

IMPLEMENTATION OF AN X-FEM SOLVER FOR THE CLASSICAL TWO-PHASE STEFAN PROBLEM

MARTIN K. BERNAUER AND ROLAND HERZOG

ABSTRACT. The classical two-phase Stefan problem in level set formulation is considered. The implementation of a solver on triangular grids is described. Extended finite elements (X-FEM) in space and an implicit Euler method in time are used to approximate the temperature. For the level set equation, a discontinuous Galerkin (DG) and a strong stability preserving (SSP) Runge-Kutta scheme are employed. Polynomial spaces of quadratic order are used. A numerical example with a change of topology is provided, and the order of convergence is studied on the Frank sphere example.

1 Introduction

The Stefan problem was posed more than a hundred years ago motivated by the formation of ice in the polar sea [46]. Extensions of the original problem can be used for the modeling of continuous casting processes, dendritic solidification and various other engineering problems. As analytic solutions can be found only in special cases, e.g., for infinite domains, the need for flexible numerical tools for the simulation of Stefan-type problems is evident.

A classical approach, see [3, 6, 8, 9, 47], is to update the underlying mesh to conform to the interface which separates the phases of the Stefan problem. Finite elements or finite differences are then used in each phase to update the temperature distribution. This moving mesh method requires the sharp resolution of the interface. As an alternative, enthalpy methods avoid the tracking of the interface by introducing an enthalpy variable that accounts for the latent heat, see [11, 17, 32] and the review article by Voller et al. [48].

Solving Stefan-type problems in level set formulation can at least be traced back to the paper by Sethian and Strain [44] who transform the energy balance equations

Date: September 2, 2011.

2000 Mathematics Subject Classification. 80A22, 35R35, 65M60.

Key words and phrases. two-phase Stefan problem, solidification, moving boundary problem, level set method, discontinuous Galerkin approximation, extended finite element method.

into boundary integral equations on the moving interface. Chen et al. [13] develop a finite difference-based approach to solving the two-phase Stefan problem in two spatial dimensions. Fried [21] extends the finite element method of Schmidt [43] who uses a parameterization of the moving interface to a level set framework. Gibou et al. [27] extend the approach of [13] to a higher-order approximation of the interface and to three spatial dimensions. Fedkiw and Gibou design a fourth order accurate finite difference scheme for the Laplace equation that is applicable to the Stefan problem, as demonstrated by the authors in [26]. Gibou et al. [12] discuss an adaptive method with supralinear convergence properties that builds upon previous work of the authors.

Probably the most recent approach to solving the Stefan problem in level set formulation is the extended finite element (X-FEM) method which avoids moving the mesh and remeshing completely. It was originally introduced for problems in computational mechanics by Dolbow [19] and was later adapted to solidification problems, see [38], [16] and [49] to name but a few references. The common idea of X-FEM approaches is to explicitly track the interface (e.g. by using the level set method) and to construct enriched finite element spaces depending on the interface position while keeping the underlying mesh fixed. Additional basis functions are designed to capture the phase change across the interface, i.e., they are continuous but their first derivatives have a jump.

We use extended finite elements for the approximation of the temperature field and the level set technique for tracking the phase-change interface. Our implementation differs from previous work in the following ways. In contrast to [16] and [49] we use a discontinuous Galerkin (DG) scheme for the spatial approximation of the level set equation which overcomes the need of using GLS stabilization and shock capturing terms. We also consider the level set equation in advective form. Our starting point is a recent DG scheme by Cheng and Shu [15] for Hamilton-Jacobi equations on rectangular grids. We extend their approach to triangular meshes which offer more flexibility for complex geometries as in the example in Section 6.3. While most other approaches used explicit constraint equations or the LATIN method [38] to enforce the isothermal interface condition, we use a weak form of this condition inspired by [33] that is incorporated into the finite element formulation by a penalty method. Another feature of our implementation is the usage of quadratic ansatz spaces for both the temperature and the level set equations, which allows for a more accurate resolution of the curvature of the interface or related terms. An

alternative higher-order extended finite element method based on a non-polygonal interface approximation and isoparametric integration is presented by Cheng and Fries [14]. A strong stability preserving Runge-Kutta scheme is employed for the solution of the level set equation, while the implicit (backwards) Euler method is used for the temperature equation. Both schemes are synchronized to ensure that the temperature and the interface position are available at the same instances of time. Finally, we examine the order of convergence. Our experiments indicate linear convergence w.r.t. the L^2 and L^∞ norms for both the temperature and the level set fields under simultaneous spatial and temporal refinement. In principle, our approach applies to problems in two and three space dimensions. While we focus on the description of the two-dimensional situation, we point out some issues related to the extension to three dimensions where they are relevant.

This paper is organized as follows. The governing equations of the Stefan problem and the level set formulation are presented in Section 2. Section 3 is devoted to the extended finite element approximation and the time discretization of the temperature variable. The discontinuous Galerkin finite element method and a corresponding time stepping scheme for solving the level set equation are discussed in Section 4. Some advanced topics concerning the implementation are reviewed in Section 5. Finally, we report on several computational experiments in Section 6.

2 Model Equations

The solidification of a fluid material in a fixed domain $D \subset \mathbb{R}^2$ (the hold-all) is modeled by the two-phase Stefan problem [31]

$$\rho c_S y_t - k_S \Delta y = f \quad \text{in } \Omega_S(t) \quad (2.1a)$$

$$\rho c_F y_t - k_F \Delta y = f \quad \text{in } \Omega_F(t) \quad (2.1b)$$

$$y(x, 0) = y_0(x) \quad \text{in } \Omega_S(0) \cup \Omega_F(0) \quad (2.1c)$$

$$y(x, t) = y_M \quad \text{on } \Gamma_I(t) \quad (2.1d)$$

$$k \frac{\partial y}{\partial n} = g \quad \text{on } \Gamma_N = \partial D \quad (2.1e)$$

on a certain time horizon $[0, T]$. The moving interface $\Gamma_I(t)$ divides D into $\Omega_S(t)$ occupied by the solid phase and $\Omega_F(t)$ occupied by the fluid phase, see Figure 2.1.

We assume the density ρ to be constant and equal in both phases so that we can ignore mass transport effects. The specific heat at constant pressure $c_{S/F}$ and the coefficient of heat conduction $k_{S/F}$ are in general different in the two phases.

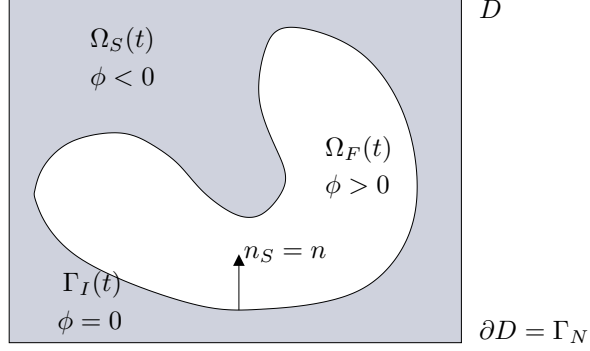


FIGURE 2.1. Setup of geometry for a two-phase Stefan problem.

$y_M \in \mathbb{R}$ is a reference temperature at which solidification takes place. In (2.1e), k denotes either k_F or k_S , depending on which of the phases is at the boundary ∂D . Depending on the application, (2.1e) might be replaced by a Newtonian cooling condition or a mixed boundary condition, see Section 6.2 for such a situation.

The Stefan condition

$$\rho L V \cdot n = k_S \frac{\partial y_S}{\partial n} - k_F \frac{\partial y_F}{\partial n} = [k \nabla y]_F^S \cdot n \quad \text{on } \Gamma_I(t) \quad (2.2)$$

states that the normal velocity $V \cdot n$ of $\Gamma_I(t)$ is proportional to the jump of the normal derivative of the temperature. L denotes the latent heat that is released upon solidification and the subscripts refer to the two phases.

Let $\phi(x, t)$ be the signed distance function to $\Gamma_I(t)$, where we use the sign convention

$$\Omega_S(t) = D \cap \{\phi(\cdot, t) < 0\}, \quad \Omega_F(t) = D \cap \{\phi(\cdot, t) > 0\}.$$

Then the outward unit normal to $\Omega_S(t)$ is given by $\frac{\nabla \phi}{|\nabla \phi|}$. As proposed by various authors, see [13], [16] and [49], we use level set techniques for the evolution of ϕ . In contrast to parametrizations of $\Gamma_I(t)$ or representations of $\Gamma_I(t)$ as the graph of a function, level set techniques handle topological changes of the phases or closed interfaces naturally.

Unlike the approaches mentioned above, we write the level set equation in advective form:

$$\phi_t + V \cdot \nabla \phi = 0 \quad \text{in } \mathbb{R}^2 \quad (2.3a)$$

$$\phi(x, 0) = \phi_0(x) \quad \text{in } \mathbb{R}^2 \quad (2.3b)$$

where the initial condition (2.3b) is defined using the signed distance function ϕ_0 to the initial position of the interface. This linear Hamilton-Jacobi equation allows for

the application of the discontinuous Galerkin scheme from [15]. This is in contrast to the nonlinear equation

$$\phi_t + V \cdot n |\nabla \phi| = 0 \quad \text{in } \mathbb{R}^2$$

which has a non-convex Hamiltonian and is usually used for evolving ϕ . Note that V in (2.3a) is a vector-valued extension of $V \cdot n$ in (2.2) from $\Gamma_I(t)$ to all of D , see Section 5.3 for more details on this point.

