A conserving discretization for the free boundary in a two-dimensional Stefan problem

Report 97-13

Guus Segal
Kees Vuik
Fred Vermolen

TU Delft
Technische Universiteit Delft
Delft University of Technology

Faculteit der Technische Wiskunde en Informatica
Faculty of Technical Mathematics and Informatics

ISSN 0922-5641

Copies of these reports may be obtained from the bureau of the Faculty of Technical Mathematics and Informatics, Julianalaan 132, 2628 BL Delft, phone $+31152784568$.
A selection of these reports is available in PostScript form at the Faculty's anonymous ftp-site. They are located in the directory /pub/publications/tech-reports at ftp.twi.tudelft.nl

A conserving discretization for the free boundary in a two-dimensional Stefan problem

Guus Segal[1], Kees Vuik, Fred Vermolen[2]

December 10, 1996

[1]Address of first two authors: Faculty of Technical Mathematics and Informatics, Delft University of Technology, P.O. Box 5031, NL 2600 GA Delft, The Netherlands

[2]Laboratory of Materials Science, Delft University of Technology, P.O. Box 5045, NL 2600 GA Delft, The Netherlands

Chapter 1

Introduction

Heat treatment of metals is often necessary to optimise their mechanical properties both for further processing and for final use. During the heat treatment, the metallurgical state of the material changes. This change can either involve the phases being present or the morphology of the various phases. Whereas the equilibrium phases often can be predicted quite accurately from thermodynamic models, there are no general models for microstructural changes nor general models for the kinetics of these changes. In the latter cases both the initial morphology and the transformation mechanisms have to be specified explicitly. One of these processes, which is both of large industrial and scientific interest and amenable to modelling, is the dissolution of second phase particles in a matrix with a uniform initial composition.

To describe this particle dissolution in rigid media several physical models have been developed, incorporating the effects of long-distance diffusion [20], [12], [14] and non-equilibrium conditions at the interface [5], [1]. The long-distance diffusion models imply that the processes at the interface between particle and matrix proceed infinitely fast. Therefore, these models provide an upper bound for the dissolution rate.

Whelan [20] considered particles dissolving in an infinite medium using the stationary interface approximation. He derived an analytical solution of the diffusion equation in an infinite medium for spherical co-ordinates by the use of the Laplace-transformation in time. The accuracy of the model increases with increasing inter-particle distance, i.e. with increasing cell size.

Mathematically particle dissolution is considered as a moving boundary problem, i.e. a Stefan-problem [3], [17], [11], [18]. All references discussed below, transform the cell in which the particle dissolves into a cell which is equally shaped as the particle, requiring the transformed cell volume to be equal to the original cell volume. This allows a one-dimensional treatment of the moving boundary problem which can be solved easily using a finite difference discretization method. This method can be used for planar, cylindrical and spherical geometry.

Baty, Tanzilli and Heckel [12] were the first authors to apply a numerical method using a finite difference method to evaluate the interface position as a function of dissolution time. Their model is also applicable to situations in which the inter-particle distance is small, i.e.

when soft impingement occurs. The model they proposed, is based on the assumption of local equilibrium at all stages of the dissolution process. They applied their numerical analysis to dissolving Al_2Cu-particles in aluminium. In their models the Al_2Cu-particles were assumed to be spherical. The poor fit of their calculations with the experimental data is probably due to the interface reactions or to the non-spherical shape of the regular particles, which were not incorporated into their numerical model.

Tundal and Ryum [14] considered the effects of a finite cell size for spherical particles as well. They too applied a numerical method using a finite difference method to predict the dissolution kinetics. Their model is based on the assumption of local equilibrium during the entire dissolution process. They introduced a lognormal distribution (logarithm according to normal distribution) for both the particle and cell size. They showed that macroscopic dissolution rates depend strongly on the shape of the particle size distribution curve and possible interactions between the neighbouring cells.

Nolfi's model [5] did not include the interface migration, but as far as we know, it is the first model which incorporated non-equilibrium conditions at the interface. In the Nolfi model non-equilibrium conditions at the interface were incorporated by the introduction of a Robbins condition, which relates the concentration gradient with the concentration at the interface. This semi-analytical solution consists of an infinite series solution for the concentration profile. Their method, however, is only accurate in the early stages of the dissolution process.

Aaron and Kotler [1] incorporated the non-equilibrium conditions at the interface too. However their approach is only applicable for those situations in which the inter-particle distance is sufficiently large, i.e. the diffusion fields do not impinge. They transformed the Robbins problem of Nolfi into a Dirichlet problem, in which the concentration is fixed at all stages of the dissolution process. Combining Whelan's [20] analytical approach for the interface velocity as a function of the annealing time, with a relation between the interface concentration and the interface position, they evaluated the interface position using a Picard-type iteration method. Aaron and Kotler also incorporated the effects of the particle curvature into their model using the Gibbs-Thomson equation [1]. In their model both the interface position and the interface concentration were taken momentarily stationary during the evaluation of the interface position as a function of time.

The effects of interfacial reactions on the rate of the dissolution of spherical particles in both infinite and finite media was examined by Vermolen and Van der Zwaag numerically [15]. Using a finite difference method it was shown, that interfacial reactions can have a significant effect on the dissolution rate and hence on the concentration profiles in the matrix during particle dissolution.

Due to surface tension, the dissolving particles may have a disk-like geometry. Hence, an algorithm capable of solving two dimensional moving boundary problems is desired. In the present work we introduce a general algorithm that can be used to solve two-dimensional problems. It is based on the application of the finite element method on two-dimensional moving boundary problems. A reason to use finite elements is that it allows the use of unstructured grids. Hence the finite element method is more flexible than other discretization

methods using structured grids only. Especially in moving boundary problems as the ones that will be demonstrated in the remainder of this report, unstructured grids are essential.

