



# Zentrum für Technomathematik

Fachbereich 3 – Mathematik und Informatik

## Solving the Stefan problem with prescribed interface using an XFEM toolbox for FENICS

Mischa Jahn

Andreas Luttmann

Report 16–03

Berichte aus der Technomathematik

Report 16–03

April 2016



# SOLVING THE STEFAN PROBLEM WITH PRESCRIBED INTERFACE USING AN XFEM TOOLBOX FOR FENICS

M. JAHN AND A. LUTTMANN

ABSTRACT. In this article, we consider the Stefan problem with prescribed interface as an example for a process with a time-dependent discontinuity. While there are various methods for solving these type of problems, the extended finite element method has grown popularity due to its flexibility. Partly based on the work of Nikbakht et al., we developed an XFEM toolbox within the FEniCS framework to solve time-dependent problems with arbitrary discontinuities. This toolbox is tested by considering different model variants of the Stefan problem. Numerical results and convergence analysis are presented which show the success of the chosen approach.

## 1. INTRODUCTION

Instationary processes with discontinuities are very common in materials science and applied physics. From a mathematical point of view, the modeling and simulation of these type of problems is very interesting and challenging. A well known example for a process with a time-dependent discontinuity is the Stefan problem describing phase transitions.

Common approaches for solving the Stefan problem numerically are moving mesh methods, based on an explicit defined sharp interface, and enthalpy methods, introducing the interface implicitly by considering the energy balance. Unfortunately, both methods have their drawbacks, see [4, 16] and references therein. In moving mesh methods for example not only is there a need for a remeshing technique but performing numerous remeshing steps during the simulation is numerically expensive, too, especially in 3D situations. Moreover, general situations including topology changes or more complex interfaces and geometries can not be considered at all. On the other hand, the enthalpy method lacks accuracy and numerical issues may arise, if problems with fluid flow and a capillary surface are considered. While there are some tricks to consider some complex situations by combining both approaches [14, 15], a method utilizing the advantages of both methods is desirable.

A method which has proven to be very suitable for all kind of problems with arbitrary discontinuities is the extended finite element method (XFEM), see e.g. [9] for an overview. XFEM is a very flexible approach which combines most of the advantages of moving mesh and enthalpy methods. The basic idea is to enrich the discrete function space(s) by additional basis functions at the interface location to consider strong discontinuities, i.e. jumps in a function, and weak discontinuities, i.e. jumps in the derivative of a function. XFEM has already been applied successfully to the Stefan problem in various publications, e.g. [6, 20].

Irrespective of the chosen method, there is a lot of implementation work to do in order to solve the modeled problem numerically. That is why automated code generation is growing more and more popularity. A very well-established framework for standard finite element approaches is the FEniCS project [18]. By using FEniCS a user can specify a problem in a specific language close to the mathematical formulation and let the software generate most of the corresponding code, e.g. quadrature methods, automatically. In this paper, we use the extended finite element method and present a toolbox partly based on [21] which enhances the FEniCS framework so that problems with arbitrary time-dependent discontinuities can be handled in the same way.

This paper is organized as the following: Starting with the governing equations for the Stefan problem, we give some brief introduction to the level set method which is a natural approach in XFEM for describing the location and movement of a discontinuity. In Section 3, we derive the time discrete formulation of the considered problem and define a weak formulation for each time step. The spatial discretization is based on Nitsche's method [23] treating the interface as internal Dirichlet boundary [7]. Some details about FEniCS and the developed toolbox used for solving the Stefan problem in XFEM formulation are presented in Section 4 and results for different model examples including a convergence study are given in Section 5. Finally, a conclusion is given.

## 2. MATHEMATICAL SETTING

**The Stefan problem.** Let  $\Omega \in \mathbb{R}^d$ , with  $\partial\Omega$  polygonally, be a fixed domain consisting for  $t \in [t_0, t_f]$  of a solid region  $\Omega_s(t)$  and a liquid region  $\Omega_l(t)$  that are separated by an interface  $\Gamma(t)$ . We assume  $\Gamma(t)$  to be sharp and sufficiently smooth for all  $t \in [t_0, t_f]$  and introduce the normal vector  $\vec{n}(t, \mathbf{x})$  to  $\Gamma(t)$  pointing from  $\Omega_l$  into  $\Omega_s$ .

The temperature field is given by  $u: \Omega \times [t_0, t_f] \rightarrow \mathbb{R}$  with  $u|_{\Omega_i} = u_i$ ,  $i \in \{s, l\}$ . Its evolution in the subdomains  $\Omega_{\{s, l\}}$  is described by

$$(2.1) \quad \rho_{\{s, l\}} c_{\{s, l\}} \frac{\partial u}{\partial t} - \nabla \cdot (\kappa_{\{s, l\}} \nabla u) = f, \quad \text{in } \Omega_s(t) \cup \Omega_l(t), \quad t \in (t_0, t_f),$$

in which we assume that  $\rho_s = \rho_l = 1$  as well as  $c_s = c_l = 1$  for simplicity and  $\kappa_{\{s, l\}}$  to be constant in each subdomain. Moreover, we expect the so-called isothermal interface condition to hold, i.e. we have  $u(\cdot, t) < u_\Gamma(\cdot, t)$  in  $\Omega_s(t)$  and  $u(\cdot, t) > u_\Gamma(\cdot, t)$  in  $\Omega_l(t)$ , and  $u(\cdot, t) = u_\Gamma$  on  $\Gamma(t)$ . For the boundary  $\partial\Omega = \Gamma_D \cup \Gamma_N$  with  $\Gamma_D \cap \Gamma_N = \emptyset$ , the following conditions are given

$$(2.2) \quad u = u_D, \quad \text{on } \Gamma_D \times (t_0, t_f),$$

$$(2.3) \quad -\kappa \frac{\partial u}{\partial \vec{n}} = g_N, \quad \text{on } \Gamma_N \times (t_0, t_f),$$

where  $\vec{n}$  denotes the outer normal to  $\partial\Omega$ . Initially, the temperature distribution on  $\Omega_s(t_0) \cup \Omega_l(t_0)$  is given by

$$(2.4) \quad u(\cdot, t_0) = u_0$$

and it is

$$(2.5) \quad \Gamma(t_0) = \{\mathbf{x} \in \Omega(t_0) \mid u(\mathbf{x}, t_0) = u_\Gamma\}$$

The last condition, which has to be considered, relates to the movement of the interface  $\Gamma$ , respectively its velocity to be more precisely, and is given by

$$(2.6) \quad [[\kappa \nabla u \cdot \vec{n}]] = LV_\Gamma \cdot \vec{n}, \quad \text{on } \Gamma$$

with  $[[\cdot]]$  denoting the jump that is defined for a function  $\phi$  by  $[[\phi]] = \phi|_{\Omega_l} - \phi|_{\Omega_s}$ . Roughly spoken, this so-called Stefan condition states that the normal velocity of  $\Gamma$  is proportional to the jump of the temperature's gradient at the interface, with  $L$  denoting the latent heat.

In general, the location of the sharp interface  $\Gamma$  is a-priori unknown and part of the solution. Hence, a representation of  $\Gamma$  is needed and there are various techniques to represent it either in an implicitly or explicitly. A very common approach in the XFEM context is to use the level set method [25, 26] for this purpose.

**The level set method.** Within the level set method [25], the location of the interface  $\Gamma$  is given by the zero level set of a continuous function  $\varphi: \Omega \times [t_0, t_f] \rightarrow \mathbb{R}$ , i.e.

$$\Gamma(t) = \{\mathbf{x} \in \Omega \mid \varphi(\mathbf{x}, t) = 0\}, \quad t \in [t_0, t_f].$$

The subdomains  $\Omega_s$  and  $\Omega_l$  can be defined by  $\mathbf{x} \in \Omega_s(t) \Leftrightarrow \varphi(\mathbf{x}, t) < 0$  and  $\mathbf{x} \in \Omega_l(t) \Leftrightarrow \varphi(\mathbf{x}, t) > 0$ . Due to this, we use the more descriptive notation  $\Omega^+ = \Omega_l$  and  $\Omega^- = \Omega_s$  in the remainder of this paper. An exemplary sketch of a 2D situation where a hold-all domain  $\Omega$  is divided by the sign of the function  $\varphi$  into subdomains  $\Omega^-(t)$  resp.  $\Omega^+(t)$  is given in Fig. 2.1a and some level sets of  $\varphi$  are indicated in Fig. 2.1b.

The level set method comes with some useful properties, e.g. it allows for an easy computation of the normal  $\vec{n}$  to  $\Gamma$ <sup>1</sup>

$$\vec{n} = -\frac{\nabla \varphi}{\|\nabla \varphi\|},$$

and the curvature  $K$  of  $\Gamma$  reads as

$$K = -\operatorname{div} \vec{n} = \operatorname{div} \frac{\nabla \varphi}{\|\nabla \varphi\|}.$$

There are various functions  $\varphi$  which can be defined and used within the level set method, however, from a numerical point of view it is important, e.g. for a stable computation of  $\vec{n}$  and  $K$ , that the gradient

<sup>1</sup>As mentioned before, the normal  $\vec{n}$  is chosen to point from  $\Omega^+$  to  $\Omega^-$ , hence we need to add the negative sign.

