Numerical solution of the Stefan problem in level set formulation with the eXtended finite element method in FEniCS

Mischa Jahn                        Timo Klock

Report 17–01
NUMERICAL SOLUTION OF THE STEFAN PROBLEM IN LEVEL SET FORMULATION WITH THE EXTENDED FINITE ELEMENT METHOD IN FENICS

M. JAHN AND T. KLOCK

Abstract. In this article, we consider the Stefan problem as an example for a process with a time-dependent discontinuity. The problem is modeled in level set formulation and discretized using the extended finite element method (XFEM) with Heaviside enrichment in combination with Nitsche's method. For efficiency, a narrow band approach is used for computing the interface's velocity based on the Stefan condition as well as maintaining the level set function. The discretized problem is solved with our XFEM library miXFEM and the associated level set toolbox, both based upon the FEniCS framework. Our approach is tested by considering different model variants of the Stefan problem with known analytical solution and numerical results and convergence studies are presented.

1. Introduction

In materials science and applied physics, processes with time-dependent discontinuities are very common. From a mathematical point of view, the modeling and simulation of these type of problems is very interesting and challenging. The Stefan problem is a well known example for such a process. Common approaches to compute a numerical solution to the Stefan problem are moving mesh methods, see e.g. [1], which are based on an explicitly defined sharp interface, and enthalpy methods, cf. for example [35], introducing the interface implicitly by considering the energy balance. Unfortunately, both methods have their drawbacks, see i.a. [9,11,21] and references therein. In moving mesh methods for example, there is not only 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 may lack accuracy near the interface 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 [20], 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. [14] for an overview. XFEM is a very flexible approach which allows for the accurate approximation of functions with strong discontinuities (jumps) and weak discontinuities (kinks) within elements by enriching the discrete function space(s) with additional basis functions at the interface location. This location is given by some indicator function whose movement is described by the level set method so it can move arbitrarily through the computational mesh.

In this article, we consider the Stefan problem in level set formulation and represent the interface by the zero level set of a signed distance function whose evolution is a-priori unknown and part of the solution. In contrast to the usual approach which is based on a weak enrichment, cf. i.a. [11,36], a Heaviside enrichment is used to enrich the function space locally for the temperature solution. As this type of enrichment generally allows for jumps in a function, we use Nitsche’s method [29] to enforce internal Dirichlet conditions [12] to get a continuous temperature distribution. For efficiency, a narrow band is introduced and the level set problem as well as the interface’s velocity is only computed for this region. Since the signed distance property may get lost during the evolution of the level set function, a reinitialization method and a mass correction approach are included.

The full problem is decoupled and solved using our previously developed XFEM toolbox miXFEM [19] and the associated level set toolbox [18] which both base the FEniCS framework [24]. FEniCS is a framework for the automated solution of PDE problems where a user can specify a problem in a specific language close to the (discretized) mathematical weak formulation and let the software generate most of the corresponding code automatically. Our toolboxes enhance the FEniCS framework so that problems with arbitrary time-dependent discontinuities can be tackled in the same way.

This paper is organized as follows: 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 and, afterwards, state the full coupled problem. Since we use different discretization methods for the Stefan problem and the level set problem, we dedicate one section to each problem. In Section 3, we derive the time discrete formulation of the Stefan problem and define a weak formulation for each time step. The spatial discretization is based on Nitsche’s method [29] treating the interface as internal Dirichlet
boundary [12]. As for the level set problem, we also derive a suitable discrete formulation and, additionally, introduce an discrete interface representation method as well as techniques to maintain the level set function in regards to the signed distance property and volume conservation. The level set method is more efficient, if used on a narrow band. This approach is also presented as well as mechanisms to compute the velocity field for the interface’s evolution. In Section 5, we provide some information concerning the FEniCS framework, our toolboxes and the specific implementation. Results for different model examples including a convergence study are given in Section 6 together with a summary of the most important aspects of this work.

2. Mathematical setting

2.1. The Stefan problem. Let \( \Omega \subset \mathbb{R}^d \), with \( \partial \Omega \) polygonally, be a fixed domain consisting for \( t \in [t_0, t_f] \) of two regions \( \Omega_1(t) \) and \( \Omega_2(t) \) that are separated by an interface \( \Gamma(t) \). We assume \( \Gamma(t) \) is sharp and sufficiently smooth for all \( t \in [t_0, t_f] \) and introduce the normal vector \( \vec{n}(t, \vec{x}) \) to \( \Gamma(t) \) pointing from \( \Omega_1 \) into \( \Omega_2 \).

The temperature field is given by \( u : \Omega \times [t_0, t_f] \rightarrow \mathbb{R} \) with \( u|_{\Omega_i} = u_i, \; i \in \{1, 2\} \) and its evolution is described by

\[
\rho \frac{\partial u}{\partial t} - \nabla \cdot (\kappa \nabla u) = f, \quad \text{in} \; \Omega_1(t) \cup \Omega_2(t), \; t \in (t_0, t_f).
\]

For simplicity, we choose \( \rho = 1 \), as well as \( c = 1 \) and only assume that \( \kappa|_{\Omega_1} = \kappa_i \) may be discontinuous with \( \kappa_i \) being piecewise constant in each subdomain\(^1\). For the boundary \( \partial \Omega = \Gamma_D \cup \Gamma_N \) with \( \Gamma_D \cap \Gamma_N = \emptyset \), the following conditions are given

\[
u = u_D, \quad \text{on} \; \Gamma_D \times (t_0, t_f),
\]

\[
\kappa \frac{\partial u}{\partial n} = g_N, \quad \text{on} \; \Gamma_N \times (t_0, t_f),
\]

where \( \vec{n} \) denotes the outer normal to \( \partial \Omega \). Initially, the temperature distribution in \( \Omega_1(t_0) \cup \Omega_2(t_0) \) is given by

\[
u(\cdot, t_0) = u_0
\]

and, moreover, we expect the so-called isothermal interface condition

\[
u(\cdot, t) = u_\Gamma \quad \text{on} \; \Gamma(t)
\]

to hold for all times \( t \in [t_0, t_f] \) and assume

\[
\nu(\cdot, t) < u_\Gamma \; \text{in} \; \Omega_1(t) \quad \text{and} \quad \nu(\cdot, t) > u_\Gamma \; \text{in} \; \Omega_2(t).
\]

As for the interface, we initially have

\[
\Gamma(t_0) = \{ \vec{x} \in \Omega : \nu(\vec{x}, t_0) = u_\Gamma \}.
\]

The movement of the interface \( \Gamma(t) \) in time for \( t > t_0 \) in terms of its normal velocity \( \vec{V} \cdot \vec{n} \) is given by

\[
[k \nabla u \cdot \vec{n}] = L \vec{V} \cdot \vec{n} \quad \text{on} \; \Gamma,
\]

with \( [\cdot] \) denoting the jump that is defined for a function \( \phi \) by \( [\phi] = \phi|_{\Omega_1} - \phi|_{\Omega_2} \). Roughly speaking, 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 material’s 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 implicitly or explicitly. As mentioned before, a very common approach in the XFEM context is to use the level set method for this purpose.

2.2. The level set method. Within the level set method [31, 34], 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.

\[
\Gamma(t) = \{ \vec{x} \in \Omega : \varphi(\vec{x}, t) = 0 \}, \quad t \in [t_0, t_f],
\]

The subdomains \( \Omega_1 \) and \( \Omega_2 \) can be defined by \( \vec{x} \in \Omega_1(t) \Leftrightarrow \varphi(\vec{x}, t) < 0 \) and \( \vec{x} \in \Omega_2(t) \Leftrightarrow \varphi(\vec{x}, t) > 0 \). An exemplary sketch of a 2D situation where a hold-all domain \( \Omega \) is divided by the sign of the function \( \varphi \) into subdomains \( \Omega_1(t) \) and \( \Omega_2(t) \) is given in Fig. 2.1.a and some more level sets of \( \varphi \) are indicated in Fig. 2.1.b.

The level set method i.a. allows for an easy computation of the normal \( \vec{n} \) to \( \Gamma \)

\[
\vec{n} = \frac{\nabla \varphi}{||\nabla \varphi||}.
\]

\(^1\)Please note that discontinuous coefficients \( \rho \) and \( c \) could easily be considered by defining the thermal diffusivity \( \tilde{\kappa} := \frac{\kappa}{\rho c} \) and appropriate scaling of the right-hand-side and the boundary conditions.
\frac{∂\varphi}{∂t} + \vec{V} \cdot \nabla \varphi = 0 \quad \text{in } \Omega \times [t_0, t_f],

\varphi(\cdot, t_0) = \varphi_0(\cdot) \quad \text{in } \Omega,

\varphi(\cdot, t) = \varphi_{in}(\cdot, t) \quad \text{on } \partial \Omega_{in} \times [t_0, t_f].