Before we derive our numerical algorithm, we present the mathematical model we will apply in Chapter 2. Actually, it turns out that this is a standard Stefan problem. Although it may be well known, we repeat the derivation of the Stefan condition at the free boundary, since we need this derivation to improve our numerical method. In Chapter 3 we derive a numerical scheme to solve the Stefan problem. It will be shown, that straightforward discretization of the equations and boundary conditions, may lead to unrealistic interfaces. These problems are investigated and a new algorithm (Chapter 4) which produces nice results is presented. This improved algorithm is based on the derivation of the Stefan boundary conditions of Chapter 2. The algorithm of Chapter 4 may be applied for example, for the dissolution of cylinders in a rectangular cell, which is a two-dimensional problem too. Besides that more particles, dissolving in one cell, may be dealt with now as well. In Chapter 5 results of the new algorithm applied to some practical problems, will be demonstrated. Finally Chapter 6 formulates the conclusions from this report.

Chapter 2

A mathematical model for a dissolution process

Consider an Al_2Cu particle in an Al-Cu alloy at a given temperature. The initial concentration of Al_2Cu in the Aluminium phase is equal to c_0 (mol/m^3), whereas c_{part} denotes the concentration of Al_2Cu in the particle. When the temperature is increased, dissolution of the Al_2Cu particle sets in. At the interface the Al_2Cu concentration is c_{sol} ($c_{part} > c_{sol} > c_0$).

To describe the mathematical model we use the geometry as given in Figure 2.1. The domain

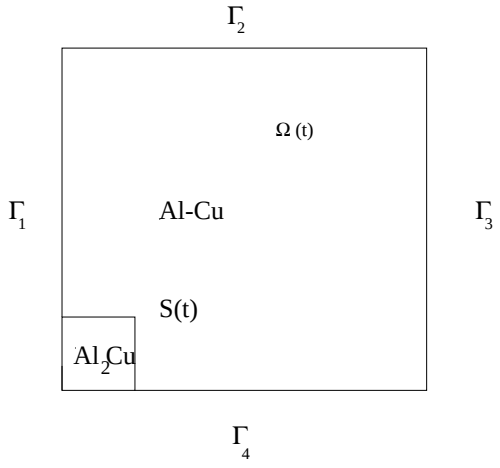

Figure 2.1: Geometry of an Al_2Cu particle in Aluminium.

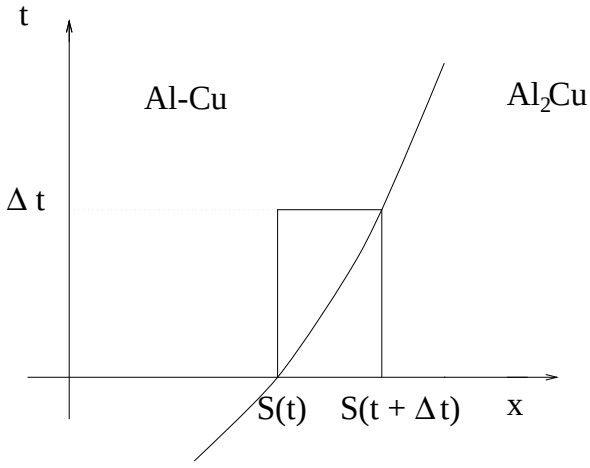

Figure 2.2: The control volume

filled with Aluminium is denoted by $\Omega(t)$. The boundary of this domain consists of the interface $S(t)$ and the outer boundaries Γ_i, $i \in \{1,2,3,4\}$. The outer boundaries are fixed in time, except the intersections of Γ_1 and Γ_4 with $S(t)$. In the Aluminium-rich phase $\Omega(t)$, the Al_2Cu concentration $c(x,y,t)$ satisfies the diffusion equation

$$\frac{\partial c}{\partial t} = \mathbb{D}\Delta c, \quad (x,y) \in \Omega(t), \quad t \in (0,T]. \tag{2.1}$$

The diffusion coefficient $\mathbb{D}$ (m^2/s) is supposed to be constant. As initial condition we use

$$c(x,y,0) = c_0(x,y), \quad (x,y) \in \Omega(0), \tag{2.2}$$

where $\Omega(0)$ is prescribed. We assume that there is no flux of Al_2Cu through the outer boundaries, so

$$\frac{\partial c}{\partial \mathbf{n}}(x,y,t) = 0, \quad (x,y) \in \Gamma_i, \quad i \in \{1,2,3,4\}, \quad t \in [0,T]. \tag{2.3}$$

At the interface the concentration satisfies the equation

$$c(x,y,t) = c_{sol}, \quad (x,y) \in S(t), \quad t \in (0,T]. \tag{2.4}$$

To determine the position of the interface one extra condition is necessary. To derive this boundary condition for a spatially three dimensional problem, we consider a small part of the interface. Suppose that the interface is smooth, which means that it can locally be described by differentiable functions. For a small time step Δt the interface moves in the direction perpendicular to the interface. The x-axis is chosen along the normal. With this choice the position of the interface is locally described by the relation $x = S(t)$. We consider a control volume of width Δy and Δz. The intersection of the control volume with the surface $y = 0, z = 0$ is shown in Figure 2.2. The balance of Al_2Cu atoms leads to the following equation:

$$(S(t+\Delta t) - S(t))\,\Delta y \Delta z c_{sol} = (S(t+\Delta t) - S(t))\,\Delta y \Delta z c_{part} - \mathbb{D}\frac{\partial c}{\partial x}\Delta y \Delta z \Delta t\ . \tag{2.5}$$