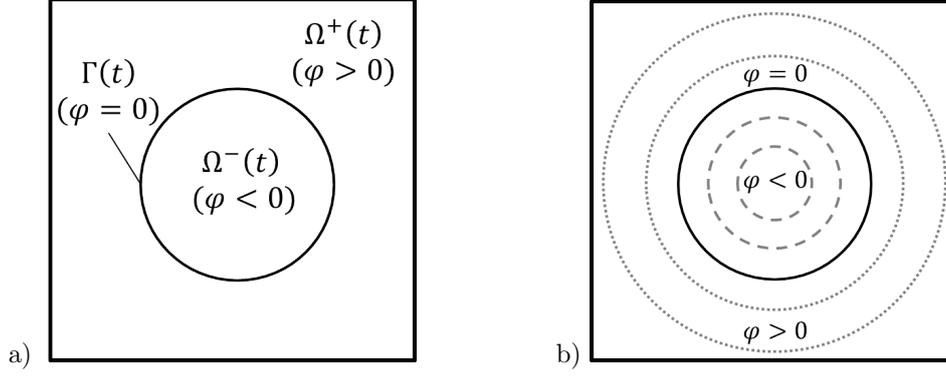


FIGURE 2.1. Visualization of the idea of the level sets method using a scalar function  $\varphi$ : a) Domains  $\Omega^-(t)$  and  $\Omega^+(t)$  are separated by the zero level set  $\Gamma(t)$  of  $\varphi$ . b) Visualization of some level sets of  $\varphi$ .

$\|\nabla\varphi\|$  does neither vanish nor become too big. Due to this, literature suggest to use a so called *signed distance function*, i.e.

$$\varphi(\mathbf{x}, t) = \begin{cases} -\min_{\tilde{\mathbf{x}} \in \Gamma(t)} \|\mathbf{x} - \tilde{\mathbf{x}}\|_2, & \text{if } \mathbf{x} \in \Omega^-(t) \\ \min_{\tilde{\mathbf{x}} \in \Gamma(t)} \|\mathbf{x} - \tilde{\mathbf{x}}\|_2, & \text{if } \mathbf{x} \in \Omega^+(t) \end{cases},$$

which satisfies  $\|\nabla\varphi\| = 1$ .

Given the initial value  $\varphi_0(\cdot) = \varphi(\cdot, t_0)$  with zero level set  $\Gamma_0 = \Gamma(t_0)$ , the evolution of the level set function  $\varphi$  and consequently of the interface  $\Gamma$  in time can be described by the transport equation

$$\varphi_t + \vec{V} \cdot \nabla\varphi = 0,$$

where  $\vec{V} = \vec{V}(\mathbf{x}, t)$  has to be a sufficiently smooth velocity field. As for the Stefan problem, this velocity field  $\vec{V}$  depends on (2.6) but is not known completely since the equation only provides us with the normal component on  $\Gamma(t)$ . Consequently more effort is needed to obtain a full velocity field, see e.g. [6].

*Remark:* In contrast to this motivation and problem formulation, we will consider the Stefan problem without solving the level set problem in this paper. The reason for this approach is to focus on the extended finite element method and neglect all technical issues arising when considering the level set method, e.g. reinitialization and mass correction techniques [13], as well as developing a strategy to tackle the full-coupled problem. Thus, we a-priori define the position and the movement of the interface and only consider the temperature field as unknown. The full coupled problem will be modeled and analyzed in a forthcoming publication.

As a result of this simplification, the full problem considered in this paper is given by: For given  $\Gamma(t)$ ,  $t \in [t_0, t_f]$ , find  $u$  sufficiently smooth, i.e.  $u(\cdot, t) \in C^2(\Omega^+(t) \cup \Omega^-(t))$ ,  $\partial_t u(\cdot, t) \in C^0(\Omega^+(t) \cup \Omega^-(t))$  and  $u \in C^0(\Omega \times [t_0, t_f])$ , such that

$$(2.7) \quad \begin{aligned} \frac{\partial u}{\partial t} - \nabla \cdot (\kappa \nabla u) &= f, & \text{in } \Omega^+(t) \cup \Omega^-(t), \quad t \in (t_0, t_f), \\ u &= u_D, & \text{on } \Gamma_D \times (t_0, t_f), \\ -\kappa \frac{\partial u}{\partial \vec{n}} &= g_N, & \text{on } \Gamma_N \times (t_0, t_f), \\ u(\cdot, t_0) &= u_0, & \text{in } \Omega^+(t_0) \cup \Omega^-(t_0), \\ u(\cdot, t) &= u_\Gamma, & \text{on } \Gamma(t), \\ \llbracket \kappa \nabla u \cdot \vec{n} \rrbracket &= LV_\Gamma \cdot \vec{n} & \text{on } \Gamma(t), \end{aligned}$$

for given data  $u_D$ ,  $g_N$ ,  $u_0$  and  $u_\Gamma$  that are assumed to be sufficiently smooth.

## 3. XFEM-DISCRETIZATION

As mentioned in [10], using space-time elements for deriving a suitable weak formulation of problem (2.7) is the natural approach. However, a weak formulation can also be introduced formally using *Rothe's method*, cf. [10, 11]. We will follow the latter in this section.

**Discretization in time.** We discretize the time interval  $[t_0, t_f]$  by  $N_t + 1$  time steps into  $t_n = t_0 + n\Delta t$ ,  $n = 0, \dots, N_t$ , with  $\Delta t$  denoting the time step size and apply the implicit Euler time discretization<sup>2</sup> to problem (2.7) which then reads: For  $n = 0, \dots, N_t$ , find  $u^{n+1} = u(\mathbf{x}, t_{n+1})$ ,  $\mathbf{x} \in \Omega^+(t_{n+1}) \cup \Omega^-(t_{n+1})$  such that

$$(3.1) \quad \begin{aligned} \frac{u^{n+1}}{\Delta t} - \nabla \cdot (\kappa \nabla u^{n+1}) &= f^{n+1} + \frac{u^n}{\Delta t}, & \text{in } \Omega^+(t_{n+1}) \cup \Omega^-(t_{n+1}), \\ u^{n+1} &= u_D^{n+1}, & \text{on } \Gamma_D(t_{n+1}), \\ -\kappa \frac{\partial u^{n+1}}{\partial \bar{n}} &= g_N^{n+1}, & \text{on } \Gamma_N(t_{n+1}), \\ u^{n+1} &= u_\Gamma, & \text{on } \Gamma(t_{n+1}), \\ [\kappa \nabla u^{n+1} \cdot \bar{n}] &= LV_\Gamma^{n+1} \cdot \bar{n} & \text{on } \Gamma(t_{n+1}). \end{aligned}$$

For a fixed  $n \in \{1, \dots, N_t\}$ , we use the notation  $\xi = \frac{1}{\Delta t}$ ,  $\Omega^+ := \Omega^+(t_{n+1})$  resp.  $\Omega^- := \Omega^-(t_{n+1})$ ,  $u := u^{n+1}$ , etc, and summarize the right-hand-side by  $\tilde{f}$ , so that, eventually, we end up with the stationary problem

$$(3.2) \quad \begin{aligned} \xi u - \nabla \cdot (\kappa \nabla u) &= \tilde{f}, & \text{in } \Omega^+ \cup \Omega^-, \\ u &= u_D, & \text{on } \Gamma_D, \\ -\kappa \frac{\partial u}{\partial \bar{n}} &= g_N, & \text{on } \Gamma_N, \\ u &= u_\Gamma, & \text{on } \Gamma, \\ [\kappa \nabla u \cdot \bar{n}] &= LV_\Gamma \cdot \bar{n} & \text{on } \Gamma, \end{aligned}$$

for each time step. Please note that the term  $\tilde{f}$  contains the expression  $\frac{u^n}{\Delta t}$  which, therefore, has to be interpolated or approximated onto the current subdomains. We will discuss this in more detail in Section 4 within the paragraph *evaluation of functions on enriched elements*.

**Weak formulation.** Since we want to solve the problem using the extended finite element method, a slightly different weak formulation of problem (3.2) is derived. Therefore, we introduce the affine space<sup>3</sup>

$$H_{u_D}^1(\Omega^+ \cup \Omega^-) := \left\{ v \in L_2(\Omega) \mid v|_{\Omega^{\{+, -\}}} \in H^1(\Omega^{\{+, -\}}), v|_{\Gamma_D} = u_D \right\},$$

where each element  $v \in H_{u_D}^1(\Omega^+ \cup \Omega^-)$  can be restricted onto a subdomain by  $v^{\{+, -\}} := v|_{\Omega^{\{+, -\}}}$ . For functions  $u, v \in H_{u_D}^1(\Omega^{\{+, -\}})$  we define

$$\begin{aligned} (u, v)_{H^1(\Omega^+ \cup \Omega^-)} &:= (u, v)_{H^1(\Omega^+)} + (u, v)_{H^1(\Omega^-)} \\ &:= \int_{\Omega^+} \nabla u^+ \cdot \nabla v^+ d\mathbf{x} + \int_{\Omega^-} \nabla u^- \cdot \nabla v^- d\mathbf{x}, \quad u, v \in H^1(\Omega^+ \cup \Omega^-). \end{aligned}$$

Using this and the  $L^2$ -norm gives us the norm

$$\left( \|\cdot\|_{L^2(\Omega)}^2 + |\cdot|_{H^1(\Omega^+ \cup \Omega^-)}^2 \right)^{1/2} =: \|\cdot\|_{H^1(\Omega^+ \cup \Omega^-)}.$$

Based on these definitions, we also define the Hilbert space

$$(3.3) \quad V_0 := \left\{ v \in H_0^1(\Omega^+ \cup \Omega^-) \mid v|_\Gamma = 0 \right\}$$

<sup>2</sup>In general, the so-called  $\theta$ -scheme is often used as time discretization technique since it allows arbitrary weighting of old and new data. However, when using  $\theta \in (0, 1)$  in the XFEM context, the formulation ends up with terms that are not well-defined and cannot be interpreted meaningfully, cf. [11]. Therefore, we use the implicit Euler scheme in this paper.

