Computing Interfaces in Diverse Applications

E. Javierre, F.J. Vermolen, C. Vuik, P. Wesseling, and S. van der Zwaag

Abstract Mathematical models and computing methods for problems involving moving interfaces are considered. These occur in a great variety of applications, and mathematical models provide a unifying framework, facilitating interdisciplinary cooperation. We discuss and propose some generic numerical methods for problems involving moving interfaces. The level set method is used for interface capturing. A Cartesian and a finite element mesh are used simultaneously. This facilitates efficient local mesh refinement and derefinement for accurate computation of physical effects occurring at the interfaces, that move and may undergo topological change. The method has been implemented in three dimensions. We present examples from materials science (homogenization) and medical technology (wound healing).

Keywords moving interface · level set method · adaptive mesh refinement · cut-cell method · epidermal wound healing

1 Introduction

In multidisciplinary research, mathematical models are the universal framework to describe phenomena from fields as diverse as physics, biology or mathematical finance, for example. Many of these models involve solving a coupled system of partial differential equations in a domain containing moving interfaces. One may think of multiphase flows [25, 28, 33], phase transformations in materials science [8, 14, 16, 20], and mathematical models for wound healing [6, 7], to mention just a few examples. These problems entail great computational challenges. Efficient

E. Javierre, F.J. Vermolen, C. Vuik, and P. Wesseling
Delft Institute of Applied Mathematics, Delft University of Technology

E. Javierre and S. van der Zwaag
Faculty of Aerospace Engineering, Delft University of Technology; E-mails: [E.JavierrePerez, F.J.Vermolen, C.Vuik]@tudelft.nl; [P.Wesseling, S.vanderZwaag]@tudelft.nl

J. Eberhardsteiner et al. (eds.), *ECCOMAS Multidisciplinary Jubilee Symposium*, Computational Methods in Applied Sciences.

numerical methods are required, and judicious simplifications in the mathematical models are necessary to reduce computing time, while keeping physical realism intact. Here we present a brief survey of numerical approximation of mathematical models involving moving interfaces, and discuss a few examples.

Requirements to be satisfied depend on the type of application. In general, we require three-dimensionality of the domain. In the materials science application to be presented, little is to be learned from two-dimensional models. In the biomedical application to be given, focus is put on closure of superficial (epidermal) wounds modelled in two dimensions. Study of more deep wounds demands coupled models involving mechanical interactions in three dimensions. Furthermore, the method must be robust and efficient in the presence of topological changes of the interface geometry and of complicated interface conditions. In the third place, certain application-dependent physical conditions must be fulfilled, such as strict mass conservation or energy minimization. Finally, numerical efficiency is required, in order to deal with three-dimensional domains and to allow sufficiently fine grids.

In Section 2 we give a brief survey of methods to deal with moving interfaces. We prefer the level set method, and go into more detail in Section 3. In Section 4 local grid refinement near the interface is discussed. Examples from materials science and mathematical biology are given in Section 5, and final remarks are presented in Section 6.

2 Survey of Methods for Moving Interfaces

There are mainly two classes of methods for computing moving interfaces: front-tracking and front-capturing methods. These we now briefly discuss.

2.1 Front-Tracking Methods

In front-tracking methods the location of interfaces is determined by keeping track of the location of points lying on the interface. The interface points may belong to a computational grid or not. In the first case, the so-called arbitrary Lagrangian-Eulerian (ALE) approach [15,20] is used to update the computational mesh together with the moving interface. In the second case, interface conditions are implemented by discretization involving fixed grid points in the vicinity of the moving interface points [2, 10, 13, 26, 32].

The drawback of these methods is that grid points must be added or removed as the interface moves, which is not a simple affair. Furthermore, in case of large changes in the shape or position of the interface, re-meshing will be necessary, which is computing-intensive in three dimensions. Merging and splitting-up of interfaces require complicated algorithms.

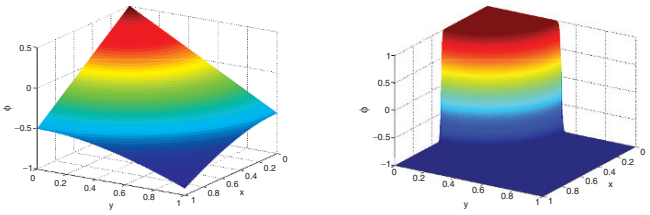

Fig. 1 The marker function ϕ in the level set (*left*) and phase field (*right*) methods. Level set methods are based on a sharp interface whereas phase field methods use a diffused interface

2.2 Front-Capturing Methods

Front-capturing methods are characterized by the incorporation of a marker function that parameterizes the computational domain. Widely used examples of front-capturing methods are the level set [18] and the phase field [3, 4] methods. For applications in materials science, see [8, 9, 11, 12, 14, 31] and in fluid dynamics, see [23, 24, 27, 28, 33].