Dividing (2.5) by $\Delta y \Delta z \Delta t$ and taking the limit $\Delta t \to 0$ one obtains

$$(c_{part} - c_{sol})\,\frac{dS}{dt} = \mathbb{D}\frac{\partial c}{\partial x}\ .$$

So the extra boundary condition at the interface is

$$(c_{part} - c_{sol})\,v_n(x,y,t) = \mathbb{D}\frac{\partial c}{\partial \mathbf{n}}(x,y,t), \quad (x,y) \in S(t), \quad t \in (0,T]\ , \tag{2.6}$$

where $\mathbf{n}$ is the unit normal vector on the interface pointing outward with respect to $\Omega(t)$ and v_n is the normal velocity of the interface. The moving boundary problem given by (2.1), (2.2), (2.3), (2.4), and (2.6) is known as a Stefan problem [3].

The normalised concentration $\hat{c} = \frac{c - c_0}{c_{sol} - c_0}$, together with a characteristic length scale L, and time scale $\frac{L^2}{\mathbb{D}}$ are used to make the problem dimensionless. After replacing $\hat{c}$ by c we obtain the following Stefan problem: find a concentration c and an interface S such that

$$\frac{\partial c}{\partial t} = \Delta c, \quad (x,y) \in \Omega(t), \quad t \in (0,T]\ , \tag{2.7}$$

$$c(x,y,0) = 0, \quad (x,y) \in \Omega(0)\ , \tag{2.8}$$

$$\frac{\partial c}{\partial \mathbf{n}}(x,y,t) = 0, \;\; (x,y) \in \Gamma_i, \;\; i \in \{1,2,3,4\}, \;\; t \in [0,T]\,, \tag{2.9}$$

$$c(x,y,t) = 1, \;\; (x,y) \in S(t), \;\; t \in (0,T]\,, \tag{2.10}$$

$$v_n(x,y,t) = \lambda \frac{\partial c}{\partial \mathbf{n}}(x,y,t), \;\; (x,y) \in S(t), \;\; t \in (0,T]\,, \tag{2.11}$$

where the dimensionless number λ is given by

$$\lambda = \frac{c_{part} - c_0}{c_{part} - c_{sol}}\,. \tag{2.12}$$

An extensive review of literature on Stefan problems is given in [13]. This bibliography contains 2500 titles on moving boundary problems. In [9] a recent overview is given of numerical methods to simulate convection/diffusion phase change problems.

Various numerical methods are known to solve Stefan problems. In Crank [3] the following types of method are distinguished: front-tracking, front-fixing, and fixed-domain methods. The latter two methods can only be used when the concentration on the interface is a constant. In the near future we want to simulate also dissolution processes with a varying concentration on the interface (e.g. first order reaction at the interface [15], or dissolution in ternary Al-alloys [16]). Therefore we choose a front-tracking method to solve our problem numerically. The front-tracking method we apply, is a two-dimensional extension of the method of Murray and Landis [8] as described in [10], [7], [6] and [19]. A detailed description of the adaptation of the interface in time is lacking in most of the literature on front-tracking methods, except [6]. We present various methods for this adaptation in Chapters 3 and 4. The results from these methods are compared in Chapter 5 by numerical experiments.

Chapter 3

Solution of the Stefan problem

As we have seen in Chapter 2, the problem to be solved can be modelled as a Stefan problem. Our method runs as follows. In the first time-step the diffusion equation (2.7) together with the initial (2.8) and boundary conditions (2.9) and (2.10) is solved. On the free boundary S only the Dirichlet condition (2.10) is used. In the next time steps first the boundary is moved using the boundary condition (2.11). This means that the co-ordinates of the free surface at time $t + \Delta t$ are approximated by:

$$\mathbf{x}(t + \Delta t) = \mathbf{x}(\mathbf{t}) + \Delta t v_n = \mathbf{x}(\mathbf{t}) + \lambda \frac{\partial c}{\partial \mathbf{n}} \Delta t \ \mathbf{n}. \tag{3.1}$$

Once the boundary is moved, the concentration c can be computed in the new region using equation (2.7). However, the computation of the concentration implies that we have to compute $\frac{c(t+\Delta t)-c(t)}{\Delta t}$. We do not know $c(t)$ in the nodal points, since due to the displacement of the boundary also all nodes have been moved [2]. So either we have to interpolate the concentration to the new nodes, or we have to make a correction for the displacement. Interpolation is of course possible, but relatively expensive. The correction is much more simple. If we compute the time-derivative based on the old and the new points, then it is clear that we actually use a material derivative

$$\frac{Dc}{Dt} = \frac{\partial c}{\partial t} + \mathbf{u}_{mesh} \cdot \nabla c, \tag{3.2}$$

where the so-called mesh velocity is defined by

$$\mathbf{u}_{mesh} = \frac{\mathbf{x}(t + \Delta t) - \mathbf{x}(t)}{\Delta t}, \tag{3.3}$$

with $\mathbf{x}$ the co-ordinate vector in a point. Substituting Equation (3.2) into Equation (2.7) leads to:

$$\frac{Dc}{Dt} - div(\nabla c) - \mathbf{u}_{mesh} \cdot \nabla c = 0. \tag{3.4}$$

The discretization of equation (3.4) is performed by a standard Galerkin finite element method

using linear triangles. The time-discretization is performed by an implicit Euler method. Appendix B discusses several alternatives with respect to the combination of Euler implicit and the moving boundary.

A straight-forward method to compute the new position of the free boundary is given by Algorithm 1.