3 Extended Finite Element Approximation

In this section, we essentially follow the work by Chessa et al. [16]. We assume that we are given a triangulation \mathcal{T}_h of our domain D . The extended finite element approximation of the temperature field is

$$y_h(x, t) = \sum_{i=1}^N v_i(x) y_i(t) + \sum_{j=1}^{N_e(t)} \psi_j(x, t) a_j(t),$$

where $v_i(x)$ are the usual Lagrangian finite element shape functions of order $k \geq 1$ on the triangulation \mathcal{T}_h and $y_i(t)$ are the corresponding time-dependent degrees of freedom, located in the points x_i . The additional degrees of freedom $a_j(t)$ are added locally around $\Gamma_I(t)$ to the $(k+1)(k+2)/2$ nodes (in 2D) of those elements intersected by $\Gamma_I(t)$, see Figure 3.1. The corresponding enrichment functions are defined as

$$\psi_j(x, t) = v_{i(j,t)}(x) (|\phi(x, t)| - |\phi(x_{i(j,t)}, t)|), \quad j = 1, \dots, N_e(t),$$

where $\phi(x_{i(j,t)}, t)$ denote the nodal values of the level set function and $i(j, t)$ maps the indices of the enriched nodes at time t to the fixed degrees of freedom. The enrichment kernel $|\phi(x, t)| - |\phi(x_{i(j,t)}, t)|$ is chosen to model the phase change across $\Gamma_I(t)$, i.e., it is continuous with a discontinuous first derivative normal to the moving boundary. This kernel is multiplied with the standard shape function $v_{i(j,t)}(x)$ to preserve the sparsity of the stiffness matrix.

As the enrichment functions $\psi_j(x, t)$ depend on the location of the interface, the underlying partial differential equations must in general be discretized in time before the enriched finite element spaces can be defined [24].

We rewrite the heat equations (2.1a)–(2.1b) in the two-phase Stefan problem as

$$\begin{aligned} y_t &= \frac{k_S}{\rho c_S} \Delta y + \frac{1}{\rho c_S} f && \text{in } \Omega_S(t), \\ y_t &= \frac{k_F}{\rho c_F} \Delta y + \frac{1}{\rho c_F} f && \text{in } \Omega_F(t). \end{aligned}$$

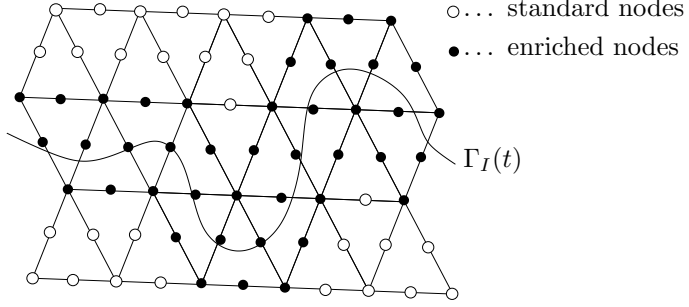


FIGURE 3.1. Illustration of local enrichment for quadratic finite elements ($K = 2$).

Formally, we interpret this as an ordinary differential equation (ODE)

$$y_t = L(t, y)$$

with the operator L defined by

$$L(t, y) := \frac{k(x, t)}{\rho c(x, t)} \Delta y(x, t) + \frac{1}{\rho c(x, t)} f(x, t).$$

Note that in this setting,

$$k(x, t) := \begin{cases} k_S & \text{if } \phi(x, t) < 0 \\ k_F & \text{if } \phi(x, t) > 0 \end{cases} \quad \text{and} \quad c(x, t) := \begin{cases} c_S & \text{if } \phi(x, t) < 0 \\ c_F & \text{if } \phi(x, t) > 0 \end{cases}$$

are interpreted as functions of space and time. In the following, the superscripts $j-1$ and j refer to instances of time, e.g., $y^j(x) = y(x, t^j)$. The application of the implicit Euler method with time step $\Delta t_y^j = t^j - t^{j-1}$ ($j = 1, \dots, N_T$) to this ODE yields the equation

$$y^j = y^{j-1} + \Delta t_y^j L(t^j, y^j)$$

which has to be solved for y^j in each time step. A multiplication of the discretized ODE with a test function δy and a subsequent integration over D yields the weak form

$$\frac{1}{\Delta t_y^j} \int_D \delta y y^j dx = \frac{1}{\Delta t_y^j} \int_D \delta y y^{j-1} dx + \int_D \frac{k^j}{\rho c^j} \Delta y^j \delta y dx + \int_D \frac{f^j}{\rho c^j} \delta y dx.$$

The Laplace operator in this weak form is integrated by parts and afterwards the test function δy is evaluated at time level j as suggested in [24], yielding the time-discrete energy balance equation in weak form:

$$\begin{aligned} & \frac{1}{\Delta t_y^j} \int_D \delta y^j y^j dx + \int_D \frac{k^j}{\rho c^j} \nabla y^j \cdot \nabla \delta y^j dx \\ &= \frac{1}{\Delta t_y^j} \int_D \delta y^j y^{j-1} dx + \int_{\partial D} \delta y^j \frac{g^j}{\rho c^j} ds + \int_D \frac{f^j}{\rho c^j} \delta y^j dx. \end{aligned}$$

Replacing δy and y by their discrete counterparts, we obtain the fully discrete system

$$\left(\frac{1}{\Delta t_y^j} \mathbf{M}^j + \mathbf{K}^j \right) Y^j = \frac{1}{\Delta t_y^j} \mathbf{M}_{j-1}^j Y^{j-1} + \mathbf{G}^j + \mathbf{F}^j \quad (3.1)$$

which has to be solved for the coordinate vector

$$Y^j = [y_1^j, \dots, y_N^j, a_1^j, \dots, a_{N_e(t^j)}^j]^\top$$

with respect to the finite element basis

$$B^j = \{v_1(x), v_2(x), \dots, v_N(x), \psi_1(x, t^j), \dots, \psi_{N_e(t^j)}(x, t^j)\}. \quad (3.2)$$

The matrices and vectors in (3.1) are defined as

$$\left. \begin{aligned} \mathbf{M}^j &= \left(\int_D \varphi_l \varphi_k dx \right)_{k,l} \\ \mathbf{M}_{j-1}^j &= \left(\int_D \varphi_l \varphi_k^{j-1} dx \right)_{k,l} \\ \mathbf{K}^j &= \left(\int_D (\nabla \varphi_l)^\top \frac{k}{\rho c} \nabla \varphi_k dx \right)_{k,l} \\ \mathbf{G}^j &= \left(\int_{\partial D} \varphi_l \frac{g}{\rho c} ds \right)_l \\ \mathbf{F}^j &= \left(\int_D \varphi_l \frac{f}{\rho c} dx \right)_l \end{aligned} \right\} \quad (3.3)$$

where $\varphi_k, \varphi_l \in B^j$, $\varphi_k^{j-1} \in B^{j-1}$, and all other quantities without a superscript are taken at time level j .

Remark 3.1 (Higher-Order Time Discretizations). *The implementation of the Crank-Nicolson scheme only requires the assembly of one additional pseudo stiffness matrix \mathbf{K}_{j-1}^j similar to \mathbf{M}_{j-1}^j , and two vectors \mathbf{G}^{j-1} and \mathbf{F}^{j-1} that represent the contributions of the Neumann boundary condition and the heat sources, respectively. Our numerical experiments on the Frank sphere example (see Section 6.1) however, have confirmed the observation that an application of the Crank-Nicolson scheme to problems involving free or moving boundaries may lead to oscillations*

in the solution, see, e.g., [18] and [40, Section 5.3]. The damping property of the implicit Euler method suppresses these oscillations, leading to better results.

Enforcing the Interface Condition (2.1d). The linear system (3.1) does not take into account the interface condition (2.1d). In [16] and [49], this condition is enforced by deriving explicit constraint equations that are then added to (3.1) as penalty terms. An alternative to this explicit method is adapted from [33]. We require only the following weak form of (2.1d):

$$\int_{\Gamma_I(t^j)} (y - y_M) v \, ds = 0 \quad \forall v \in B^j,$$

which yields the linear system

$$\mathbf{P}^j Y^j = Y_I^j$$

that is added as a penalty term to (3.1):

$$\begin{aligned} & \left(\frac{1}{\Delta t_y^j} \mathbf{M}^j + \mathbf{K}^j + \beta (\mathbf{P}^j)^\top \mathbf{P}^j \right) Y^j \\ & = \frac{1}{\Delta t_y^j} \mathbf{M}_{j-1}^j Y^{j-1} + \mathbf{G}^j + \mathbf{F}^j + \beta (\mathbf{P}^j)^\top Y_I^j \end{aligned} \quad (3.4)$$

for some $\beta > 0$.

4 Solving the Level Set Equation

As the level set equation (2.3) is a first order Hamilton-Jacobi (HJ) equation, its spatial and temporal discretizations have to be undertaken with care to obtain the viscosity solution. There is a well developed set of finite difference schemes that have to be combined with total variation diminishing Runge-Kutta methods, see [28] and [41]. In a finite element framework, one can either use a remeshing technique as in [7] or resort to GLS stabilization as in [16] and [49]. Discontinuous Galerkin (DG) methods provide a modern and powerful framework for the spatial approximation of hyperbolic conservation laws. Cheng and Shu recently proposed a DG scheme for HJ equations on rectangular grids [15]. We extend this approach to triangular meshes which offer more flexibility for complex geometries as in the example in Section 6.3.

4.1. Spatial Approximation. The main idea of the DG scheme in [15] for a general Hamilton-Jacobi equation

$$u_t + H(x, \nabla u) = 0$$

with a convex Hamiltonian $H(x, p)$ is to add Roe-type stabilization terms to the discretized weak form of this equation on all edges e of the rectangular mesh, on which $\nabla_p H \cdot n_e < 0$. Here, n_e denotes the outward unit normal of an element of the mesh on e .

