patience with me and the knoledge in Finite Elements and programming he gave me.
- all the staff of the Department of Civil and Environmental Engineering for his help in the administrative approach.
- the International House, especially Lisa Giragosian who provided me a friendly place to stay.

References

Budd, T. (1998). *Data Structures in C++, using the Standard Template Library*. Addison Wesley.

Champaney, L. (1996). *Une nouvelle approche modulaire pour l'analyse d'assemblages de structures tridimensionnelles*. Ph. D. thesis, Ecole Normale Superieure de Cachan, France.

Daux, C., N. Moës, J. Dolbow, N. Sukumar, and T. Belytschko (2000). Arbitrary branched and intersecting cracks with the extended finite element method. *International Journal for Numerical Methods in Engineering 48*, 1741 – 1760.

Deitel and Deitel (1998). *C++, How to program*. Eyrolles.

Dolbow, J. (1999). *An extended finite element method with discontinuous enrichment for applied mechanics*. Ph. D. thesis, Northwestern University.

Dolbow, J., N. Moës, and T. Belytschko (2000). An extended finite element method for modeling crack growth with frictional contact. *Computer Methods in Applied Mechanics and Engineering*. Submitted for publication.

Ghosh, S. and S. Moorthy (1993). An arbitrary lagrangian-eulerian finite element model for heat transfer analysis of solidification processes. *Numerical Heat Transfer: Part B 23*, 327–350.

Ladevèze, P. (1998). *Nonlinear Computational Structural Mechanics*. New York: Springer-Verlag.

Lynch, D. and K. O'Neill (1981). Continuously deforming finite elements for the solution of parabolic problems, with and without phase change. *International Journal of Numerical Methods in Engineering 17*, 81–96.

Melenk, J. M. and I. Babuška (1996). The partition of unity finite element method: Basic theory and applications. *Computer Methods in Applied Mechanics and Engineering 39*, 289–314.

Moës, N., J. Dolbow, and T. Belytschko (1999). A finite element method for crack growth without remeshing. *International Journal of Numerical Methods in Engineering 46*, 131–150.

Sampath, R. and N. Zabaras (1999). An object oriented implementation of a front tracking finite element method for directional solidification processes. *International Journal of Numerical Methods in Engineering 44*, 1227–1265.