Computing (2013) 95 (Suppl 1):S553–S572
DOI 10.1007/s00607-012-0247-3

Application of the level-set method to a mixed-mode driven Stefan problem in 2*D* and 3*D*

D. den Ouden · A. Segal · F. J. Vermolen · L. Zhao · C. Vuik · J. Sietsma

Received: 28 September 2012 / Accepted: 20 November 2012 / Published online: 2 December 2012

Abstract This study focusses on the growth of small precipitates within a matrix phase (see also den Ouden et al., Comput Mater Sci 50:2397–2410, 2011). The growth of a precipitate is assumed to be affected by the concentration gradients of a single chemical element within the matrix phase at the precipitate/matrix boundary and by an interface reaction, resulting into a mixed-mode formulation of the boundary condition on the precipitate/matrix interface. Within the matrix phase we assume that the standard diffusion equation applies to the concentration of the considered chemical element. The formulated Stefan problem is solved using a level-set method (J Comput Phys 79:12–49, 1988) by introducing a time-dependent signed-distance function for which the zero level-set describes the precipitate/matrix interface. All appearing hyperbolic partial differential equations are discretised by the use of Streamline-Upwind Petrov–Galerkin finite-element techniques (Comput Vis Sci 3:93–101, 2000). All level-set related equations are solved on a background mesh, which is enriched with interface nodes located on the zero-level of the signed-distance function. The diffusion equation is solved in the diffusive phase. Simulations with the implemented methods for the

In memory of our late colleague Jeroen Colijn.

D. den Ouden (✉) · L. Zhao
Materials innovation institute, Mekelweg 2,
2628 CD Delft, The Netherlands
e-mail: d.denouden@m2i.nl

A. Segal · F. J. Vermolen · C. Vuik
Delft University of Technology, DIAM, Mekelweg 4,
2628 CD Delft, The Netherlands

J. Sietsma
Delft University of Technology, MS&E, Mekelweg 2,
2628 CD Delft, The Netherlands

growth of various precipitate shapes show that the methods employed in this study correctly capture the evolution of the precipitate/matrix interface including topological changes. At the final stage of growth/dissolution physical equilibrium is attained. We also observe that our solutions show mass conservation as the time-step and element-size tend to zero.

Keywords Level-set · Finite-element · Stefan problem · Diffusion · Precipitate growth

Mathematics Subject Classification (2000) 35K57 · 65M60 · 74A50 · 74N25 · 80A22 · 82C24

1 Introduction

Metalworking of alloys is a widely used and complex process that involves several physical phenomena, such as dislocation movement, grain recrystallisation and secondary phase precipitation [14], that influence the workability and applicability of the object. These phenomena and their influences have been studied and documented using mainly "trial and error"-based experiments and by experience. These experimentally obtained results could be verified by an analytical investigation of the studied aspects and hence they improve the understanding of the physical behaviour of alloys during metalworking.

Several models that describe the nucleation and growth of precipitates in alloys have been proposed in the last decades, see for example [5,8,12,15,19,23]. A drawback of the majority of these models is the assumption of a predefined, such as spherical, shape for the precipitates. These precipitate geometries include cuboids, needles, plates and other shapes. Modelling the growth and dissolution of arbitrarily-shaped precipitates is the subject of our study. We also focus on the effect of various physical properties on the growth/dissolution process, such as interfacial tension and reaction and diffusion rates.

Various models exist in literature for the description of the growth of precipitates within a diffusive domain. The phase field method (See for example [6]) considers the interface between the two phases as a smooth interface, where the evolution of the interface is governed by the minimization of the free energy of the entire system. The Kampmann–Wagner-Numerical method, introduced in [8], models a complete statistical distribution of precipitates, in which the precipitates are assumed spherical and growth is modelled using a Zener approximation [24]. The Monte Carlo method has also been used to model the nucleation and growth of precipitates [19], in which the behaviour of the precipitates follows from the behaviour of single atoms. In [23] the moving mesh method is used to model the topological changes associated with the growth and dissolution of precipitates using an explicit representation of the interface. Finally the level-set method [10] has been applied, which uses a sharp interface representation and which can handle topological changes of the precipitate with ease.

In this paper we will first give a short overview of the moving boundary model used for the description of growth and dissolution in a binary alloy. Subsequently, we discuss some numerical aspects of our approach and finally we discuss the results

obtained with our method with regard to convergence and the dependence of the model on the studied physical properties.

2 The Stefan problem

In this section, we describe the models used to simulate the dissolution and growth of a precipitate in a diffusive phase driven by a single chemical element. The level-set method introduced in [10] is used for the description of the boundary, whereas diffusive and reactional fluxes physically determine the movement of the interface.

2.1 Evolution of the concentration

Our model is based on the original Stefan problem described by Stefan in 1890 (see [4]). Consider a diffusive phase $\Omega_D(t)$ in which a precipitate $\Omega_P(t)$ has nucleated at some point. Here $\Omega_D(t)$ and $\Omega_P(t)$ are open domains. Let $\Gamma(t)$ denote the interface between the two phases, which represents the moving boundary in our model. Let the concentration c_p within the precipitate $\Omega_P(t)$ be fixed and assume the concentration $c(\mathbf{x}, t)$ within the diffusive phase $\Omega_D(t)$ to be described by the standard diffusion equation

$$\frac{\partial c}{\partial t}(\mathbf{x}, t) = \nabla \cdot (D(\mathbf{x}, t)\nabla c(\mathbf{x}, t)), \quad \text{for } \mathbf{x} \in \Omega_D(t),\ t > 0, \tag{1}$$

where D is the diffusivity of the diffusing chemical element. At the outer boundary of Ω_D, i.e. $\partial\Omega_D(t) \setminus \Gamma(t)$, we assume a no-flux condition, which results into an homogeneous Neumann boundary condition. Furthermore, let $\Omega(t)$ be the open domain defined by

$$\Omega(t) = (\Omega_D(t) \cup \Omega_P(t)) \setminus \Gamma(t). \tag{2}$$

At the precipitate/matrix interface $\Gamma(t)$ three physical phenomena occur in sequence during dissolution:

1. Detachment of atoms from the lattice structure of the precipitate phase;
2. Crossing of atoms from within the precipitate into the matrix;
3. Long-range diffusion of atoms into the matrix.

These phenomena occur during growth in the reverse order. In both cases all phenomena put restrictions on the speed at which the interface can move. Many models assume that the diffusive phenomenon is rate-limiting and hence neglect the possible influence of the reaction at the interface given by the first two phenomena. In [22] it has been shown for a plate-like precipitate that the interface reaction can have a significant impact on the dissolution kinetics. Similar to the model in [22] we model the flux of atoms $J_r(\mathbf{x}, t)$ across the interface by a first-order reaction:

$$J_r(\mathbf{x}, t) = K(\mathbf{x}, t)\,(c_s(\mathbf{x}, t) - c(\mathbf{x}, t)), \quad \text{for } \mathbf{x} \in \Gamma(t), t > 0. \tag{3}$$

The flux at the interface within the diffusive phase $\Omega_D(t)$ consists of two parts, the flux $J_m(\mathbf{x}, t)$ due to movement of the interface itself

$$J_m(\mathbf{x}, t) = c(\mathbf{x}, t) v_n(\mathbf{x}, t), \quad \text{for } \mathbf{x} \in \Gamma(t), t > 0, \tag{4}$$

and the diffusive flux $J_d(\mathbf{x}, t)$

$$J_d(\mathbf{x}, t) = D(\mathbf{x}, t) \frac{\partial c}{\partial n}(\mathbf{x}, t), \quad \text{for } \mathbf{x} \in \Gamma(t), t > 0. \tag{5}$$

In these definitions $K(\mathbf{x}, t)$ is the interface-reaction speed, $c_s(\mathbf{x}, t)$ the local equilibrium concentration and $v_n(\mathbf{x}, t)$ denotes the speed of the interface in the outward normal direction $\mathbf{n}(\mathbf{x}, t)$ from the domain $\Omega_D(t)$ at $\Gamma(t)$. Combining Eqs. (3), (4) and (5), we arrive at the flux boundary condition

$$K(\mathbf{x}, t) \left(c_s(\mathbf{x}, t) - c(\mathbf{x}, t)\right) = D(\mathbf{x}, t) \frac{\partial c}{\partial n}(\mathbf{x}, t) + c(\mathbf{x}, t) v_n(\mathbf{x}, t), \quad \text{for } \mathbf{x} \in \Gamma(t), t > 0. \tag{6}$$