For the linear level set equation

$$\phi_t + V \cdot \nabla \phi = 0 \quad \text{in } \mathbb{R}^2,$$

this idea translates to a triangulation \mathcal{T}_h of \mathbb{R}^2 (or D) as follows. We define the space of trial functions

$$\mathcal{P}_{\text{pw}}^k := \{v \mid v \in P^k(K) \text{ for all } K \in \mathcal{T}_h\}.$$

The scheme then reads:

Find $\phi_h \in \mathcal{P}_{\text{pw}}^k$ such that

$$0 = \int_K ((\phi_h)_t + V \cdot \nabla \phi_h) v_h dx + \sum_{i=1}^3 \int_{e_i} \min(V \cdot n_{e_i}, 0) [\phi_h]_{e_i} v_h^- ds \quad (4.1)$$

for all $v_h \in \mathcal{P}_{\text{pw}}^k$ and for all $K \in \mathcal{T}_h$.

The first term, which is obtained by multiplying (2.3a) with a test function v_h and integrating this product over an element K of the triangulation, guarantees the accuracy of the scheme. The boundary integral terms are added for stabilization purposes on all inflow edges, i.e., on all edges on which $V \cdot n_{e_i} < 0$. To ensure that the scheme approximates a continuous solution (the viscosity solution), the jump of ϕ_h across e_i ,

$$[\phi_h]_{e_i} := \phi_h^+(x, t) - \phi_h^-(x, t), \quad \text{where} \quad \phi_h^\pm(x, t) := \lim_{\varepsilon \searrow 0} \phi_h(x \pm \varepsilon n_{e_i}, t),$$

is added to the scheme. Note that this stabilization approach does not require the tuning of parameters. In a three dimensional setting, edges are replaced by faces of tetrahedra.

TABLE 4.1. Butcher array of the optimal SSP(3,3) Runge-Kutta scheme

| | | | |
|---------------|---------------|---------------|---------------|
| 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 |
| $\frac{1}{2}$ | $\frac{1}{4}$ | $\frac{1}{4}$ | 0 |
| | $\frac{1}{6}$ | $\frac{1}{6}$ | $\frac{2}{3}$ |

4.2. Time Stepping Scheme. The DG spatial approximation has to be combined with strong stability preserving (SSP) Runge-Kutta methods in time [29] to ensure stability of the fully discrete scheme. For an ansatz space of polynomial degree k in (4.1), a Runge-Kutta method of order $k + 1$ should be used. In the numerical examples, we use quadratic ansatz spaces and employ the third order SSP scheme defined by the Butcher array in Table 4.1.

Unfortunately, linear stability of the resulting method for the solution of (2.3) can only be guaranteed if a CFL condition on the time step is kept. These conditions are easy to formulate in the one-dimensional case and for rectangular grids in two dimensions but straightforward extensions to triangular meshes are in general not sufficient for the stability. Following Kubatko et al. [35], we formulate the CFL condition as

$$\Delta t_{\text{CFL}} = \frac{h_{\min}}{\sqrt{2}} \frac{1}{5 \cdot 2^{1/(k+1)}} \min \left\{ \frac{1}{\|V\|_{\infty}}, \frac{1}{2} \right\}, \quad (4.2)$$

where h_{\min} is length of the shortest element edge of the triangulation \mathcal{T}_h .

5 Details of the Implementation

In this section, we elaborate on some issues concerning the implementation of the proposed finite element approximations (3.4) and (4.1). In particular, we discuss the combination of these schemes. Algorithm 1 summarizes our approach. Apart from an increased implementation effort, it carries over to the three dimensional case.

5.1. Location of the Interface. For linear ansatz functions, the discrete interface is determined by its (at most) two intersection points with the edges of each element. This representation does in general not result in a good approximation of expressions that are related to second derivatives of the level set function, e.g., curvature. In [22], it was even shown that using linear finite elements might prevent approximations of curvature from converging. This is remedied by using a higher

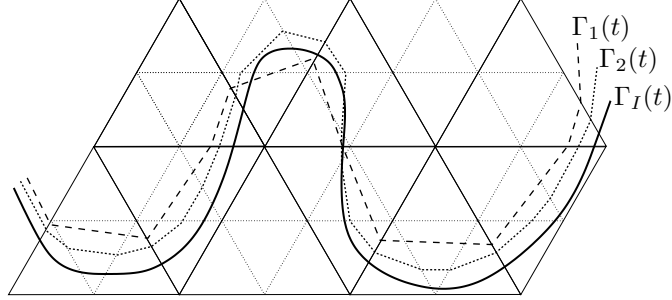


FIGURE 5.1. Determining the position of the discrete interface using linear ($\Gamma_1(t)$) and quadratic ($\Gamma_2(t)$) approximation spaces.

order, e.g. a quadratic, ansatz space for the level set function for which the discrete interface is obtained via a two-level grid as suggested in [30]. Let $\mathcal{T}_{h/2}$ be the triangulation obtained from \mathcal{T}_h by one regular refinement. We define $\mathcal{I}(\phi_h)$ as the continuous piecewise linear function on $\mathcal{T}_{h/2}$ that interpolates ϕ_h at the vertices of all four triangles that form a subdivision of an element of \mathcal{T}_h . Then the location of the discrete interface is determined as

$$\Gamma_2 = \{ x \in D \mid \mathcal{I}(\phi_h)(x) = 0 \},$$

see Figure 5.1. If the discrete interface found in this way consists of several non-connected parts, we ignore those parts whose diameter is less than the length of the longest edge in the triangulation.

Remark 5.1 (Alternative Discrete Interface Approximations). *Fried [21, Section 6.2], and Cheng and Fries [14] develop more elaborate strategies for determining the location of the discrete interface. The latter approach requires the application of isoparametric integration routines, see Section 5.2. However, in our numerical experiments, we did not encounter any stability problems with the discussed interpolation routine. Moreover, the proposed approach carries over directly to the three-dimensional case.*

5.2. Numerical Integration. As the kernels of the enrichment functions are not differentiable across $\Gamma_I(t)$, standard numerical integration routines might not produce reasonable approximations to the integrals in (3.3). This is remedied as follows.

- To compute the entries of \mathbf{M}^j , \mathbf{K}^j , \mathbf{G}^j and \mathbf{F}^j , we first subdivide the intersected elements according to the position of the discrete interface as indicated by Figure 5.2(a). We then apply standard numerical integration routines of appropriate order on each of these sub-triangles. This approach

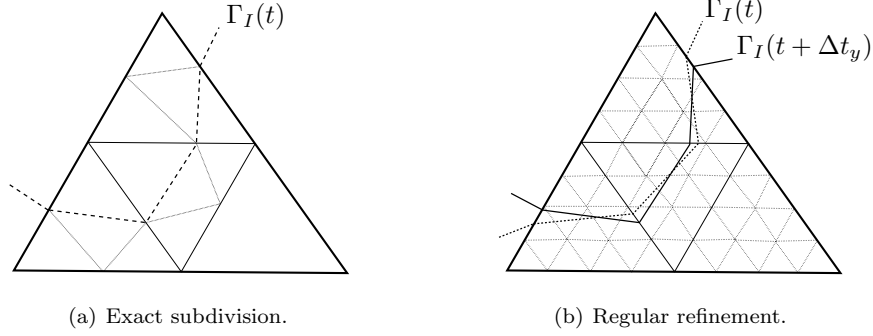


FIGURE 5.2. Numerical integration on intersected elements.

was proposed in [39] in the context of computational mechanics and is also used by Chessa et al. [16] for the treatment of phase change problems. Note that this exact subdivision approach naturally suggests how to choose the coefficients c and k on each of the triangles involved.

- In case of \mathbf{M}_{j-1}^j this exact subdivision is not applicable as an element might be intersected by two interfaces. Therefore, we regularly refine these elements and apply numerical integration schemes on the resulting triangulation. This approach was first proposed in [19] and is adapted by Zabaras et al. [49]. This situation is illustrated by Figure 5.2(b). Note that the entries of \mathbf{M}_{j-1}^j are independent of c and of k .

5.3. Extension Velocity. When formulating the level set equation we tacitly made the assumption that the velocity field V is known on all of D . Obviously this is not the case as the Stefan condition (2.2) defines only the normal velocity $V \cdot n$ on the interface.

There are several (equivalent) ways to construct a vector valued velocity field with global support from this scalar quantity.

- (1) The first approach is to make the natural identification

$$V_i = \frac{1}{\rho L} \left(k_S \frac{\partial y_S}{\partial x_i} - k_F \frac{\partial y_F}{\partial x_i} \right) \quad \text{on } \Gamma_I(t) \quad (5.1)$$

and to extend this velocity field to all of D . This approach requires two evaluations of the gradient of y .

- (2) Alternatively, one can set

$$V = (V \cdot n) n \quad \text{on } \Gamma_I(t) \quad (5.2)$$

and use an extension of this velocity vector. This approach requires only the evaluation of $V \cdot n$ which can be implemented using a simple jump calculation, see Section 5.4, and should thus be cheaper than (1). As the velocity field defined by (5.1) has no tangential component due to (2.1d), it can be written as in (5.2) and thus (1) and (2) are equivalent.

- (3) Finally, one can rotate the coordinate system in a way that the x_1 -axis transforms to the outer normal of $\Gamma_I(t)$. The jump of the gradient (as in (5.1)) in this new coordinate system can then be easily evaluated as it is given by $V \cdot n$ in the direction corresponding to x_1 and zero in the direction corresponding to x_2 . In the original coordinate system, we obtain the representation (5.2) and therefore, also (2) and (3) are equivalent.

As mentioned above, we consider (2) and (3) to be more efficient than (1). Thus, we discuss the evaluation of the Stefan condition (2.2) in the next section in more detail. We now turn to the question of how we compute the extension of V to all of D .