2.3. Coupled problem. Combining both models, the full coupled Stefan problem in level set formulation with non-prescribed interface as considered in this paper is given by:

Find \( \varphi \in C^1(\Omega \times [t_0, t_f]) \cap C^0(\Omega \times [t_0, t_f]) \) and \( u \) sufficiently smooth, i.e. \( u \in C^0(\Omega \times [t_0, t_f]), u(\cdot, t)|_{\Omega_i} \in C^2(\Omega_i(t)) \) and \( \partial_i u(\cdot, t) \in C^0(\Omega_i(t) \cup \Omega_2(t)) \) for \( t \in [t_0, t_f] \) and \( i = 1, 2 \), such that

\begin{align}
(2.5a) & \quad \frac{∂u}{∂t} - \nabla \cdot (\kappa \nabla u) = f, \quad \text{in } \Omega_1(t) \cup \Omega_2(t), \quad t \in (t_0, t_f), \\
(2.5b) & \quad u = u_D, \quad \text{on } \Gamma_D \times (t_0, t_f), \\
(2.5c) & \quad \frac{∂u}{∂n} = g_N, \quad \text{on } \Gamma_N \times (t_0, t_f), \\
(2.5d) & \quad u(\cdot, t_0) = u_0, \quad \text{in } \Omega_1(t_0) \cup \Omega_2(t_0), \\
(2.5e) & \quad u(\cdot, t) = u_T, \quad \text{on } \Gamma(t), \\
(2.6) & \quad [\kappa \nabla u \cdot \vec{n}] = L\vec{V} \cdot \vec{n}, \quad \text{on } \Gamma(t),
\end{align}
for given data $u_D$, $g_N$, $u_0$, $u_I$, $\varphi_0$ and $\varphi_I$ that are assumed to be sufficiently smooth. Thereby, equation (2.6) couples heat equation and level set problem and only provides the normal component of $\vec{V}(t)$ on $\Gamma(t)$. Consequently, more effort is needed to obtain a full velocity field which is needed for the solution of the transport problem (2.7). We will discuss this in Section 4.7.

3. DISCRETIZATION OF THE STEFAN PROBLEM

As mentioned in [15], using space-time elements for deriving a suitable weak formulation of problem (2.5) is a natural approach. However, a weak formulation can also be introduced formally by using Rothe’s method, cf. [15, 16], as we will do in this section.

3.1. Discretization in time. The interval $[t_0, t_f]$ is discretized by $N_t + 1$ time steps into $t_n = t_0 + n\Delta t$, $n = 0, \ldots, N_t$, with $\Delta t$ denoting the time step size, and the implicit Euler time discretization is applied to problem (2.5) which then reads: For $n = 0, \ldots, N_t$, find $u^{n+1} \approx u(\cdot, t_{n+1})$ such that

\begin{align}
(3.1) & \quad \frac{u^{n+1}}{\Delta t} - \nabla \cdot (\kappa \nabla u^{n+1}) = f^{n+1} + \frac{u^n}{\Delta t}, \quad \text{in } \Omega_1(t_{n+1}) \cup \Omega_2(t_{n+1}), \\
(3.2) & \quad u^{n+1} = u_D^{n+1}, \quad \text{on } \Gamma_D(t_{n+1}), \\
(3.3) & \quad \kappa \frac{\partial u^{n+1}}{\partial n} = g_N^{n+1}, \quad \text{on } \Gamma_N(t_{n+1}), \\
(3.4) & \quad u^{n+1} = u_I, \quad \text{on } \Gamma(t_{n+1}), \\
(3.5) & \quad [\kappa \nabla u^{n+1} \cdot \vec{n}] = \vec{L} \cdot \vec{n}, \quad \text{on } \Gamma(t_{n+1}).
\end{align}

For a fixed $n \in \{1, \ldots, N_t\}$, we use the notation $\xi = \frac{1}{\Delta t}$ and obviate the time dependency by setting $\Omega_1 := \Omega_1(t_{n+1})$, $\Omega_2 := \Omega_2(t_{n+1})$, $\bar{u} := u^{n+1}$, et cetera, and summarize the right-hand-side in (3.1) by $\bar{f}$ so that, eventually, we end up with the stationary problem

\begin{align}
(3.6) & \quad \xi \bar{u} - \nabla \cdot (\kappa \nabla \bar{u}) = \bar{f}, \quad \text{in } \Omega_1 \cup \Omega_2, \\
(3.7) & \quad \bar{u} = u_D, \quad \text{on } \Gamma_D, \\
(3.8) & \quad \kappa \frac{\partial \bar{u}}{\partial n} = g_N, \quad \text{on } \Gamma_N, \\
(3.9) & \quad \bar{u} = u_I, \quad \text{on } \Gamma, \\
(3.10) & \quad [\kappa \nabla \bar{u} \cdot \vec{n}] = \vec{L} \cdot \vec{n}, \quad \text{on } \Gamma,
\end{align}

for each time step. By using this simplified notation we are able to drop the time as an argument of the function space, however, we want to stress that all XFEM function spaces depend on $\Gamma(t)$ and, consequently, on time, cf. Section 5.

3.2. Weak formulation. Since we want to solve the problem using the extended finite element method, we introduce the affine space

$$H^1_d(\Omega_1 \cup \Omega_2) := \{ u \in L_2(\Omega) : u|_{\Omega_i} \in H^1(\Omega_i), i = 1, 2, u|_{\Gamma_D} = u_D \},$$

where each element $u \in H^1_d(\Omega_1 \cup \Omega_2)$ can be restricted onto a subdomain by $v_i := u|_{\Omega_i}$. While we choose the general setting with $u_D$ as Dirichlet boundary condition (in trace sense), setting $u_D = 0$ leads to the more familiar Hilbert space $H^1_d(\Omega_1 \cup \Omega_2)$ which will be used in (3.11).

For functions $u, v \in H^1_d(\Omega_1 \cup \Omega_2) \subset H^1(\Omega_1 \cup \Omega_2)$, we define

$$\langle u, v \rangle_{H^1(\Omega_1 \cup \Omega_2)} := \langle u, v \rangle_{H^1(\Omega_1)} + \langle u, v \rangle_{H^1(\Omega_2)}.$$
with
\[(u_i, v_i)_{H^1(\Omega_i)} := \int_{\Omega_i} \nabla u_i \nabla v_i \, dx, \quad i = 1, 2.\]

By using this definition and the \(L^2\)-norm we end up with the norm
\[\left( \| \cdot \|_{L^2(\Omega)}^2 + \| \cdot \|_{H^1(\Omega_1 \cup \Omega_2)}^2 \right)^{1/2} =: \| \cdot \|_{H^1(\Omega_1 \cup \Omega_2)}\]
and define the corresponding Hilbert space

\[(3.6) \text{ is then given by: For }\]

\[V_0 := \{ v \in H_0^1(\Omega_1 \cup \Omega_2) : v|_\Gamma = 0 \}\]

and the affine space\(^3\)

\[(3.12) \quad V_\Gamma := \{ v \in H_0^1(\Omega_1 \cup \Omega_2) : v|_\Gamma = u_\Gamma \}\]

where the interface conditions are introduced in a trace sense, as before. A weak formulation of the problem \((3.6)\) is then given by: For \(\xi, \kappa \in L^\infty(\Omega), f \in L^2(\Omega)\) and \(g_N \in L^2(\Gamma_N)\) find \(u \in V_\Gamma\) s.t.

\[(3.13) \quad (\xi u, v)_{L^2(\Omega_1 \cup \Omega_2)} + (\kappa u, v)_{H^1(\Omega_1 \cup \Omega_2)} = (f, v)_{L^2(\Omega)} + (g_N, v)_{L^2(\Gamma_N)} \]

for all \(v \in V_0\). Using the theorem of Lax-Milgram, one can show that there exists an unique solution to \((3.13)\).

### 3.3. Discretization in space based on Nitsche’s method.

Now, let \(\{S_h\}_{h>0}\) be a family of shape regular triangulations consisting of \(d\)-simplices, with \(d\) denoting the dimension, and \(h\) is the maximum diameter \(h = \max_{S \in S_h} \text{diam}(S)\). Furthermore, let \(\Omega_{h,i}, i = 1, 2\), be the discrete counterparts of \(\Omega_i\) separated by an (at this point arbitrary) approximation \(\Gamma_h\) of \(\Gamma\) and \(S_i := \Omega_{h,i} \cap \Omega_{h,2}\). In our approach, the interface \(\Gamma_h\) is not considered explicitly as facets within the triangulation, so the condition \(v|_{\Gamma_h} = u_\Gamma\) can not be included into the discrete function space in the same way as the outer Dirichlet condition on \(\Gamma_D\). Thus, we introduce the function space

\[(3.14) \quad V_{h,D}^k := \{ v \in H_0^1(\Omega_{1,h} \cup \Omega_{2,h}) : v_i \in C^0(\Omega_i), v|_{S_i} \in P_k, i = 1, 2, S \in S_h \}\]

for \(k \in \mathbb{N}\), where we only request that for a function \(v\) only the restriction \(v_i\) has to be continuous on \(\Omega_{i,h}, i = 1, 2\) but not on \(\Omega_h = \Omega_{1,h} \cup \Omega_{2,h}\). Please note that the corresponding function space without Dirichlet condition is denoted by \(V_{h}^k\). To consider the internal Dirichlet condition \(u_h = u_\Gamma\) on \(\Gamma_h\), Nitsche’s method [29] is used to include this condition weakly into the discrete problem formulation. Hence, we end up treating \((3.13)\) as two “independent” problems.