As we have introduced a new unknown, the interface velocity $v_n(\mathbf{x}, t)$, we must complete our definition by another boundary condition on $\Gamma(t)$. Using a mass balance on a growing/dissolving precipitate, we arrive at the familiar Stefan condition

$$c_p v_n(\mathbf{x}, t) = D \frac{\partial c}{\partial n}(\mathbf{x}, t) + c(\mathbf{x}, t) v_n(\mathbf{x}, t), \quad \text{for } \mathbf{x} \in \Gamma(t), t > 0. \tag{7}$$

By substracting Eq. (7) from Eq. (6) we see that the interface velocity $v_n(\mathbf{x}, t)$ is given by

$$v_n(\mathbf{x}, t) = \frac{K(\mathbf{x}, t)}{c_p} \left(c_s(\mathbf{x}, t) - c(\mathbf{x}, t)\right), \quad \text{for } \mathbf{x} \in \Gamma(t), t > 0, \tag{8}$$

Substituting the above result in either Eq. (6) or Eq. (7), yields that the normal diffusive flux at the interface is given by

$$D(\mathbf{x}, t) \frac{\partial c}{\partial n}(\mathbf{x}, t) = \frac{K(\mathbf{x}, t)}{c_p} \left(c_s(\mathbf{x}, t) - c(\mathbf{x}, t)\right) \left(c_p - c(\mathbf{x}, t)\right), \quad \text{for } \mathbf{x} \in \Gamma(t), t > 0. \tag{9}$$

From Eq. (8) we see that the determination of the interface velocity does not involve computing the normal diffusive fluxes at the interface, as opposed to the model used in for example [7]. A drawback is the introduction of a nonlinear boundary condition on $\Gamma(t)$ for the diffusion problem, in contrast to the simpler Dirichlet condition

$$c(\mathbf{x}, t) = c_s(\mathbf{x}, t), \quad \text{for } \mathbf{x} \in \Gamma(t), t > 0, \tag{10}$$

used in for example [7]. Inspection of Eq. (8) shows that if the value of K is large, we will have fast dissolution/growth of the precipitate, indicating diffusion controlled kinetics, whereas a lower value of K leads to slow dissolution/growth, indicating reaction-controlled kinetics.

In this paper we assume that the solubility of the considered element at the precipitate/matrix interface inside the diffusive phase, $c_s(\mathbf{x}, t)$, is known and modelled using the Gibbs–Thomson effect [13,14]

$$c_s(\mathbf{x}, t) = c_s^{\infty}(t)\exp\left(\zeta\kappa(\mathbf{x}, t)\right), \tag{11}$$

where $c_s^{\infty}(t)$ is the solubility of the considered element, ζ a positive physical factor and $\kappa(\mathbf{x}, t)$ the sum of the principle curvatures of the interface $\Gamma(t)$. The solubility $c_s^{\infty}(t)$ can be derived from thermodynamic databases such as ThermoCalc [1]. The parameter ζ is defined as

$$\zeta = \frac{\gamma V_m}{R_g T}, \tag{12}$$

with γ the interface energy, V_m the molar volume of the precipitate, R_g the gas constant and T the temperature. For a sphere the derivation of Eq. (11) can be found in [13], leading to $\kappa = 2/R$ where R is the radius of the sphere. By Eq. (11) the equilibrium concentration c_s^{∞} increases for locally convex interfaces, which have positive curvature, and decreases for locally concave interfaces, which have negative curvature. This amplification/dampening will cause the precipitate to grow/dissolve to the configuration with the lowest overall surface tension, i.e. the total energy of the system will be minimised.

2.2 The level-set method

In [10] the level-set method was introduced, which captures the motion of an arbitrary interface using a signed-distance function. As we are interested in the movement of the precipitate/matrix interface $\Gamma(t)$ we will employ this method to describe the interface. Define the signed-distance function $\phi(\mathbf{x}, t)$ by

$$\phi(\mathbf{x}, t) = \begin{cases} + \min\limits_{\mathbf{y}\in\Gamma(t)} \|\mathbf{y} - \mathbf{x}\|_2 & \text{if } \mathbf{x} \in \overline{\Omega}_P(t) \setminus \Gamma(t), \\ 0 & \text{if } \mathbf{x} \in \Gamma(t), \\ - \min\limits_{\mathbf{y}\in\Gamma(t)} \|\mathbf{y} - \mathbf{x}\|_2 & \text{if } \mathbf{x} \in \overline{\Omega}_D(t) \setminus \Gamma(t). \end{cases} \tag{13}$$

Using this definition of $\phi(\mathbf{x}, t)$, the normal $\mathbf{n}(\mathbf{x}, t)$ of the interface $\Gamma(t)$ can be calculated directly from $\phi(\mathbf{x}, t)$ by

$$\mathbf{n}(\mathbf{x}, t) = \frac{\nabla\phi(\mathbf{x}, t)}{\|\nabla\phi(\mathbf{x}, t)\|_2}, \tag{14}$$

which will point into $\Omega_P(t)$ and out of $\Omega_D(t)$ due to our chosen orientation of $\phi(\mathbf{x}, t)$. Equation (11) uses the sum of the principal curvatures $\kappa(\mathbf{x}, t)$, which can be derived from $\phi(\mathbf{x}, t)$, similar to the normal $\mathbf{n}(\mathbf{x}, t)$, using

$$\kappa(\mathbf{x}, t) = -\nabla \cdot \frac{\nabla\phi(\mathbf{x}, t)}{\|\nabla\phi(\mathbf{x}, t)\|_2}. \tag{15}$$

Note that we introduced a minus sign in the definition of $\kappa(\mathbf{x}, t)$ as opposed to the definition introduced in [10] due to the orientation of our signed-distance function $\phi(\mathbf{x}, t)$. The movement of the interface is captured by evolving the signed-distance function $\phi(\mathbf{x}, t)$ using the convection equation [11]

$$\frac{\partial\phi}{\partial t}(\mathbf{x}, t) + v_n^{\text{ex}}(\mathbf{x}, t)\|\nabla\phi(\mathbf{x}, t)\|_2 = 0, \quad \text{for } \mathbf{x} \in \Omega, t > 0. \tag{16}$$

The velocity $v_n^{\text{ex}}(\mathbf{x}, t)$ in Eq. (16) is an extension of the normal interface velocity $v_n(\mathbf{x}, t)$ given by Eq. (8). Various approaches exist for this extension routine, see for example [2,11], which are based on the evolution of convection equation in pseudo-time and a Dirichlet condition on the interface. We however choose to use a simpler approach to reduce the computational effort needed for this extension. We solve the Laplace equation

$$\Delta v_n^{\text{ex}}(\mathbf{x}, t) = 0, \quad \text{for } \mathbf{x} \in \Omega(t), t > 0, \tag{17}$$

with a homogeneous Neumann boundary condition on $\partial\Omega_D(t) \setminus \Gamma(t)$ and the Dirichlet condition

$$v_n^{\text{ex}}(\mathbf{x}, t) = v_n(\mathbf{x}, t), \quad \text{for } \mathbf{x} \in \Gamma(t), t > 0, \tag{18}$$

where $v_n(\mathbf{x}, t)$ is given by Eq. (8).

In [20] it has been shown that advancing fronts are only captured correctly if the function $\phi(\mathbf{x}, t)$ remains a signed-distance function throughout the simulations. To enforce this constraint, we employ the technique of reinitialisation. Various reinitialisation techniques have been developed which either use a partial differential equation or a front marching technique (see for an overview [18,20]). In this article we will use another approach, which will be explained in the next section.