The support of a quantity ψ that is known on an interface only can be enlarged to all of D or at least to the vicinity of the interface by means of a constant extension in normal direction. This is achieved by solving

$$\text{sign}(\phi) \nabla \psi \cdot \nabla \phi = 0 \quad \text{in } D \quad (5.3a)$$

$$\psi = \hat{\psi} \quad \text{on } \Gamma_I(t). \quad (5.3b)$$

As in [16], we add the sign term to ensure that the characteristics of this equation emanate from $\Gamma_I(t)$ which enables us to use the given data $\hat{\psi}$ on $\Gamma_I(t)$ as a boundary condition, see Figure 5.3(a). If a narrow band approach is used (see Section 5.6), D in (5.3a) must be replaced by the corresponding local domain of computation. This procedure is applied to both components V_i of the velocity vector. This choice of extension velocity moves the interface with the correct speed and preserves the signed distance property of ϕ [2].

In contrast to [16], we use the fast marching method of Kimmel and Sethian [34] to construct a signed distance function to the current position of the interface and simultaneously solve (5.3) for both components of the velocity field on the triangular mesh. The key idea of the fast marching scheme is to subdivide the grid points into three sets, see Figure 5.3(b). All points for which the initial condition is known (or can be explicitly computed) are tagged as *alive* and remain unchanged. All points that are adjacent to *alive* points are tagged as *close* and all others are

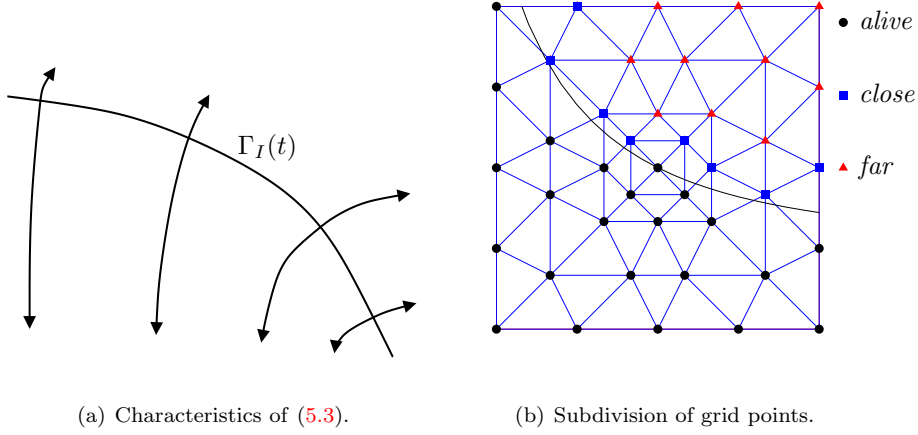


FIGURE 5.3. Construction of extension velocities by the fast marching scheme.

tagged as *far*. The method then iteratively moves points tagged as *close* to the *alive* nodes and at the same time moves nodes from the set of *far* points to the *close* nodes. This procedure is implemented to monotonically move the front of *close* nodes and update the signed distance function in an upwind way. During the update procedure several triangles might contribute to the value of the signed distance function at a certain node. In this case, the triangle yielding the smallest value is taken as the upwind triangle. We use the same triangle to update both components of the extension velocity. We finally remark that all fast marching computations are carried out on the finer level of the grid which is also used to determine the current interface position, see Section 5.1.

5.4. Evaluating the Stefan Condition. The Stefan condition

$$\rho L V \cdot n = k_S \frac{\partial y_S}{\partial n} - k_F \frac{\partial y_F}{\partial n} = [k \nabla y]_F^S \cdot n \quad \text{on } \Gamma_I(t) \quad (2.2)$$

requires the evaluation of ∇y on $\Gamma_I(t)$. Due to the discontinuity of ∇y across $\Gamma_I(t)$ a straightforward evaluation is not possible. Since we are seeking only the normal component of the velocity, the domain integral method of Ji and Dolbow [33] could be applied. A more efficient approach to evaluating the normal velocity is the jump calculation used by Zabararas et al. [49], which we adapt here.

In addition to the point x_d on $\Gamma_I(t)$, the temperature is evaluated at four points, denoted by $x_S^{\delta/4}$, $x_S^{\delta/2}$, $x_S^{3\delta/4}$, x_S^δ and $x_F^{\delta/4}$, $x_F^{\delta/2}$, $x_F^{3\delta/4}$, x_F^δ , respectively, that are located on normals through x_d to $\Gamma_I(t)$ on each side of the interface. The distances

of these points to $\Gamma_I(t)$ are $\frac{1}{4}\delta$, $\frac{1}{2}\delta$, $\frac{3}{4}\delta$ and δ , where δ is chosen as the average element size. The slopes of the least squares lines through the pairs

$$\left\{ (0, y(x_d)), \left(\frac{\delta}{4}, y(x_{S/F}^{\delta/4})\right), \left(\frac{\delta}{2}, y(x_{S/F}^{\delta/2})\right), \left(\frac{3}{4}\delta, y(x_{S/F}^{3\delta/4})\right), (\delta, y(x_{S/F}^{\delta})) \right\}$$

are then taken as approximations to the temperature gradients in normal direction.

Thus, the normal velocity of the interface is evaluated according to

$$V \cdot n = \frac{1}{\rho L} \left[k_S \frac{2}{5} \frac{2y(x_d) + y(x_S^{\delta/4}) - y(x_S^{3\delta/4}) - 2y(x_S^{\delta})}{\delta} - k_F \frac{2}{5} \frac{2y(x_F^{\delta}) + y(x_F^{3\delta/4}) - y(x_F^{\delta/4}) - 2y(x_d)}{\delta} \right]$$

at the point x_d on $\Gamma_I(t)$. Note that both $x_S^{\delta/2}$ and $x_F^{\delta/2}$ drop out in this formula. This means that for the proposed evaluation procedure of the normal velocity, we need three points on either side of the interface and one point on the interface for each evaluation point.

This construction is of course more involved and also slightly more costly than the simple jump calculation proposed in [49], but all numerical experiments have confirmed that this choice of the normal velocity is robust, reliable and, most important, reasonably accurate.

Remark 5.2. *The fast marching method described in Section 5.3 needs as input data the values of the velocity field at those nodes of the grid that are adjacent to the interface. To determine these values, we project each of these nodes onto the piecewise linear approximation of the interface. The velocity field is then evaluated at these projected points (which correspond to the x_d nodes in the above formula) and the values obtained by the jump calculation are copied to the corresponding nodes of the grid.*

5.5. Taylor Basis. The discontinuous Galerkin scheme (4.1) is implemented using a Taylor basis [36] for the space \mathcal{P}_{pw}^k . This basis has the advantage over the standard Lagrangian basis that setting all off-diagonal entries to zero is a conservative mass lumping strategy. Moreover, the Taylor basis allows for the efficient solution of the adjoint systems in an optimal control approach to motion planning for the two-phase Stefan problem in level set formulation, see [10].

5.6. Narrow Band Level Set Method. The major drawback of level set techniques is the considerable additional effort that is introduced by embedding the

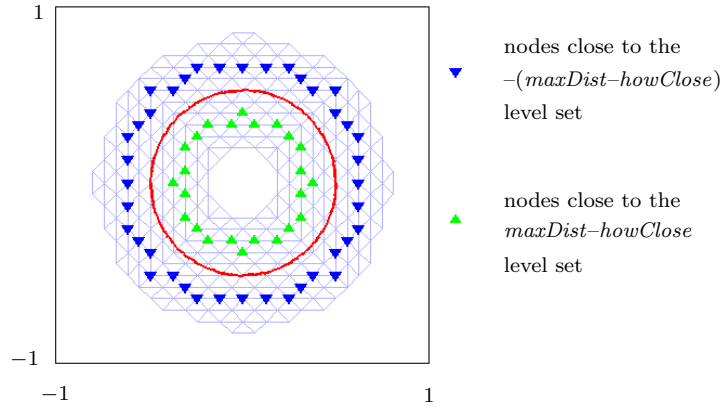


FIGURE 5.4. The interface (red line) embedded into the narrow band (gray) that collects all nodes whose distance from the interface is less than $maxDist$.

moving interface into a higher dimensional object whose evolution is described by a partial differential equation. In cases where there is no physical quantity that can be used as the velocity field V in (2.3a), such a velocity field has to be constructed as discussed in Section 5.3. These problems are at least partly overcome by using a narrow band technique that solves the level set equation, and the corresponding extension PDE (5.3), only in the vicinity of the interface [1, 42], see Figures 5.4 and 6.4. Such a localized computational domain is constructed, e.g., by collecting all the nodes of the grid that are less than a predefined threshold $maxDist$ away from the interface. A monitoring procedure controls the distance $howClose$ of the interface from the boundary of the current narrow band (tube). If this distance is getting too small, the tube has to be rebuilt which requires also the construction of a signed distance function to the current interface. To implement the monitoring procedure, we determine the $\pm(maxDist-howClose)$ level sets of the current signed distance function. All nodes that belong to triangles that are intersected by these level sets and whose signed distance to the interface is larger than $maxDist-howClose$ or smaller than $-(maxDist-howClose)$, respectively, are tagged. Finally, the values of the signed distance function at these nodes are checked in each time step, and, if any of these values changes sign, the tube is rebuilt. This idea is illustrated in Figure 5.4.