**Spatial discretization.** Following [12], the discrete formulation of \((3.6)\) is given by: Find \(u_h \in V_{h,D}^k\) s.t.

\[(3.15) \quad a(u_h, v_h) + a_1(u_h, v_h) + a_2(u_h, v_h) = L(v_h) + L_1(v_h) + L_2(v_h) \]

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

\[a(u_h, v_h) = \int_{\Omega_{1,h} \cup \Omega_{2,h}} \xi u_h v_h \, dx + \int_{\Omega_{1,h} \cup \Omega_{2,h}} \kappa \nabla u_h \nabla v_h \, dx\]

\[a_1(u_h, v_h) = - \int_{\Gamma_h} \kappa_1 \nabla v_{1,h} \cdot \vec{n} \bar{v}_{1,h} \, d\Gamma - \int_{\Gamma_h} \kappa_1 \nabla v_{1,h} \cdot \vec{n} \bar{u}_{1,h} \, d\Gamma + \int_{\Gamma_h} \lambda u_{1,h} v_{1,h} \, d\Gamma\]

\[a_2(u_h, v_h) = \int_{\Gamma_h} \kappa_2 \nabla v_{2,h} \cdot \vec{n} \bar{v}_{2,h} \, d\Gamma + \int_{\Gamma_h} \kappa_2 \nabla v_{2,h} \cdot \vec{n} \bar{u}_{2,h} \, d\Gamma + \int_{\Gamma_h} \lambda u_{2,h} v_{2,h} \, d\Gamma\]

\[L(v_h) = \int_{\Omega_{1,h} \cup \Omega_{2,h}} \vec{f} v_h \, dx + \int_{\Gamma_{N,h}} g_N v_h \, ds\]

\[L_1(v_h) = - \int_{\Gamma_h} \kappa_1 \nabla v_{1,h} \cdot \vec{n} \bar{u}_{1,h} \, d\Gamma + \int_{\Gamma_h} \lambda u_{1,h} v_{1,h} \, d\Gamma, \]

\[L_2(v_h) = \int_{\Gamma_h} \kappa_2 \nabla v_{2,h} \cdot \vec{n} \bar{u}_{2,h} \, d\Gamma + \int_{\Gamma_h} \lambda u_{2,h} v_{2,h} \, d\Gamma\]

cf. Appendix for more details. Here, \(0 < \lambda \in \mathbb{R}\) is a stability parameter which has to be chosen large enough and can be derived analytically for some situations [12]. Please note that the signs of the terms in \(a_i\) and \(L_i, i = 1, 2,\)

\(^3\)In this particular situation, one could also use \(H^1_0(\Omega)\) resp. \(H^1_{\Gamma,D}(\Omega)\) 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 strongly discontinuities as they may occur in more general problems.

\(^4\)We want to stress that due to our \(\Omega\) with \(\partial \Omega\) polygonal, we have \(\Omega_h = \Omega\) without any further requirements on the boundary of the \(d\)-simplices.
result from the direction of the normal vector $\vec{n}_h$ pointing from $\Omega_{2,h}$ to $\Omega_{1,h}$. 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 (3.15).

3.4. XFEM representation of the function space. The general idea of the extended finite element method is to represent a function by a standard and an enriched part, where the enrichment should be locally restricted. Thus, most of the simplices and degrees of freedom can be considered just as in the standard finite element context while only a minor subset needs special attention so that the assembled matrices and vectors are still sparse. While there are various approaches possible for a concrete representation of a basis of the function space (3.14), we choose a Heaviside enrichment, cf. i.a. [14,26,28], so that $u_h \in V_{k,h}$, is given by

$$u_h = \sum_{i \in N} u_i v_i + \sum_{j \in \hat{N}} \hat{u}_j H v_j$$

with basis functions $v_i$, $i \in N$, of the associated standard Lagrangian function space $\hat{V}_h^k = \{ v_h \in C(\Omega_h) : v_h|_S \in P_k, \forall S \in \mathcal{S}_h \}$ and corresponding coefficients $u_i$. The index set of enriched basis functions $\hat{N}$ is defined by

$$\hat{N} := \{ i \in N : \text{meas}_{d-1}(\Gamma_h \cap \text{supp}(v_i)) > 0, v_i \in \hat{V}_h^k \},$$

$\hat{u}_j$ are the enriched coefficients, and

$$H(x) = \begin{cases} 1, & \text{for } x \in \Omega_{2,h} \\ 0, & \text{else} \end{cases}$$

is the Heaviside function. The advantage of using a strong enrichment, like the presented Heaviside enrichment, is its flexibility as it can be used for problems with strong and weak discontinuities by adding corresponding conditions using Nitsche’s technique.

Although using curved intersection segments and the corresponding adaption of the quadrature rules are possible as shown in [10], $k = 1$ is chosen as polynomial degree in this paper so that $\Gamma_h$ is an linear approximation of $\Gamma$, thus, making the intersecting segments $S_i$ linear, too. Fig. 3.1 shows a visualization of a standard and the corresponding enriched basis function for a 2D setting.

Remark: As shown in [5], this enrichment is equivalent to the method proposed by [17] which is called cut cell method and based on duplicating nodes of intersected elements.

4. Discretization of the level set problem

The level set problem can be discretized by both, the method of lines and Rothe’s method. Since using the method of lines is more common, we continue with this approach and, firstly, derive a suitable weak formulation before discretizing the problem. Please note that this subsection is just a brief overview of the approaches considered in our previous work [18] and the references therein.

4.1. Weak formulation. A weak formulation of the level set problem (2.4) can be easily derived using the time dependent function space

$$W(t)_{\vec{\varphi},\varphi_D} = \{ v \in L^2(\Omega) : \vec{V}(\cdot, t) \cdot \nabla v \in L^2(\Omega) \land v|_{\partial \Omega_{h,\nu}(t)} = \varphi_D(\cdot, t) \}.$$ 

By multiplying (2.3) with an arbitrary test function $v \in L^2(\Omega)$ and integrating over $\Omega$, we end up with the weak formulation of the level set problem (2.4): For $t \in (t_0, t_f)$ find $\varphi(\cdot, t) \in W_{\vec{\varphi},\varphi_D}(t)$ with $\frac{\partial \varphi}{\partial t} \in L^2(\Omega)$ s.t.
\( \varphi(\cdot, t_0) = \varphi_0 \) and
\[
(4.1) \quad \left( \frac{\partial \varphi}{\partial t}, v \right)_{L^2} + (\vec{V} \cdot \nabla \varphi, v)_{L^2} = 0, \quad \forall v \in L^2(\Omega), \ t \in [t_0, t_f].
\]

### 4.2. Discretization in space.

For the triangulations \( \{S_h\}_{h>0} \) we introduce the standard Lagrangian finite element space
\[
(4.2) \quad W_h^I = \{v_h \in C(\Omega_h) : v_h|_S \in \mathcal{P}_1, \forall S \in S_h \},
\]
and for functions with Dirichlet boundary conditions we define for \([t_0, t_f] \) the affine space
\[
(4.3) \quad W_{h,\varphi_D}(t) = \{v_h \in C(\Omega_h) : v_h|_S \in \mathcal{P}_1, \forall S \in S_h, v(x) = \varphi_D(x, t), \forall x \in \partial\Omega_{in,h}(t) \},
\]
with \( l \geq 1 \) and \( \partial\Omega_{in,h}(t) \) being the discrete influx boundary\(^5\). Using these function spaces, (4.1) discretized in space reads: For \( t \in [t_0, t_f] \) find \( \varphi(\cdot, t) \in W_{h,\varphi_D}^I \) with \( \vec{V}(t) \in L^\infty(\Omega_h) \) and \( \frac{\partial \varphi}{\partial t} \in L^2(\Omega_h) \) such that
\[
(4.4) \quad \sum_{S \in S_h} \left( \frac{\partial \varphi_h}{\partial t} + \vec{V} \cdot \nabla \varphi_h, v_h \right)_{L^2(S)} = 0, \quad \forall v_h \in W_h^I.
\]

In this paper as well as in many other applications like multi-phase flow, the polynomial degree \( l = 2 \) is chosen for the finite-dimensional function space (4.3). This is due to different reasons, for example the quality of the curvature approximation of the level set function containing second derivatives, as pointed out in [16]. Moreover, using quadratic basis functions has the additional advantage that the degrees of freedom coincide with the degrees of freedom of linear basis functions on a regularly refined mesh. This will be extensively exploited for characterizing the interface \( \Gamma \) discretely and by the reinitialization technique, see Section 4.5.

**Remark:** It is well known, that solving hyperbolic PDEs with standard finite element methods can be unstable, especially for high velocities \( \vec{V} \). An approach to overcome this issue is using a stabilization method [33] to slightly reformulate the discretized problem to enforce stability. A method well known in literature is the Streamline-Upwind/Petrov-Galerkin (SUPG) stabilization [8]. In this paper however, we do not use any stabilization technique due to the small velocities and the absence of an advection term in (2.1). Further reference on this topic and the adjusted formulation of (4.4) can be found in [16] and [18].

### 4.3. Discretization in time.

For time discretization of (4.4), the so-called \( \theta \)-scheme is used. Since the time discretization of the level set problem may differ in comparison to the discretization described in Section 3.1, we now discretize the interval \([t_0, t_f] \) by \( \bar{N}_t + 1 \) time steps \( t_n = t_0 + n\Delta t \), \( n = 0, \ldots, \bar{N}_t \) with \( \Delta t \) denoting the time step. Let \( \theta \in [0,1] \) be a parameter\(^6\) and \( \varphi_h(\cdot, t_n) \) be an approximation of the level set function \( \varphi \) at time \( t_n \). The completely discretized level set problem reads
\[
(4.5) \quad \sum_{S \in S_h} \left( \frac{\varphi_h^{n+1} - \varphi_h^n}{\Delta t} + \theta \vec{V}^{n+1} \cdot \nabla \varphi_h^{n+1} + (1 - \theta) \vec{V}^n \cdot \nabla \varphi_h^n, v_h \right)_{L^2(S)} = 0, \quad \forall v_h \in W_h^I.
\]

### 4.4. Representation of \( \Gamma \).

An important aspect of the discretization is the discrete approximation of the interface \( \Gamma \). While we assumed \( \Gamma_h \) to be an arbitrary approximation of \( \Gamma \) for the formal definition of the function spaces and the introduction of the discrete formulation of the Stefan problem in Section 3, we now describe the approach chosen in this article which follows the idea presented in [16].