Algorithm 1 Straight-forward computation of the displacement of the free boundary.

Compute the constant gradient of the concentration in the elements connected to the free boundary.
Compute the gradient of the concentration in the vertices at the free boundary by averaging over neighbouring elements.
Define the normal in the vertices by
for all i **do**
$\hat{\mathbf{n}}_i = \frac{1}{2}(\mathbf{n}_{i-\frac{1}{2}} + \mathbf{n}_{i+\frac{1}{2}})$; $\mathbf{n}_i = \frac{\hat{\mathbf{n}}_i}{||\hat{\mathbf{n}}_i||}$.
end for
Define $\frac{\partial c}{\partial \mathbf{n}}$ in the vertices by the inner product of the gradient and the normal vector.
Apply formula (3.1) to compute the new positions of the vertices.

At the start of the computation we are faced with the fact that the normal of the interface is not defined in the middle point of S (see Figure 2.1), where we have a corner of 90°. Note that in fact the normal is not defined in vertices at the boundary unless the two surrounding line elements have the same direction. For that reason the outward normal in a vertex point is defined as the normalized mean value of the normals at the two line elements adjacent to that point. Figure 3.1 shows the definition of these tangential and normal vectors for a curved boundary. Using this definition of the normal vector, the displacement of the vertices of the

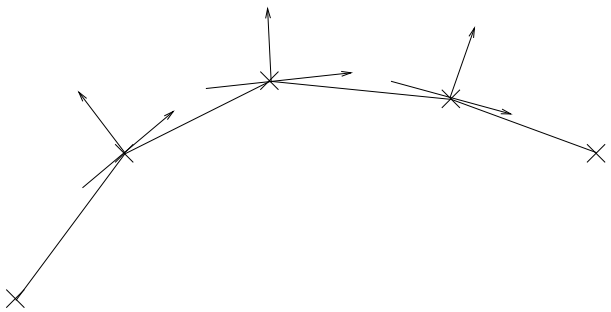

Figure 3.1: Definition of tangential and normal vectors in vertices of boundary

boundary can immediately be deducted from equation (3.1).
A different way of computing the normal in the vertices is presented by Lynch [6]. He proposes to take a weighted average of the surrounding normals, where the weights are defined

by the length of the adjacent boundary elements. A clear disadvantage of this approach is that the normal depends on the local grid size. For example, if at a 90° corner the mesh size at one side is larger than at the other one, the normal will not be defined under an angle of 45° as in our method.

Figure 3.2 shows the free boundary during the first 10 time steps. Only the boundary in the neighbourhood of the left-hand corner is plotted, the rest of the boundary remains unchanged. It is clear that in the mid point of the free boundary, where we start with the 90° corner, the free boundary moves more slowly than in the other points. However, from the physical point of view one would expect this point to move faster than the surrounding points. To inves-

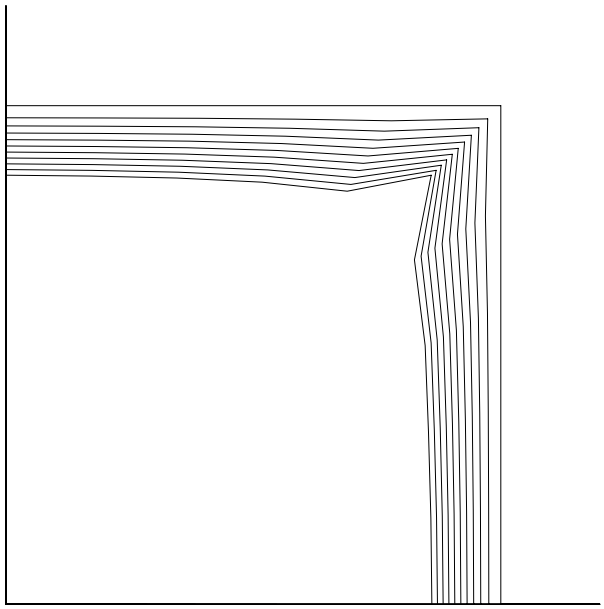

Figure 3.2: Free boundary in the first 10 time steps

tigate this strange behaviour we consider the artificial case where all the equi-concentration lines are parallel to the free boundary. Figure 3.3 shows the boundary (solid line) and one such a concentration line. The dashed line denotes the new position of the free boundary, one would intuitively expect, after one time step $\Delta t = 1$ with the parameter λ equal to 1. Let the difference in concentration between free boundary and the plotted equi-concentration line be equal to Δc. Let the distance between the points `x` and `x'` be equal to Δx. Then the value of $\frac{\partial c}{\partial \mathbf{n}}$ is approximately $\frac{\Delta c}{\Delta x}$. However, the distance between the points `y` and `y'` is equal to $\sqrt{2}\Delta x$. Hence along the line `x''`, `x'` the value of $\frac{\partial c}{\partial \mathbf{n}}$ is approximately $\frac{\Delta c}{\sqrt{2}\Delta x}$. Since the average normal is equal to $\mathbf{n} = \frac{\sqrt{2}}{2}\begin{pmatrix} 1 \\ 1 \end{pmatrix}$ this implies that the free boundary in the point `y` does not move to `y''`, but only to the point precisely in the middle of `y` and `y''`. Although in practice the concentration lines are not parallel to the free boundary; the point `y'` will be

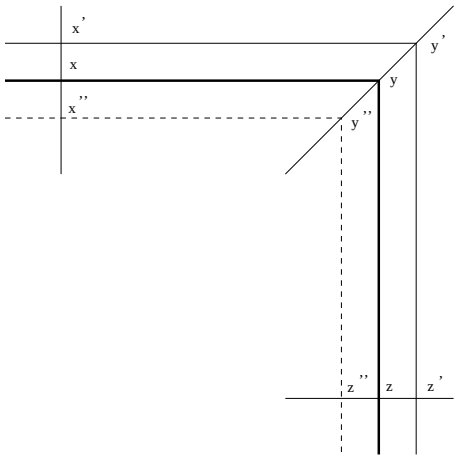

Figure 3.3: Assumed position of free boundary in case of parallel concentration lines

closer to y, still this effect is visible in Figure 3.2. The reason for this inconsistency is in fact, that the Stefan boundary condition is derived under the assumption of a smooth boundary. In the case of a non-differentiable boundary, the limit does not exist and only the integral form of the Stefan boundary condition makes sense.