Level set methods use a level set function ϕ, whose zero level set is the interface. The interface motion is recovered by incorporating a transport equation for ϕ. Usually, ϕ is the signed distance to the interface in the neighbourhood of the interface. Phase field methods identify the phases by a phase field function ϕ that has a certain value for each phase and changes rapidly in a small neighbourhood of the interface. The interface motion follows from minimization of a free energy functional, so that a direct implementation of interface jump conditions is avoided. The interface is not sharp but smeared out over a narrow zone. The level set and phase field functions are illustrated in Fig. 1.

A great advantage of front-capturing methods is that the implicit representation of interfaces makes topological changes of interfaces (merging and splitting-up) easy to handle. Moreover, these methods are easily generalized to problems involving more than two phases, simply by introducing additional artificial marker functions. A disadvantage of phase field methods is that they give rise to strongly nonlinear systems that impose severe time step restrictions. Furthermore, they involve a very large number of physical parameters, that are usually obtained by fitting simulations to experiments, and in general cannot yet be derived from thermodynamic databases. Here we restrict ourselves to level set methods. We present some details and ways to overcome certain weaknesses of level set methods.

3 The Level Set Method

For introductions to the level set method, see [17, 21]. A brief description is given below. We include an enhancement to increase the generality and the accuracy of the method, namely the simultaneous use of a Cartesian mesh and a locally refined finite element mesh that is fitted to the interface.

3.1 The Level Set Equation

The interface, denoted by Γ, is the zero level set of an artificial differentiable function ϕ, called the level set function. Initially, ϕ is defined as follows:

$$\phi(\mathbf{x},0) = \begin{cases} +\operatorname{dist}(\mathbf{x},\Gamma(0)), & \text{if } \mathbf{x} \in \Omega_1, \\ 0, & \text{if } \mathbf{x} \in \Gamma(0), \\ -\operatorname{dist}(\mathbf{x},\Gamma(0)), & \text{if } \mathbf{x} \in \Omega_2, \end{cases} \tag{1}$$

where dist is the distance to the interface, and Ω_i is the domain occupied by phase i. As Γ moves, it remains the zero level set of ϕ, so that ϕ must satisfy

$$\frac{\partial \phi}{\partial t} + v_n ||\nabla \phi|| = 0 \quad \text{at} \quad \mathbf{x} \in \Gamma, \tag{2}$$

where v_n denotes the normal component of the interface velocity $\mathbf{v}$. This equation is continued to the whole domain by continuous but otherwise arbitrary extension of $\mathbf{v}$. Given such $\mathbf{v}$, ϕ is found from the so-called level set equation:

$$\frac{\partial \phi}{\partial t} + \mathbf{v} \cdot \nabla \phi = 0. \tag{3}$$

The curvature κ and the normal vector $\mathbf{n}$ of the interface follow from

$$\mathbf{n} = \frac{\nabla \phi}{||\nabla \phi||}, \quad \kappa = \nabla \cdot \mathbf{n}. \tag{4}$$

3.2 Velocity Extension

In fluid dynamics, for example, it is natural to let $\mathbf{v}$ in Eq. (3) be the flow velocity. In other applications, such as in materials science, artificial continuation of v_n from the interface to the domain is required. Adalsteinsson and Sethian [1] continue $\mathbf{v}$ in the direction of increasing $|\phi|$ such that the solution of (3) is a distance function. In applications where v_n is readily available, such as in solidification [11] or epidermal wound healing [7], simple constant extrapolation of v_n suffices. Advection of v_n a few grid points away from the interface [17], by performing a few pseudo-time steps with

$$\frac{\partial v_n}{\partial \tau} + \mathrm{S}(\phi)\mathbf{n} \cdot \nabla v_n = 0, \tag{5}$$

where S denotes the sign function, is also attractive because of low computing cost. If computing v_n at the interface is more complicated, such as in phase transformations in multi-component alloys [8], it is strongly advisable to continue the Cartesian components of $\mathbf{v}$ independently [5], by performing a few pseudo-time steps of

$$\frac{\partial v_i}{\partial \tau} + \mathrm{S}(\phi) v_i \frac{\partial \phi}{\partial x_i} = 0, \quad i = 1,2,3, \tag{6}$$

where v_i and x_i are Cartesian components. Note that (5) and (6) leave v_n unaltered at the interface, and advect v_n away from the interface in the right direction.

3.3 Distance Function and Re-initialization of the Level Set Function

Initially, ϕ is specified to be the distance to the interface and satisfies $||\nabla\phi|| = 1$. But this property gradually gets lost during time stepping. Flat or steep gradients of ϕ may develop, leading to inaccurate approximation of the interface location, and often as a consequence its velocity. This numerical problem can be remedied by re-initialization of ϕ to a distance function after a few time steps with (3).