For \( t_n = t_0 + n\Delta t, \ n = 0, \ldots, \bar{N}_t \), let \( \varphi_h(\cdot, t_n) \in W_h^I \) be the finite element approximation of the level set function \( \varphi \) at time \( t_n \). The completely discretized level set problem reads
\[
(4.6) \quad \mathcal{S}_h^{\Gamma} := \{ S \in S_h : \text{meas}_{d-1}(S \cap \Gamma_h) > 0 \}
\]
the set of simplices containing \( \Gamma_h \). We drop the time \( t_n \) as an argument in this paragraph in our notation for simplicity and define \( \mathcal{S}_h^{\Gamma/2} \) as the set consisting of all simplices that are obtained, if the elements in \( \mathcal{S}_h^{\Gamma} \) are regularly refined.

The finite element approximation \( \varphi_h \) of \( \varphi \) is then linearly interpolated by \( I\varphi_h \) using standard Lagrange interpolation on the patch of refined elements \( S \in \mathcal{S}_h^{\Gamma/2} \) and the discrete approximation of \( \Gamma \) is given by
\[
\Gamma_h := \{ x \in \Omega : I\varphi_h(x) = 0 \},
\]
as shown in Fig. 4.1 for a 2D setting. A detailed investigation about the approximation quality and the discretization error of this discrete interface representation is given in [16].

\(^5\)Note that the time dependency of this function space is only introduced by the influx boundary and the corresponding boundary condition. If both are time independent, the function space \( W_{h,\varphi_D} \) is also time independent.

\(^6\)Note that \( \theta = 0 \) leads to the explicit Euler-scheme while \( \theta = 1 \) results in the implicit Euler-scheme.
Remark: While a high order approximation of the interface $\Gamma$ is possible, cf. [10], using the presented approach to get a discrete representation $\Gamma_h$ has several advantages. First of all, the degrees of freedom (DOFs) of $\varphi_h$ and $I\varphi_h$ coincide due to the choice of $\varphi_h \in W^2_h,$ allowing for a fast interpolation. More important, the segments of $\Gamma_S, S \in \mathcal{S}_{h/2},$ are straight resp. planar which makes the computation of intersection points very easy. This fact is heavily utilized during the computation of the distances in the reinitialization method presented in the following section. Another advantage of this approach is that the discrete counterparts
\[ \Omega_{1,h} := \Omega_{1,h}(\varphi_h) = \{x \in \Omega : I\varphi_h(x, t_n) < 0\} \]
and
\[ \Omega_{2,h} := \Omega_{2,h}(\varphi_h) = \{x \in \Omega : I\varphi_h(x, t_n) > 0\} \]
at time $t_n = t_0 + n\Delta t, n = 0, \ldots, N_t,$ can be easily (re-) constructed.

4.5. Maintaining techniques. As mentioned earlier, it is beneficial to have a level set function $\varphi_h$ which is close to a signed distance function. Unfortunately, this property may be lost during the evolution of the level set function in time due to various reasons, e.g. discretization errors, insufficient approximation of the curvature and topological changes. To regain the signed distance property, the level set function is reinitialized with a variant [16] of the Fast Marching Method (FMM) [34], providing a signed distance approximation $\tilde{\varphi}_h$ of $\varphi_h.$ Since the FMM slightly distorts the interface $\Gamma_h$ and, consequently, is not volume-preserving, we present a volume correction algorithm in Section 4.5.2 which can be applied during the reinitialization process. Please note that a more detailed description of these techniques can also be found in [16] and our previous work [18].

4.5.1. Reinitialization via FMM. Given $\varphi_h \in W^2_h$ on $\mathcal{S}_h,$ we firstly compute the linear interpolation $I\varphi_h$ of $\varphi_h$ on the regularly refined triangulation $\mathcal{S}_{h/2},$ cf. Section 4.4. Let $\mathcal{V}(S)$ denote the set of vertices given on a simplex $S \in \mathcal{S}_{h/2}$ and $\mathcal{V} := \mathcal{V}(\mathcal{S}_{h/2})$ be the (discrete) set of all vertices of $\mathcal{S}_{h/2}.$ The patch of order $l + 1, l \geq 1,$ of elements related to a vertex $v \in \mathcal{V}(S), S \in \mathcal{S}_{h/2}$ is given by
\[ \mathcal{P}^{l+1}(v) := \{S \in \mathcal{S}_{h/2} : \mathcal{V}(S) \cup \mathcal{V}(\mathcal{P}^l(v)) \neq \emptyset\} \]
with
\[ \mathcal{P}^1(v) := \{S \in \mathcal{S}_{h/2} : v \in \mathcal{V}(S)\}. \]
The basic idea of the FFM, consisting of two phases, is to compute distance values $\hat{d}(v)$ for any vertex $v \in \mathcal{V}$ taking thereby advantage of the fact that all information will only propagate outwards from the zero level set of the function.

In the first, so-called initialization phase, distance values $\hat{d}(v)$ are then calculated for $v \in \mathcal{V}(\mathcal{S}_{h/2})$ by
\[ \hat{d}(v) = \min_{S \in \mathcal{P}^1(v)} \text{dist}(v, \Gamma_h, S). \]
Please note that for stability reasons, cf. [16], we do not consider the patch $\mathcal{P}^1(v)$ but the extended patch $\mathcal{P}^2(v)$ including also all second neighbor simplices for this computation.

In the second phase referred to as extension phase, the initially computed values are propagated into the far field. Therefore, we introduce the finished set
\[ \mathcal{V}_F = \{v \in \mathcal{V}(\mathcal{S}_{h/2}) : \hat{d}(v) \text{ is computed}\}, \]
which stores already processed vertices, as well as an active set
\[ \mathcal{V}_A = \{v \in \mathcal{V}(\mathcal{S}_{h/2}) : \text{At least one neighbor vertex of } v \text{ is in } \mathcal{V}_F\}. \]

\footnote{Please note that by choosing a linear Lagrangian basis, vertices and degrees of freedom coincide.}
storing vertices, which are likely to be dealt with next since they are neighbors of already processed vertices. The process of choosing the vertex \(v\) storing vertices, which are likely to be dealt with next is based on a tentative distance function

\[
d(v) = \min \{d_S(v) : S \in \mathcal{P}^1(v) \text{ with } \mathcal{V}(S) \cap \mathcal{V}_F = \emptyset \}
\]

with \(d_S(v)\) defined as

\[
d_S(v) = d(P_{\mathcal{W}(S)}(v)) + \|v - P_{\mathcal{W}(S)}(v)\|.
\]

Thereby, the function \(P_{\mathcal{W}(S)}(v)\) is the minimum distance projection of \(v\) onto the convex hull \(W(S) = \mathcal{V}(S) \cap \mathcal{V}_F\), i.e.

\[
P_{\mathcal{W}(S)}(v) = \arg\min_{x \in \text{conv}(W(S))} \|v - x\|.
\]

With this construction, the value \(d_S(v)\) approximates the distance of \(v\) to the discrete interface \(\Gamma_h\) by using the simplex \(S\) (which has at least one vertex in \(\mathcal{V}_F\)) as an information propagator. Out of all these possible values, the minimum value \(d_S(v)\) is used as the tentative distance value respectively the most likely distance value, if all processed vertices/simplices are considered as information propagators.

Once values \(d(v)\) can be calculated, the extension phase works by extracting the current nearest vertex \(v^* = \arg\min_{v \in \mathcal{V}_A} d(v)\), setting \(d(v^*) = d(v^*)\), removing this vertex from the active set \(\mathcal{V}_A\) and adding it to the finished set \(\mathcal{V}_F\). Then the active set \(\mathcal{V}_A\) and the tentative values \(d(v)\), \(v \in \mathcal{V}_A\) are updated according to the updated finished set and the procedure is repeated until all vertices are processed.

As the result of the FMM, we obtain unsigned distance values \(d(v)\) for every vertex \(v\) of the refined triangulation. These uniquely define a reinitialized piecewise quadratic level set function \(\tilde{\varphi}_h \in W_h^2\) due to the one-to-one relation between vertices in \(S_{h/2}\) and DOFs in \(W_h^2\). Note that the values \(d\) are unsigned, hence a multiplication with the correct sign is necessary before using them as new DOFs of \(\tilde{\varphi}_h\).

### 4.5.2. Volume conservation

While the reinitialized level set function \(\tilde{\varphi}_h\) is close to a signed distance function, its zero level set \(\Gamma_h\) does not coincide with the zero level set \(\Gamma_h\) of the original level set function \(\varphi_h\) any more. Consequently, the volume is not preserved during reinitialization. This is also true for the linearized level set function \(\varphi_h\) and \(\tilde{\varphi}_h\) on \(S_{h/2}\).