3 Numerical methodology

Both Eqs. (1) and (16) are solved using finite-element techniques. For Eq. (1) we will employ the standard Galerkin finite-element technique, whereas for Eq. (16) we will use a SUPG finite-element technique, which is in detail discussed in [21]. We have chosen this SUPG stabilization, as the stabilization parameter is dependent on the time-step and the current convection speed and direction of ϕ. No extra artificial parameters need to be chosen in this SUPG technique. We combine this discretisation scheme with the Explicit-Euler time-integration for the signed-distance function $\phi(\mathbf{x}, t)$ and the Implicit-Euler time-integration for the concentration $c(\mathbf{x}, t)$. The extension of the normal velocity with Eq. (17) is done using the standard Galerkin finite-element technique. All finite-element techniques are applied on meshes with linear elements, of which the generation will be discussed next. After this we will discuss the discretisation

of Eq. (1) and the reinitialisation of the level-set function ϕ in more detail. Finally an overview of the entire algorithm will be presented.

3.1 Mesh generation

To solve the problem formulated above, we will generate several meshes in order to deal with the moving interface. The approach taken to construct these meshes in $2D$ is given here, but is readily extended to $3D$ (see for example [7]). First we define a background mesh $\mathcal{T}$ on which we assume a finite-element approximation $\phi(t^n)$ at the discrete time t^n is known. Figure 1a shows such a $2D$ mesh including the zero-level of $\phi(t^n)$ and the classification of the domains. Let $\mathcal{X}$ denote all points from the background mesh $\mathcal{T}$ and $\mathcal{E}$ the set of edges from $\mathcal{T}$.

Upon inspection of Eqs. (17) and (18), we see that in order to determine the velocity $\mathbf{v}_n^{ex}(t^n)$, a Dirichlet boundary condition must be applied on the interface $\Gamma(t^n)$, which is embedded in the domain $\Omega(t)$. Analysis of Eqs. (1)–(9) gives that a restriction of $\mathcal{T}$ to $\Omega_D(t^n)$ and boundary elements that represent the interface $\Gamma(t^n)$ are needed. To introduce these, we will introduce two new meshes, $\mathcal{T}_E^n$ and $\mathcal{T}_D^n$. The background mesh $\mathcal{T}$ is enriched with nodes on the interface $\Gamma(t^n)$ and contains explicitly edges on the interface $\Gamma(t^n)$, giving the enriched mesh $\mathcal{T}_E^n$. The mesh $\mathcal{T}_D^n$ is based on the enriched mesh $\mathcal{T}_E^n$, and only contains those triangles that are located within $\Omega_D(t^n)$.

Let $\mathcal{E}^{\pm} \subset \mathcal{E}$ represent the edges of the background mesh $\mathcal{T}$ over which $\phi(t^n)$ changes sign, as the zero-level of $\phi(t^n)$, and consequently $\Gamma(t^n)$, must be located along such an edge. Using our linear approximation of the level-set function $\phi(\mathbf{x}, t^n)$ on each edge, we can define the local variable τ^e for each edge $e \in \mathcal{E}^{\pm}$ as

$$\tau^e = \frac{-\phi_1^e(t^n)}{\phi_2^e(t^n) - \phi_1^e(t^n)}, \tag{19}$$

where $\phi_i^e(t^n), i = 1, 2$ represent the values of the first and second point of the edge e over which $\phi(t^n)$ changes sign at t^n. Note that $\phi_1(t^n)\phi_2(t^n) < 0$. Using this variable we approximate the location $\mathbf{x}^e$ of $\Gamma(t^n)$ along the edge e as

$$\mathbf{x}^e = (1 - \tau^e)\mathbf{x}_1^e + \tau^e \mathbf{x}_2^e, \tag{20}$$

with $\mathbf{x}_i^e, i = 1, 2$ the global locations of the first and second point of the edge e. If $\tau^e \in (\delta, 1 - \delta)$, where δ is some value between 0 and $\frac{1}{2}$, we add $\mathbf{x}^e$ to the set $\mathcal{X}^{\pm}$, which represents the set[1] of points to be added to the mesh $\mathcal{T}$. If however $\tau^e \leq \delta$ we do not add $\mathbf{x}^e$ to $\mathcal{X}^{\pm}$ but set

[1] The $\pm$ symbol represents the fact that the points in the set $\mathcal{X}^{\pm}$ come from an edge over which $\phi(t^n)$ changes sign.

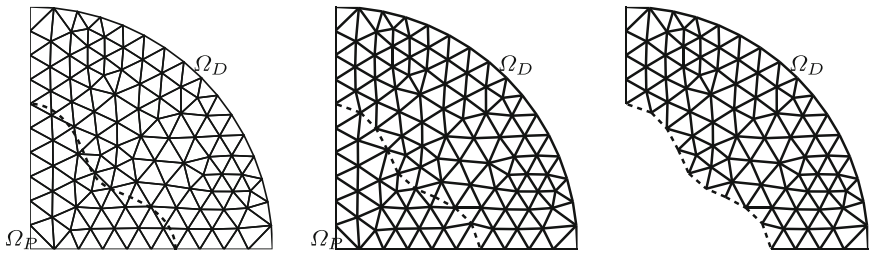


Fig. 1 $2D$ examples of the generated meshes in our approach. The *dashed curve* corresponds to **a** the given/prescribed interface $\Gamma(t^n)$, **b**, **c** the set of interface edges $\mathcal{E}^0$. The value $\delta = 0.3$ has been used. **a** Background mesh $\mathcal{T}$. **b** Enriched mesh $\mathcal{T}_E^n$. **c** Diffusive phase mesh $\mathcal{T}_D^n$

$$\mathbf{x}_1^e = \mathbf{x}^e \quad \text{and} \quad \phi_1^e = 0, \tag{21}$$

i.e. we shift an original mesh point of the background mesh to the interface $\Gamma(t^n)$, and if $\tau^e \geq 1 - \delta$, we apply the same principle by setting

$$\mathbf{x}_2^e = \mathbf{x}^e \quad \text{and} \quad \phi_2^e = 0. \tag{22}$$

If an original mesh point $\mathbf{x}$ can be shifted to several locations, we choose the interface location $\mathbf{x}^e$ closest to the mesh point $\mathbf{x}$. The next step in construction of the enriched mesh $\mathcal{T}_E^n$ is the construction of the boundary elements. To this end define $\mathcal{X}^0$ as the union of all (possibly shifted) background points $\mathcal{X}$ where $\phi(t^n) = 0$ and all points from $\mathcal{X}^{\pm}$. Next we construct a set of edges $\mathcal{E}^0$ through $\mathcal{X}^0$, to approximate the interface $\Gamma(t^n)$. Finally we construct the enriched mesh $\mathcal{T}_E^n$ on the union $\mathcal{X}_E$ of the (possibly shifted) background points $\mathcal{X}$ and the interface points $\mathcal{X}^{\pm}$, such that the set of edges $\mathcal{E}^0$ is present within the mesh. Figure 1b shows the enriched mesh $\mathcal{T}_E^n$ with the edges $\mathcal{E}^0$ that results from applying this approach to the background mesh $\mathcal{T}$ and interface $\Gamma(t^n)$ of Fig. 1a.

Construction of the mesh $\mathcal{T}_D^n$ can easily be accomplished by noting that $\overline{\Omega}_D(t^n)$ can only contain those points where $\phi(t^n)$ is non-positive due to Eq. (13). So after determination of those points in $\mathcal{X}_E$ where $\phi(t^n)$ is positive, we delete any simplex from $\mathcal{T}_E^n$ that uses at least one of those points. This leads to the diffusive phase mesh $\mathcal{T}_D^n$, which due to the derivation of the enriched mesh $\mathcal{T}_E^n$ will contain boundary elements along $\Gamma(t^n)$. Figure 1c shows the diffusive phase mesh $\mathcal{T}_D^n$ with the edges $\mathcal{E}^0$ that results from restricting the enriched mesh $\mathcal{T}_E^n$ to $\overline{\Omega}_D(t^n)$. In this article we have chosen for the parameter δ the value 0.3.