<sup>3</sup>Here we chose the general setting with  $u_D$  as Dirichlet boundary condition (in trace sense). Using  $u_D = 0$  leads to the Hilbert space  $H_0^1(\Omega^+ \cup \Omega^-)$  which will be used in 3.3.

and the affine space <sup>4</sup>

$$(3.4) \quad V_\Gamma := \{v \in H_{u_D}^1(\Omega^+ \cup \Omega^-) \mid v|_\Gamma = u_\Gamma\}.$$

A weak formulation of the problem (3.2) is then given by: For  $\xi, \kappa \in L_\infty(\Omega)$ ,  $\tilde{f} \in L^2(\Omega)$  and  $g \in L^2(\Gamma_N)$  find  $u \in V_\Gamma$  s.t.

$$(3.5) \quad (\xi u, v)_{L_2(\Omega^+ \cup \Omega^-)} + (\kappa u, v)_{H^1(\Omega^+ \cup \Omega^-)} = (\tilde{f}, v)_{L^2(\Omega)} + (g, v)_{L^2(\Gamma_N)}$$

for all  $v \in V_0$ . Using the theorem of Lax-Milgram, one can show that there exists an unique solution to (3.5).

*Remark.* Since the the Stefan problem only includes a weak discontinuity, i.e. the temperature  $u$  is continuous but there is a jump in the temperature gradients at  $\Gamma$  one could also introduce the function spaces

$$(3.6) \quad \tilde{V}_0 := \{v \in H_0^1(\Omega^+ \cup \Omega^-) \mid \llbracket v \rrbracket_\Gamma = 0\},$$

and

$$(3.7) \quad \tilde{V}_\Gamma := \{v \in H_{u_D}^1(\Omega^+ \cup \Omega^-) \mid \llbracket \kappa \nabla v \vec{n} \rrbracket_\Gamma = L V_\Gamma \vec{n}\}.$$

which include the Stefan condition in their definition and end up with a different weak formulation. For the spatial discretization, this weak discontinuity could be considered for example by a *modified-abs* enrichment [9]. However, there is still need to consider the interface condition  $u = u_\Gamma$  due to numerical inaccuracies. Anyway, since our approach bases on strong enriched function spaces and the Nitsche approach [23], see below and Section 4, we use the definitions given above.

### Discretization in space based on Nitsche's method.

*Preliminaries.* We introduce some notation and assumptions following the work of [12]: Let  $\{\mathcal{S}_h\}_{h>0}$  be a family of shape regular triangulations consisting of  $d$ -simplices and  $h$  is the maximum diameter  $h = \max_{S \in \mathcal{S}_h} \text{diam}(S)$ . Furthermore, let  $\Gamma_h$  be a discrete approximation of  $\Gamma$  with  $\Omega_h^{\{+,-\}}$  as discrete counterparts of  $\Omega^{\{+,-\}}$ <sup>5</sup>. For any element  $S \in \mathcal{S}_h$  let  $S^{\{+,-\}} := S \cap \Omega_h^{\{+,-\}}$  be the part of  $S$  in  $\Omega_h^{\{+,-\}}$ . The set of elements being intersected by the interface  $\Gamma_h$  is given by  $\mathcal{S}_h^{\Gamma_h} := \{S \in \mathcal{S}_h \mid S \cap \Gamma_h \neq \emptyset\}$  and the intersecting part is denoted by  $\Gamma_S := S \cap \Gamma_h$ .

Since we do not want to consider the interface  $\Gamma_h$  explicitly as edges within the triangulation, we can not include the condition  $v|_{\Gamma_h} = u_\Gamma$  into the discrete function space as we do for the outer Dirichlet condition on  $\Gamma_D$ . Thus, we introduce the function space<sup>6</sup>

$$(3.8) \quad V_{h,u_D}^k := \left\{ v \in H_{u_D}^1(\Omega_h^+ \cup \Omega_h^-) \mid v^{\{+,-\}} \in C^0(\Omega^{\{+,-\}}), v|_{S^{\{+,-\}}} \in \mathbb{P}_k, S \in \mathcal{S}_h \right\}$$

for  $k \in \mathbb{N}$ . To consider the internal Dirichlet condition  $u_h = u_\Gamma$  on  $\Gamma_h$ , Nitsche's method [23] is used to include this condition weakly into the discrete problem formulation. Hence, we end up treating (3.5) as two "independent" problems.

*Spatial discretization.* Following [7], the discrete formulation of (3.2) is given by: Find  $u_h \in V_{h,u_D}^k$  s.t.

$$(3.9) \quad a(u_h, v_h) + a^+(u_h, v_h) + a^-(u_h, v_h) = L(v_h) + L^+(v_h) + L^-(v_h)$$

for all  $v_h \in V_{h,0}^k$ . The bilinear forms and linear forms are defined as

$$(3.10) \quad a(u_h, v_h) = \int_{\Omega_h^+ \cup \Omega_h^-} \xi u_h v_h d\mathbf{x} + \int_{\Omega_h^+ \cup \Omega_h^-} \kappa \nabla u_h \nabla v_h d\mathbf{x}$$

$$(3.11) \quad a^+(u_h, v_h) = - \int_{\Gamma_h} \kappa^+ \nabla u_h^+ \cdot \vec{n}_h v_h^+ d\mathbf{c} - \int_{\Gamma_h} \kappa^+ \nabla v_h^+ \cdot \vec{n}_h u_h^+ d\mathbf{c} + \int_{\Gamma_h} \lambda u_h^+ v_h^+ d\mathbf{c}$$

<sup>4</sup>In this particular situation, one could also use  $H_0^1(\Omega)$  resp.  $H_{u_D}^1(\Omega)$  in the definitions of  $V_0$  resp.  $V_\Gamma$  since no jumps are allowed across  $\Gamma$  and only a weak discontinuity, i.e. a jump in the gradients, is present. However, the present approach can easily be extended to allow for strong discontinuities as they may occur in more general problems.

<sup>5</sup>We want to stress that due to our  $\Omega$  with  $\partial\Omega$  polygonal, we have  $\Omega_h = \Omega$

<sup>6</sup>Evidently, in contrast to the standard Lagrangian function space we now request that for a function  $v$  only the restriction  $v^{\{+,-\}}$  has to be continuous on  $\Omega^{\{+,-\}}$ .

$$(3.12) \quad a^-(u_h, v_h) = + \int_{\Gamma_h} \kappa^- \nabla u_h^- \cdot \vec{n}_h v_h^- d\mathbf{c} + \int_{\Gamma_h} \kappa^- \nabla v_h^- \cdot \vec{n}_h u_h^- d\mathbf{c} + \int_{\Gamma_h} \lambda u_h^- v_h^- d\mathbf{c}$$

$$(3.13) \quad L(v_h) = \int_{\Omega^+ \cup \Omega^-} \tilde{f} v_h d\mathbf{x} + \int_{\Gamma_N} g v_h d\mathbf{s}$$

$$(3.14) \quad L^+(v_h) = - \int_{\Gamma_h} \kappa^+ \nabla v_h^+ \cdot \vec{n}_h u_h d\mathbf{c} + \int_{\Gamma_h} \lambda u_h v_h^+ d\mathbf{c}$$

$$(3.15) \quad L^-(v_h) = \int_{\Gamma_h} \kappa^- \nabla v_h^- \cdot \vec{n}_h u_h d\mathbf{c} + \int_{\Gamma_h} \lambda u_h v_h^- d\mathbf{c},$$

cf. Appendix for more details. Therein,  $0 < \lambda \in \mathbb{R}$  is a stability parameter which has to be chosen big enough and can be derived analytically for some situations [7]. Please note that the sign of the terms in  $a^{\{+,-\}}$  and  $L^{\{+,-\}}$  result from the direction of the normal vector  $\vec{n}_h$  pointing from  $\Omega_h^+$  to  $\Omega_h^-$ . In contrast to the continuous situation, it is not trivial to show that the sum of all bilinear forms is coercive. In fact, this property depends heavily on the choice of  $\lambda$ . In this paper, we do not comment and investigate this issue further but assume that there is a unique solution of problem (3.9).

#### 4. IMPLEMENTATION ASPECTS

The problem (3.9) is solved by using an XFEM toolbox for **FEniCS** which has been developed within our work group. It uses the **FEniCS** framework and is partly based on the **PUM** toolbox [21, 22]. **FEniCS** is a collaborative project of researchers who develop tools for automated scientific computing, especially in the field of finite element methods for the solution of partial differential equations [18]. It consists of a collection of components, described in various articles, e.g. [1, 2, 17–19, 24], which make the numerical computation of solutions for PDE-based problems very easy. Furthermore, the **FEniCS** framework can be extended by new modules such as our XFEM toolbox, which, as its precursor, bases on added features for the domain specific language **UFL**, an extended **FEniCS** Form Compiler (**FFC**), and an extension of the **DOLFIN** library.

**4.1. XFEM toolbox.** In contrast to its precursor **PUM**, our XFEM toolbox can handle time-dependent problems with multiple, arbitrary discontinuities that may evolve and also intersect each other. A detailed description of its features will be given in an upcoming publication. Here, we just want to point out the most relevant aspects regarding the solution of the Stefan problem in level set formulation.

**Enrichment scheme.** A principal characteristic of the extended finite element method is that the effect of the enrichment is locally restricted. Thus, we can consider most of the elements and degrees of freedom just as in the standard finite element context while only a minor subset needs special attention. Hence, the assembled matrices and vectors are still sparse. There are various approaches to enrich a function space which, for a start, differ whether the enrichment is made to handle strong discontinuities, i.e. jumps in the function, or weak discontinuities, that is to say the function itself is continuous but the gradients differ on each side of the interface  $\Gamma_h$ .

In our toolbox, we follow [21, 22] and use a strong enrichment so that an enriched function  $u_h^{\text{XFEM}} \in V_h^{\text{XFEM}}$ , e.g.  $V_h^{\text{XFEM}} = V_h^k$ , is given by

$$(4.1) \quad u_h^{\text{XFEM}} = \sum_{i \in \mathcal{N}} u_i v_i + \sum_{j \in \tilde{\mathcal{N}}} \tilde{u}_j H v_j$$

with basis functions  $v_i$ ,  $i \in \mathcal{N}$ , of the standard Lagrangian function space  $V_h^{\text{FEM}}$  and corresponding coefficients  $u_i$ . The index set  $\tilde{\mathcal{N}}$  is defined by

$$(4.2) \quad \tilde{\mathcal{N}} := \{i \in \mathcal{N} \mid \text{meas}_{d-1}(\Gamma_h \cap \text{supp}(v_i)) > 0, v_i \in V_h^{\text{FEM}}\}$$

and

$$(4.3) \quad H(\mathbf{x}) = \begin{cases} 1, & \text{for } \mathbf{x} \in \Omega^+ \\ 0, & \text{else} \end{cases}$$

is the Heaviside function.