One way to overcome this issue is to apply a local volume conservation algorithm after the initialization phase of the FMM in which the new (unsigned) distance values \(d(v)\) for \(v \in \mathcal{V}(S_{h/2})\) are computed. In this article, we use the localized correction algorithm of [6] that is briefly explained in the following. For a more detailed description, we refer to [6] and [18].

As before, we only consider the linear level set functions \(I\varphi_h\) and \(I\tilde{\varphi}_h\) on \(S_{h/2}\). For arbitrary functions \(\phi, \psi\), we define the volume functional

\[
\Delta V(\phi_h, \psi_h, S) = \int_{\mathcal{V}(x \in \Omega_h : \phi < 0)} dx - \int_{\mathcal{V}(x \in \Omega_h : \psi < 0)} dx, \quad S \in S_{h/2}.
\]

Let \(\Omega_{\Gamma_h} = \bigcup_{S \in S_{h/2}} S\) be the domain of all intersected simplices and \(W_{h/2}(\Omega_{\Gamma_h})\) be the corresponding function space, cf. Eq. (4.2). The values \(d(v)\) computed in the initialization phase of the FMM then define a tentative level set function \(\tilde{\varphi}_{\text{ten}} \in W_{h/2}^1(\Omega_{\Gamma_h})\). To apply the volume conservation, we adjust the values of this function in the following four steps:

1. Calculate an offset \(C_S \in \mathbb{R}\) for every \(S \in S_{h/2}^\Gamma\) such that

\[
\Delta V(I(\varphi_h), \tilde{\varphi}_{\text{ten}}(\cdot) + C_S, S) = 0
\]

is satisfied. Thereby the addition \(\tilde{\varphi}_{\text{ten}}(\cdot) + C_S\) is understood as a DOF-wise addition, i.e. \(C_S\) is added to every DOF of \(\tilde{\varphi}_{\text{ten}}\).

2. Compute a continuous, piecewise linear offset function \(\phi_{\text{corr}} \in W_{h/2}^1(\Omega_{\Gamma_h})\) by averaging values \(C_S\) on \(\mathcal{P}^1(v)\) so that a value \(\phi_{\text{corr}}(v)\) for \(v\) connected to \(S \in S_{h/2}^\Gamma\) is given as

\[
\phi_{\text{corr}}(v) = \frac{1}{|S|} \sum_{S \in \mathcal{P}^1(v) \cap S_{h/2}^\Gamma} C_S.
\]

3. Search a global multiplier \(C_{\Omega} \in \mathbb{R}\) such that the equation

\[
\Delta V(I(\varphi_h), \tilde{\varphi}_{\text{ten}}(\cdot) + C_{\Omega}\phi_{\text{corr}}(\cdot), \Omega_h) = 0
\]

is satisfied. Since this is already the second optimization, the resulting \(C_{\Omega}\) is usually close to 1.
(4) Adjust the $\hat{d}(v)$ values for vertices that are processed in the FMM initialization phase by setting them to

$$\hat{d}(v) = \begin{cases} 
\hat{d}(v) - C_{\Omega} \phi_{\text{corr}}(v), & \text{if } v \in \Omega_{1,h}, \\
\hat{d}(v) + C_{\Omega} \phi_{\text{corr}}(v), & \text{if } v \in \Omega_{2,h}, 
\end{cases}$$

and proceed the extension phase of the FMM with these modified $\hat{d}(v)$ values.

The roots in steps 1 and 3 can e.g. be computed by using the regula falsi algorithm in the Anderson/Björk variant [4] as shown in [18].

4.6. **Narrow band approach.** A major drawback of the level set method as an interface representation technique is the inherent computational effort which is caused by using a higher dimensional object (the level set function) to represent the lower dimensional object (the interface). To overcome this drawback, the narrow band level set method [32] can be used. The basic idea in this is to restrict the steps of the interface evolution, namely the PDE solution and the reinitialization, to a small narrow band around the current interface.

4.6.1. **Construction of the narrow band(s).** Two sets of vertices are defined via a small $\gamma \in \mathbb{Z}^+$, namely the inner narrow band (4.7)

$$\mathcal{V}_{\text{INB}} = \{ v \in \mathcal{V}(S_h) : \varphi_h(v) < \gamma_h \},$$

with $\gamma_h = \gamma h$ and $h = \max_{S \in S_h} \text{diam}(S)$, and the outer narrow band

$$\mathcal{V}_{\text{ONB}} = \mathcal{V}_{\text{INB}} \cup \left( \bigcup_{v \in \mathcal{V}_{\text{INB}}} \bigcup_{S \in P(v)} \mathcal{V}(S) \right),$$

which corresponds to all vertices of the inner narrow band set as well as all vertices of the first neighbor patch of all simplices in the inner narrow band domain. An exemplary visualization of both sets can be seen in Figure 4.2. Using these set, we define the corresponding domains $\Omega_{\text{INB}} := \{ S \in S_h : \mathcal{V}(S) \subset \mathcal{V}_{\text{INB}} \}$ and $\Omega_{\text{ONB}} := \{ S \in S_h : \mathcal{V}(S) \subset \mathcal{V}_{\text{ONB}} \}$.

Remark: The narrow band method is based on the assumption that the level set function $\varphi_h(v)$ is an approximate signed distance function, cf. (4.7) so that the DOF values in the inner narrow band are be approximately equal to the exact distance from a vertex to the interface, making reinitialization even more important.

4.6.2. **Modifications to the level set problem.** Based on these definitions, the concept of the narrow band level set method is to solve (4.5) on $\Omega_{\text{INB}}$, reinitialize the solution on $\Omega_{\text{ONB}}$ and extend the function with a constant value $\pm(\gamma h + \varepsilon)$ outside of $\Omega_{\text{ONB}}$. Unfortunately, solving the original level set problem (4.5) on $\Omega_{\text{INB}}$ and reinitializing the solution on $\Omega_{\text{ONB}}$ often exhibits oscillations at the boundary $\partial \Omega_{\text{INB}}$ as shown in [32]. Due to this, a modified level set problem is introduced reading

$$\frac{\partial \varphi}{\partial t} + \zeta(\varphi) \bar{V} \cdot \nabla \varphi = 0 \quad \text{in } \Omega \times [t_0, t_f],$$

Figure 4.2. Different vertices in the narrow band level set method for a two-dimensional problem: $\mathcal{V}_{\text{INB}}$ in blue, $\mathcal{V}_{\text{ONB}}$ in blue and red.
where
\begin{equation}
\zeta(\varphi) = \zeta(\varphi(x, t)) = \begin{cases} 
1 & \text{if } |\varphi(x, t)| \leq \beta_{I,h}, \\
(\varphi(x, t) - \beta_{O,h})^2 \frac{2\varphi(x, t) + \beta_{O,h} - \beta_{I,h}}{\beta_{O,h} - \beta_{I,h}} & \text{if } \beta_{I,h} < |\varphi(x, t)| \leq \beta_{O,h}, \\
0 & \text{if } |\varphi(x, t)| > \beta_{O,h},
\end{cases}
\end{equation}

is a cutoff function that slowly decreases the influence of the advection term towards the boundary of $\Omega_{INB}$. The parameters $\beta_{I,h} = \beta_I h$ and $\beta_{O,h} = \beta_O h$ with $\beta_I < \beta_O < \gamma$ divide the inner narrow band layer into three sublayers, s.t. the cutoff parameter is equal to 1 in the innermost layer, tends to zero in the middle layer and is equal to zero in the outermost layer of the inner narrow band

As for the discretization of (4.8), we treat (4.9) explicitly w.r.t. time by defining $\zeta_h^{n+1} = \zeta((\varphi_h^n)^2)$ to avoid the task of solving a non-linear equation. For doing so, the innermost narrow band layer width $\beta_I h$ has to be chosen to be sufficiently large. Using this approach, the discretized level set problem on the narrow domain $\Omega_{INB}$ is given as
\begin{equation}
\sum_{S \in S_h} \left( \frac{\varphi_h^{n+1} - \varphi_h^n}{\Delta t} + \theta \zeta_h^{n+1} \tilde{V}^{n+1} \cdot \nabla \varphi_h^{n+1} + (1 - \theta) \zeta_h^n \tilde{V}^n \cdot \nabla \varphi_h^n, v_h \right)_{L^2(S)} = 0,
\end{equation}

for $v_h \in W_h^2$.

4.6.3. **CFL conditions.** When using the narrow band level set method, one has to consider two CFL conditions:

- Since $\zeta$ decreases the transport of the level set function everywhere but in the most inner band, the velocity $\tilde{V}$ must not exceed a value which would make the interface $\Gamma_h$ leave this region. Therefore, the CFL condition
\begin{equation}
\Delta t \| \tilde{V}^n \|_{L_{\infty}(\Omega_{INB})} < \beta_{I,h}, \quad \forall n \in \{0, \ldots, N_t\}
\end{equation}

must hold.

- For constructing $\Omega_{INB}^{n+1} \subset \Omega_{ONB}^n$, we need $\varphi_h$ to be close to a signed distance function on $\Omega_{ONB}^n$. If the velocity transporting the interface is too big, we may end up considering the constant values $\pm(\gamma_h + \varepsilon)$ during the solution and reinitialization process. To avoid this, the condition
\begin{equation}
\Delta t \| \tilde{V}^n \|_{L_{\infty}(\Omega_{INB})} < h, \quad \forall n \in \{0, \ldots, N_t\}
\end{equation}

has to be respected.