In cases where ϕ is used only to determine the interface location, such as multiphase flow and Stefan problems, it is required to have $||\nabla\phi|| \approx 1$ only near the interface. In such cases it is recommended to re-initialize ϕ only in grid points near the interface. This can be done by carrying out a few pseudo-time steps with

$$\frac{\partial\phi}{\partial\tau} = \hat{\mathrm{S}}(\phi)\big(1 - ||\nabla\phi||\big), \tag{7}$$

until $\big|1 - ||\nabla\phi||\big| < \varepsilon$, where $\hat{S}$ is given by (see [19])

$$\hat{\mathrm{S}}(\phi) = \frac{\phi}{\sqrt{\phi^2 + ||\nabla\phi||^2\Delta x^2}}, \tag{8}$$

where Δx denotes the grid size.

In other cases one may need to know approximately the distance to the interface, in order to identify regions with different physical effects. Then it is convenient to let ϕ be an approximation to the distance to the interface in a large part of the domain. For example, in modelling wound healing, we need to distinguish the wound region Ω_w, the active layer Ω_{al} and the undamaged region Ω_u. If ϕ is an approximation of the distance to the interface, then these regions can be conveniently specified as follows [7]:

$$\begin{aligned}
\Omega_w(t) &= \{\, \mathbf{x} \in \Omega \mid 0 < \phi(\mathbf{x},t) \,\}, \\
\Omega_{al}(t) &= \{\, \mathbf{x} \in \Omega \mid -\delta < \phi(\mathbf{x},t) < 0 \,\}, \\
\Omega_u(t) &= \{\, \mathbf{x} \in \Omega \mid \phi(\mathbf{x},t) < -\delta \,\}.
\end{aligned} \tag{9}$$

where δ denotes the thickness of the active layer. Here it is important to specify the active layer accurately, since this is where things happen: the production of certain proteins that trigger cell mitosis and migration. In cases like this, where re-initialization is required in a sizable part of the domain, it is computationally efficient to solve the Eikonal equation $||\nabla\phi|| = 1$ by a direct method, such as the Fast Marching Method [22].

4 Composite Grid

In solving moving interface problems, partial differential equations have to be solved in domains that change due to motion of the interface. The velocity with which the interface moves depends on the solutions of the partial differential equations involved. For discretization near the arbitrarily shaped interface, we opt for the finite element method, using adaptive local grid refinement where called for. For the level set equation, a finite difference method on a Cartesian grid is accurate and efficient. We have found a combination of finite elements and finite differences that is efficient and flexible in various applications. To make communication between the grids cheap, the two grids use the same grid points. The finite element grid thus obtained is called the basis finite element grid. This grid is locally and dynamically enriched with points on the interface, and possibly with local refinement patches.

4.1 Local Grid Refinement and Derefinement

Often, the physical processes taking place at the interface need to be computed accurately, in order to find the motion of the interface with sufficient precision. To achieve this, we apply local grid refinement, if necessary. Let us refine the elements within a certain distance δ from the interface. All elements with one or more vertices e_i satisfying $|\phi(e_i)| < \delta$ get level of refinement LOR $= 2$. Because of mesh consistency, elements not having LOR $= 2$ and adjacent to one element with LOR $= 2$ get LOR $= 1$, but if they are adjacent to two elements with LOR $= 2$ they get LOR $= 2$. Other elements get LOR $= 0$. The refinement procedure is illustrated in Fig. 2. For efficiency, it is necessary to apply derefinement as the interface moves away sufficiently far from a refined cell. The time step is restricted such that LOR never changes by more than 1. This makes it easy to derefine cells whose LOR decreases.

The refined grid is nested in the Cartesian grid, as illustrated in Fig. 3, so that we obtain a locally refined Cartesian grid as well. This has the advantage that the velocity continuation and the solution of the level set equation can be carried out on the refined Cartesian grid only. If necessary the level set function can be continued onto the coarse part of the Cartesian grid by the Fast Marching Method [22].

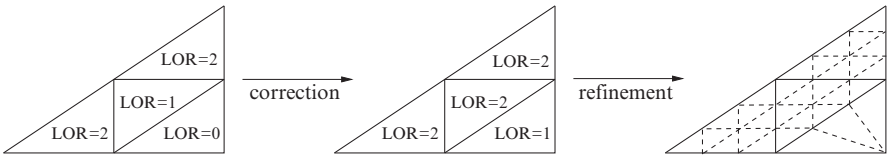

Fig. 2 Correction of the LOR and subsequent subdivision of the elements, with refinement ratio equal to 3 (i.e. each edge of an element with LOR $= 2$ is divided into three equally-sized subedges)