In combination with the Heaviside function, definition (4.2) takes care to avoid linear dependency since the enriched degrees of freedom only exists in a small region around the interface  $\Gamma_h$  and vanish elsewhere. The advantage of using a strong enrichment is its flexibility as it can be used for the modeling

and simulation of different processes with both, strong and weak discontinuities since for the latter, continuity of a function can be enforced by adding corresponding conditions using the Nitsche technique as presented in Section 3 or by including jump conditions [12].

*Remark:* As shown in [3], this enrichment is equivalent to the method proposed by [12] which is called *cut cell method* and based on duplicating nodes of intersected elements.

**Interpolation.** A fundamental challenge in XFEM is the interpolation of an arbitrary function onto the discrete function space. It is well known that standard Lagrange finite element basis functions fulfill the partition of unity property in each degree of freedom, i.e. for a function space  $V_h^{\text{FEM}}$  with basis functions  $v_i, i \in \mathcal{N}$ , we have

$$\sum_{i \in \mathcal{N}} v_i(\mathbf{x}) = 1, \quad \forall \mathbf{x} \in \Omega,$$

and, therefore, the nodal interpolation property

$$(4.4) \quad u_h^{\text{FEM}}(\mathbf{x}_k) = \sum_{i \in \mathcal{N}} u_i v_i(\mathbf{x}_k) = u_k$$

holds for all nodes  $\mathbf{x}_k \in \Omega$  with coefficients  $u_i, i \in \mathcal{N}$ .

Unfortunately, this is not true for enriched degrees of freedom in the XFEM case: Due to (4.1), we have for  $u_h^{\text{XFEM}} \in V_h^{\text{XFEM}}$

$$(4.5) \quad u_h^{\text{XFEM}}(\mathbf{x}_k) = \sum_{i \in \mathcal{N}} u_i v_i(\mathbf{x}_k) + \sum_{j \in \tilde{\mathcal{N}}} \tilde{u}_j H(\mathbf{x}_k) v_j(\mathbf{x}_k) = u_k + H(\mathbf{x}_k) \tilde{u}_k$$

and, hence,  $u_h^{\text{XFEM}}(\mathbf{x}_k) \neq u_k$  for all  $k \in \tilde{\mathcal{N}}$  with  $x_k \in \Omega_h^+$ , cf. definitions (4.2) and (4.3). Consequently for interpolating an arbitrary function onto an enriched function space, a more sophisticated interpolation scheme has to be implemented.

Alternatively, the  $L^2$ -projection can be used for this purpose, i.e. solve

$$(4.6) \quad \int_{\Omega} (P_h^{\text{XFEM}} f) v dx = \int_{\Omega} f v dx, \quad \forall v \in V_h^{\text{XFEM}},$$

to project an arbitrary  $f$  onto  $V_h^{\text{XFEM}}$  with  $(P_h^{\text{XFEM}} f)$  denoting the corresponding XFEM function. This variant is very convenient since the FEniCS framework offers the possibility to just add this equation to our problem formulation within the UFL file and generate the respective code automatically using the extended FEniCS Form Compiler. Moreover, if we have to interpolate multiple functions onto the same function space, we only need to assemble the left-hand-side in (4.6) once and can re-use it.

*Remark:* Obviously, the  $L^2$ -projection approximates a given function as good as possible only in  $L^2$  sense. Consequently, the nodal values may differ from the values which would be computed using a nodal interpolation scheme. While in general this is of no importance for a function, there may arise numerical issues if considering the gradient of a function if the support of a degree of freedom is barely intersected. This is due to the fact that the domain on which the quadrature is performed can be tiny, making the value at a corresponding degree of freedom almost arbitrary. This may result in deviations and spikes in convergence plots when analyzing the convergence behavior for the gradient. The described issue can be prevented by deactivating the respective degrees of freedom, however, if for error computation the numerical and analytical solution are interpolated onto a function space with higher polynomial degree, the deactivation of DOFs may cause additional noise. Therefore, one may add a penalty term to punish small errors at the interface and, hence, reduce the error at this domain. We will address this issue further in Section 5.

**Interface representation and local mesh.** Within our implementation, a given level set function  $\varphi$  is approximated piecewise quadratically by  $\varphi_h$  while its zero level set  $\Gamma_h$  is approximated polygonally using Lagrangian interpolation, i.e.

$$\Gamma_h := \{\mathbf{x} \in \Omega_h \mid I_{\text{lin}} \varphi_h(\mathbf{x}) = 0\}.$$

Hence, the intersection segments  $\Gamma_S = S \cap \Gamma_h$  are linear. In order to perform the quadrature on all elements  $S \in \mathcal{S}_h$  being intersected by  $\Gamma_h$ , we use the same idea as in [21, 22], i.e. dividing them into sub-elements by firstly divide them into two parts by using the linear segment  $\Gamma_S$  of the interface as facet and create new sub-simplices using the element's vertices, the intersection points, and the centroids of the newly created subdomains. A sketch of the idea for a 2D situation is given in Figure 4.1. The same

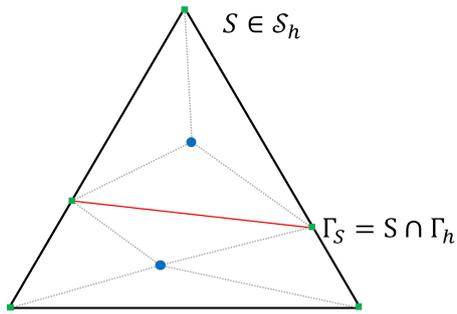


FIGURE 4.1. Creating sub-elements of a cell  $S \in \mathcal{S}_h$  using  $\Gamma_S$  and the centroids of the resulting subdomains.

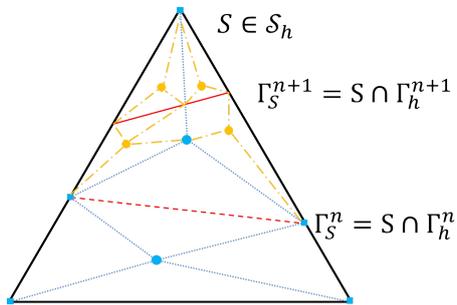


FIGURE 4.2. Creating sub-elements of a cell  $S \in \mathcal{S}_h$  for time-dependent problems considering  $\Gamma_S^n$  and  $\Gamma_S^{n+1}$ .

approach is used for 3D situations where additionally the facets are subdivided in order to maintain the connectivity and proximity relationship.

For time-dependent problems, this concept of creating a local mesh is extended, so that now both interfaces  $\Gamma_h^n = \Gamma_h(t_n)$  and  $\Gamma_h^{n+1} = \Gamma_h(t_{n+1})$  are considered for the subdivision of an element: Firstly, we divide the mesh accordingly to  $\Gamma_h^n$ , as described previously, and then refine those newly created sub-elements of the adapted mesh who are intersected by  $\Gamma_h^{n+1}$ , see Figure 4.2 for a 2D visualization. The resulting sub-mesh of each element is stored for both situations in a so-called *local mesh* which is used for doing the quadrature, visualization of the results and so on.

*Remark:* As shown in [5], it is also possible to use curved intersection segments for the subdivision of an element into sub-elements. By using their method with an appropriate quadrature scheme, one can archive optimal convergence rates for  $k > 1$ , if using strong enrichments. Up to now, this concept is not included into our work. The consequences of this can be seen in Example 2 in Section 5.

**Quadrature and function evaluation on enriched elements.** As mentioned before, the FEniCS approach is to specify the problem's weak formulation within an UFL formulation and generate most of the related code automatically using the FFC. Among other things, the corresponding formulas for performing quadrature for all kind of finite elements are generated this way.

In general, the assembling strategy in FEniCS is to compute the integral over each element and add the result to a global system matrix resp. vector. As usual, the quadrature rule is based on evaluating the function to be integrated in its quadrature points by using the basis functions and the corresponding coefficients. However for doing quadrature in the XFEM context, we need to adapt the rules since special care is needed for enriched elements.

Using the *local mesh*, we have a partition of each intersected element into sub-elements respecting the interface. Since these sub-elements are either completely in  $\Omega^+$  or in  $\Omega^-$ , we can simply use a transformed standard quadrature rule to compute the contribution of each sub-element. For doing this accurately, we need to evaluate every function with respect to their particular enrichment. This means, we have to distinguish for each quadrature point  $\mathbf{x} \in \Omega$  whether the enriched basis function is *active*, i.e.  $\varphi_h(\mathbf{x}) > 0$  and consequently  $H(\mathbf{x}) = 1$ , or it is *inactive*, i.e. we have  $\varphi_h(\mathbf{x}) < 0$  and therefore  $H(\mathbf{x}) = 0$ .

As for the implementation and the automated code generation, this is especially complex for computing integrals of type

$$(4.7) \quad \int_{\Omega} f v_h d\mathbf{x},$$

if we have  $f \notin V_h^{\text{XFEM}}$  but  $f \in W_h^{\text{XFEM}}$  instead, i.e the function space resp. the enrichment of a function  $f$ , which has to be integrated, differs from the function space of the test function  $v_h \in V_h^{\text{XFEM}}$ . Since different function spaces lead to dissimilar enriched elements and degrees of freedom, we have to consider multiple enrichments at the same time. This is especially of concern, if more functions with different enrichments are considered. Please note that the discussed terms naturally arise due to the time discretization since for time dependent problems with  $\Gamma_h^{n+1} \neq \Gamma_h^n$  we have

$$(4.8) \quad \int_{\Omega^{n+1}} u_n^n v_h^{n+1} d\mathbf{x},$$

cf. the first equation in (3.1).