A natural solution of the problem mentioned above is not to use the gradient and the normal in the vertices, but only those in the mid-side points of the elements. Since the gradient of the concentration is constant per triangle, the value of both the normal and the normal derivative in these mid-side points, is unique. So one can compute the displacement of the mid-side points by equation (3.1) and the only remaining problem is to define how the vertices must be moved. An obvious choice is to choose the new positions of the vertices right in the middle of the new positions of the mid-side points. The displacement in the two corner points can be chosen exactly equal to the displacement of the corresponding mid point. This method is described in Algorithm 2.

Algorithm 2 Displacement of the free boundary based on the mid-side points

---

Compute the constant gradient of the concentration in the elements connected to the free boundary.
Define $\frac{\partial c}{\partial \mathbf{n}}$ in the mid-side points of the elements at the free boundary, by the inner product of the gradient and the normal vector.
Apply formula (3.1) to compute the new positions of the mid-side points.
Define the new positions of the vertices by interpolation between mid-side points.

Figure 3.4 indicates the results of this action for the artificial case of Figure 3.3. It is clear that in this case it is very unlikely that the erratic behaviour of Figure 3.2 will appear. Indeed computations show a very smooth progress of the free boundary. However, there is one serious

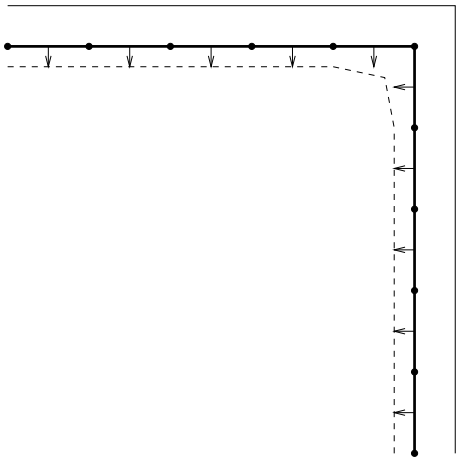

Figure 3.4: Assumed position of the free boundary in case of the adapted movement of the vertices

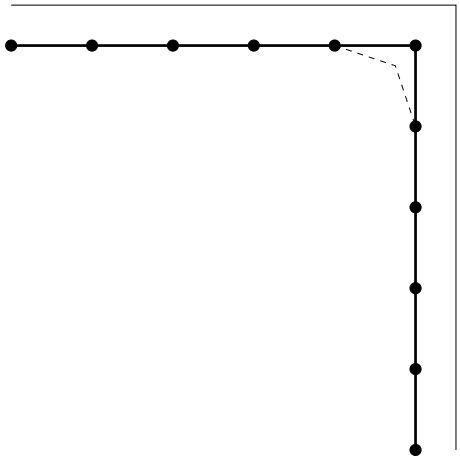

Figure 3.5: Position of the free boundary in absence of a displacement of the mid-side points

drawback in this method. Suppose that none of the mid-side points are moved, for example because $\lambda = 0$. In that case one would expect that the vertices also will be kept at their old positions. However, as can be seen immediately from Figure 3.5, after the first step the vertex corresponding to the 90° corner has been moved inwardly. In the next step an extra smoothing is applied and if this process is repeated we end up with a straight line. For an initial corner larger than 180°, the boundary even moves into the wrong direction. So we are looking for a method that keeps the vertices fixed as long as the mid-side points do not move, but also lacks the behaviour shown in Figure 3.2. In the next chapter we will introduce such a method.

Before considering our new method we consider a variant of a method proposed by Lynch [6]. Lynch suggests to use a weighted average method to compute the position of the new boundary. However, for an equidistant mesh size along the boundaries his method is identical to our Algorithm 1 and for non-equidistant mesh sizes it is even worse. Nevertheless, his derivation based on a weak formulation of the Stefan condition 2.11, inspired us to use a Galerkin approximation of Equation (2.11). The idea is that this is more an integral approach than a differential approach. The Galerkin formulation of Equation (2.11) is given by

$$\int_S v_n \phi d\Gamma = \int_S \lambda \frac{\partial c}{\partial n} \phi d\Gamma, \tag{3.5}$$

with ϕ an arbitrary test function. Exact integration of Equation (3.5) leads to a tridiagonal system of equations, where the matrix to be inverted is the standard mass matrix along the free boundary. Lumping of this matrix reduces this method to Algorithm 1. Figure 3.6 shows the free boundary during the first 10 time steps, using the consistent mass matrix. Indeed the boundary is better than the one shown in Figure 3.2, but still the corner problem is visible.