3.2 Discretisation of Eq. (1)

As mentioned above, we discretise Eq. (1) combined with the boundary conditions in Eqs. (6) and (7) using the standard Galerkin finite-element technique and Implicit-Euler time-integration. The finite-element technique will transform Eq. (1) to a system of nonlinear ordinary differential equations

$$M(t)\frac{\mathrm{d}\mathbf{c}(t)}{\mathrm{d}t} = S(t,\mathbf{c}(t))\mathbf{c}(t) + \mathbf{f}(t,\mathbf{c}(t)), \tag{23}$$

where $\mathbf{c}(t)$ represents the finite-element approximation at time t. The mass matrix $M(t)$, the stiffness matrix $S(t,\mathbf{c}(t))$ and vector $\mathbf{f}(t,\mathbf{c}(t))$ contain integrals over the domain $\Omega_D(t)$ and the interface $\Gamma(t)$. Due to the nonlinearity of Eq. (9), which is used in the derivation of the weak formulation, we have a stiffness matrix $S(t,\mathbf{c}(t))$ and vector $\mathbf{f}(t,\mathbf{c}(t))$ that dependent on the solution vector $\mathbf{c}(t)$. The term

$$\frac{K(\mathbf{x},t)}{c_p}\left(c_p - c(\mathbf{x},t)\right), \tag{24}$$

of Eq. (9) is incorporated in the matrix $S(t,\mathbf{c}(t))$ and the remaining term

$$-\frac{K(\mathbf{x},t)}{c_p}\left(c_p - c(\mathbf{x},t)\right)c_s(\mathbf{x},t), \tag{25}$$

is incorporated in the vector $\mathbf{f}(t,\mathbf{c}(t))$.

Application of Implicit-Euler time-integration on Eq. (23) leads to the matrix-vector equation

$$\left(M(t^{n+1}) - \Delta t\ S(t^{n+1},\mathbf{c}^{n+1})\right)\mathbf{c}^{n+1} = M(t^{n+1})\mathbf{c}^n + \Delta t\ \mathbf{f}(t^{n+1},\mathbf{c}^{n+1}), \tag{26}$$

where the superscript n refers to the evaluation of the variable at the discrete time t^n and Δt is the chosen time step. Under the assumption that the diffusive phase mesh $\mathcal{T}_D^n$ contains N^n points and since the time-consecutive meshes $\mathcal{T}_D^n$ and $\mathcal{T}_D^{n+1}$ can contain different numbers of points, i.e. $N^{n+1} \neq N^n$, we see a possible dimensional mismatch in the matrix-vector product between $M(t^{n+1})$ and $\mathbf{c}^n$.

To overcome this possible dimensional mismatch, we extend the solution $\mathbf{c}^n$ from the mesh $\mathcal{T}_D^n$ to the mesh $\mathcal{T}_D^{n+1}$. For any point $\mathbf{x}_i \in \mathcal{T}_D^{n+1}$, $i \in \{1,\ldots,C^{n+1}\}$ we first find the point $\mathbf{y}_j \in \mathcal{T}_D^n$, $j \in \{1,\ldots,C^n\}$ such that $\|\mathbf{x}_i - \mathbf{y}_j\|_2$ is minimal and define the index-to-index function $I : i \to j$. We then define the extended solution vector $\mathbf{c}_{\mathrm{ex}}^n$ of length C^{n+1} by setting

$$c_{\mathrm{ex},i}^n = c_{I(i)}^n. \tag{27}$$

Note that if a point $\mathbf{x}_i$ originates from the background mesh $\mathcal{T}$ and has not been shifted in both diffusive phase meshes $\mathcal{T}_D^n$ and $\mathcal{T}_D^{n+1}$, we will have $\mathbf{x}_i = \mathbf{y}_{I(i)}$. Furthermore if $\mathcal{T}_D^{n+1} = \mathcal{T}_D^n$, we will have consistency as $\mathbf{x}_i = \mathbf{y}_{I(i)}$. Using the index-to-index function $I(i)$ we can also assign a mesh velocity $\mathbf{v}_i^{\mathrm{mesh}}(t^{n+1})$ to each point $\mathbf{x}_i \in \mathcal{T}_D^{n+1}$, $i = 1,\ldots,N^{n+1}$, which is given by the function

$$\mathbf{v}_i^{\text{mesh}}(t^{n+1}) = \frac{\mathbf{y}_{I(i)} - \mathbf{x}_i}{\Delta t}, \quad i = 1, \ldots, N^{n+1}, \tag{28}$$

where $\mathbf{y}_{I(i)} \in \mathcal{T}_D^n$ for all i.

Using the above extension, we must replace the partial derivative with respect to time in Eq. (1) with a material derivative. The material derivative is defined as

$$\frac{\mathrm{D}c}{\mathrm{D}t}(\mathbf{x}, t) = \frac{\partial c}{\partial t}(\mathbf{x}, t) + \frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t}(t) \cdot \nabla c(\mathbf{x}, t), \tag{29}$$

which transforms Eq. (1) to

$$\frac{\mathrm{D}c}{\mathrm{D}t}(\mathbf{x}, t) = \nabla \cdot (D(\mathbf{x}, t)\nabla c(\mathbf{x}, t)) + \frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t}(t) \cdot \nabla c(\mathbf{x}, t), \quad \text{for } \mathbf{x} \in \Omega_D(t), t > 0. \tag{30}$$

We approximate the velocity of a point $\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t}(t^{n+1})$ with our derived mesh velocity of Eq. (28) based on $\mathcal{T}_D^n$ and $\mathcal{T}_D^{n+1}$. We note that this approach is similar to the approach used in [16].

After application of the standard Galerkin finite-element technique and Implicit-Euler time-integration, we arrive at the natural fixed-point problem

$$\mathbf{c}^{n+1} = A(\mathbf{c}^{n+1})^{-1}\mathbf{g}\left(\mathbf{c}^{n+1}\right), \tag{31}$$

where the matrix A and vector g are defined by

$$A(\mathbf{q}) = M(t^{n+1}) - \Delta t\ S(t^{n+1}, \mathbf{q}), \tag{32}$$

$$\mathbf{g}(\mathbf{q}) = M(t^{n+1})\mathbf{c}_{\text{ex}}^n + \Delta t\ \mathbf{f}(t^{n+1}, \mathbf{q}), \tag{33}$$

and the matrix $S(t^{n+1}, \mathbf{q})$ now also contains the mesh velocity $\mathbf{v}^{\text{mesh}}(t^{n+1})$. We solve this fixed-point problem using the fixed-point iteration

$$\begin{cases} \mathbf{c}_0^{n+1} = \mathbf{c}_{\text{ex}}^n, \\ \mathbf{c}_{p+1}^{n+1} = A(\mathbf{c}_p^{n+1})^{-1}\mathbf{g}\left(\mathbf{c}_p^{n+1}\right), \quad p = 1, 2, 3, \ldots \end{cases} \tag{34}$$

We iterate until the error ϵ has dropped below some predefined tolerance `FPT`, where the error ϵ is given by

$$\epsilon = \frac{\left\| \mathbf{c}_{p+1}^{n+1} - \mathbf{c}_p^{n+1} \right\|_\infty}{\left\| \mathbf{c}_{p+1}^{n+1} \right\|_\infty}. \tag{35}$$

In this article we have chosen $\texttt{FPT} = 10^{-6}$. Using the fixed-point iterations we have observed a linear rate of convergence.

Instead of the fixed-point problem in Eq. (31) we could also have chosen to solve the equivalent root-finding problem

$$A(\mathbf{c}^{n+1})\mathbf{c}^{n+1} - \mathbf{g}\left(\mathbf{c}^{n+1}\right) = 0, \tag{36}$$

with the same matrices and vectors. This problem can be solved by using for example a Newton–Raphson solver. We have not chosen for this approach, as it requires the determination of (an approximation of) the Jacobian, which for our model is complex or leads to multiple function evaluations per iteration.

3.3 Reinitialisation

As shown in [20], it is necessary to maintain the signed-distance property of the level-set function $\phi(\mathbf{x}, t)$ during the computations. To this extent, the technique of reinitialisation has been introduced, which can be done in several ways [18,20]. We compared several methods and we have chosen to use an efficient direct approach in this research, which will be described next.