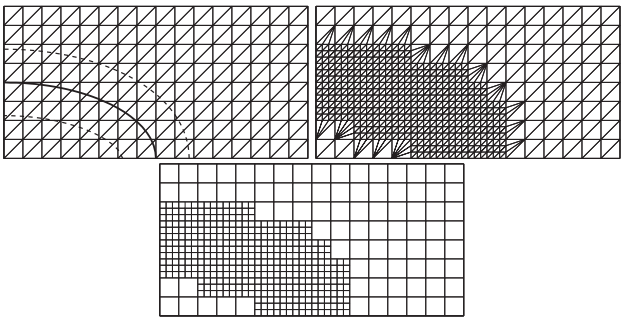

Fig. 3 *Left*: Fixed base FE mesh with the interface position $\phi = 0$ (solid curve) and the contours $\phi = \pm dist$ (dashed curves). The elements within these contours are to be refined. *Center*: Refined FE mesh. *Right*: The nested Cartesian grids

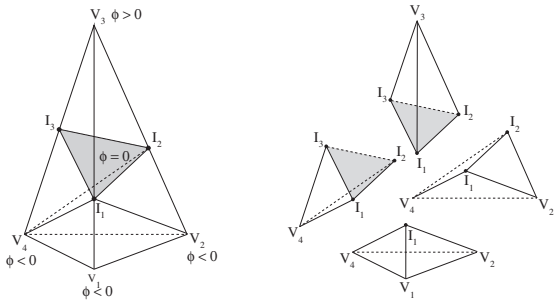

Fig. 4 The cut-cell method applied to a tetrahedron with three intersection points: division of the faces (*left*) and division of the element (*right*)

4.2 The Cut-Cell Method

To facilitate accurate discretization near and at the interface the finite element grid is adapted to conform to the interface. Elements of the basis finite element grid or of the locally refined grid that are intersected by the interface are cut such that the intersection with the interface becomes an element face. Linear finite elements allow easy local elementwise adaptation to the interface. An element is intersected by the interface if and only if the level set function ϕ changes sign at the element vertices. The points of intersection of the interface and the element edges are easily determined by linear interpolation of ϕ between the element vertices. These intersection points are added to the vertices of the finite element mesh. The element is cut in two by the interface, and further cuts are made so that we end up again with tetrahedra. The procedure is straightforward, and is illustrated in Fig. 4.

5 Applications

The method described in Sections 3 and 4 is applied here to two applications: the dissolution of precipitates in multi-component alloys under homogenization conditions and the closure of epidermal wounds by cell mitosis and migration. Results show the relevance of geometry in the transformation kinetics, proving the necessity of efficient higher dimensional computational tools.

5.1 Precipitate Dissolution Under Homogenization Conditions

Homogenization refers to a thermal treatment applied to as-cast alloys in order to eliminate small precipitates that impair the mechanical properties and limit further treatments. Annealing the as-cast microstructure above the so-called A_1-temperature leads to the dissolution of the precipitates into the surrounding matrix. Due to the appreciable energetical requirements of this process, good approximations of the homogenization times depending on precipitate volume fraction, composition and morphology are of great value.

We use a vector Stefan model [30] to simulate precipitate dissolution in multi-component alloys. The hypotheses of this model are: (1) Fickian diffusion of the chemical species into the matrix:

$$\frac{\partial c_i}{\partial t} = D_i \Delta c_i \quad \text{in } \Omega_{dm}, \tag{10}$$

where Ω_{dm} denotes the diffusive matrix surrounding the precipitate, (2) precipitate stoichiometry preservation:

$$c_i = c_i^{part} \quad \text{in } \Omega_p, \tag{11}$$

$$f(c_1^{sol}, \ldots, c_p^{sol}) = 0 \quad \text{in } \Gamma, \tag{12}$$

where Ω_p denotes the precipitate, c_i^{sol} the interfacial concentration and f describes the solubility product at the interface Γ, and (3) mass conservation of the chemical species in the alloy:

$$\frac{\partial c_i}{\partial \mathbf{n}} = 0 \quad \text{in} \overline{\Omega}_{dm} \backslash \Gamma, \tag{13}$$

$$\left(c_i^{part} - c_i^{sol}\right) v_n = D_i \frac{\partial c_i}{\partial \mathbf{n}} \quad \text{in } \Gamma. \tag{14}$$

Each time step, the interfacial c_i^{sol} and matrix c_i concentrations are computed using a fixed-point iteration after the position of the interface is determined. Subsequently, the interface velocity is calculated and the interface position updated.

5.1.1 Effect of Morphology in Dissolution Kinetics

A family of Mg_2Si particles dissolving in a aluminium alloy at 560°C are studied here. The shape of these particles vary from a circle to a very elongated ellipse, being all of the same area. Hence, the differences in the dissolution time observed in Fig. 5 (left) are entirely due to the morphology: elongated particles allow larger outflux of atoms into the matrix resulting into shorter dissolution times. Furthermore, analysis of the precipitate morphology during the whole transformation shows that dissolution of isolated precipitates is shape preserving, see Fig. 5 (right).