5.7. Boundary Conditions. To make the level set equation (2.3) well-posed on a finite domain, boundary conditions on the inflow boundaries, i.e., on those parts of

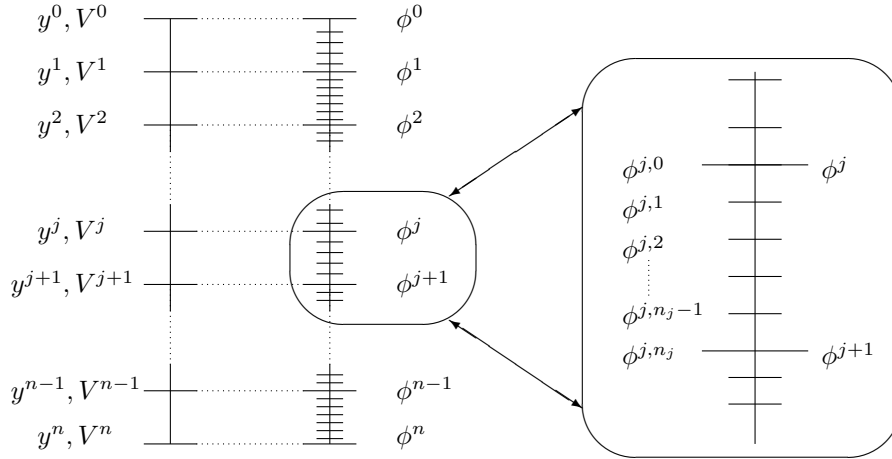


FIGURE 5.5. Interplay of the time stepping schemes.

the boundary of the computational domain on which $V \cdot n < 0$, where n is the outer unit normal to the boundary of the domain, need to be specified. This is reflected by the scheme (4.1) that takes into account only the jumps across those edges on which the inflow condition is fulfilled. Since there are no physical boundary conditions on ϕ given, we impose artificial Dirichlet boundary conditions on the time-discrete level by prescribing ϕ^j as the required Dirichlet data in the time step from ϕ^j to ϕ^{j+1} , see Figure 5.5. Numerical experiments confirm that the influence of these boundary conditions on the evolution of ϕ is negligible if reasonably small time steps and a frequent reinitialization of the level set function ϕ to a signed distance function are used.

5.8. Synchronizing the Time-Stepping Schemes. While the energy equation (2.1) is discretized in time using an implicit Euler method, a Runge-Kutta scheme is used for the level set equation (2.3). The corresponding time steps must be synchronized. Δt_{CFL} from (4.2) determines the step length for the level set equation. In practice, this could be smaller than the desired step size for the heat equation. In this case, several time steps for the level set equation must be executed to arrive at the next point of the time grid for the heat equation. This procedure is illustrated by Figure 5.5.

6 Numerical Results

In this section, we investigate the order of convergence of Algorithm 1 on the Frank Sphere example with respect to simultaneous spatial and temporal refinement. In

Algorithm 1 Solver for Two-Phase Stefan Problem

Input: $D, \Gamma_I(0), y_0, y_M, \rho, c_S, c_F, k_S, k_F, L, f, g, \{t^j\}_{j=1}^{N_T}$ **Output:** $y^j = y(t^j), \Gamma_I^j = \Gamma_I(t^j), \quad j = 1, \dots, N_T$

- 1: Set up ϕ^0, V^0 and Δt_{CFL}^0 and initialize the narrow band, $j \rightarrow 1$.
 - 2: **while** $j \leq N_T$ **do**
 - 3: $t_\phi \rightarrow 0, m \rightarrow 1$
 - 4: **while** $t_\phi < t^j - t^{j-1}$ **do**
 - 5: Compute $\phi^{j-1, m}$ using $\Delta t_\phi = \min \{ \Delta t_{\text{CFL}}^{j-1}, t^j - t^{j-1} - t_\phi \}$.
 - 6: Update the narrow band if necessary.
 - 7: Reinitialize ϕ if necessary.
 - 8: $m \rightarrow m + 1, t_\phi \rightarrow t_\phi + \Delta t_\phi$
 - 9: **end while**
 - 10: Determine Γ_I^j and the enriched nodes.
 - 11: Assemble and solve (3.4) for y^j .
 - 12: Evaluate V^j and reinitialize ϕ^j .
 - 13: Compute Δt_{CFL}^j .
 - 14: $j \rightarrow j + 1$
 - 15: **end while**
-

addition, we solve the benchmark problem of solidification in an infinite corner, and we also demonstrate the capability of the method to track topological changes. In the examples in Sections 6.1 and 6.2, $c_S = c_F = k_S = k_F$ holds and the exact solutions (or at least the interface position in the second example) are known. Unfortunately, we are not aware of an example with known solution in which either $c_S \neq c_F$ or $k_S \neq k_F$ holds. Such an example would enable us to reliably check the order of convergence of our solver in that case.

All our computations are based on quadratic ansatz spaces for both the temperature field and the level set field. We always chose an equidistant time grid for the temperature, i.e., $\Delta t_y^j = \Delta t_y$. The time grid for the level set equation is constructed automatically during the computation as specified in Algorithm 1. The main part of the solver is implemented as a prototype in MATLAB. Some time critical components have been implemented in the C++ programming language.

6.1. The Frank Sphere Example. The growing Frank Sphere [20] is a good benchmark example (used, e.g., in [4, 12, 13, 27]) to verify the implementation of

Algorithm 1 and to determine the order of convergence, as an exact solution for both, the temperature and the interface position, is known. The Frank sphere example has been found to be ill-posed [26], making its accurate numerical solution a challenging task. The setup is such that a solid nucleus which is initially a circle of radius s_0 at the equilibrium temperature $y_M = 0$ grows into an undercooled liquid in the infinite domain \mathbb{R}^2 . The initial radius of the nucleus, s_0 , is related to the temperature at infinity, y_∞ , by the equation

$$y_\infty = \frac{s_0 F(s_0)}{2 F'(s_0)}.$$

The value $s_0 = 0.25$ corresponds to $y_\infty = -0.05709187113307$ [12]. At time t , the exact temperature distribution is

$$y(x, t) = \tilde{y}(s) = \begin{cases} y_\infty \left(1 - \frac{F(s)}{F(s_0)}\right) & s > s_0, \\ 0 & s \leq s_0, \end{cases} \quad (6.1)$$

where the similarity variable $s = \frac{r}{\sqrt{t}}$ depends on the radius $r = \sqrt{x_1^2 + x_2^2}$, and

$$F(s) = \int_{s^2/4}^{\infty} \frac{e^{-t}}{t} dt.$$

Thus, at any instance of time, the interface is the boundary of

$$\Omega_S(t) = \{(x_1, x_2) \in \mathbb{R}^2 \mid \sqrt{x_1^2 + x_2^2} \leq s_0 \sqrt{t}\},$$

i.e., it is a circle with radius $s_0 \sqrt{t}$.

To solve this problem numerically in the time interval $[1, 2]$, we use the hold-all $D = [-1, 1]^2$ and the parameters

$$\rho = L = c_S = c_F = k_S = k_F = 1, \quad y_M = 0.$$

To estimate the experimental order of convergence, we uniformly refine the spatial discretization, starting from 144 triangles ($h = 0.4$) up to 36864 triangles ($h = 0.025$). Simultaneously, the temporal grid is refined, starting from 20 time steps ($\Delta t_y = 0.05$) up to 100 time steps ($\Delta t_y = 0.01$).

Remark 6.1. *The choice of the temporal discretization may seem a bit arbitrary. In fact, numerical experiments with different couplings of the form $\Delta t_y = \mathcal{O}(h^\alpha)$, $\alpha > 0$, have shown that we obtain at most linear convergence w.r.t. h on the Frank Sphere example. However, this linear rate can be obtained by different combinations of temporal and spatial refinements.*

The coupling of the spatial and the temporal discretizations allows us to interpret the errors in the temperature and in the interface position as functions of the spatial grid size parameter h alone. Consequently, the experimental order of convergence (eoc), for instance, of the interface position (see (6.3)) for two different grid sizes h_1, h_2 can be estimated as

$$\text{eoc} = \frac{\ln E_T^\infty(\phi_{h_1}) - \ln E_T^\infty(\phi_{h_2})}{\ln h_1 - \ln h_2},$$

assuming that the error has a representation of the form

$$E_T^\infty(\phi_h) = c h^p$$

with a suitable constant $c > 0$. For a sequence of decreasing grid sizes $(h_n)_{n=1}^N$ and corresponding errors $(E_T^\infty(\phi_{h_n}))_{n=1}^N$, the order of convergence can be estimated by a least-squares approach. In this way, the last lines in Tables 6.1–6.2 were computed. Exact Dirichlet boundary conditions are imposed on ∂D using (6.1). The error indicators for the temperature at $t = T$,

$$E_T^2(y_h) := \left(\int_D |y_h(x, T) - y(x, T)|^2 dx \right)^{1/2}$$

and

$$E_T^\infty(y_h) := \sup_D |y_h(x, T) - y(x, T)|,$$

are of L^2 and L^∞ type, respectively. Here, y_h denotes the numerical solution, and y is the exact solution as given in (6.1). Similarly, we use the L^2 measure

$$E_T^2(\phi_h) := \left(\int_{\Gamma_I(T)} \phi_h^2 ds \right)^{1/2} \quad (6.2)$$

as well as the L^∞ measure

$$E_T^\infty(\phi_h) := \sup_{\Gamma_I(T)} |\phi_h| \quad (6.3)$$

to estimate the error in the interface position. As documented in Table 6.1 and Table 6.2, the convergence rate in all four cases is slightly less than linear. Potential reasons for this sub-optimal behavior are that the reinitialization of the level set function, as well as the construction of the extension velocity are only of first order, and that we use a polygonal (as opposed to piecewise quadratic) interface approximation. Nevertheless, compared with the results in [12], where the same parameters are used, our solver exhibits reasonable accuracy. Using the corrected X-FEM of Fries [23] can help to improve the observed convergence rates. In addition, a higher-order version of the fast marching method on triangular grids [45] or a

TABLE 6.1. Errors and estimated orders of convergence for the temperature in the Frank sphere example at $t = T$.