Remark: A typical parameter choice includes $\beta_I > 1$ so that (4.11) is automatically fulfilled, if (4.12) holds, making this the limiting condition. Please also note that even though the method’s description assumes the reinitialization procedure to be applied after every time step, it might be better to apply reinitialization and update the narrow band after every $m$-th time step instead. This results in a more restrictive CFL condition given by
\begin{equation}
m \Delta t \| \tilde{V}^n \|_{L_{\infty}(\Omega_{INB})} < h, \quad \forall n \in \{0, \ldots, N_t\}.
\end{equation}

4.7. **Construction of a velocity field.** The solution of the level set problem is based on knowing the velocity $\tilde{V}$. For the Stefan problem, this velocity field $\tilde{V}^n \in (W_h^1)^d$ can be computed in two steps by using the Stefan condition (2.2), which discretely reads
\begin{equation}
[k \nabla u_h^n \cdot \tilde{n}_h] = L \tilde{V}^n \cdot \tilde{n}_h \quad \text{on } \Gamma^n, \quad n \in \{1, \ldots, N_t\}.
\end{equation}

In the first step, (4.14) is used to compute the velocity at the interface which is then extended to the whole narrow band in a second step, making this approach very similar to the previously presented Fast Marching Method. However, please note that the velocity field is not calculated on the regularly refined mesh since we do not need a piecewise quadratic velocity function.

4.7.1. **Initialization phase.** First of all, we compute the projections $w_j$, $j \in N \subset N$, of $v$ onto the discrete interface $\Gamma_n$ such that $\|v - w_j\| = \text{dist}(v, \Gamma_n)$ holds for all $j$. As there can be multiple points that satisfy the minimum distance requirement, we may have multiple projections $w_j$. Now, we present two approaches that can be used to compute the corresponding values $\tilde{V}^n(v)$ for $v \in V(S), \ S \in S_h^\Gamma$.

\footnote{A viable choice for these parameters is $\beta_I = 2$, $\beta_O = 4$, $\gamma = 6$.}
Figure 4.3. Evaluation points of in the DSCE velocity calculation method

(1) **Direct gradient evaluation (DGE)** We compute the discrete temperature gradient\(^9\) \(\nabla u^n_{1,h}\), which is a piecewise constant, vectorial XFEM function, and the velocity field at the projections \(w_j\) directly using

\[
(\tilde{V}^n(w_j))_k = (\nabla(u^n_{1,h})(w_j))_k - (\nabla(u^n_{2,h})(w_j))_k,
\]

with index \(k = 1, \ldots, d\) denoting the respective component. The velocity vector at \(v\) is then defined by averaging over all contributions \((\tilde{V}^n(w_j))_k\) of all projections \(w_j\) that are found in the previous step.

(2) **Discretized Stefan condition evaluation (DSCE)** [7] For every projection \(w \in \{w_j : j \in \mathcal{N}\}\), we use point-value tuples

\[
\left( w^\pm \frac{\delta_h}{2}, u^n_{1,h} \left( w^\pm \frac{\delta_h}{2} \right) \right) \quad \text{and} \quad \left( w^\pm \frac{\delta_h}{2}, u^n_{2,h} \left( w^\pm \frac{\delta_h}{2} \right) \right), \quad l = 0, \ldots, 4,
\]

with

\[
w^{r \delta_h} = w \pm r \delta_h \vec{n}_h(w)
\]

and \(\delta_h = \delta_{h_{\text{max}}}\) a step-width parameter to perform a linear least-squares regression through these five points on each separate side, cf. Figure 4.3. The slope of these regressions is taken to approximate the gradient in normal direction at \(w\) and the resulting normal velocity is given as

\[
(\tilde{V}^n \cdot \vec{n}_h)(w) = \frac{2}{L} \left[ \frac{2u^n_{1,h}(w)}{\delta_h} + u^n_{1,h} \left( w^\pm \frac{\delta_h}{2} \right) - u^n_{1,h} \left( w^\mp \frac{\delta_h}{2} \right) - 2u^n_{1,h} \left( w^\pm \frac{\delta_h}{2} \right) \right]
\]

By multiplying with \(\vec{n}_h\) a velocity field \(\tilde{V}^n(w) = (\tilde{V}^n \cdot \vec{n}_h)(w) \cdot \vec{n}_h(w)\) can be obtained from this expression. At the end, we set the velocity field at \(v\) to the average of all contributions from the projections \(w \in \{w_j : j \in \mathcal{N}\}\).

**Remark:** In numerical studies, we sometimes observe stability issues for the DGE method in situations with “barely intersected” simplices, i.e. for situations where we have either a very large or a very small volume ratio \(|S \cap \Omega_{2,h}|/|S \cap \Omega_{1,h}|\). To overcome these problems, we neglect any simplex \(S\) and the corresponding discrete interface \(\Gamma_{h,S}\) where the ratio of one subvolume to the complete simplex volume is below a small tolerance.

4.7.2. **Extension phase.** To propagate the initialized velocity values into the far field, we can make extensive use of the already presented FMM algorithm. Since the algorithm propagates (distance) values into the far field in the reinitialization, we can basically step through the same procedure and propagate velocity values additionally to the distance values. Concretely, we conduct the following 2 steps:

(1) Calculate distance values for vertices \(v \in \mathcal{V}(S), S \in \mathcal{S}_h\) so that initialized distance and velocity field values are given after this step.

---

\(^9\)Note that this gradient is a piecewise constant, vectorial XFEM function.
Remark: Note that though (2) coincides highly with what we do in the reinitialization process, there is actually a minor difference that needs to be considered: There can be multiple minimizing simplices \( S_{\text{min}} \) the current \( \hat{S} \) we need to set the velocity to be the average of all contributions, i.e. since the distance value coincide for all these minimizing simplices, the velocity values can differ. In this case, we need to set the velocity to be the average of all contributions, i.e.

\[
\vec{V}^n(v^*) = \frac{1}{|S_{\text{min}}|} \sum_{S \in S_{\text{min}}} \vec{V}^n(P_W(S)(v^*))
\]

where \( S_{\text{min}} \) captures all simplices that minimize the tentative distance function. Also note that there is no such thing as a tentative velocity function in this procedure. The vertex that is processed next is still defined by the vertex that minimizes the tentative distance function \( \hat{d} \) on the current active set \( \mathcal{V}_a \). In other words, the velocity field values is information that is propagated alongside but does not interfere with the FMM procedure itself.

5. Implementation aspects

The considered problem is solved with \texttt{miXFEM} and a level set toolbox, both developed within our work group for the \texttt{FEniCS} framework.

5.1. The \texttt{FEniCS} project. \texttt{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 [24]. It consists of a collection of core components such as

(1) the Unified Form Language \texttt{UFL} [3], which is a domain-specific language to specify finite element discretizations of differential equations using variational formulations close to the mathematical notation,

(2) the \texttt{FEniCS} Form Compiler \texttt{FFC} [23,30], which analyzes given \texttt{UFL} code and, in combination with \texttt{Instant} and \texttt{FIAT} [22], generates \texttt{C++} code for arbitrary finite elements on simplices based on the variational forms specified in the \texttt{UFL} file,

(3) \texttt{DOLFIN} [25], the main problem solving environment and user interface whose functionality integrates the other \texttt{FEniCS} components and handles communication with external libraries or toolboxes such as \texttt{miXFEM}.

5.2. \texttt{miXFEM} - an XFEM toolbox for \texttt{FEniCS}. \texttt{FEniCS} provides a lot of useful classes, structures and other utilities for solving PDE based problems with FEM. However, the \texttt{FEniCS} framework has to be extended by new modules for considering non-standard problems. In order to solve problems with arbitrary time-dependent discontinuities that may evolve and also intersect each other, we developed an XFEM toolbox [19] partly based on the \texttt{PUM} toolbox [27,28].

\texttt{miXFEM} adds features to the domain specific language \texttt{UFL} in order to define enriched function spaces. Additionally a new syntax for integrals on arbitrary interfaces is introduced. The \texttt{UFL} file is compiled using an extended \texttt{FeniCS} Form Compiler, which understands and interprets the new features, to generate the corresponding \texttt{C++} code. Based on this code, the problems can be solved numerically using an extension of the \texttt{DOLFIN} library, implemented in \texttt{C++}. While some key features are presented in [19], a detailed technical description of all features will be given in an upcoming publication.

5.3. Level set toolbox. Another extension to \texttt{FEniCS} used to solve problem (2.5)-(2.7) is a level set toolbox, which is used to compute the evolution of the level set function resp. the discontinuity. The toolbox consists of discretized weak formulations for different time stepping schemes formulated in \texttt{UFL} and compiled with the \texttt{FFC} which are used by a \texttt{C++} library. This library provides an implementation of the presented Fast Marching Method and the volume correction approach which can be used for various problems. A detailed description is given in [18]. Recently, the narrow band approach has been included to improve the efficiency of the implemented methods.