5.1.2 Spheroidization of Lamellar Structures

The metallurgical application studied in this section is inspired by an AISI 52100 steel. The diffusion of chromium is disregarded here, thus the pearlite structure is

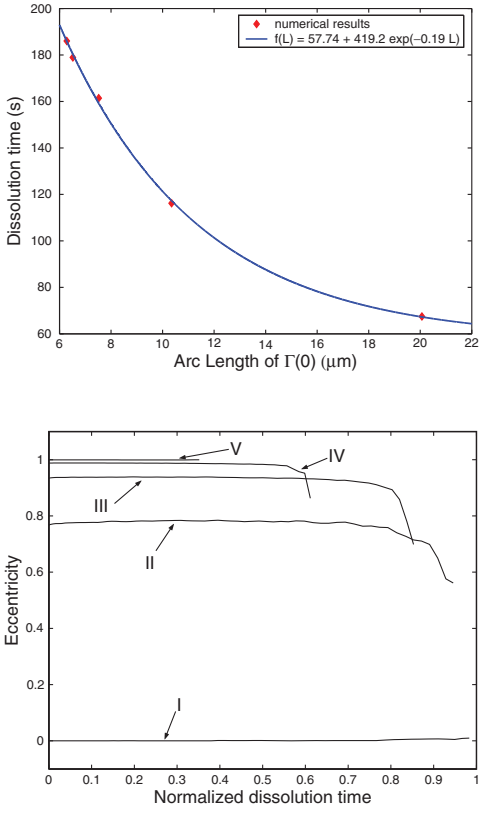

Fig. 5 Dissolution of an Mg_2Si particle of different elongations: dissolution time as function of the length of the initial interface (*left*) and particle eccentricity during dissolution (*right*)

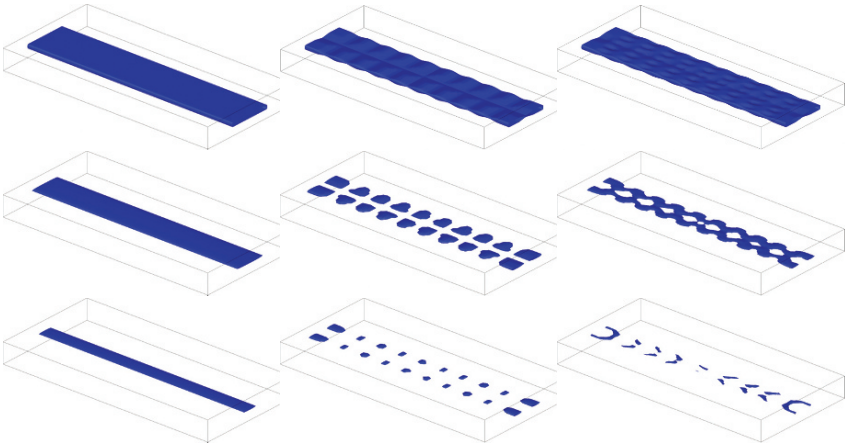

Fig. 6 Dissolution of a cementite particle upon different initial morphologies: unperturbed plate (*left*), plate with a linear crack (*center*), plate with a sinusoidal crack (*right*)

treated as a binary alloy, considering only the diffusion of carbon. Because of symmetry, we consider one cementite plate surrounded by the ferrite matrix. The initial volume fraction of the cementite phase is 8.33%, and the transformation is chosen to occur at 800°C, which leads to the complete dissolution of the cementite.

The evolution of the plate is plotted in the left column of Fig. 6. Due to the azimuthal dimensions of the computational domain, dissolution progresses further in the other directions until it is completed. Moreover, the lamellar shape of the cementite is preserved during all the stages of dissolution. However, if a slight perturbation is applied onto the surface of the cementite plate, the dissolution kinetics are completely changed. The center column of Fig. 6 shows the evolution of the plate when a planar perturbation or 'crack' is added, and the right column of Fig. 6 shows the evolution when a sinusoidal perturbation is applied. In both cases, the plate splits up into an array of subparticles that dissolve in time. However, the time at which the plate breaks up and the shape of these subparticles is completely different. The spheroidization of cementite is obtained for the planar perturbation, whereas more irregular and extended particles are obtained when the sinusoidal perturbation is used.

The dissolution time is also strongly affected by the type of perturbation applied to the surface. The normalized volume of the cementite plate during dissolution is presented in Fig. 7. The dimensions of the plate are adjusted according to the crack to preserve the initial volume. Results show that any perturbation of the configuration used here gives a faster dissolution, even having an array of 20 separated particles next to each other, where the soft-impingement effects are strong. Furthermore, the dissolution speed is increased by almost a factor three when the thickness of the plate is not modified.