| h | Δt_y | $E_T^2(y_h)$ | eoc | $E_T^\infty(y_h)$ | eoc |
|-------|--------------|--------------|------|-------------------|------|
| 0.400 | 0.0500 | 5.5883e-03 | – | 8.2476e-03 | – |
| 0.200 | 0.0250 | 3.3103e-03 | 0.76 | 5.1098e-03 | 0.69 |
| 0.100 | 0.0167 | 1.5653e-03 | 1.08 | 2.5916e-03 | 0.98 |
| 0.050 | 0.0125 | 7.6895e-04 | 1.03 | 1.2687e-03 | 1.03 |
| 0.025 | 0.0100 | 4.0455e-04 | 0.93 | 6.3618e-04 | 1.00 |
| eoc | | | 0.97 | | 0.94 |

TABLE 6.2. Errors and estimated orders of convergence for the interface position in the Frank sphere example at $t = T$.

| h | Δt_y | $E_T^2(\phi_h)$ | eoc | $E_T^\infty(\phi_h)$ | eoc |
|-------|--------------|-----------------|------|----------------------|------|
| 0.400 | 0.0500 | 1.3867e-01 | – | 1.1295e-01 | – |
| 0.200 | 0.0250 | 7.9418e-02 | 0.80 | 6.6132e-02 | 0.77 |
| 0.100 | 0.0167 | 3.7637e-02 | 1.08 | 3.1993e-02 | 1.05 |
| 0.050 | 0.0125 | 1.8434e-02 | 1.03 | 1.4870e-02 | 1.11 |
| 0.025 | 0.0100 | 9.6054e-03 | 0.94 | 7.3170e-03 | 1.02 |
| eoc | | | 0.98 | | 1.00 |

different reinitialization PDE [42] can be used to improve the overall order of the proposed scheme at the expense of a higher computational cost.

6.2. Solidification in a Corner. In this section we validate Algorithm 1 by considering the benchmark problem of a material solidifying in an infinite corner that has been treated by a lot of researchers, see [5, 16, 37, 49]. We simulate this problem by solving the two-phase Stefan problem in the domain $D = [0, 14] \times [0, 14]$. The parameters are chosen as

$$\rho = 1, L = 0.25, c_S = c_F = k_S = k_F = 1$$

resulting in equal thermal diffusivities in both phases. The melting temperature is $y_M = 0$ while the initial temperature is $y_0(x) = 0.3$. As in the example above we neglect heat sources ($f = 0$) but now the boundary conditions are given by the

Dirichlet condition $y = -1$ on the lower and the left part of ∂D and the insulation condition $\frac{\partial y}{\partial n} = 0$ on the upper and the right part of ∂D .

As cited in [5], the interface position in dimensionless coordinates

$$x_i^* = \frac{x_i}{\sqrt{4at}}$$

is constant in time and given by

$$x_2^* = \left(\lambda^u + \frac{C}{(x_1^*)^u - \lambda^u} \right)^{1/u} \quad (6.4)$$

for an infinite region. The constants $C \approx 0.159$, $u \approx 5.02$ and $\lambda \approx 0.70766$ are determined numerically and a is the thermal diffusivity that has to be equal in both phases:

$$a = \frac{k_S}{\rho c_S} = \frac{k_F}{\rho c_F} = 1.$$

Figure 6.1 shows this dimensionless interface in comparison with interface positions obtained by Algorithm 1 at four different time steps. The prediction of the interface is very close to the analytic solution at all four time steps. In addition, Figure 6.2 shows the temperature distribution for the solidification process corresponding to the four time steps in Figure 6.1.

6.3. Hold-All with a More Complex Geometry. Our main motivation for using the level set method for representing the interface $\Gamma_I(t)$ is that this method is capable of tracking topological changes. In this section we construct an example that demonstrates this feature. In addition, this last example involves a more complex geometrical setup of the hold-all D that can not be easily treated by using a rectangular grid.

The domain we consider has a star-like shape and the initial position of $\Gamma_I(t)$ is given by a slightly scaled and rotated version of the standard quatrefoil [25], that is sometimes used as a benchmark example for testing solvers for evolving interfaces. Just like in the examples before, the fluid phase, which corresponds to the interior of the quatrefoil, is enclosed by the solid phase, and the initial temperature distribution is given by

$$y_0(x) = \begin{cases} \phi_0(x) & x \in \Omega_S(0), \\ 2\phi_0(x) & x \in \Omega_F(0). \end{cases}$$

We use the parameters

$$\rho = 1, L = 1, y_M = 0, k_F = 0.5, k_S = 1, c_F = 1, c_S = 1,$$

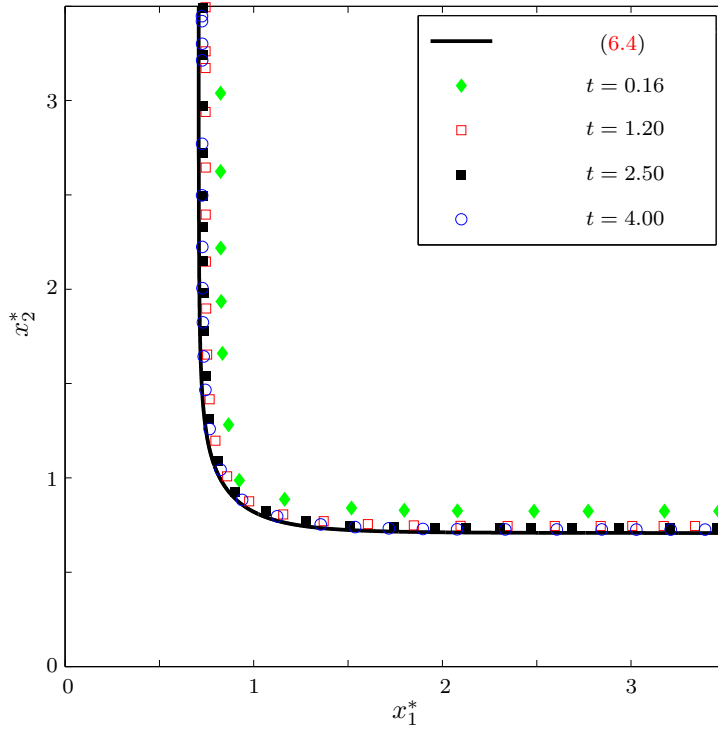


FIGURE 6.1. Analytic and numerical solution for the corner solidification problem at four time steps in dimensionless coordinates.

and prescribe piecewise constant Neumann boundary conditions that will cause a nonsymmetric inwards solidification. Eventually, the interface $\Gamma_I(t)$ vanishes and all of D is occupied by the solid phase $\Omega_S(t)$. The results are shown in Figure 6.5 at six different time steps. Note the pinch-off in the upper left corner, shown in Figure 6.5(b). This topological change is handled nicely by our method.

Acknowledgments

This work was financially supported by FWF (<http://www.fwf.ac.at>) under project grant P19918-N14. The authors would like to thank Karsten Eppler for suggesting the choice of V in (5.2) and Frank Schmidt for providing the least-squares code for estimating the order of convergence.

References

- [1] D. Adalsteinsson and J. A. Sethian. A fast level set method for propagating interfaces. *Journal of Computational Physics*, 118(2):269–277, 1995. doi:10.1006/jcph.1995.1098.
- [2] D. Adalsteinsson and J. A. Sethian. The fast construction of extension velocities in level set methods. *Journal of Computational Physics*, 148:2–22, 1999. doi:10.1006/jcph.1998.6090.

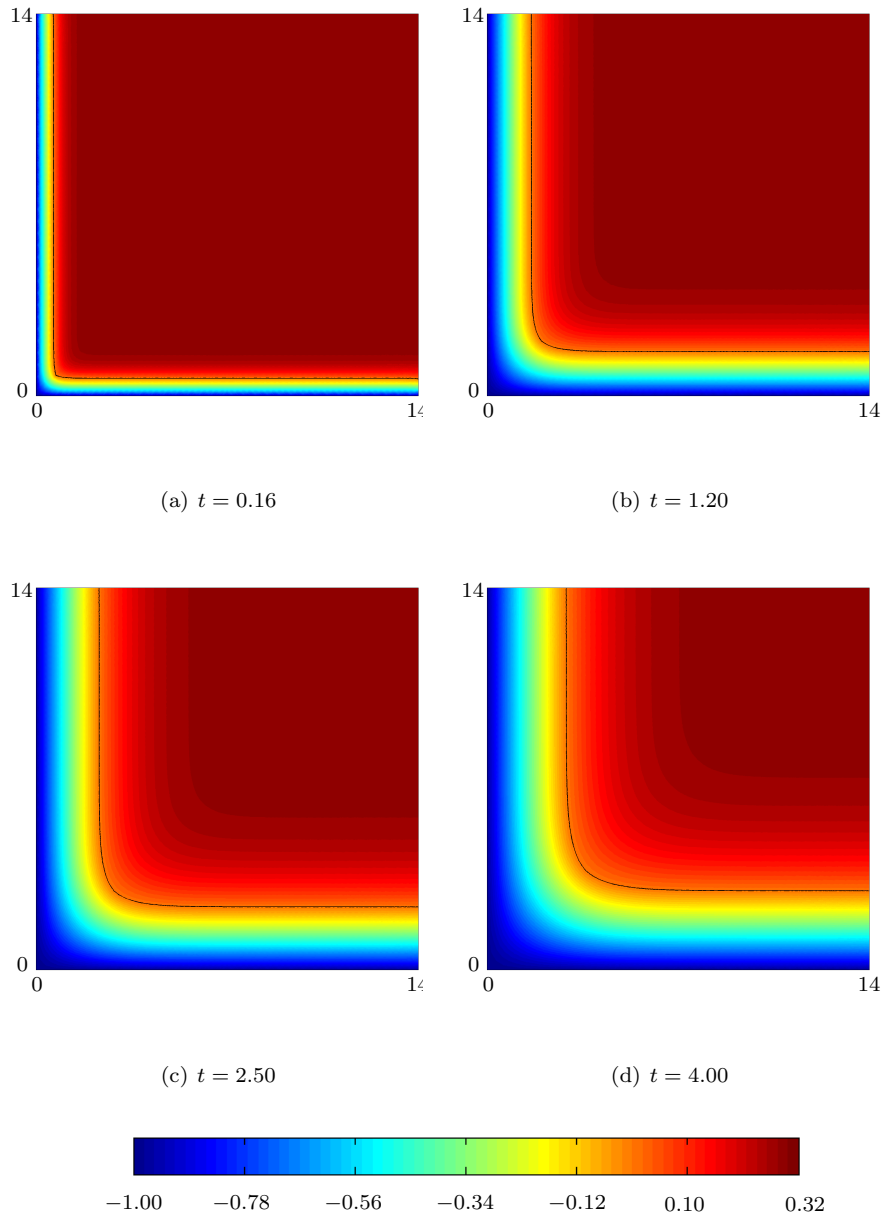


FIGURE 6.2. Temperature distribution for the corner solidification problem at four time steps.