*Remark:* In regards to interface integrals, the proposed subdivision of the elements by linear interface segments is beneficial as well since we can easily compute the integrals on a  $d - 1$  simplex.

**Boundary conditions.** As mentioned in [9], applying boundary conditions in XFEM context, especially of type Dirichlet, can cause issues since not only standard degrees of freedom but also enriched ones need to be considered. For problems where this issue may arise, we use weak Dirichlet boundary conditions, i.e. we incorporate them into the weak formulation, similar to the approach shown in Section 3, and mark the respective boundary domain with marker functions provided by DOLFIN. In contrast to Dirichlet boundary conditions, applying Neumann type boundary conditions can be done without any problems.

**Error computation.** For the analysis of the convergence behavior, various errors can be considered, including the standard  $L^2$ -error

$$(4.9) \quad \|u(\cdot, t) - u_h(\cdot, t)\|_{L^2(\Omega)},$$

the  $H^1$ -error

$$(4.10) \quad \|\nabla u(\cdot, t) - \nabla u_h(\cdot, t)\|_{L^2(\Omega)},$$

and the many more. These error values are computed using additional forms corresponding to the errors into the problem's UFL file. Thereby the forms respect both, the analytical given interface and the computed interface. All results are visualized automatically using `gnuplot` [27].

## 5. RESULTS

In this section, two academical examples with known analytical solutions  $u$  are examined. In particular, a numerical convergence analysis is performed with respect to different time step sizes  $\Delta t$ , varied maximum cell diameters  $h$  and the local polynomial degrees  $k$  of the XFEM-space. The considered errors for  $u - u_h$  are the  $L^2$ -error

$$\|u - u_h\|_{L^\infty(L^2)} := \max_{t \in (t_0, t_f)} \|u(\cdot, t) - u_h(\cdot, t)\|_{L^2(\Omega)}$$

and the  $H^1$ -error

$$\|\nabla u - \nabla u_h\|_{L^2(L^2)} := \sqrt{\int_{t_0}^{t_f} \|u(\cdot, t) - u_h(\cdot, t)\|_{L^2(\Omega)}^2 dt}.$$

The order of convergence for each error is determined for  $\kappa = 1, 2$  fixed while either  $\Delta t$  or  $h$  are varied but not both. The results are compared to the convergence rates using a standard finite element method for the heat equation using the implicit Euler scheme which at best, cf. [8], are given by

$$\|u - u_h\|_{L^\infty(L^2)} = \begin{cases} \mathcal{O}(h^{k+1}) \\ \mathcal{O}(\Delta t) \end{cases}$$

and

$$\|\nabla u - \nabla u_h\|_{L^2(L^2)} = \begin{cases} \mathcal{O}(h^k) \\ \mathcal{O}(\Delta t) \end{cases}.$$

For both examples, a regular structured triangular mesh is used. The range of chosen cell diameters  $h$  and time step sizes  $\Delta t$  are specified for each example individually in the respective subsection. The local polynomial degree  $k$  is chosen as either 1 or 2, as mentioned above.

**5.1. Example 1: straight interface.** Choosing  $\Omega := (0, 1)^2$ ,  $\Gamma_D := \{(x, y) \in \partial\Omega \mid y = 0 \vee y = 1\}$ ,  $\Gamma_N := \partial\Omega \setminus \Gamma_D$  for the geometry and  $\kappa^- := 1$ ,  $\kappa^+ := 2$ ,  $L := 1$ ,  $u_\Gamma := 0$  for the material parameters, an analytical solution to problem (2.7) on the time interval  $(t_0, t_f) := (0, \frac{1}{6})$  is given by

$$u(\mathbf{x}, t) = \begin{cases} \cos\left(\frac{\pi x}{2}\right) \sin\left(\frac{\pi \varphi(\mathbf{x}, t)}{y - \varphi(\mathbf{x}, t)}\right) + \varphi(\mathbf{x}, t) & \text{on } \Omega^-(t) \\ \cos\left(\frac{\pi x}{2}\right) \sin\left(\frac{\pi \varphi(\mathbf{x}, t)}{2(y - \varphi(\mathbf{x}, t))}\right) + \frac{1}{2}\varphi(\mathbf{x}, t) + e^y + \frac{3}{3t-2} & \text{on } \Omega^+(t) \end{cases},$$

where  $y$  denotes the second component of  $\mathbf{x} \in \Omega$  and the interface  $\Gamma(t)$  is a straight horizontal line moving downwards.  $\Gamma(t)$  can be characterized by the level set function

$$\varphi(\mathbf{x}, t) := y - \ln\left(\frac{3}{2-3t}\right).$$

The source term  $f$ , the boundary functions  $u_D$  and  $g_N$  and the initial conditions have been chosen with respect to the specified analytical solution and can be easily computed. The analytical solution  $u$  is shown in Figure 5.1 at different time instants.

Since  $\varphi$  is a linear function in  $\mathbf{x}$  at each time instant  $t$ , using  $\varphi_h$  and the corresponding normal vector  $\vec{n}_h$  instead of  $\varphi$  and  $\vec{n}$  results in no additional approximation error. Results of the convergence test for  $k = 1$  resp.  $k = 2$  are shown in Figure 5.2.

In each subfigure, the interdependence of the errors with respect to space and time discretization can be seen since the ideal convergence rates regarding the space discretization are only achieved for small values of  $\Delta t$ , which is accompanied with a relatively small time discretization error. The same holds for the time discretization error and small values of  $h$ . In particular, for  $k = 1$  it can be seen from Figure 2(a) and 2(b) that the ideal convergence rate  $\|u - u_h\|_{L^\infty(L^2)} = \mathcal{O}(h^2)$  is barely achieved for the smallest value of  $\Delta t$ , whereas the ideal convergence rate  $\|\nabla u - \nabla u_h\|_{L^2(L^2)} = \mathcal{O}(h)$  is achieved for nearly all values chosen for  $\Delta t$ . In Figure 2(c) and 2(d), the ideal convergence rate  $\mathcal{O}(\Delta t)$  can only be seen for the  $L^2$ -error. For the  $H^1$ -error, the space discretization error is still dominant for the smallest value of  $h$ .

This behavior changes for  $k = 2$ . As shown in Figure 2(g) and 2(h), the ideal convergence rate  $\mathcal{O}(\Delta t)$  is achieved for both, the  $L^2$ -error as well as the  $H^1$ -error. Due to the dominance of the time error in this case, no convergence in 2(e) and 2(f) except for larger values of  $h$  and the  $H^1$ -error can be observed. Not surprisingly, when comparing the Figures for  $k = 1$  with the corresponding figures for  $k = 2$ , it can be seen that the error for  $k = 2$  is always smaller due to the smaller space discretization error. The biggest difference can be seen for Figures 2(d) and 2(h) where the space discretization error is dominant.

**5.2. Example 2: circular interface.** Choosing  $\Omega := (-1, 1)^2$ ,  $\Gamma_D := \partial\Omega$  for the geometry and  $\kappa^- := \kappa^+ := 1$ ,  $L := 1$ ,  $u_\Gamma := 0$  for the material parameters, an analytical solution to problem (2.7) on the time interval  $(t_0, t_f) := (0, \frac{1}{6})$  is given by

$$u(\mathbf{x}, t) = \begin{cases} \cos\left(\frac{\pi|\mathbf{x}|}{2R(t)}\right) - e^{|\mathbf{x}|} + \frac{3}{2-3t} & \text{on } \Omega^-(t) \\ \cos\left(\frac{\pi|\mathbf{x}|}{2R(t)}\right) & \text{on } \Omega^+(t) \end{cases}.$$

In this example, the interface  $\Gamma(t)$  corresponds to an expanding circle with radius  $R(t) := \ln\left(\frac{3}{2-3t}\right)$  which is centered at the origin and can be characterized by the level set function

$$\varphi(\mathbf{x}, t) := R^2(t) - |\mathbf{x}|^2.$$

Again, the source term  $f$ , the Dirichlet boundary function  $u_D$  and the initial conditions must be chosen with respect to the specified analytical solution and can be derived easily. The analytical solution  $u$  is shown in Figure 5.3 at different time instants. While  $\varphi$  can be exactly approximated by  $\varphi_h$  when using quadratic basis functions, the polygonal approximation  $\Gamma_h(t)$  of the circular interface  $\Gamma(t)$  introduces an additional error, which is different compared to the situation in Example 1. The discrete unit normal vector to  $\Gamma_h$  is approximated by

$$\vec{n}_h := -\frac{\nabla \varphi_h}{\|\nabla \varphi_h\|} \quad \text{on } \Gamma_h(t)$$

using  $\varphi_h$ . Results of the convergence test for  $k = 1$  resp.  $k = 2$  are shown in Figure 5.4.