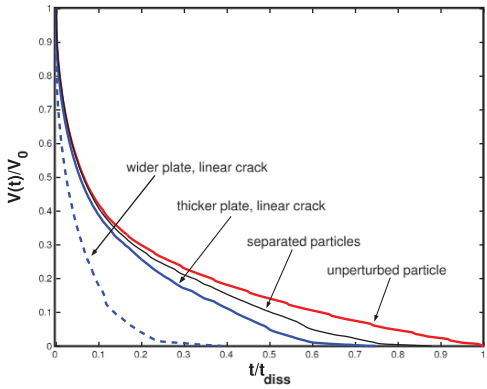

Fig. 7 Influence of the initial morphology in the dissolution kinetics

5.2 Epidermal Wound Closure

Wound healing proceeds by a combination of several interdependent and self-controlled mechanochemical processes: angiogenesis, re-epithelialization, extracellular matrix synthesis and wound contraction. Cells are activated by certain proteins (growth factors) and provide the adequate response. Mathematical modelling combined with computer simulations can help understanding the healing process and reveal conditions for optimal treatment.

Closure of epidermal wounds is entirely due to cell mitosis and migration. The increased cellular activity in the proximity of the wound edge triggers the production of epidermal growth factors (EGF) that activate cell motility. The concentration c of EGF is governed by the reaction-diffusion equation

$$\frac{\partial c}{\partial t} - D\Delta c + \lambda c = P\chi_{\Omega_{al}(t)}, \tag{15}$$

where χ denotes the characteristic function, restricting the production of EGF to the active layer Ω_{al}. The computational domain Ω is taken sufficiently large to justify the choice of homogeneous Neumann boundary conditions. Furthermore, prior to damage, there is no EGF present, i.e. $c(\mathbf{x},0) = 0$. The closure rate is proportional to the local curvature of the wound edge:

$$v_n = \big(\alpha + \beta\kappa\big)H(c-\theta), \tag{16}$$

where H denotes the Heaviside function and θ the threshold value below which cells are non-motile.

5.2.1 Influence of Morphology on Healing

The evolution of the wound edge and the local refinement is shown in Fig. 8 for a starfish-shaped wound. Due to the choice of the parameters ($\alpha = 1\,\text{cm/s}$, $\beta = 5 \times 10^{-2}\,\text{cm}^2/\text{s}$ and $\theta = 10^{-3}$), the initial wound divides into five smaller sub-wounds that heal independently. We have observed that modifying β or θ leads to significantly different healing patterns, involving local retreat to the wound at the concave areas in the extreme cases [7].

5.2.2 Accuracy and Computational Cost

The accuracy and efficiency of the adaptive grid strategy is evaluated here. The EGF concentration of a circular wound after 45 min of incubation period (i.e. before the threshold value θ is reached) is compared to the analytic solution [29]. The L^2

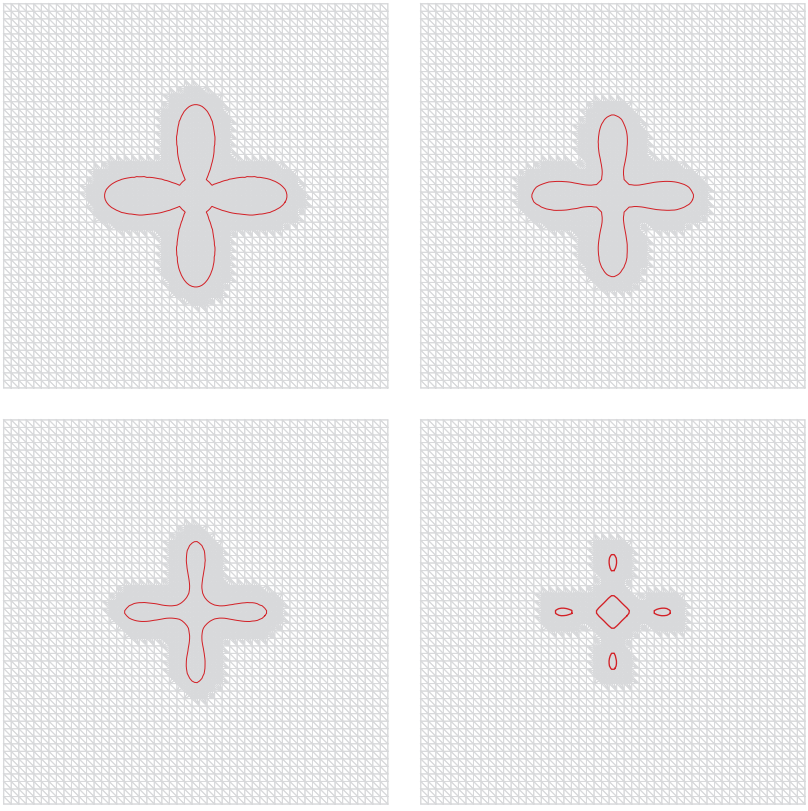

Fig. 8 Snapshots of the wound edge location and the local grid refinement. Time evolution follows from left to right and from top to bottom, and correspond to the initial wound and the wound after 30%, 60% and 90% of its initial area is healed respectively