- [3] M. R. Albert and K. O'Neill. Moving boundary-moving mesh analysis of phase change using finite elements with transfinite mappings. *International Journal for Numerical Methods in Engineering*, 23(4):591–607, 1986. doi:10.1002/nme.1620230406.
- [4] R. Almgren. Variational algorithms and pattern formation in dendritic solidification. *Journal of Computational Physics*, 106(2):337–354, 1993. doi:10.1016/S0021-9991(83)71112-5.

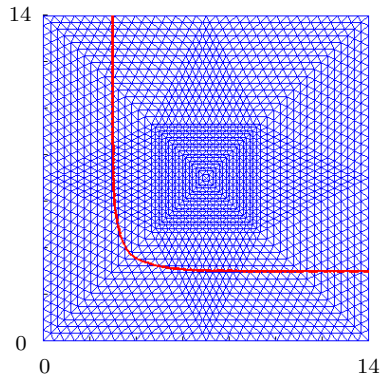


FIGURE 6.3. Final interface position and global mesh with 4096 triangles.

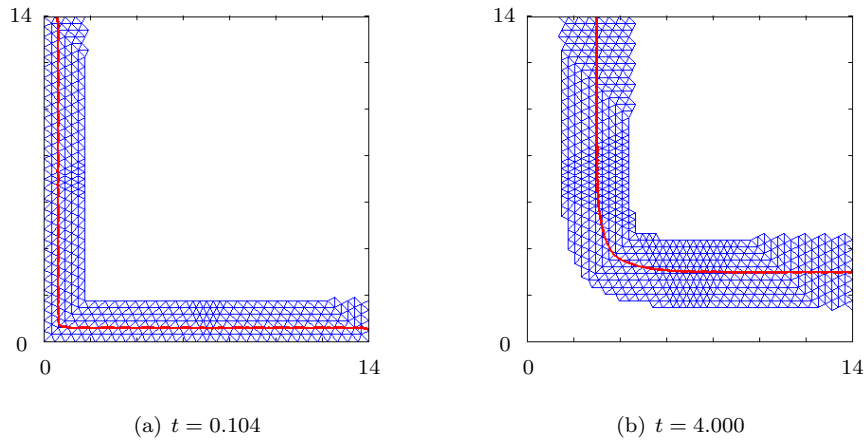


FIGURE 6.4. Interface position and corresponding narrow band mesh at two different time steps.

- [5] A. Ayasoufi and T. Keith. Application of the conservation element and solution element method in numerical modeling of heat conduction with melting and/or freezing. *International Journal of Numerical Methods for Heat & Fluid Flow*, 13:448–472, 2003. doi:[10.1108/09615530310475902](https://doi.org/10.1108/09615530310475902).
- [6] M. Baines, M. Hubbard, and P. Jimack. A moving mesh finite element algorithm for the adaptive solution of time-dependent partial differential equations with moving boundaries. *Applied Numerical Mathematics*, 54(3-4):450–469, 2005. doi:[10.1016/j.apnum.2004.09.013](https://doi.org/10.1016/j.apnum.2004.09.013).
- [7] T. Barth and J. A. Sethian. Numerical schemes for the Hamilton-Jacobi and level set equations on triangulated domains. *Journal of Computational Physics*, 145:1–40, 1998. doi:[10.1006/jcph.1998.6007](https://doi.org/10.1006/jcph.1998.6007).
- [8] G. Beckett, J. A. Mackenzie, A. Ramage, and D. M. Sloan. Computational solution of two-dimensional unsteady pdes using moving mesh methods. *Journal of Computational Physics*, 182(2):478–495, 2002. doi:[10.1006/jcph.2002.7179](https://doi.org/10.1006/jcph.2002.7179).

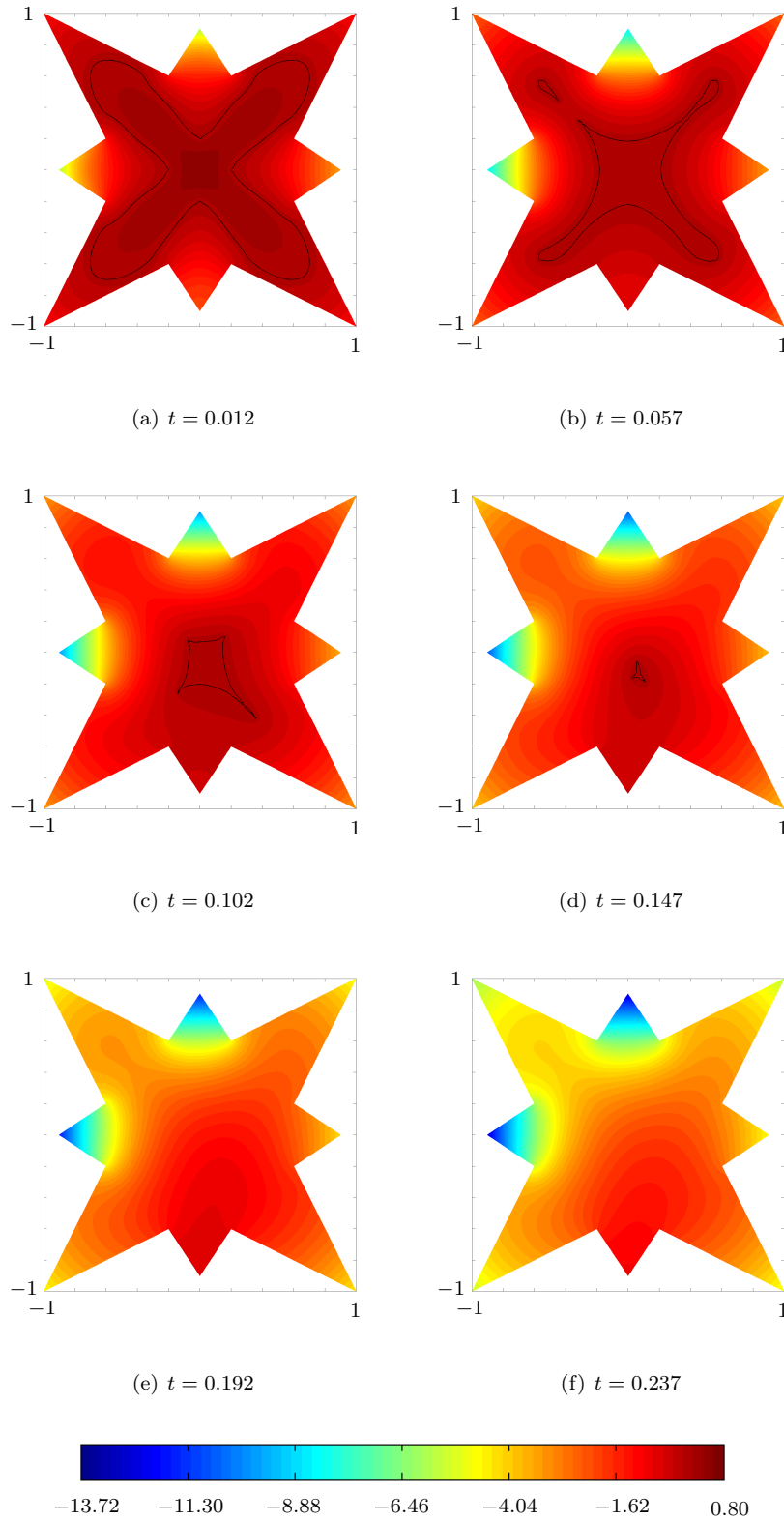


FIGURE 6.5. Temperature distribution in a star-like domain at six time steps.