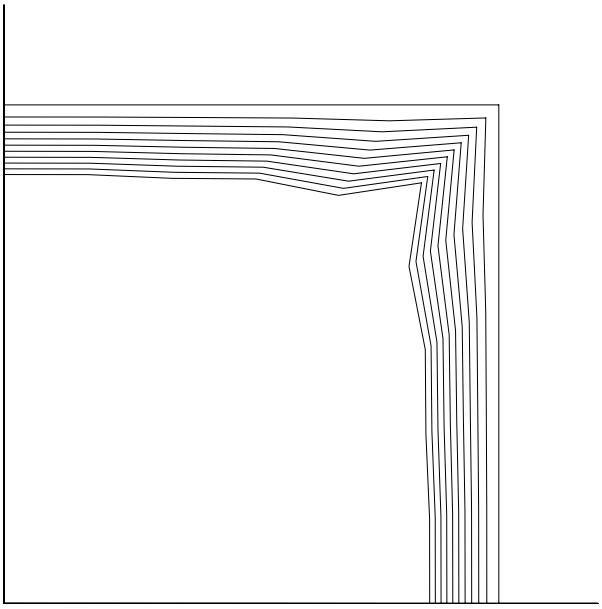

Figure 3.6: Free boundary in the first 10 time steps computed by the "weak approach"

Chapter 4

A new method to compute the displacement of the boundary

Before deriving our new approach for the computation of the free boundary, we first formulate two demands a good numerical method should satisfy.

- Let Δx be a measure for the displacement of the boundary, then it is necessary that the new boundary converges to the old boundary as $\Delta x \to 0$. It would be even better if the error between computed and true position is of order Δx^2.
- Suppose that the corner between two adjacent elements at the moving boundary, measured with respect to the dissolving material, is equal to α. If $\alpha < 180°$ then one expects that this corner increases as soon as the material dissolves. The reason for such a behaviour is that in the case of a straight boundary, diffusion can take place in one direction only, whereas in a corner, diffusion is possible in various directions. So a numerical method should satisfy this property too.

It is clear that the methods mentioned in Chapter 3 do not fulfil these two demands at the same time. Algorithm 1 does not satisfy the second demand, whereas Algorithm 2 does not fulfil the first demand.
We propose the following method.

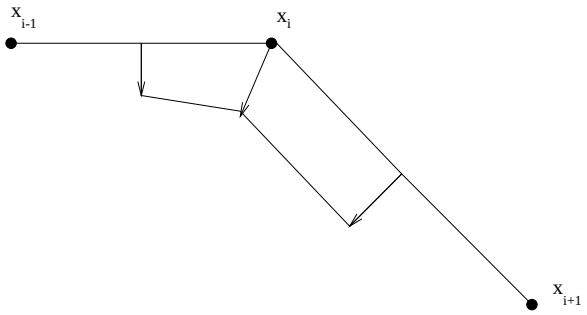

Figure 4.1: Area occupied by the region defined by the displacement of the vertex

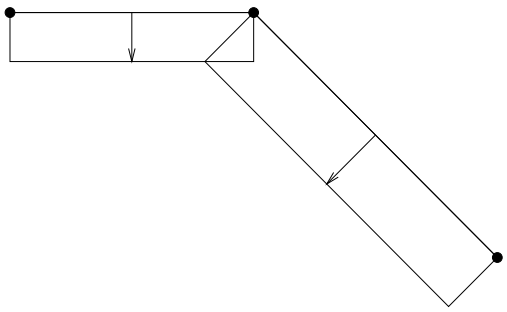

Figure 4.2: Area occupied by the region defined by the displacement of both mid-side points

---

Algorithm 3 Improved scheme for the computation of the free boundary

---

Compute the constant gradient of the concentration in the elements connected to the free boundary.
Define $\frac{\partial c}{\partial \mathbf{n}}$ in the mid-side points of the elements at the free boundary, by the inner product of the gradient and the normal vector.
Apply formula (3.1) to compute the new positions of the mid-side points.
Define the new positions of the vertices along the normal, in such a way that, the area occupied by the region defined by the displacement of the vertex and its surrounding mid-side points (Figure 4.1) is equal to the mean values of the area defined by the displacement of both mid-side points (Figure 4.2).

This method combines the smoothing properties of Algorithm 2 with the non-displacement of the vertex in case the mid-side points remain unchanged.

The motivation for this approach is the following. From the derivation of the Stefan boundary condition it follows that the area of the particle that has been dissolved is equal to the amount of diffused material. The flux through the element (x_{i-1}, x_i) is approximately equal to:

$$\mathbb{D}\frac{\partial c}{\partial \mathbf{n}}(x_{i-\frac{1}{2}})l_i\Delta t, \tag{4.1}$$

with l_i the length of the line element (x_{i-1}, x_i). Hence the amount of diffused material through the boundary $(x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}})$ is equal to

$$\frac{\Delta t}{2}(\mathbb{D}\frac{\partial c}{\partial \mathbf{n}}(x_{i-\frac{1}{2}})l_i + \mathbb{D}\frac{\partial c}{\partial \mathbf{n}}(x_{i+\frac{1}{2}})l_{i+1}). \tag{4.2}$$

The amount M of material dissolved, is approximately equal to $(c_{part} - c_{sol})O$, where O is the area defined in Figure 4.1. Due to the balance of atoms M must be equal to the amount

of diffused material given in Equation (4.2). Making this equation dimensionless this is equal to (see (2.12))

$$O = \frac{\lambda \Delta t}{2}\left(\frac{\partial c}{\partial \mathbf{n}}(x_{i-\frac{1}{2}})l_i + \frac{\partial c}{\partial \mathbf{n}}(x_{i+\frac{1}{2}})l_{i+1}\right). \tag{4.3}$$

The right-hand side of (4.3) is the mean value of the areas defined in Figure 4.2.