Table 1 Accuracy and computational cost of the local grid refinement, where N denotes the number of nodes per Cartesian direction

N	Ref. ratio	L^2 error	Nr. of elements	CPU-time in re-initialization
	–	$1.61 \cdot 10^{-1}$	3,200	1.31 s
41	2	$7.55 \cdot 10^{-2}$	+778	+0.34 s
	3	$6.09 \cdot 10^{-2}$	+2,032	+0.92 s
	–	$8.07 \cdot 10^{-2}$	12,800	6.55 s
81	2	$3.75 \cdot 10^{-2}$	+1,562	+0.83 s
	3	$2.87 \cdot 10^{-2}$	+4,080	+2.28 s
	–	$4.11 \cdot 10^{-2}$	51,200	46.44 s
161	2	$2.15 \cdot 10^{-2}$	+3,094	+2.20 s
	3	$1.52 \cdot 10^{-2}$	+8,096	+6.18 s

norm of the relative error is presented in Table 1 for different levels of refinement. Due to the discontinuous production function χ, the finite element approximation looses one order of accuracy. However, the accuracy of the globally refined grid is recovered and even improved upon by the local refinement of the coarse grid at the interface. Furthermore, the number of elements (and hence the computational cost of the finite element calculations) grows only linearly if the basis grid is locally refined. On the other hand, the most expensive calculation over the Cartesian grids is the reinitialization, for which computational time is also given in Table 1. Note that there are two contributions to this term: the reinitialization time in the refined band and the reinitialization time in the coarse grid. In both cases, a second order accurate Fast Marching Method [22] has been used. The changes of the computational times are entirely due to the different number of nodes within the refined band. Results show that using a uniformly fine background grid is computationally expensive. However, using a coarser basis grid with a refinement ratio of 3 improves the accuracy and decreases the cost of reinitialization by nearly a factor 5.

6 Conclusions

Numerical solution methods for problems involving moving interfaces in two different applications have been discussed. The level set method is chosen instead of alternative methods, because of its superior performance when the interface develops topological changes. The typical issues concerning this representation are discussed, and numerical methods are proposed for a wide range of applications. It is shown that a composite grid algorithm such as presented here facilitates the implementation of solution methods for the hyperbolic equations introduced by the level set formulation, and allows straightforward local grid refinement and derefinement and boundary fitting techniques, leading to an improvement of the accuracy. Results

for dissolution of precipitates in multi-component alloys and closure of epidermal wounds show the flexibility of the algorithm and highlights the strong influence of morphology on the transformation kinetics.

References

1. D. Adalsteinsson and J. A. Sethian, *The fast construction of extension velocities in level set methods*, J. Comput. Phys., 148 (1999), pp. 2–22.
2. J. U. Brackbill, D. B. Kothe, and C. Zemach, *A continuum method for modeling surface tension*, J. Comput. Phys., 100 (1992), pp. 335–354.
3. G. Caginalp, *An analysis of a phase field model of a free boundary*, Arch. Ration. Mech. Anal., 92 (1986), pp. 205–245.
4. G. Caginalp, *Stefan and Hele-Shaw type models as asymptotic limits of the phase-field equations*, Phys. Rev. A, 39(3) (1989), pp. 5887–5896.
5. S. Chen, B. Merriman, S. Osher, and P. Smereka, *A simple level set method for solving Stefan problems*, J. Comput. Phys., 135 (1997), pp. 8–29.
6. C. S. Hogea, B. T. Murray, and J. A. Sethian, *Simulating complex tumor dynamics from avascular to vascular growth using a general level-set method*, J. Math. Biol., 53 (2006), pp. 86–134.
7. E. Javierre, F. J. Vermolen, C. Vuik, and S. van der Zwaag, *A mathematical analysis of physiological and morphological aspects of wound closure*, J. Math. Biol., submitted (2008), p. 18 pages.
8. E. Javierre, C. Vuik, F. J. Vermolen, and A. Segal, *A level set method for three dimensional vector Stefan problems: dissolution of stoichiometric particles in multi-component alloys*, J. Comput. Phys., 224 (2007), pp. 222–240.
9. E. Javierre Pérez, *Numerical methods for vector Stefan models of solid-state alloys*, Ph.D. thesis, Delft University of Technology, 2006.
10. D. Juric and G. Tryggvason, *A front-tracking method for dendritic solidification*, J. Comput. Phys., 123 (1996), pp. 127–148.
11. Y.-T. Kim, N. Goldenfeld, and J. Dantzig, *Computation of dendritic microstructures using a level set method*, Phys. Rev. E, 62 (2000), pp. 2471–2474.
12. I. Kovačević and B. Šarler, *Solution of a phase-field model for dissolution of primary particles in binary aluminium alloys by an r-adaptive mesh-free method*, Mater. Sci. Eng. A, 413–414 (2005), pp. 423–428.
13. M.-C. Lai and C. S. Peskin, *An immersed boundary method with formal second-order accuracy and reduced numerical viscosity*, J. Comput. Phys., 160 (2000), pp. 705–719.
14. B. T. Murray, A. A. Wheeler, and M. E. Glicksman, *Simulations of experimentally observed dendritic growth behavior using a phase-field model*, J. Cryst. Growth, 154 (1995), pp. 386–400.
15. W. D. Murray and F. Landis, *Numerical and machine solutions of transient heat conduction problems involving melting or freezing*, Transactions ASME (C), J. Heat Transf., 245 (1959), pp. 106–112.
16. B. Nedjar, *An enthalpy-based finite element method for nonlinear heat problems involving phase change*, Comput. Struct., 80 (2002), pp. 9–21.
17. S. Osher and R. Fedkiw, *Level set methods and dynamic implicit surfaces*, vol. 153 of Applied Mathematical Sciences, Springer, New York, 2003.
18. S. Osher and J. A. Sethian, *Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations*, J. Comput. Phys., 79 (1988), pp. 12–49.
19. D. Peng, B. Merriman, S. Osher, H. Zhao, and M. Kang, *A PDE-based fast local level set method*, J. Comput. Phys., 155 (1999), pp. 410–438.