After obtaining the numerical solution $\tilde{\phi}^{n+1}$ on the enriched mesh $\mathcal{T}_E^n$ by solving Eq. (16), we map this solution back onto the background mesh $\mathcal{T}$. From this numerical solution we generate the enriched mesh $\mathcal{T}_E^{n+1}$, which will contain an explicit linear representation of the interface $\Gamma(t^{n+1})$ consisting of vertices and edges in $2D$ and vertices and triangles in $3D$. This representation can easily be used to recompute the distance between any point in the enriched mesh $\mathcal{T}_E^{n+1}$ and the interface $\Gamma(t^{n+1})$. In the following paragraphs we assume that we are working in $3D$.

Denote by $\mathcal{T}^0$ the set of triangles describing the interface $\Gamma(t^n)$ through the interface points $\mathcal{X}^0$ and by $\mathcal{E}^0$ the edges defined by the triangulation $\mathcal{T}^0$. For any point $\mathbf{x} \in \Omega$, the orthogonal projection on the plane defined by the triangle $\triangle ABC \in \mathcal{T}^0$ is given by

$$\mathbf{x}^p_{\triangle ABC}(\mathbf{x}) = \mathbf{x} - \left((\mathbf{x} - \mathbf{x}_A) \cdot \mathbf{n}_{\triangle ABC}\right) \mathbf{n}_{\triangle ABC}, \tag{37}$$

with $\mathbf{n}_{\triangle ABC}$ a unit normal of the plane and $\mathbf{x}_A$ the coordinates of point A. For the same point $\mathbf{x}$, the projection to the line defined by the edge $\overline{AB} \in \mathcal{E}^0$ is given by

$$\mathbf{x}^p_{\overline{AB}}(\mathbf{x}) = \mathbf{x}_A + (\mathbf{d}_{\overline{AB}} \cdot \mathbf{x})\mathbf{d}_{\overline{AB}}, \tag{38}$$

with $\mathbf{d}_{\overline{AB}}$ the unit direction vector of the line. The trivial projection of $\mathbf{x}$ onto the point A is given by

$$\mathbf{x}^p_A(\mathbf{x}) = \mathbf{x}_A. \tag{39}$$

To find the minimal distance between the point $\mathbf{x}_i \in \mathcal{T}_E^{n+1}$ and the triangulated surface $\Gamma(t^{n+1})$, we project $\mathbf{x}_i$ onto all triangles in $\mathcal{T}^0$, all edges in $\mathcal{E}^0$ and all points $\mathcal{X}^0$. We then obtain a set of projection points, of which we keep all projections that fall on the defining triangle, edge or point. We then set $|\phi_i^{n+1}|$ to the minimum of all

absolute distances and let the sign of ϕ_i^{n+1} be determined by the previous value. This procedure can be described by the following set of equations:

$$\mathcal{T}_p^0 = \left\{ t \in \mathcal{T}^0 \mid \mathbf{x}_t^p(\mathbf{x}_i) \in t \right\}, \quad \mathcal{E}_p^0 = \left\{ e \in \mathcal{E}^0 \mid \mathbf{x}_e^p(\mathbf{x}_i) \in e \right\}, \tag{40}$$

$$d_\triangle = \min_{t \in \mathcal{T}_p^0} \left\| \mathbf{x}_i - \mathbf{x}_t^p(\mathbf{x}_i) \right\|_2, \quad d_/ = \min_{e \in \mathcal{E}_p^0} \left\| \mathbf{x}_i - \mathbf{x}_e^p(\mathbf{x}_i) \right\|_2, \tag{41}$$

$$d_\bullet = \min_{\mathbf{y} \in \mathcal{X}^0} \|\mathbf{x}_i - \mathbf{y}\|_2, \quad \phi_i^{n+1} = \operatorname{sgn}(\tilde{\phi}_i^{n+1}) \min \left\{ d_\triangle, d_/, d_\bullet \right\}. \tag{42}$$

The values $d_\triangle$, $d_/$ and $d_\bullet$ are the optima over respectively the triangles, the edges and the points within $\mathcal{T}^0$. If $\mathcal{T}_p^0$ is equal to the empty set $\emptyset$, we set $d_\triangle = \infty$. Similar, if $\mathcal{E}_p^0 = \emptyset$, we set $d_/ = \infty$.

To reduce the computational effort of this approach, we only apply Eqs. (40)–(42) to all mesh points $\mathbf{x}_i \in \mathcal{X}$ upto the 2-th neighbor of $\mathcal{X}^0$ within $\mathcal{T}_E^{n+1}$. Further, we only apply the above algorithm on those triangles, edges and points of which the points lie at maximum a distance of $2.5h$ from $\mathbf{x}_i$, with h the average mesh coarseness. For $2D$ problems, the approach is similarly obtained by projection on all edges and points representing $\Gamma(t^{n+1})$, acceptance of all projections that fall on the defining edges and points and taking again the minimum over all the accepted distances.

The approach given by Eqs. (40)–(42) gives an exact value for ϕ^{n+1} if the interface $\Gamma(t^{n+1})$ is exactly described by a polyhedron in $3D$ or a polygon in $2D$. If however the surface is not captured exactly by a polyhedron/polygon, we obtain an experimental order of convergence with respect to h close to quadratic over the entire domain Ω. This approach is applicable since no reinitialisation over all of Ω is needed. A benefit is that no termination criterion based on the gradient of ϕ^{n+1} is introduced. A drawback of this approach is the computational effort needed to update ϕ^{n+1}, if one would incorporate mesh points further away from the interface. This can be avoided by exploiting the fact that an update for a grid point is independent of all other grid points, which allows parallelisation of the approach. The computational cost per updated point is of $\mathcal{O}(1)$ in $\mathbb{R}^d$ due to the restriction of the used triangles, edges and points in the update. Due to our limitation of the reinitialisation on a narrow band around the interface $\mathcal{O}(h^{1-d})$ points are updated, leading to an overall computational cost of $\mathcal{O}(h^{1-d})$, which is of lower order than the computational cost associated with the method in [20], which is based on the pseudo-time evolution of a partial differential equation.

3.4 The algorithm

In the previous sections we discussed three major aspects of our numerical solution technique. In Algorithm 1 the approach to solve the problem is given in pseudo-code. One can see at line 10 of Algorithm 1 that we have used the Courant–Friedrichs–Lewy condition (CFL) [3] to limit the time-step Δt. This limitation arises from the discretisation of Eq. (16) with the SUPG finite-element and the application of the Explicit-Euler

time-integration. In this article we have used $\texttt{CFL} = 0.25$. This algorithm is implemented into the commercial software package SEPRAN [17].

Algorithm 1 The algorithm in pseudo-code.

```
Define background mesh T with coarseness h;
Define initial level-set function φ^0 on T;
Generate enriched mesh T_E^0;                          // See Section 3.1.
Generate diffusive phase mesh T_D^0;                   // See Section 3.1.
Define initial concentration c^0 on T_D^0;
Set t = 0;
Set n = 0;
WHILE t < t_end DO
   Calculate v_n^ex on T_E^n;                          // See Equation (17).
   Set Δt = CFL h / max ||v_n^ex||_∞;
   Calculate φ^{n+1} on T_E^n;                         // See Equation (16).
   Map φ^{n+1} back to T;
   Generate enriched mesh T_E^{n+1};                   // See Section 3.1.
   Reinitialise φ^{n+1} on T_E^{n+1};                  // See Section 3.3.
   Generate diffusive phase mesh T_D^{n+1};            // See Section 3.1.
   Calculate c_ex^n;                                   // See Equation (27).
   Calculate v_i^m, i = 1, ..., C^{n+1};               // See Equation (28).
   Set p = 1;
   Set c_p^{n+1} = c_ex^n;
   Set ε = ∞;
   WHILE ε ≥FPT DO
      Calculate c_{p+1}^{n+1};                         // See Equation (34).
      Calculate ε;                                     // See Equation (35).
      Set p = p + 1;
   END WHILE
   Set c^{n+1} = c_p^{n+1};
   Set n = n + 1;
   Set t = t + Δt;
END WHILE
```

4 Computer simulations

4.1 The experimental accuracy: dissolution of planar and circular precipitates

To investigate the experimental accuracy of our method, we simulate the dissolution of a planar precipitate with initial width 0.615 in the computational domain $[0, 1]^n$ for $n = 2, 3$. Within the precipitate the concentration is given by $c_p = 0.45$, whereas the initial concentration in Ω_D is given by $c_0 = 0.3$ and the equilibrium concentration by $c_s^\infty = 0.33$. We take $\zeta = 0$ to exclude curvature effects and assume $D = 1$ and $K = 10^3$, leading to diffusion-controlled dissolution. We take a regular background mesh $\mathcal{T}$ with point spacing $h = 1/N$ for several values of N. A simple mass balance gives that under the current settings the equilibrium width of the precipitate is given by 0.51875. We simulate until $t_{\text{end}} = 3$.