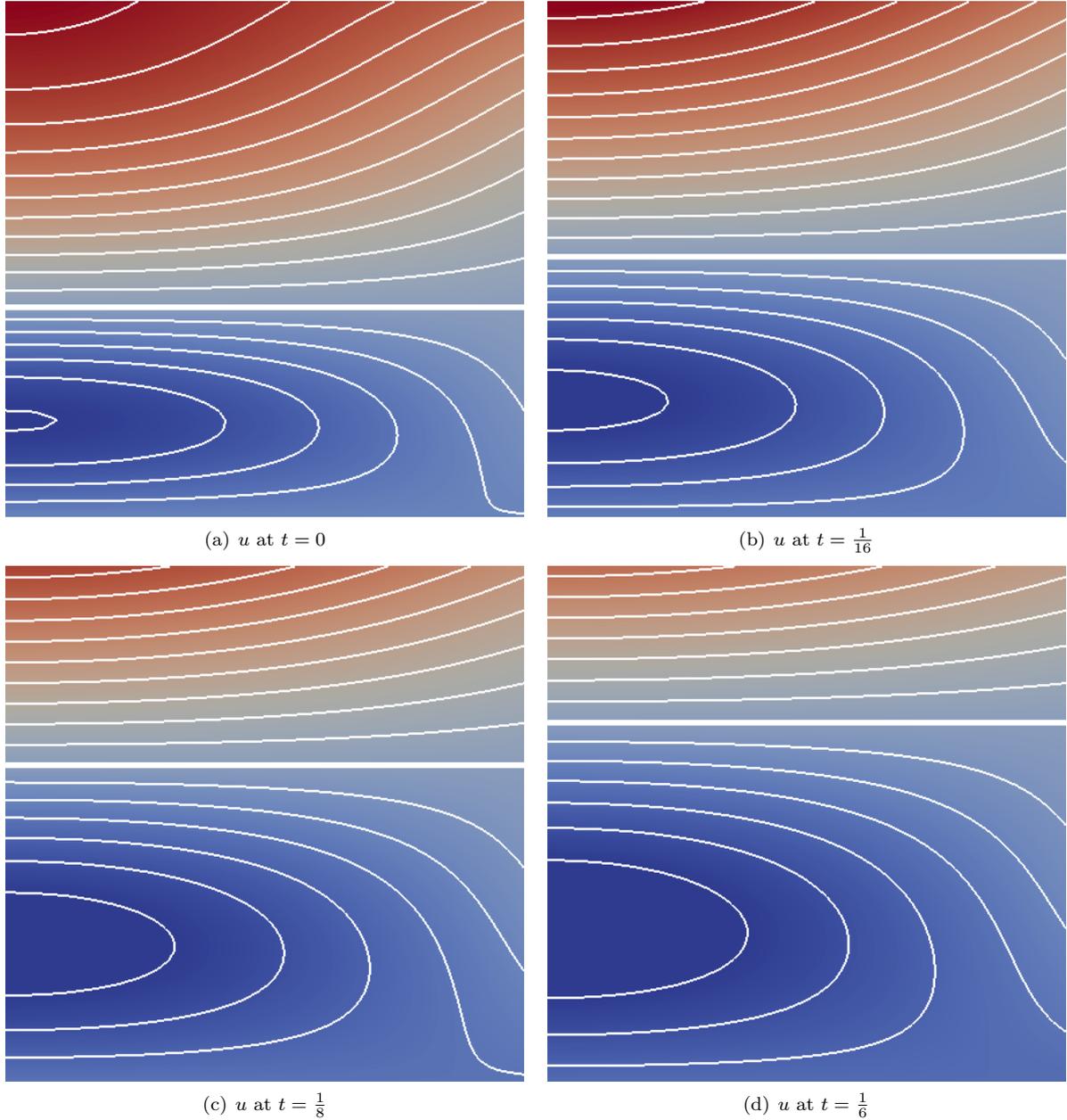


FIGURE 5.1. Example 1: analytical solution  $u$  at different time instants  $t$ .  $u$  is represented by its level sets  $\{\mathbf{x} \in \Omega \mid \exists z \in \mathbb{Z} : u(\mathbf{x}, t) = 0.1 \cdot z\}$ . The interface  $\Gamma(t)$  coinciding with the zero level set  $\{\mathbf{x} \in \Omega \mid u(\mathbf{x}, t) = 0\}$  is visualized using an increased line width.

Despite the additional approximation error introduced by  $\Gamma_h$  and  $\vec{n}_h$ , the results are qualitatively the same as for Example 1. For Figure 4(a), it should be pointed out, that the ideal convergence  $\|u - u_h\|_{L^\infty(L^2)} = \mathcal{O}(h^2)$  can only be seen by neglecting the smaller values of  $h$  for the smallest value of  $\Delta t$  which is due to the relatively large time discretization error.

## 6. CONCLUSION

In our paper, we have discussed a numerical approach to solve the Stefan problem based on XFEM by using the FEniCS framework. An XFEM toolbox partly based on [21, 22] has been developed, which is used for computing solutions of two academical examples. The numerical results show that optimal convergence rates can be archived for piecewise linear elements. The named toolbox can be combined

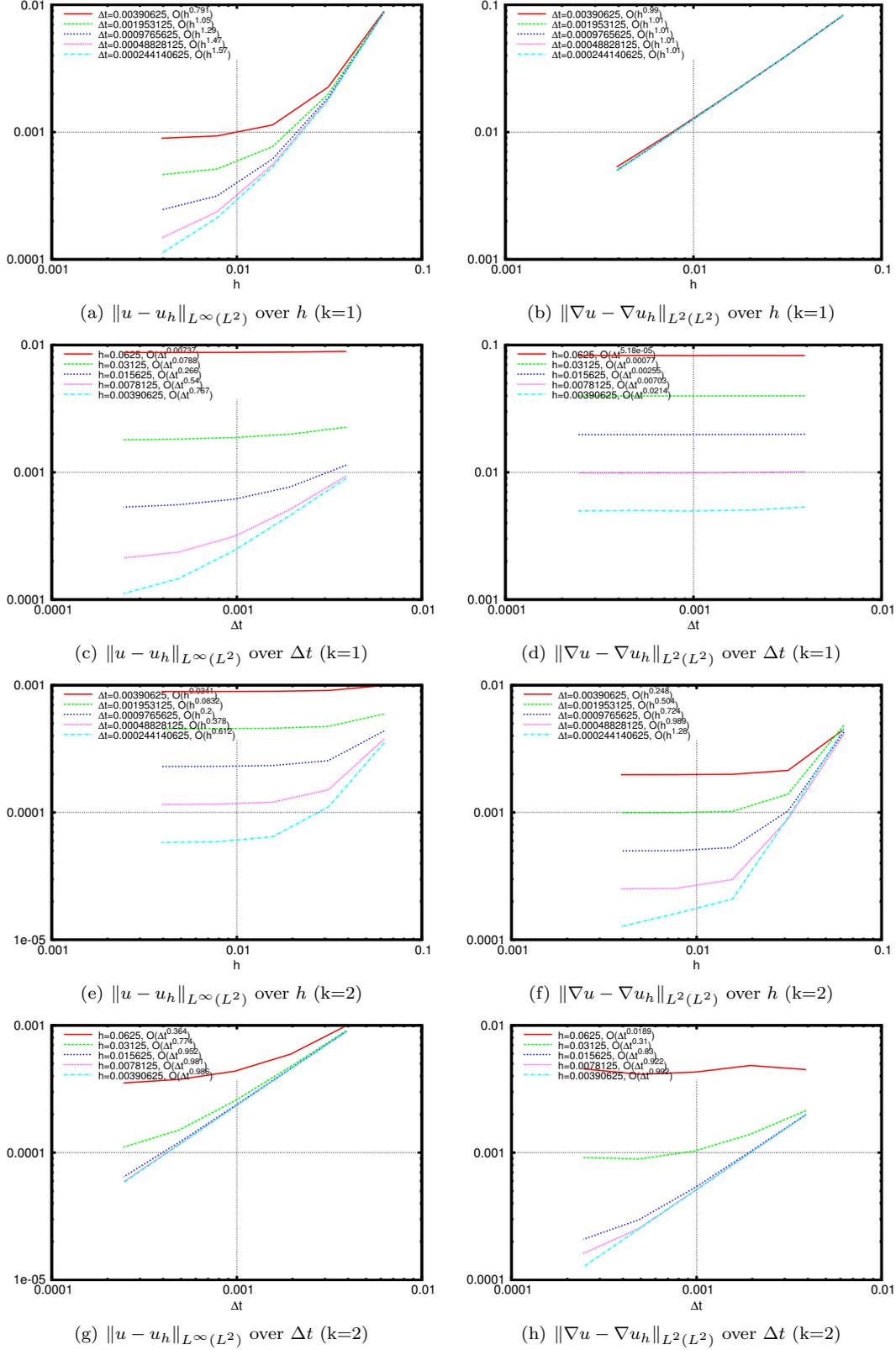


FIGURE 5.2. Example 1 convergence tests:  $\|u - u_h\|_{L^\infty(L^2)}$  and  $\|\nabla u - \nabla u_h\|_{L^2(L^2)}$  for  $\Delta t \in \{2^{-8}, 2^{-9}, 2^{-10}, 2^{-11}, 2^{-12}\}$ ,  $n_x \in \{16, 32, 64, 128, 256\}$  and  $k = 1$ (a-d) resp.  $k = 2$  (e-h). An approximated order of convergence is shown in the legend for each plot.