5.4. Problem related aspects for solving the Stefan problem in level set formulation.
Decoupling of thermal problem and level set problem. Since the problems (2.5) and (2.7) are coupled by the Stefan condition (2.6), a numerical decoupling strategy is needed to solve the complete problem. In this article, we use the simple approach to solve the problems in succession: Given all data for $t_n$, we firstly solve the level set problem to obtain $\varphi_{n+1}^h$ and the new interface $\Gamma_{n+1}^h$ based on the old data of the thermal problem. The computed interface $\Gamma_{n+1}^h$ is then used to construct new XFEM function spaces and to solve the thermal problem for $u_{n+1}$. Using the approaches presented in Section 4.7, the interface’s normal velocity $\vec{V}_{n+1} \cdot \vec{n}$ and the velocity field $\vec{V}$ are computed. In our approach, we use the implicit Euler scheme for time discretization in both subproblems, however, we still may need intermediate time steps for solving the level set problem due to the CFL conditions, as explained in the next paragraph.

Time stepping. As mentioned before, the discretization of the time interval $[t_0, t_f]$ for the thermal problem (2.5) in Section 3.1 and for the transport problem (2.7) in Section 4.3 do not necessarily have to coincide. The reason for this can be that, firstly, we may use different time stepping methods for both subproblems, e.g. the Crank-Nicolson method for the level set problem and the implicit Euler method for the thermal problem, and, secondly, we may need smaller time steps due to the arising CFL conditions when using the narrow band approach. Hence, we have to synchronize the time step sizes for the subproblems in order to compute a numerical solution of the coupled problem.

For this purpose, we use the time step size $\Delta t$ of the thermal problem as major time step size and the values $t_n = t_0 + n \Delta t$ with $n \in \{0, \ldots, N \}$, cf. Section 3.1, as so-called synchronization points. Based on this, we adjust the time step size $\Delta t_\varphi$ for the level set problem, if necessary, so that the discretization method with time step size $\Delta t$ and the CFL condition(s) with time step size $\Delta t_{CFL}$ are respected. This means, we have may have to introduce potentially non-equidistant intermediate time steps $t_{n,i}$ in order to reach the (next) synchronization point(s). The described procedure is illustrated in Fig. 5.1. As we will see in Example 2, this can influence the solution process and the convergence behavior.

Now, the complete numerical approach for solving the Stefan problem in level set formulation is shown in Algorithm 1.

6. Results

The presented numerical approach consists of a level set solver including maintaining techniques, an XFEM framework, and some methods for tackling the Stefan problem. Since we already verified the level set toolbox, cf. [18], and the XFEM framework for time-dependent problems, see [19], we now focus on the overall results of the complete approach. In particular, a numerical convergence study is performed with respect to different time step sizes $\Delta t$ and varied maximum cell diameters $h$. Similar to our previous work with prescribed interface [19], two academical examples with known analytical solutions $u$ are examined and 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 (semi) $H^1$-error

$$\|\nabla u - \nabla u_h\|_{L^2(L^2)} := \sqrt{\int_{t_0}^{t_f} \|\nabla u(\cdot, t) - \nabla u_h(\cdot, t)\|_{L^2(\Omega)}^2 \, dt}.$$
Algorithm 1 Solver for the two-phase Stefan problem in level set formulation

Input: $\Omega, \Gamma_D, \Gamma_N, S_h, \varphi_0, u_0, u_T, f, \kappa, L, u_D, g_N, \Delta t, \beta_D, \beta_I, \gamma$.
Output: $u^n_h, \Gamma^n_h, \varphi^n_h, \vec{V}_n^h$ for $n = 0, \ldots, N_T$.

Initialization:

- Obtain $\varphi_0^h$ by reinitializing the given level set function $\varphi_0$.
- Construct the function space $V_1^{h,u_D}(t_0)$ with $\varphi_0^h$.
- if the narrow band method is applied then
  - Initialize $\Omega_{INB}$ and $\Omega_{ONB}$ with $\varphi_0^h$.
end if

- Interpolate $u_0$ onto $V_1^{h,u_D}$ to obtain the initial temperature $u_0^h$.

Time stepping:

for $n = 0, \ldots, N_T - 1$ do

- Derive the velocity field $\vec{V}$ from $u_n^h$ with one of the methods that are presented in Section 4.7.
- Assign $t_\varphi = n\Delta t$ (current simulation time) and $\varphi^{t_\varphi}_n$.
- while $t_\varphi < (n+1)\Delta t$ do
  - Calculate time step for level set propagation $\Delta t_\varphi = \min\{\Delta t_{CFL}, (n+1)\Delta t - t_\varphi\}$.
  - Propagate level set function to obtain an updated level set function $\varphi^{n+1}_n$.
  - if reinitialization is necessary then
    - Replace $\varphi^{n+1}_n$ by its reinitialized version.
    - if narrow band method is applied then
      - Update inner and outer narrow band regions $\Omega_{INB}, \Omega_{ONB}$ with the reinitialized function $\varphi^{n+1}_n$.
    end if
  end if
- Set $t_\varphi = t_\varphi + \Delta t_\varphi$.
end while

- Set $\varphi^{n+1}_n = \varphi^{n+1}_n$.
- Construct the new discrete interface $\Gamma^{n+1}_h$ from $\varphi^{n+1}_h$.
- Construct the new XFEM space $V_1^{h,u_D}(t_{n+1})$ with $\Gamma^{n+1}_h$.
- Solve for the new temperature approximation $u^{n+1}_h$.
end for

The order of convergence for each error is as usual determined for varied $\Delta t$ and $h$ and 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. [13], 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. In regards to the time stepping scheme, we use the implicit Euler method in both examples for all subproblems. While we present examples for 2D situations only, please note that the same methods can be used in a straightforward way for 3D problems as well.

6.1. Example 1: Straight interface. Choosing $\Omega := (0,1)^2$ with $\Gamma_D := \{(x,y) \in \partial \Omega \mid y = 0 \vee y = 1\}$ and $\Gamma_N := \partial \Omega \backslash \Gamma_D$ for the geometry and $\kappa_1 := 1, \kappa_2 := 2, L := 2, u_T := 0$ for the material parameters, an analytical
solution to problem (2.5)-(2.7) for \([t_0,t_f] := [0, 5 \cdot 2^{-4}]\) is given by

\[
u(x,t) = \begin{cases} \cos\left(\frac{\pi x}{2}\right) \sin\left(\frac{\pi \varphi(x,t)}{y - \varphi(x,t)}\right) + \varphi(x,t) & \text{on } \Omega_1(t) \\ \cos\left(\frac{\pi x}{2}\right) \sin\left(\frac{\pi \varphi(x,t)}{y - \varphi(x,t)}\right) + \frac{1}{2} \varphi(x,t) + e^y + \frac{11}{11t+5} & \text{on } \Omega_2(t) \end{cases},
\]

with \(x = (x,y) \in \Omega\). The corresponding interface \(\Gamma(t)\) to this solution is a straight horizontal line moving downwards which is characterized by the zero level set of the level set function

\[
\varphi(x,t) := y - \ln \left(\frac{11}{11t+5}\right).
\]

The source term \(f\) for the right-hand-side, the boundary functions \(u_D\) and \(g_N\) and the initial conditions have to be chosen with respect to the specified analytical solution and can be easily computed. The analytical solution \(u\) is shown in Figure 6.1 at different time instants and the results of the convergence analysis are shown in Figure 6.2.

In the latter, one can see that the problem is dominated by the spatial error reaching the optimal convergence order 2, cf. Fig 6.2(a). Moreover, an interesting effect can be seen in Fig. 6.2(b) (left) for the coarsest mesh size is shown in Figure 6.1 at different time instants and the results of the convergence analysis are shown in Figure 6.2.

Example 2: Circular interface. On \(\Omega := (-1,1)^2\) with \(\Gamma_N := \partial \Omega\) consider for \([t_0,t_f] := [0,\frac{3}{4}]\) the function

\[
u(x,t) = \begin{cases} A \left(\|x\|^2 - R(t)^2\right) & \text{on } \Omega_1(t) \\ A \left(\|x\|^2 - R(t)^2\right) - R(t) \left(\|x\| - R(t)\right) + \frac{R(t)(R(t)+\frac{1}{2}\sin(\pi t))}{2\|x\| - R(t)^2} & \text{on } \Omega_2(t) \end{cases},
\]

with \(A > \frac{R(t)}{2\pi R(t)}\) to ensure \(u < 0\) on \(\Omega_1(t)\) and \(u > 0\) on \(\Omega_2(t)\), and \(R(t) := R_0 + \frac{1}{2}\sin(\pi t)\). Choosing \(R_0 = 0.3\), \(A = 4.1\) as well as \(\kappa_1 := \kappa_2 := 1\), \(L := 1\) and \(u_F := 0\) for the parameters, this is a solution to problem (2.5)-(2.7) where the interface \(\Gamma(t)\) corresponds to a circle with radius \(R(t)\), which is centered at the origin and is expanding till \(t = 0.5\) and then shrinking again. \(\Gamma(t)\) can be characterized by the level set function

\[
\varphi(x,t) := R^2(t) - \|x\|^2.
\]

As before, the right-hand-side term \(f\), the Neumann boundary function \(g_N\) and the initial conditions have to be chosen with respect to the specified analytical solution and can be derived easily. The analytical solution \(u\) is shown in Figure 6.3 at different time instants.
\[ \Gamma^\prime(t) \]\n
(a) \( \| u - u_h \|_{L^\infty(L^2)} \) (left) and \( \| \nabla u - \nabla u_h \|_{L^2(L^2)} \) (right) over \( h \) for different time step sizes

(b) \( \| u - u_h \|_{L^\infty(L^2)} \) (left) and \( \| \nabla u - \nabla u_h \|_{L^2(L^2)} \) (right) over \( \Delta t \) for different \( h \)

**Figure 6.2.** Convergence tests for example 1 including approximated orders of convergences.

The convergence behavior is visualized in Fig. 6.4. Therein one can see that this problem is also highly dominated by the spacial error and qualitatively similar to Example 1. In regards to the convergence order, however, we can only archive suboptimal results. One reason for this is that although \( \varphi \) can be exactly approximated by \( \varphi_h \) if using quadratic basis functions, the linear polygonal approximation \( \Gamma_h(t) \) of the circular interface \( \Gamma(t) \) introduces an additional error propagating through the solution process, which is different compared to the
situation in Example 1. Additionally, the same problem as described in the previous section arises, i.e. for small
time steps we have to reinitialize more often introducing thereby a secondary approximation error. However,
an even more significant impact on the convergence behavior has the narrow band approach, respectively the
CFL condition (4.12): In addition to using the reinitialization procedure after each time step, the CFL condition
may also request intermediate time steps, see Section 5.4, especially for big ∆t. In such a situation, the narrow
band has to be updated and the update process relies on the signed distance property making an additional
reinitialization step mandatory for every intermediate time step as well as very regular one. As a consequence,
the introduced errors also propagate through the solution process and, for some situations, prevent a convergent
behavior.