There is a drawback of Algorithm 3, which can be explained as follows. Once the displacement of the vertices has been computed by means of Algorithm 3, the mid-side points are moved to new positions, since they are always in the middle of two vertices. As a consequence the amount of dissolved material is no longer equal to M. If we want to have a displacement which gives an amount of dissolved material that is exactly equal to M, the situation is more complex. Consider two adjacent line elements (x_{i-1}, x_i), and (x_i, x_{i+1}), with length l_i and l_{i+1} respectively (Figure 4.3). The mid-side points of these elements are denoted by $x_{i-\frac{1}{2}}$ and

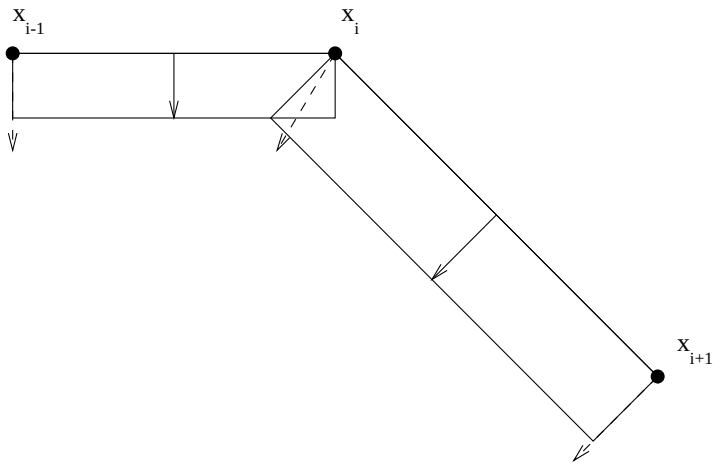

Figure 4.3: Two adjacent elements in the free boundary

$x_{i+\frac{1}{2}}$. Let the from formula (3.1) computed displacement in the mid-side points, be equal to $\delta x_{i-\frac{1}{2}}$ and $\delta x_{i+\frac{1}{2}}$. The new position of vertex x_i is denoted by $\hat{\mathbf{x}}_i$. The length of the displacement is given as $\Delta x_i = ||\hat{\mathbf{x}}_i - \mathbf{x}_i||$. Once the displacement in the vertices is computed, also the displacements in the mid-side points change. In order to get both a local and global equilibrium in the amount of dissolved material, it is necessary, that the new area is identical to

$$\frac{1}{2}l_i\delta x_{i-\frac{1}{2}} + \frac{1}{2}l_{i+1}\delta x_{i+\frac{1}{2}}. \tag{4.4}$$

The area O depends on Δx_i, $\Delta x_{i-\frac{1}{2}}$ and $\Delta x_{i+\frac{1}{2}}$, where $\Delta x_{i-\frac{1}{2}}$ is the adapted length of the displacement in $x_{i-\frac{1}{2}}$. Since $\Delta x_{i-\frac{1}{2}}$ and $\Delta x_{i+\frac{1}{2}}$ depend on Δx_{i-1}, Δx_i and Δx_{i+1} the relation is non-linear. In order to solve this system of non-linear equations we propose the following algorithm:

Algorithm 4 "Exact" satisfaction of balance between dissolution and diffusion

for all i **do**
  Compute Δx_i from $\delta x_{i-\frac{1}{2}}$ and $\delta x_{i+\frac{1}{2}}$ according to *Algorithm 3*.
  $\hat{\mathbf{x}}_i = \mathbf{x}_i + \Delta x_i \mathbf{n}_i$.
end for
while not converged **do**
  for all i **do**
    $\hat{\mathbf{x}}_{i-\frac{1}{2}} = \frac{1}{2}(\hat{\mathbf{x}}_{i-1} + \hat{\mathbf{x}}_i)$.
    Compute Δx_i from *Algorithm 3* using the areas occupied by the known quadrilaterals $(\mathbf{x}_{i-\frac{1}{2}}, \mathbf{x}_i, \hat{\mathbf{x}}_i, \hat{\mathbf{x}}_{i-\frac{1}{2}})$ and $(\mathbf{x}_{i+\frac{1}{2}}, \mathbf{x}_i, \hat{\mathbf{x}}_i, \hat{\mathbf{x}}_{i+\frac{1}{2}})$.
    $\hat{\mathbf{x}}_i = \hat{\mathbf{x}}_i + \omega(\mathbf{x}_i + \Delta x_i \mathbf{n}_i - \hat{\mathbf{x}}_i)$.
  end for
end while

All formulae from this algorithm can be found in appendix A.

Numerical experiments showed that, in the case of a relaxation parameter ω equal to 1, Algorithm 4 did not converge. The free surface switched between two states, for successive iterations. This behaviour is caused by an overestimation of the correction in each iteration. To solve this problem an under-relaxation factor $\omega < 1$ has been tried. Practical computations show that the iteration process is rather insensitive for the value of the ω. For $\omega = 0.5$ we usually got convergence within 5 iterations. The process is stopped as soon as the difference between the area due to the mid-side displacements, as defined in Equation (4.2) and the final area is less than 1%.

In first instance we did not expect much difference between Algorithm 3 and Algorithm 4. But from Figures 4.4 and 4.5 it is clear that, especially during the first time-steps, the behaviour is quite different. To get an idea of the mesh, Figure 4.6 shows the original coarse mesh in the neighbourhood of the free surface. In order to compare the accuracy of both algorithms we halved the space-step, i.e. we increased the number of elements by a factor 4. In Figure 4.8 the free boundary computed by Algorithm 4 is plotted, whereas Figure 4.7 gives the free boundary after the first time-step, using Algorithm 3. It is clear from this picture, that the boundary of Algorithm 3 is unacceptable, whereas that of Algorithm 4 seems reasonably good. The reason for this strange behaviour is that the displacement of the two mid-side points around the sharp corner, is very close to the line $x = y$ (Figure 4.9). As a consequence the first estimate is very inaccurate. Iteration as used in Algorithm 4, however, is able to solve this problem nicely.