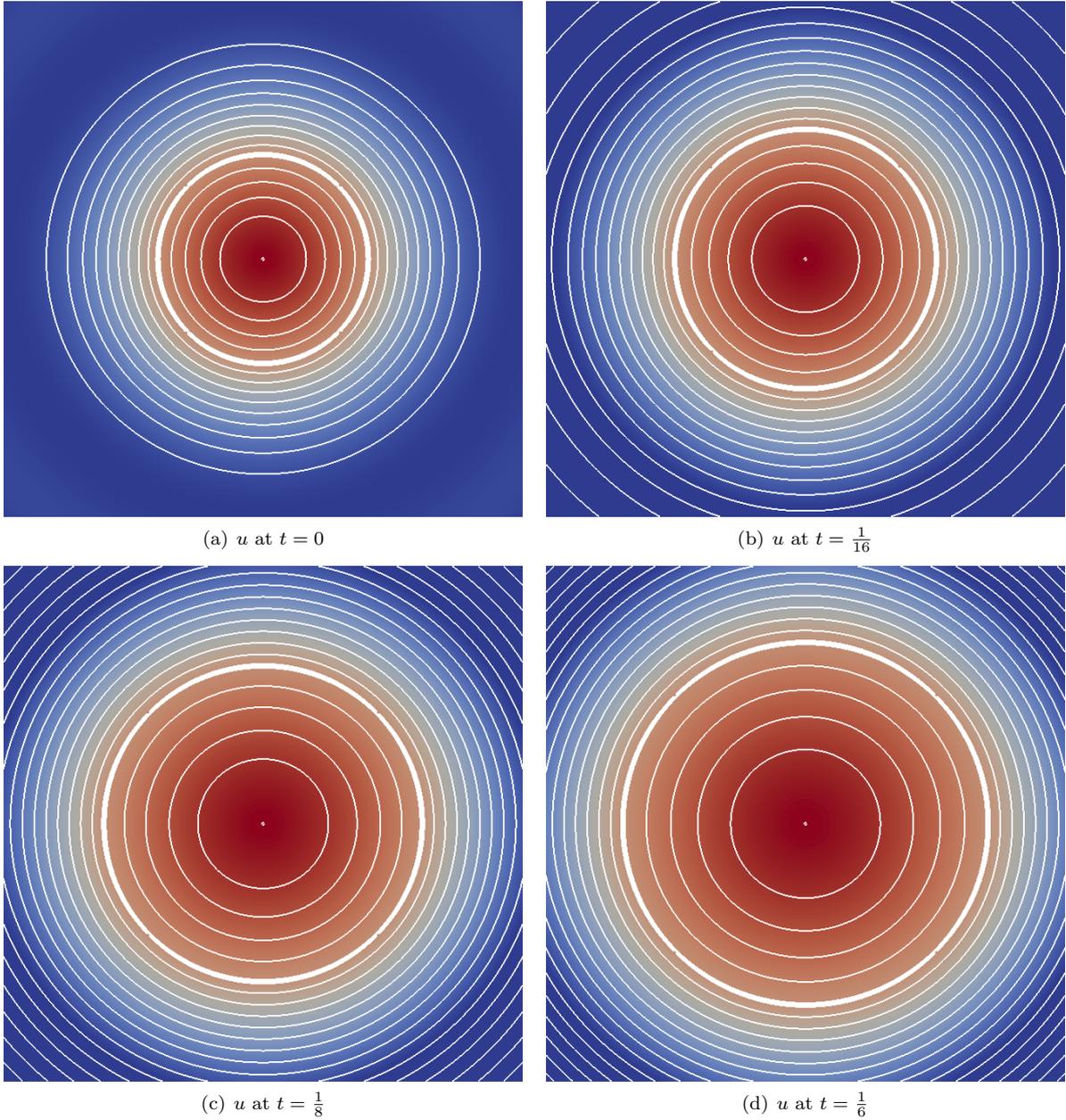


FIGURE 5.3. Example 2: analytical solution  $u$  at different time instants  $t$ .  $u$  is represented by its level sets  $\{\mathbf{x} \in \Omega \mid \exists z \in \mathbb{Z} : u(\mathbf{x}, t) = 0.1 \cdot z\}$ . The interface  $\Gamma(t)$  coinciding with the zero level  $\{\mathbf{x} \in \Omega \mid u(\mathbf{x}, t) = 0\}$  is visualized using an increased line width.

with our level set toolbox [13] in order to solve the full coupled problem. This full coupled problem will be investigated in an upcoming article.

#### ACKNOWLEDGEMENT

The authors gratefully acknowledge the financial support by the DFG (German Research Foundation) for the subproject A3 within the Collaborative Research Center SFB 747 “Mikroaltumformen - Prozesse, Charakterisierung, Optimierung”.

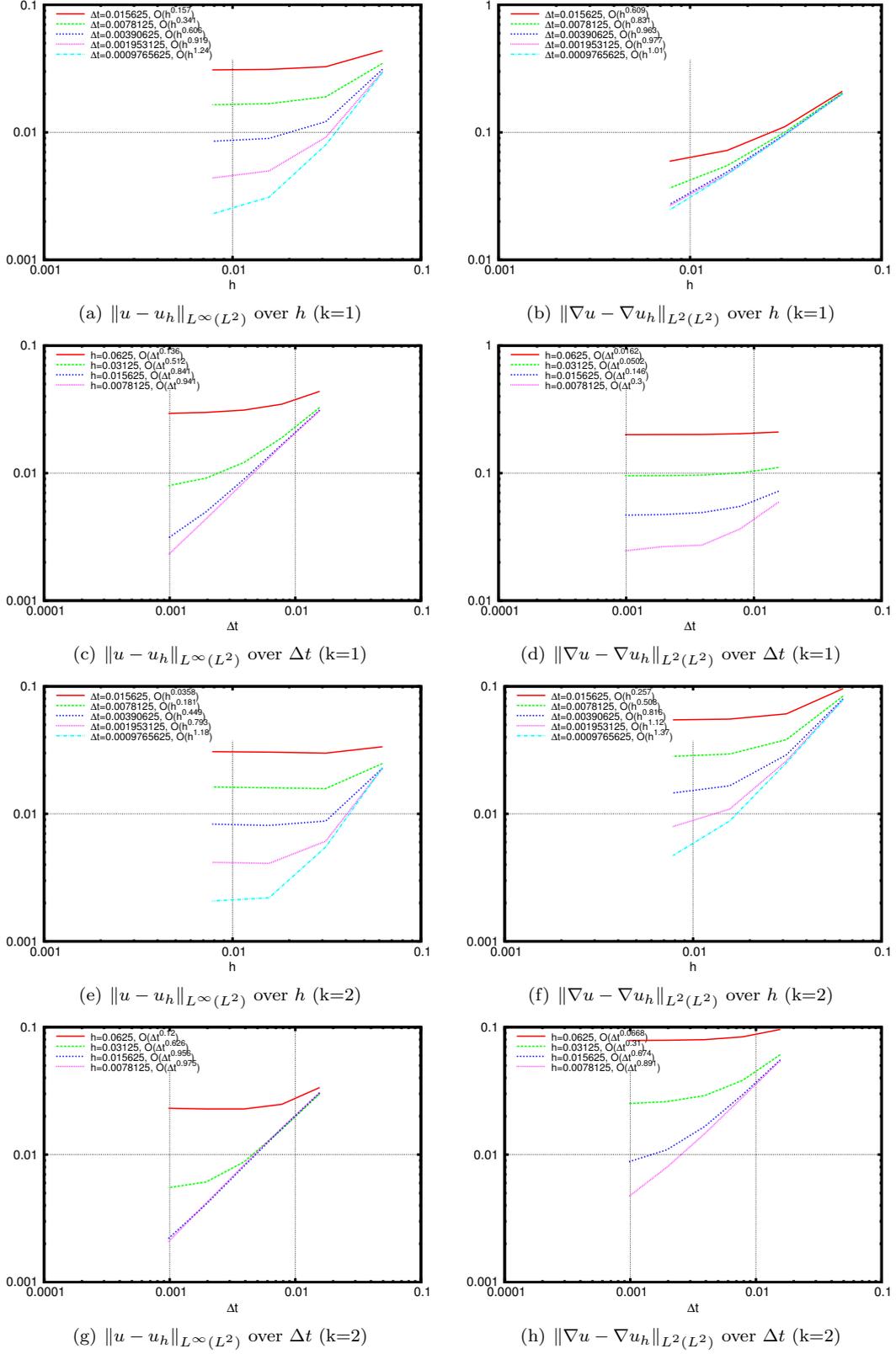


FIGURE 5.4. Example 2 convergence tests:  $\|u - u_h\|_{L^\infty(L^2)}$  and  $\|\nabla u - \nabla u_h\|_{L^2(L^2)}$  for  $\Delta t \in \{2^{-6}, 2^{-7}, 2^{-8}, 2^{-9}, 2^{-10}\}$ ,  $n_x \in \{16, 32, 64, 128\}$  and  $k = 1$ (a-d) resp.  $k = 2$  (e-h). An approximated order of convergence is shown in the legend for each plot.

## 7. APPENDIX

The discrete formulation of (3.2) is derived as follows: Multiply (3.2) with a test function  $v \in V_h$  and integrate over the domain  $\Omega_+ \cup \Omega_-$

$$(7.1) \quad \int_{\Omega_+ \cup \Omega_-} \xi u_h v_h d\mathbf{x} - \int_{\Omega_+ \cup \Omega_-} \nabla \cdot (\kappa \nabla u_h) v_h d\mathbf{x} = \int_{\Omega_+ \cup \Omega_-} \tilde{f} v_h d\mathbf{x}.$$

Integration by parts leads to

$$\begin{aligned} & \int_{\Omega_+ \cup \Omega_-} \xi u_h v_h d\mathbf{x} + \int_{\Omega_+ \cup \Omega_-} \kappa \nabla u_h \nabla v_h d\mathbf{x} - \int_{\partial(\Omega_+ \cup \Omega_-)} \kappa \nabla u_h \cdot \vec{n}_h v_h d\mathbf{c} \\ &= \int_{\Omega_+ \cup \Omega_-} \xi u_h v_h d\mathbf{x} + \int_{\Omega_+ \cup \Omega_-} \kappa \nabla u_h \nabla v_h d\mathbf{x} - \int_{\Gamma_D} \underbrace{\kappa \nabla u_h \cdot \vec{n}}_{=0} v_h d\mathbf{c} \\ & - \int_{\Gamma_N} \underbrace{\kappa \nabla u_h \cdot \vec{n}}_{=g} v_h d\mathbf{c} - \int_{\Gamma_h} \kappa^+ \nabla u_h^+ \cdot \vec{n}_h v_h^+ d\mathbf{c} + \int_{\Gamma_h} \kappa^- \nabla u_h^- \cdot \vec{n}_h v_h^- d\mathbf{c} \\ &= \int_{\Omega_+ \cup \Omega_-} \tilde{f} v_h d\mathbf{x}. \end{aligned}$$