Consider exemplary the graph of the error ∥u − u_h∥_{L^\infty(L^2)} for ∆t = 2^{-4} in Fig. 6.4(a)(left): As the mesh
size decreases, (4.12) requests more intermediate time steps introducing thereby more reinitialization procedures
which in turn modify the interface position and, hence, the solution. Due to the complex character of the example,
small deviations of the interface position have a high impact on the solution of the thermal problem so that,
finally, we end up with bigger errors on finer meshes then on coarser meshes. Alternatively, it can be seen in Fig.
6.4(b) (left) that for a fixed spacial discretization at some point the error ∥u − u_h∥_{L^\infty(L^2)} does not decrease any
more but sightly increase as a results of the more and more reinitialization procedures.

7. Summary

In this article, we present a numerical approach to solve the Stefan problem using the level set method and
XFEM. The problem is decoupled by solving the level set problem and the thermal problem in succession. For
this purpose, the thermal problem is discretized using Nitsche’s method for internal Dirichlet boundaries and a
(local) strong enrichment based on a Heaviside function. The evolution of the interface is described by a level set
problem restricted to a narrow band region. In addition to maintaining techniques for the shape of the level set
function, two approaches are presented to compute the corresponding propagation velocity field using the Stefan
condition.

All described methods are implemented using our toolbox miXFEM for the FEniCS framework and are used for
solving two academical examples of different complexities with known analytical solutions. In both examples,
good results can be achieved with optimal rates of convergence (Example 1) or suboptimal rates of convergence
(Example 2).

Due to the modular implementation, maintaining methods for the level set problem, more precisely, the
reinitialization, which is mandatory for the narrow band approach, could be identified as “the weakness of the
approach”. However, this is just a technical issue arising within the convergence studies where very different
mesh sizes and time step sizes are considered.

Last but not least, we want to point out that due to the general approach, this numerical method is not limited
to the Stefan problem but works for all problems with arbitrary time-dependent discontinuities, e.g. multi-phase
flow.
The authors gratefully acknowledge the financial support by the DFG (German Research Foundation) for the subproject A3 within the Collaborative Research Center SFB 747 “Mikrokontaktflächen - Prozesse, Charakterisierung, Optimierung”.

Figure 6.4. Convergence tests for example 2 including approximated orders of convergences.
The discrete formulation of (3.6) is derived as follows: Multiply (3.6) with a test function \( v \in V_h \) and integrate over the domain \( \Omega_{1,h} \cup \Omega_{2,h} \):

\[
\int_{\Omega_{1,h} \cup \Omega_{2,h}} \xi u_h v_h \, dx - \int_{\Omega_{1,h} \cup \Omega_{2,h}} \nabla \cdot (\kappa \nabla u_h) v_h \, dx = \int_{\Omega_{1,h} \cup \Omega_{2,h}} \tilde{f} v_h \, dx.
\]

Integration by parts to

\[
\int_{\Omega_{1,h} \cup \Omega_{2,h}} \xi u_h v_h \, dx + \int_{\Omega_{1,h} \cup \Omega_{2,h}} \kappa \nabla u_h \nabla v_h \, dx - \int_{\partial(\Omega_{1,h} \cup \Omega_{2,h})} \kappa \nabla u_h \cdot \tilde{n}_h v_h \, dc
\]

\[
= \int_{\Omega_{1,h} \cup \Omega_{2,h}} \xi u_h v_h \, dx + \int_{\Omega_{1,h} \cup \Omega_{2,h}} \kappa \nabla u_h \nabla v_h \, dx - \int_{\Gamma_h} \kappa \nabla u_h \cdot \tilde{n}_h v_h \, dc
\]

\[
- \int_{\Gamma_N} \kappa \nabla u_h \cdot \tilde{n}_h v_h \, dc - \int_{\Gamma_h} \kappa_1 \nabla u_{1,h} \cdot \tilde{n}_h v_{1,h} \, dc + \int_{\Gamma_h} \kappa_2 \nabla u_{2,h} \cdot \tilde{n}_h v_{2,h} \, dc
\]

\[
= \int_{\Omega_{1,h} \cup \Omega_{2,h}} \tilde{f} v_h \, dx.
\]

Now, we add the “artificial” terms

\[
0 = \pm \int_{\Gamma_h} \kappa_{1,2} \nabla v_{1,2,\gamma} \cdot \tilde{n}_h u_{1,2,\gamma} \, dc \pm \int_{\Gamma_h} \kappa_{1,2} \nabla v_{1,2,\gamma} \cdot \tilde{n}_h u_{1,2,\gamma} \, dc
\]

and

\[
0 = \int_{\Gamma_h} \lambda_{1,2} u_{1,2,\gamma} v_{1,2,\gamma} \, dc - \int_{\Gamma_h} \lambda_{1,2} u_{1,2,\gamma} v_{1,2,\gamma} \, dc
\]

so the equation reads

\[
\int_{\Omega_{1,h} \cup \Omega_{2,h}} \xi u_h v_h \, dx + \int_{\Omega_{1,h} \cup \Omega_{2,h}} \kappa \nabla u_h \nabla v_h \, dx
\]

\[
- \int_{\Gamma_h} \kappa_1 \nabla u_{1,h} \cdot \tilde{n}_h v_{1,h} \, dc - \int_{\Gamma_h} \kappa_1 \nabla v_{1,h} \cdot \tilde{n}_h u_{1,h} \, dc + \int_{\Gamma_h} \kappa_1 \nabla u_{1,h} \cdot \tilde{n}_h u_{1,h} \, dc
\]

\[
+ \int_{\Gamma_h} \kappa_2 \nabla v_{1,h} \cdot \tilde{n}_h u_{1,h} \, dc + \int_{\Gamma_h} \kappa_2 \nabla v_{1,h} \cdot \tilde{n}_h u_{2,h} \, dc - \int_{\Gamma_h} \kappa_2 \nabla v_{2,h} \cdot \tilde{n}_h u_{2,h} \, dc
\]

\[
+ \int_{\Gamma_h} \lambda_1 u_{1,h} v_{1,h} \, dc - \int_{\Gamma_h} \lambda_1 u_{1,h} v_{1,h} \, dc + \int_{\Gamma_h} \lambda_2 u_{2,h} v_{2,h} \, dc - \int_{\Gamma_h} \lambda_2 u_{2,h} v_{2,h} \, dc
\]

\[
= \int_{\Omega_{1,h} \cup \Omega_{2,h}} \tilde{f} v_h \, dx + \int_{\Gamma_N} g_N v_h \, dc
\]

Now, we put one of each artificial term onto the right-hand-side and make use of the condition \( u_{1,2,\gamma} = u_{\gamma} \) on \( \Gamma_h \) and define \( \lambda_1 = \lambda_2 \)

\[
\int_{\Omega_{1,h} \cup \Omega_{2,h}} \xi u_h v_h \, dx + \int_{\Omega_{1,h} \cup \Omega_{2,h}} \kappa \nabla u_h \nabla v_h \, dx
\]

\[
- \int_{\Gamma_h} \kappa_1 \nabla u_{1,h} \cdot \tilde{n}_h v_{1,h} \, dc - \int_{\Gamma_h} \kappa_1 \nabla v_{1,h} \cdot \tilde{n}_h u_{1,h} \, dc + \int_{\Gamma_h} \lambda u_{1,h} v_{1,h} \, dc
\]

\[
= \int_{\Omega_{1,h} \cup \Omega_{2,h}} \tilde{f} v_h \, dx + \int_{\Gamma_N} g_N v_h \, dc
\]

\[
= L_1(v_h)
\]

\[
- \int_{\Gamma_h} \kappa_2 \nabla v_{1,h} \cdot \tilde{n}_h u_{1,h} \, dc + \int_{\Gamma_h} \lambda u_{1,h} v_{1,h} \, dc + \int_{\Gamma_h} \kappa_2 \nabla v_{1,h} \cdot \tilde{n}_h u_{2,h} \, dc + \int_{\Gamma_h} \lambda u_{2,h} v_{2,h} \, dc
\]

\[
= L_2(v_h)
\]

\[
= L_1(v_h)
\]

\[
= L_2(v_h)
\]
REFERENCES