- [9] G. Beckett, J. A. Mackenzie, and M. L. Robertson. A moving mesh finite element method for the solution of two-dimensional stefan problems. *Journal of Computational Physics*, 168(2):500–518, 2001. doi:10.1006/jcph.2001.6721.
- [10] M. K. Bernauer and R. Herzog. Optimal control of the classical two-phase Stefan problem in level set formulation. *SIAM Journal on Scientific Computing*, 33(1):342–363, 2011. doi:10.1137/100783327.
- [11] J. Caldwell and C.-C. Chan. Numerical solutions of the Stefan problem by the enthalpy method and the heat balance integral method. *Numerical Heat Transfer, Part B: Fundamentals: An International Journal of Computation and Methodology*, 33:99–117, 1998. doi:10.1080/10407799808915025.
- [12] H. Chen, C. Min, and F. Gibou. A numerical scheme for the Stefan problem on adaptive Cartesian grids with supralinear convergence rate. *Journal of Computational Physics*, 228(16):5803–5818, 2009. doi:10.1016/j.jcp.2009.04.044.
- [13] S. Chen, B. Merriman, S. Osher, and P. Smereka. A simple level set method for solving Stefan problems. *Journal of Computational Physics*, 135(1):8–29, 1997. doi:10.1006/jcph.1997.5721.
- [14] K. Cheng and T. Fries. Higher-order XFEM for curved strong and weak discontinuities. *International Journal for Numerical Methods in Engineering*, 82(5):564–590, 2010. doi:10.1002/nme.2768.
- [15] Y. Cheng and C.-W. Shu. A discontinuous Galerkin finite element method for directly solving the Hamilton-Jacobi equations. *Journal of Computational Physics*, 223(1):398–415, 2007. doi:10.1016/j.jcp.2006.09.012.
- [16] 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. doi:10.1002/nme.386.
- [17] A. J. Chorin. Curvature and solidification. *Journal of Computational Physics*, 57(3):472–490, 1985. doi:10.1016/0021-9991(85)90191-3.
- [18] D. M. Christopher. Comparison of interface-following techniques for numerical analysis of phase-change problems. *Numerical Heat Transfer, Part B: Fundamentals: An International Journal of Computation and Methodology*, 39(2):189–206, 2001. doi:10.1080/10407790150503503.
- [19] J. Dolbow. *An Extended Finite Element Method with Discontinuous Enrichment for Applied Mechanics*. PhD thesis, Northwestern University, 1999. Available from: <http://dolbow.cee.duke.edu/phd.html>.
- [20] F. Frank. Radially symmetric phase growth controlled by diffusion. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, 201(1067):586–599, 1950. doi:10.1098/rspa.1950.0080.
- [21] M. Fried. *Niveauflächen zur Berechnung zweidimensionaler Dendrite*. PhD thesis, Universität Freiburg, 1999. Available from: <http://www.mathematik.uni-freiburg.de/IAM/homepages/micha/publications/diss.pdf>.

- [22] M. Fried. A level set based finite element algorithm for the simulation of dendritic growth. *Computing and Visualization in Science*, 2(2):97–110, 2004. doi:[10.1007/s00791-004-0141-4](https://doi.org/10.1007/s00791-004-0141-4).
- [23] T. Fries. A corrected XFEM approximation without problems in blending elements. *International Journal for Numerical Methods in Engineering*, 75(5):503–532, 2008. doi:[10.1002/nme.2259](https://doi.org/10.1002/nme.2259).
- [24] T. Fries and A. Zilian. On time integration in the XFEM. *International Journal for Numerical Methods in Engineering*, 79(1):69–93, 2009. doi:[10.1002/nme.2558](https://doi.org/10.1002/nme.2558).
- [25] P. Frolkovič and K. Mikula. Flux-based level set method: a finite volume method for evolving interfaces. *Applied Numerical Mathematics. An IMACS Journal*, 57:436–454, 2007. doi:[10.1016/j.apnum.2006.06.002](https://doi.org/10.1016/j.apnum.2006.06.002).
- [26] F. Gibou and R. Fedkiw. A fourth order accurate discretization for the Laplace and heat equations on arbitrary domains, with applications to the Stefan problem. *Journal of Computational Physics*, 202(2):577–601, 2005. doi:[10.1016/j.jcp.2004.07.018](https://doi.org/10.1016/j.jcp.2004.07.018).
- [27] F. Gibou, R. Fedkiw, R. Caflisch, and S. Osher. A level set approach for the numerical simulation of dendritic growth. *Journal of Scientific Computing*, 19(1–3):183–199, 2003. doi:[10.1023/A:1025399807998](https://doi.org/10.1023/A:1025399807998).
- [28] S. Gottlieb and C.-W. Shu. Total variation diminishing Runge-Kutta schemes. *Mathematics of Computation*, 67(221):73–85, 1998. doi:[10.1090/S0025-5718-98-00913-2](https://doi.org/10.1090/S0025-5718-98-00913-2).
- [29] S. Gottlieb, C.-W. Shu, and E. Tadmor. Strong stability-preserving high-order time discretization methods. *SIAM Review*, 43(1):89–112, 2001. doi:[10.1137/S003614450036757X](https://doi.org/10.1137/S003614450036757X).
- [30] S. Groß, V. Reichelt, and A. Reusken. A finite element based level set method for two-phase incompressible flows. *Computing and Visualization in Science*, 9:239–257, 2006. doi:[10.1007/s00791-006-0024-y](https://doi.org/10.1007/s00791-006-0024-y).
- [31] S. Gupta. *The Classical Stefan Problem. Basic Concepts, Modelling and Analysis*, volume 45 of *Applied Mathematics and Mechanics*. North-Holland, Amsterdam, 2003.
- [32] R. H. W. Hoppe. A globally convergent multi-grid algorithm for moving boundary problems of two-phase Stefan type. *IMA Journal of Numerical Analysis*, 13(2):235–253, 1993. doi:[10.1093/imanum/13.2.235](https://doi.org/10.1093/imanum/13.2.235).
- [33] H. Ji and J. Dolbow. On strategies for enforcing interfacial constraints and evaluating jump conditions with the extended finite element method. *International Journal for Numerical Methods in Engineering*, 61:2508–2535, 2004. doi:[10.1002/nme.1167](https://doi.org/10.1002/nme.1167).
- [34] R. Kimmel and J. A. Sethian. Computing geodesic paths on manifolds. *Proceedings of the National Academy of Sciences of the United States of America*, 95(15):8431–8435, 1998.
- [35] E. Kubatko, C. Dawson, and J. Westerink. Time step restrictions for Runge-Kutta discontinuous Galerkin methods on triangular grids. *Journal of Computational Physics*, 227:9697–9710, 2008. doi:[10.1016/j.jcp.2008.07.026](https://doi.org/10.1016/j.jcp.2008.07.026).
- [36] D. Kuzmin. A vertex-based hierarchical slope limiter for p -adaptive discontinuous Galerkin methods. *Journal of Computational and Applied Mathematics*, 233(12):3077–3085, 2010. doi:[10.1016/j.cam.2009.05.028](https://doi.org/10.1016/j.cam.2009.05.028).

- [37] A. Lazaridis. A numerical solution of the multidimensional solidification (or melting) problem. *International Journal of Heat and Mass Transfer*, 13(9):1459–1477, 1970. doi:10.1016/0017-9310(70)90180-8.
- [38] R. Merle and J. Dolbow. Solving thermal and phase change problems with the extended finite element method. *Computational Mechanics*, 28:339–350, 2002. doi:10.1007/s00466-002-0298-y.
- [39] N. Moës, J. Dolbow, and T. Belytschko. A finite element method for crack growth without remeshing. *International Journal for Numerical Methods in Engineering*, 46:131–150, 1999. doi:10.1002/(SICI)1097-0207(19990910)46:1<131::AID-NME726>3.0.CO;2-J.
- [40] R. Müller. *Numerische Simulation dendritischen Kristallwachstums*. PhD thesis, Otto-von-Guericke-Universität Magdeburg, 2005. Available from: <http://diglib.uni-magdeburg.de/Dissertationen/2005/ruemueller.pdf>.
- [41] S. Osher and C.-W. Shu. High-order essentially nonoscillatory schemes for Hamilton-Jacobi equations. *SIAM Journal on Numerical Analysis*, 28(4):907–922, 1991. doi:10.1137/0728049.
- [42] D. Peng, B. Merriman, H. Zhao, S. Osher, and M. Kang. A PDE-based fast local level set method. *Journal of Computational Physics*, 155:410–438, 1999. doi:10.1006/jcph.1999.6345.
- [43] A. Schmidt. *Die Berechnung dreidimensionaler Dendriten mit Finiten Elementen*. PhD thesis, Universität Freiburg, 1993. http://www.mathematik.uni-freiburg.de/IAM/homepages/alfred/paper_diss.html.
- [44] J. A. Sethian and J. Strain. Crystal growth and dendritic solidification. *Journal of Computational Physics*, 98(2):231–253, 1992. doi:10.1016/0021-9991(92)90140-T.
- [45] J. A. Sethian and A. Vladimirovsky. Fast methods for the eikonal and related Hamilton-Jacobi equations on unstructured meshes. *Proceedings of the National Academy of Sciences of the United States of America*, 97:5699–5703, 2000. doi:10.1073/pnas.090060097.
- [46] J. Stefan. Über die Theorie der Eisbildung, insbesondere über die Eisbildung im Polarmeere. *Sitzungsberichte der Kaiserlichen Akademie der Wissenschaften, Wien, Abteilung 2*, 98:965–983, 1889.
- [47] J. M. Sullivan Jr., D. R. Lynch, and K. O’Neill. Finite element simulation of planar instabilities during solidification of an undercooled melt. *Journal of Computational Physics*, 69(1):81–111, 1987. doi:10.1016/0021-9991(87)90157-4.
- [48] V. R. Voller, C. R. Swaminathan, and B. G. Thomas. Fixed grid techniques for phase change problems: A review. *International Journal for Numerical Methods in Engineering*, 30:875–898, 1990. doi:10.1002/nme.1620300419.
- [49] N. Zabaras, B. Ganapathysubramanian, and L. Tan. Modelling dendritic solidification with melt convection using the extended finite element method. *Journal of Computational Physics*, 218:200–227, 2006. doi:10.1016/j.jcp.2006.02.002.

CHEMNITZ UNIVERSITY OF TECHNOLOGY, FACULTY OF MATHEMATICS, D-09107 CHEMNITZ, GERMANY

E-mail address: `martin.bernauer@mathematik.tu-chemnitz.de`

URL: `http://www.tu-chemnitz.de/mathematik/part_dgl/people/bernauer/index.en.php`

CHEMNITZ UNIVERSITY OF TECHNOLOGY, FACULTY OF MATHEMATICS, D-09107 CHEMNITZ, GERMANY

E-mail address: `roland.herzog@mathematik.tu-chemnitz.de`

URL: `http://www.tu-chemnitz.de/herzog`