We also simulate the dissolution of a circular precipitate with initial radius 0.615 on the unit circle in $2D$. We take the same values for the physical and numerical constants as for the planar dissolution. We discretize the computation domain using linear triangles with coarseness $h = 1/N$ for several values of N. Under the assumption that

Table 1 Experimental accuracy of dissolving a planar and circular precipitate in $2D$

N	Planar			Circular		
	Mass error	Width error	Deviation	Mass error	Radius error	Deviation
16	2.5187×10^{-2}	1.5869×10^{-1}	1.3010×10^{-4}	2.9987×10^{-2}	2.2515×10^{-1}	1.9309×10^{-3}
32	1.4130×10^{-2}	8.9000×10^{-2}	8.1082×10^{-5}	1.7354×10^{-2}	1.2327×10^{-1}	5.4712×10^{-4}
64	7.6502×10^{-3}	4.8202×10^{-2}	5.3468×10^{-5}	9.4015×10^{-3}	6.4816×10^{-2}	1.9345×10^{-4}
128	4.2333×10^{-3}	2.6672×10^{-2}	1.8035×10^{-5}	5.1214×10^{-3}	3.4774×10^{-2}	7.8181×10^{-5}
256	2.3582×10^{-3}	1.4856×10^{-2}	7.8939×10^{-6}	2.8028×10^{-3}	1.8861×10^{-2}	1.3807×10^{-5}
512	1.2236×10^{-3}	7.7096×10^{-3}	3.5917×10^{-6}			

Table 2 Experimental accuracy of dissolving a planar precipitate in $3D$

N	Mass error	Width error	Deviation
8	4.3963×10^{-2}	2.7683×10^{-1}	1.3539×10^{-3}
16	2.4420×10^{-2}	1.5385×10^{-1}	2.6967×10^{-4}
32	1.4304×10^{-2}	9.0121×10^{-2}	6.2047×10^{-5}
64	7.6492×10^{-3}	4.8194×10^{-2}	3.1252×10^{-5}

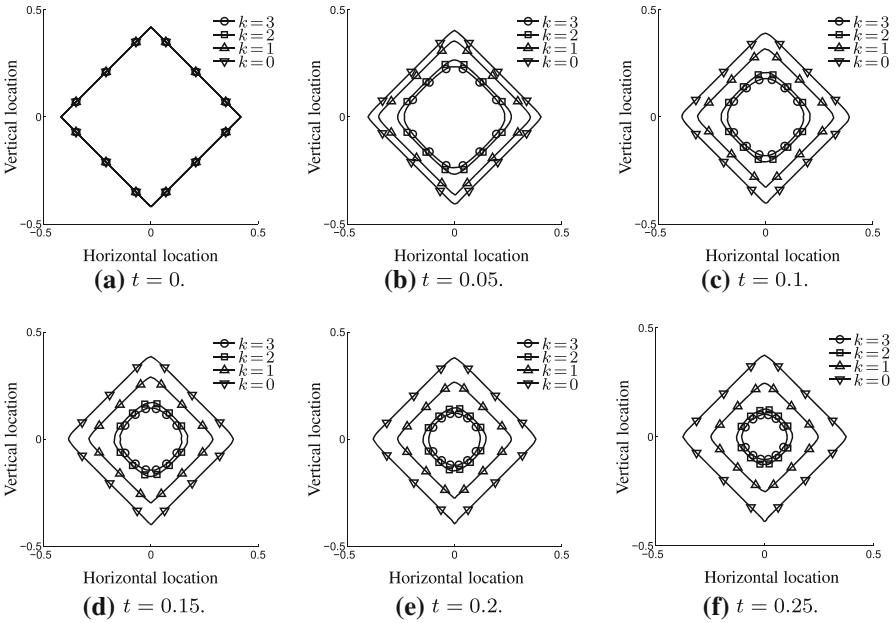
the precipitate stays circular, a mass balance gives that the equilibrium radius equals 0.47200. We again simulate until $t_{\text{end}} = 3$.

Tables 1 and 2 show the results obtained for the dissolution of the planar and circular precipitates. The mass error is defined as the error in the total mass at $t = t_{\text{end}}$ relative to the initial total mass. The width/radius error is defined as the error in the mean width/radius at $t = t_{\text{end}}$ relative to the equilibrium width/radius. The columns labelled "deviation" represent the standard deviation of the width/radius at $t = t_{\text{end}}$.

The errors in Tables 1 and 2 indicate that Algorithm 1 is first-order accurate in h. This is supported by the notion that we have used first-order accurate time-integration techniques and that h and Δt are linearly coupled by the CFL condition. We furthermore see from the deviations given in Tables 1 and 2 that for the planar precipitate there is good convergence in the spread of the points around the mean width, indicating that the shape of the precipitate is correctly preserved. The last column of Table 1 gives the deviation of the radius of the circular precipitate. Here we see that we also have conservation of the circular shape, although we do not obtain the same convergence as for the planar precipitates. We think that this originates from the use of an irregular mesh for the simulation of the circular particle, which decreases the control over the element size in the vicinity of the interface.

4.2 Mixed-mode dissolution

To investigate the balance between the reactive and diffusive fluxes, we simulate the dissolution of a square precipitate with initial diagonal 0.42 in the computational



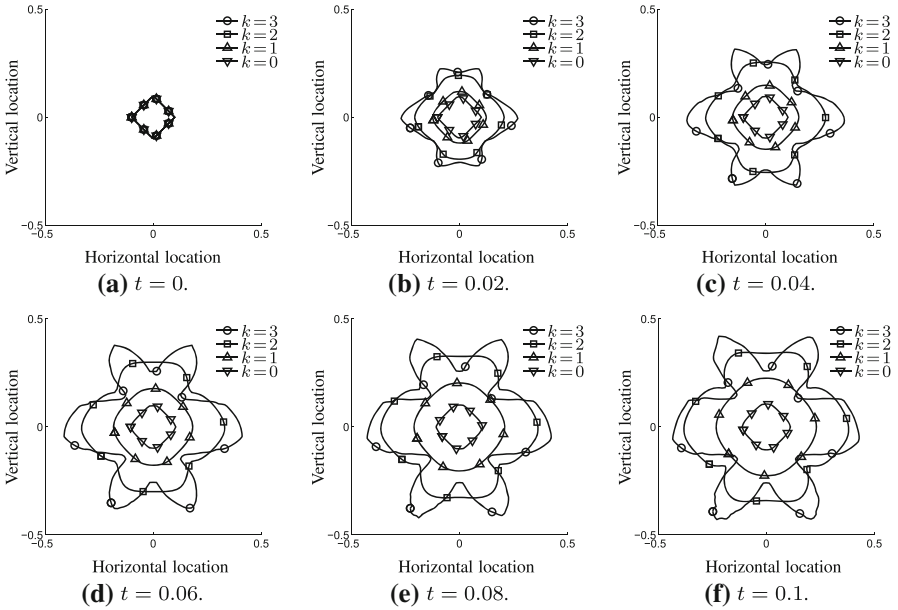
(a) $t = 0.$ **(b)** $t = 0.05.$ **(c)** $t = 0.1.$

(d) $t = 0.15.$ **(e)** $t = 0.2.$ **(f)** $t = 0.25.$

Fig. 2 $2D$ mixed-mode dissolution results for the values of $K = 10^k, k \in \{0, 1, 2, 3\}$ at 6 simulation times, where $\zeta = 0$ and $D = 1$

domain $[-\frac{1}{2}, \frac{1}{2}]^2$. The center of the precipitate is located at the origin $(0, 0)$ and is rotated over 45°. Within the precipitate the concentration is given by $c_p = 0.45$, whereas the initial concentration in Ω_D is given by $c_0 = 0.3$ and the equilibrium concentration by $c_s^\infty = 0.35$. We take $\zeta = 0$ to exclude curvature effects and take $D = 1$. We take an irregular background mesh $\mathcal{T}$ with element size $h = 1/64$. We simulate until $t_{\text{end}} = 0.25$. For the value of the interface-reaction speed K we take $K = 10^k, k \in \{0, 1, 2, 3\}$.