Now, we add the “artificial” terms

$$0 = \mp \int_{\Gamma_h} \kappa^{\{+,-\}} \nabla v_h^{\{+,-\}} \cdot \vec{n}_h u_h^{\{+,-\}} d\mathbf{c} \pm \int_{\Gamma_h} \kappa^{\{+,-\}} \nabla v_h^{\{+,-\}} \cdot \vec{n}_h u_h^{\{+,-\}} d\mathbf{c}$$

and

$$0 = \int_{\Gamma_h} \lambda^{\{+,-\}} u_h^{\{+,-\}} v_h^{\{+,-\}} d\mathbf{c} - \int_{\Gamma_h} \lambda^{\{+,-\}} u_h^{\{+,-\}} v_h^{\{+,-\}} d\mathbf{c}$$

so the equation reads

$$\begin{aligned} & \int_{\Omega_+ \cup \Omega_-} \xi u_h v_h d\mathbf{x} + \int_{\Omega_+ \cup \Omega_-} \kappa \nabla u_h \nabla v_h d\mathbf{x} \\ & - \int_{\Gamma_h} \kappa^+ \nabla u_h^+ \cdot \vec{n}_h v_h^+ d\mathbf{c} - \int_{\Gamma_h} \kappa^+ \nabla v_h^+ \cdot \vec{n}_h u_h^+ d\mathbf{c} + \int_{\Gamma_h} \kappa^+ \nabla v_h^+ \cdot \vec{n}_h u_h^+ d\mathbf{c} \\ & + \int_{\Gamma_h} \kappa^- \nabla u_h^- \cdot \vec{n}_h v_h^- d\mathbf{c} + \int_{\Gamma_h} \kappa^- \nabla v_h^- \cdot \vec{n}_h u_h^- d\mathbf{c} - \int_{\Gamma_h} \kappa^- \nabla v_h^- \cdot \vec{n}_h u_h^- d\mathbf{c} \\ & + \int_{\Gamma_h} \lambda^+ u_h^+ v_h^+ d\mathbf{c} - \int_{\Gamma_h} \lambda^+ u_h^+ v_h^+ d\mathbf{c} + \int_{\Gamma_h} \lambda^- u_h^- v_h^- d\mathbf{c} - \int_{\Gamma_h} \lambda^- u_h^- v_h^- d\mathbf{c} \\ & = \int_{\Omega_+ \cup \Omega_-} \tilde{f} v_h d\mathbf{x} + \int_{\Gamma_N} g v_h d\mathbf{c} \end{aligned}$$

Now, we put one of each artificial term onto the right-hand-side and make use of the condition  $u^{\{+,-\}} = u_\Gamma$  on  $\Gamma_h$  and define  $\lambda^+ = \lambda^-$

$$\begin{aligned}
& \underbrace{\int_{\Omega_+ \cup \Omega_-} \xi u_h v_h d\mathbf{x} + \int_{\Omega_+ \cup \Omega_-} \kappa \nabla u_h \nabla v_h d\mathbf{x}}_{=:a(u_h, v_h)} \\
& - \underbrace{\int_{\Gamma_h} \kappa^+ \nabla u_h^+ \cdot \vec{n}_h v_h^+ d\mathbf{c} - \int_{\Gamma_h} \kappa^+ \nabla v_h^+ \cdot \vec{n}_h u_h^+ d\mathbf{c} + \int_{\Gamma_h} \lambda u_h^+ v_h^+ d\mathbf{c}}_{=:a_+(u_h, v_h)} \\
& + \underbrace{\int_{\Gamma_h} \kappa^- \nabla u_h^- \cdot \vec{n}_h v_h^- d\mathbf{c} + \int_{\Gamma_h} \kappa^- \nabla v_h^- \cdot \vec{n}_h u_h^- d\mathbf{c} + \int_{\Gamma_h} \lambda u_h^- v_h^- d\mathbf{c}}_{=:a_-(u_h, v_h)} \\
& = \underbrace{\int_{\Omega_+ \cup \Omega_-} \tilde{f} v_h d\mathbf{x} + \int_{\Gamma_N} g v_h d\mathbf{c}}_{=:L(v_h)} \\
& - \underbrace{\int_{\Gamma_h} \kappa^+ \nabla v_h^+ \cdot \vec{n}_h u_\Gamma d\mathbf{c} + \int_{\Gamma_h} \lambda u_\Gamma v_h^+ d\mathbf{c}}_{=:L_+(v_h)} + \underbrace{\int_{\Gamma_h} \kappa^- \nabla v_h^- \cdot \vec{n}_h u_\Gamma d\mathbf{c} + \int_{\Gamma_h} \lambda u_\Gamma v_h^- d\mathbf{c}}_{=:L_-(v_h)}.
\end{aligned}$$

## REFERENCES

- [1] M. S. Alnaes, A. Logg, K.-A. Mardal, O. Skavhaug, and H. P. Langtangen. Unified framework for finite element assembly. *International Journal of Computational Science and Engineering*, 4(4):231–244, 2009.
- [2] M. S. Alnaes, A. Logg, K. B. Olgaard, M. E. Rognes, and G. N. Wells. Unified form language: A domain-specific language for weak formulations of partial differential equations. *ACM Trans. Math. Softw.*, 40(2):9:1–9:37, March 2014.
- [3] P. M.A. Areias and T. Belytschko. A comment on the article “a finite element method for simulation of strong and weak discontinuities in solid mechanics” by a. hansbo and p. hansbo [comput. methods appl. mech. engrg. 193 (2004) 3523–3540]. *Computer Methods in Applied Mechanics and Engineering*, 195(9–12):1275 – 1276, 2006.
- [4] E. Bänsch, J. Paul, and A. Schmidt. An ALE finite element method for a coupled Stefan problem and Navier–Stokes equations with free capillary surface. *International Journal for Numerical Methods in Fluids*, 2012.
- [5] K. W. Cheng and T.-P. Fries. Higher-order XFEM for curved strong and weak discontinuities. *International Journal for Numerical Methods in Engineering*, 82(5):564–590, 2010.
- [6] J. Chessa, P. Smolinski, and T. Belytschko. The extended finite element method (xfem) for solidification problems. *International Journal for Numerical Methods in Engineering*, 53(8):1959–1977, 2002.
- [7] J. Dolbow and I. Harari. An efficient finite element method for embedded interface problems. *International Journal for Numerical Methods in Engineering*, 78(2):229–252, 2009.
- [8] G. Dziuk. *Theorie und Numerik partieller Differentialgleichungen*. De-Gruyter-Studium. De Gruyter, 2010.
- [9] T.-P. Fries and T. Belytschko. The extended/generalized finite element method: An overview of the method and its applications. *International Journal for Numerical Methods in Engineering*, 84(3):253–304, 2010.
- [10] T.-P. Fries and A. Zilian. On time integration in the xfem. *International Journal for Numerical Methods in Engineering*, 79(1):69–93, 2009.
- [11] S. Gross and A. Reusken. *Numerical Methods for Two-phase Incompressible Flows*. Springer Series in Computational Mathematics. Springer, 2011.
- [12] A. Hansbo and P. Hansbo. An unfitted finite element method, based on nitsche’s method, for elliptic interface problems. *Computer Methods in Applied Mechanics and Engineering*, 191(47–48):5537 – 5552, 2002.
- [13] M. Jahn and T. Klock. A level set toolbox including reinitialization and mass correction algorithms for FEniCS. Technical Report 16-01, ZeTeM, Bremen, 2016.
- [14] M. Jahn, A. Luttmann, and A. Schmidt. Finite element simulation for material accumulation and welding processes including a free melt surface. *PAMM*, 13(1):235–236, 2013.
- [15] M. Jahn, A. Luttmann, A. Schmidt, and J. Paul. Finite element methods for problems with solid-liquid-solid phase transitions and free melt surface. *PAMM*, 12(1):403–404, 2012.
- [16] M. Jahn and A. Schmidt. Finite element simulation of a material accumulation process including phase transitions and a capillary surface. Technical Report 12-03, ZeTeM, Bremen, 2012.
- [17] R. C. Kirby and A. Logg. A compiler for variational forms. *ACM Trans. Math. Softw.*, 32(3):417–444, September 2006.
- [18] A. Logg, K.-A. Mardal, and G. N. Wells, editors. *Automated Solution of Differential Equations by the Finite Element Method*, volume 84 of *Lecture Notes in Computational Science and Engineering*. Springer, 2012.
- [19] A. Logg and G. N. Wells. DOLFIN: Automated finite element computing. *ACM Trans Math Software*, 37(2):20:1–20:28, 2010.
- [20] R. Merle and J. Dolbow. Solving thermal and phase change problems with the extended finite element method. *Computational Mechanics*, 28(5):339–350.
- [21] M. Nikbakht. *Automated Solution of Partial Differential Equations with Discontinuities using the Partition of Unity Method*. PhD thesis, 2012.
- [22] M. Nikbakht and GN Wells. Automated modelling of evolving discontinuities. *Algorithms*, 2:1008–1030, 2009.
- [23] J. Nitsche. Über ein variationsprinzip zur lösung von dirichlet-problemen bei verwendung von teilräumen, die keinen randbedingungen unterworfen sind. *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg*, 36(1):9–15, 2013.
- [24] K. B. Olgaard and G. N. Wells. Optimizations for quadrature representations of finite element tensors through automated code generation. *ACM Trans. Math. Softw.*, 37(1):8:1–8:23, January 2010.
- [25] S. Osher and J. A. Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on hamilton-jacobi formulations. *Journal of Computational Physics*, 79(1):12 – 49, 1988.
- [26] J. A. Sethian. A fast marching level set method for monotonically advancing fronts. *Proceedings of the National Academy of Sciences*, 93(4):1591–1595, 1996.
- [27] T. Williams, C. Kelley, and many others. Gnuplot 5.0: An interactive plotting program. <http://www.gnuplot.info>, 2015.

THE CENTER FOR INDUSTRIAL MATHEMATICS AND MAPEX CENTER FOR MATERIALS AND PROCESSES, UNIVERSITY OF BREMEN, 28359 BREMEN

*E-mail address:* `mischa@math.uni-bremen.de`

*E-mail address:* `andreas1@math.uni-bremen.de`