20. G. SEGAL, K. VUIK, AND F. VERMOLEN, *A conserving discretization for the free boundary in a two-dimensional Stefan problem*, J. Comput. Phys., 141 (1998), pp. 1–21.
21. J. A. SETHIAN, *Level set methods*, vol. 3 of Cambridge Monographs on Applied and Computational Mathematics, Cambridge University Press, Cambridge, 1996. Evolving interfaces in geometry, fluid mechanics, computer vision, and materials science.
22. J. A. SETHIAN, *Theory, algorithms, and applications of level set methods for propagating interfaces*, vol. 5 of Acta Numerica, Cambridge University Press, Cambridge, 1996, pp. 309–395.
23. M. SUSSMAN AND E. FATEMI, *An efficient, interface-preserving level set redistancing algorithm and its application to interfacial incompressible fluid flow*, SIAM J. Sci. Comput., 20 (1999), pp. 1165–1191 (electronic).
24. M. SUSSMAN, P. SMEREKA, AND S. OSHER, *A level set approach for computing solutions to incompressible two-phase flow*, J. Comput. Phys., 114 (1994), pp. 146–159.
25. G. TRYGGVASON, B. BUNNER, A. ESMAEELI, D. JURIC, N. AL-RAWAHI, W. TAUBER, J. HAN, S. NAS, AND Y.-J. JAN, *A front-tracking method for the computations of multiphase flow*, J. Comput. Phys., 169 (2001), pp. 708–759.
26. H. S. UDAYKUMAR, R. MITTAL, P. RAMPUNGGOON, AND A. KHANNA, *A sharp interface Cartesian Grid method for simulating flows with complex moving boundaries*, J. Comput. Phys., 174 (2001), pp. 345–380.
27. S. P. VAN DER PIJL, *Computation of bubbly flows with a mass-conserving level-set method*, Ph.D. thesis, Delft University of Technology, 2005.
28. S. P. VAN DER PIJL, A. SEGAL, C. VUIK, AND P. WESSELING, *A mass-conserving Level-Set method for modelling of multi-phase flows*, Int. J. Numer. Meth. Fluid., 47 (2005), pp. 339–361.
29. F. J. VERMOLEN AND E. JAVIERRE, *Qualitative Analysis of a simple mathematical model for epidermal wound closure*, in preparation, (2008).
30. F. J. VERMOLEN, C. VUIK, AND S. VAN DER ZWAAG, *The dissolution of a stoichiometric second phase in ternary alloys: a numerical analysis*, Mater. Sci. Eng. A, 246 (1998), pp. 93–103.
31. A. A. WHEELER, B. T. MURRAY, AND R. J. SCHAEFER, *Computation of dendrites using a phase field model*, Physica D, 66 (1993), pp. 243–262.
32. Y. YANG AND H. S. UDAYKUMAR, *Sharp interface Cartesian grid method III: solidification of pure materials and binary solutions*, J. Comput. Phys., 210 (2005), pp. 55–74.
33. H.-K. ZHAO, T. CHAN, B. MERRIMAN, AND S. OSHER, *A variational level set approach to multiphase motion*, J. Comput. Phys., 127 (1996), pp. 179–195.