Figure 2 shows the results obtained with Algorithm 1 for the various values of K at $t \in \{0, 0.05, 0.1, 0.15, 0.2, 0.25\}$. These results show two effects of the value of K on the dissolution kinetics of the precipitate. First we clearly see that if K decreases in magnitude, the overall dissolution rate of the precipitate slows down, resulting in larger precipitates at $t = 0.25$ for lower values of K. This indicates that for lower values of K the reaction at the interface becomes limiting for the dissolution kinetics, which agrees with what have already shown in the analysis of the model. Secondly, the shape of the precipitate during dissolution is influenced by the magnitude of K, where a smaller K preserves the initial shape and a larger K causes rounding of the corners of the precipitate. This is a direct consequence of the fact that diffusion is the rate-determining step if K is large. Diffusion processes generally smoothen solutions and thereby make the interface smooth. This means that if precipitates observed in physical experiments maintain sharp corners during dissolution, these precipitates could be described best by a mixed-mode model with a low value of K.



(a) $t = 0.$ **(b)** $t = 0.02.$ **(c)** $t = 0.04.$

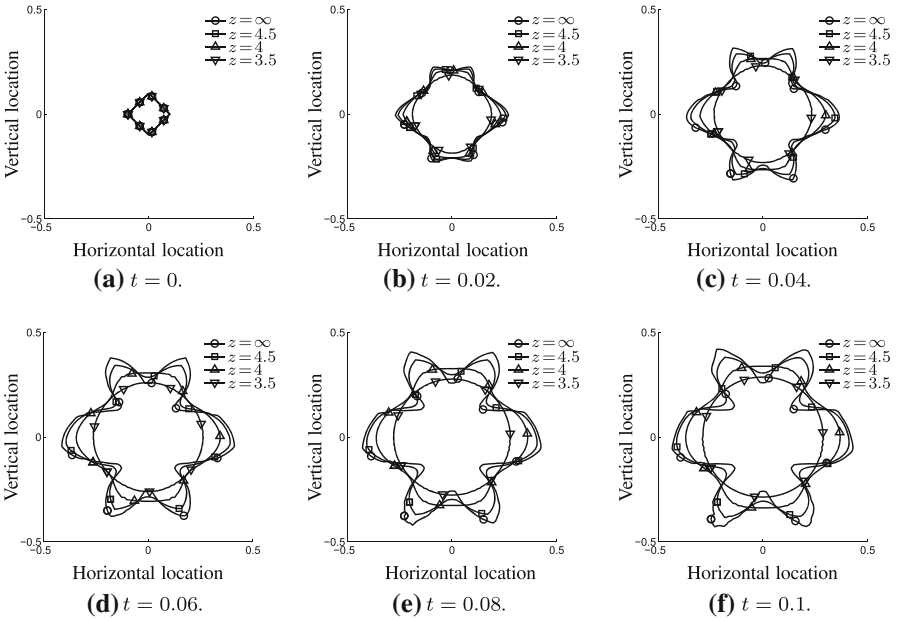
(d) $t = 0.06.$ **(e)** $t = 0.08.$ **(f)** $t = 0.1.$

Fig. 3 $2D$ mixed-mode growth results for the values of $K = 10^k$, $k = 0, 1, 2, 3$ at six simulation times, where $\zeta = 0$ and $D = 1$

4.3 Mixed-mode growth

Mullins and Sekerka [9] have shown that the growth of precipitate is an unstable process if perturbations of the interface are present. It was furthermore assumed that Eq. (10) holds instead of Eq. (7). To investigate these instabilities in our mixed-mode model, we simulate the growth of a square precipitate with initial diagonal 0.1 in the computational domain $[-\frac{1}{2}, \frac{1}{2}]^2$. The center of the precipitate is located at the origin $(0, 0)$ and is rotated over 45°. Within the precipitate the concentration is given by $c_p = 0.45$, whereas the initial concentration in Ω_D is given by $c_0 = 0.3$ and the equilibrium concentration by $c_s^\infty = 0.2$. We take $\zeta = 0$ to exclude curvature effects and assume $D = 1$. We take an irregular background mesh $\mathcal{T}$ with coarseness $h = 1/64$. We simulate until $t_{\text{end}} = 0.1$. For the value of the interface-reaction speed K we take $K = 10^k$, $k = 0, 1, 2, 3$.

Figure 3 shows the results obtained with Algorithm 1 for the various values of K at $t \in \{0, 0.02, 0.04, 0.06, 0.08, 0.1\}$. Similar to the results shown in Fig. 2 for the dissolution of the square particle, the value of K has two influences on the growth kinetics. First the precipitate grows slower for smaller values of K, as expected from Eq. (8). Secondly higher values of K cause instabilities to occur in the shape of the interface. As a high value of K corresponds with a diffusion-controlled process, Eq. (7) could be replaced by Eq. (10). This indicates that our model reproduces the analytical instabilities shown in [9] correctly. We believe that the initial instabilities that are amplified during the simulation are caused by small numerical fluctuations in the concentrations at the interface which are related to the topology of the mesh.



(a) $t = 0$. **(b)** $t = 0.02$. **(c)** $t = 0.04$.

(d) $t = 0.06$. **(e)** $t = 0.08$. **(f)** $t = 0.1$.

Fig. 4 $2D$ mixed-mode growth results for the values of $\zeta = 10^{-z}$, $z = \infty, -4.5, -4, -3.5$ at 6 simulation times, where $K = 10^3$ and $D = 1$

4.4 Curvature-influenced growth

In [9], it has also been shown that if the curvature of a precipitate is taken into account during growth, the perturbations should cancel out. To investigate this effect in the mixed-mode model, we again simulate the precipitate from Sect. 4.3, but now take $K = 1,000$ and take $\zeta = 0, 10^{-4.5}, 10^{-4}, 10^{-3.5}$. All other settings are kept the same.

Figure 4 shows the results obtained with Algorithm 1 for the various values of ζ at $t \in \{0, 0.02, 0.04, 0.06, 0.08, 0.1\}$. Similar to the results shown in Figs. 2 and 3, the value of ζ has a significant influence on the growth kinetics. One can see that the amplification of the perturbations reduces significantly for higher values of ζ, thereby suppressing the creation of concave interface parts. If however concave interface parts occur, these grow faster for higher values of ζ, thereby reducing the concavity. Furthermore, we see that for larger values of ζ, there is no shape preservation, but rounding of the precipitate. This indicates that a minimal value of ζ could exist such that the shape of the precipitate stays convex. This minimal value may depend on the physical and geometrical application.

4.5 Precipitate breakup

During dissolution of precipitates might break into several pieces due to the geometry of the initial precipitate. To test the numerical scheme on handling with these topological changes, we have modelled the dissolution of a dumbbell-shaped precipitate,

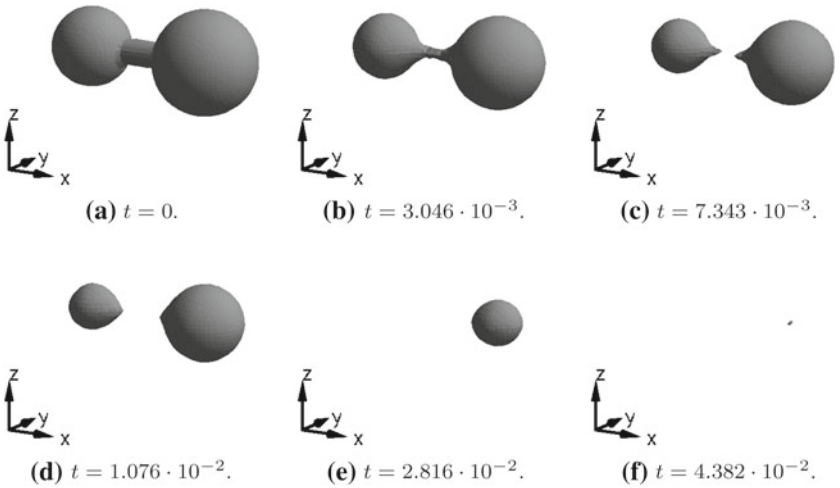


Fig. 5 $3D$ dissolution of a dumbbell-shaped precipitate at six simulation times, where $K = 10^3$, $\zeta = 0$ and $D = \sqrt{2}$

consisting of a sphere with radius 0.15 and center $(-0.275, 0, 0)$, a sphere with radius 0.2 and center $(0.21, 0, 0)$ and a connecting cylinder with radius 0.05 with the axis aligned with the x-axis. The computational domain is $[-\frac{1}{2}, \frac{1}{2}]^3$, the diffusivity $D = \sqrt{2}$, the initial concentration $c_0 = 0$, the precipitate concentration c_p, the solubility $c_s^\infty = 1$, the interface reaction speed $K = 10^3$ and the interfacial parameter $\zeta = 0$. We take a regular mesh with point spacing $1/64$. We simulate until full dissolution is obtained. Figure 5 shows the results of these simulations at 6 discrete times.

The dissolution process as shown in Fig. 5 clearly shows the ability of the level-set method to handle topology changes, as it does not discriminate between one, two or more precipitates, but only considers a sharp interface between two phases. The breakup of the particle is handled without any trouble. Further our algorithm, as presented in Algorithm 1, has no problems with the cusps created along the x-axis at the moment of dissolution of the connecting cylinder.

5 Discussion and conclusion

We described a physical model for the mixed-mode dissolution and growth of precipitates within a diffusive phase. The kinetics governing growth and dissolution are influenced by the concentration gradient of a single chemical element near the precipitate interface and an interface reaction. The model also incorporates the local curvature of the interface and the influence of this curvature on the equilibrium between the precipitate and the diffusive phase. The shape and topology of the precipitate are modelled using the level-set method [10], where the moving interface is treated as sharp.

Using the current level-set function a background triangulation is transformed to two new meshes, which contains an explicit representation of the interface between

the precipitate and the diffusive phase. On these meshes the appearing hyperbolic partial differential equations have been discretised using Streamline-Upwind Petrov–Galerkin finite-element techniques [21] and a first-order time integration. The needed reinitialisation of the level-set function [18,20] has been done using an efficient direct approach based on the explicit representation of the interface.

It has been shown that our model and accompanying discretisation achieves first order accuracy with respect to the mesh coarseness for the dissolution of planar and circular precipitates, which agrees with the theory due to the coupling of the mesh coarseness and the time step by the CFL condition [3].

The introduced interface reaction speed K has been shown to influence the mixed-mode dissolution of precipitates, where higher values of K causes diffusion-limited dissolution and loss of the topological shape, whereas lower values of K cause reaction-limited dissolution and topological shape preservation.

Our model is able to reproduce the analytical instability which was studied by Mullins and Sekerka in [9], where the onset of the initial instabilities of the interface result from local fluctuations of the numerical error from the finite-element discretisation. These instabilities can be suppressed by two different parameters, the interface reaction speed K and the interfacial parameter ζ. The first parameter causes dampening of the instabilities using the interface reaction and has, similar to mixed-mode dissolution, a shape preserving effect. The parameter ζ introduces indirectly via Eq. (11) a curvature-driven growth, where concave regions grow faster than convex regions. This curvature driven growth causes rounding of the precipitate.

It has also been shown that the topological changes associated with the dissolution of a dumbbell-shaped precipitate are captured accurately with ease by using the level-set method. The cusps appearing on the interface during the breakup of the precipitate in two smaller precipitate are handled correctly, as is the full dissolution of one of the two remaining precipitates after breakup.

Acknowledgments This research was carried out under the project number M41.5.09341 in the framework of the Research Program of the Materials innovation institute M2i (http://www.m2i.nl).

References

1. Andersson JO, Helander T, Höglund L, Shi P, Sundman B (2002) Thermo-Calc and DICTRA, computational tools for materials science. Calphad 26:273–312
2. Chen S, Merriman B, Osher S, Smereka P (1997) A simple level set method for solving Stefan problems. J Comput Phys 135:8–29
3. Courant R, Friedrichs K, Lewy H (1928) Über die partiellen Differenzengleichungen der mathematischen Physik. Math Ann 100:32–74
4. Crank J (1984) Free and moving boundary problems. Clarendon Press, Oxford
5. Deschamps A, Brechet Y (1999) Influence of predeformation and ageing of an Al–Zn–Mg Alloy—II. Modeling of precipitation kinetics and yield stress. Acta Mater 47:293–305
6. Hu S, Henager CH Jr (2009) Phase-field simulations of Te-precipitate morphology and evolution kinetics in Te-rich CdTe crystals. J Crystal Growth 311:3184–3194
7. Javierre E, Vuik C, Vermolen FJ, Segal A (2007) A level set method for three dimensional vector Stefan problems: dissolution of stoichiometric particles in multi-component alloys. J Comput Phys 224:222–240
8. Kampmann R, Wagner R (1991) Materials science and technology—a comprehensive treatment, vol 5. VCH, Weinheim

9. Mullins WW, Sekerka RF (1963) Morphological stability of a particle growing by diffusion or heat flow. J Appl Phys 34:323–329
10. Osher SJ, Sethian JA (1988) Fronts propagating with curvature-dependent speed: algorithms based on Hamilton–Jacobi formulations. J Comput Phys 79:12–49
11. Osher SJ, Fedkiw RP (2002) Level set methods and dynamic implicit surfaces, 1st edn. Springer, The Netherlands
12. den Ouden D, Vermolen FJ, Zhao L, Vuik C, Sietsma J (2011) Modelling of precipitate nucleation and growth in binary alloys under elastic deformation: an application to a Cu-0.95%Co alloy. Comput Mater Sci 50:2397–2410
13. Perez M (2005) Gibbs–Thomson effects in phase transformations. Scr Mater 52:709–712
14. Porter DA, Easterling KE (1992) Phase transformations in metals and alloys, 2nd edn. Chapmann and Hall, London
15. Robson JD (2004) Modelling the evolution of precipitate size distribution during nucleation, growth and coarsening. Mater Sci Technol 20:441–448
16. Segal A, Vuik C, Vermolen FJ (1998) A conserving discretization for the free boundary in a two-dimensional Stefan problem. J Comput Phys 141:1–21
17. Segal A (2009) Sepran introduction. Ingenieursbureau SEPRA. Den Haag, The Netherlands
18. Sethian JA (1999) Fast marching methods. SIAM Rev 41:199–235
19. Soisson F, Barbu A, Martin G (1996) Monte Carlo simulation of copper precipitation in dilute iron–copper alloys during thermal ageing and under electron radiation. Acta Mater 44:3789–3800
20. Sussman M, Smereka P, Osher SJ (1994) A level set approach for computing solutions to incompressible two-phase flow. J Comput Phys 114:146–159
21. Tornberg AK, Engquist B (2000) A finite element based level-set method for multiphase flow applications. Comput Vis Sci 3:93–101
22. Vermolen FJ (2007) On similarity solutions and interface reactions for a vector-valued Stefan problem. Nonlinear Anal Model Control 12:269–288
23. Vermolen FJ, Javierre E, Vuik C, Zhao L, van der Zwaag S (2007) A three-dimensional model for precipitate dissolution in binary alloys. Comput Mater Sci 39:767–774
24. Zener C (1949) Theory of growth of spherical precipitates from solid solution. J Appl Phys 20:950–953