If we enlarge the time-step somewhat more, the displacement of the mid-side points may even cross the line $x = y$. In that case both algorithms fail, since they produce a negative displacement for the vertex at $y = x$. Hence reducing the space-step, implies that we also have to decrease the time-step. Figures 4.10 and 4.11 show the free boundaries for the finer

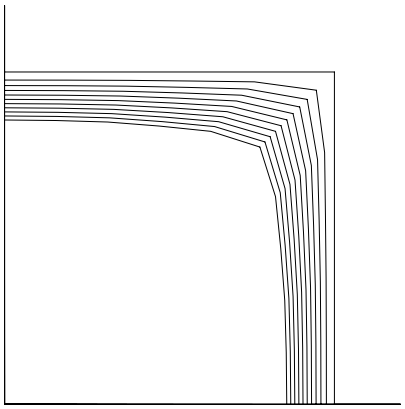

Figure 4.4: Free boundary during the first 10 time-steps using Algorithm 3

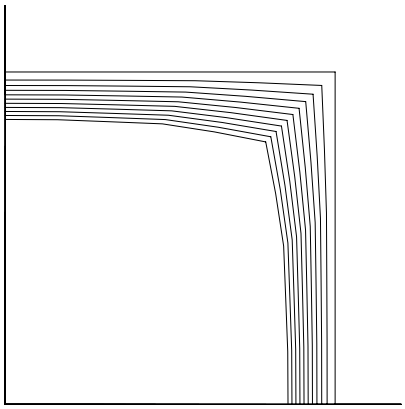

Figure 4.5: Free boundary during the first 10 time-steps using Algorithm 4

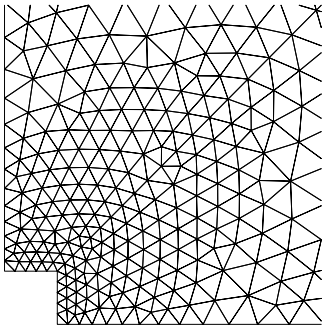

Figure 4.6: Coarse mesh in the neighbourhood of the free boundary

mesh with the time-step divided by two. The boundaries are plotted at the same time-levels as the ones in Figures 4.4 and 4.5. Further halving of the space- and time-steps leads to nice pictures for Algorithm 4. However, Algorithm 3 produces in the first time-step, a free boundary as the one depicted in Figure 4.7. Of course, at that moment the mesh generator fails to create a new mesh.

Some final remarks concern the generation of the mesh and the free boundaries during the time-stepping process.

At the start of the computation a mesh is generated using the initial boundary. In each time-step the boundary is updated according to one of the algorithms of this report. The new free boundary is approximated by a spline. The number of nodes is not changed. The nodes along this spline are distributed in such a way that local refinement in the initial free boundary is kept. Hence the relative distribution of nodes along the initial boundary is maintained

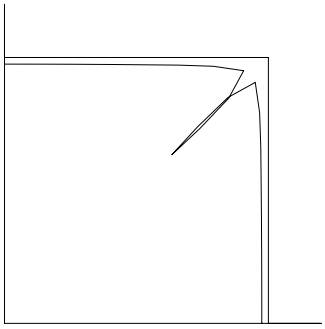

Figure 4.7: Free boundary during the first time-step using Algorithm 3, mesh-size has been halved

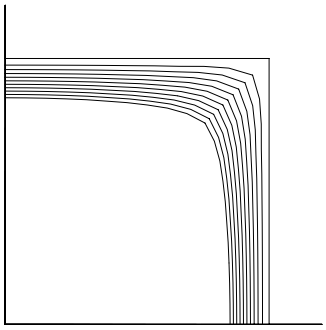

Figure 4.8: Free boundary during the first 10 time-steps using Algorithm 4, mesh-size has been halved

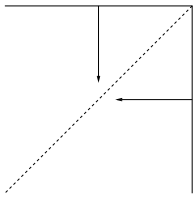

Figure 4.9: Displacement of the mid-side points is too close to the line $x = y$

during the whole process. Once the boundary is changed, the mesh is updated. In first instance all points in the mesh are repositioned by taking the mean value of the co-ordinates of neighbouring nodes. This averaging process is performed in a number of steps in a Gauss-Seidel like procedure.

When the mesh is created, the quality of the mesh and the distances at the free boundary are checked. If the angles of the triangles in the new mesh are too large, or if the distances between nodes at the free boundary differ too much from the original distances, the mesh is completely regenerated. In that case also the nodes at the free boundary are recomputed and the number of nodes at this boundary may be changed. After that, the just computed solution is interpolated to the new mesh. Of course this remeshing technique is quite expensive and so it is only carried out if necessary.

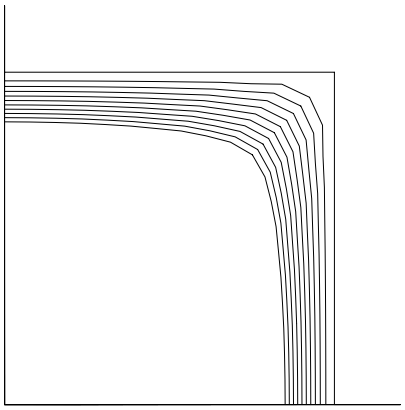

Figure 4.10: Free boundary during the first 10 even time-steps using Algorithm 3; mesh-size and time-step have been halved

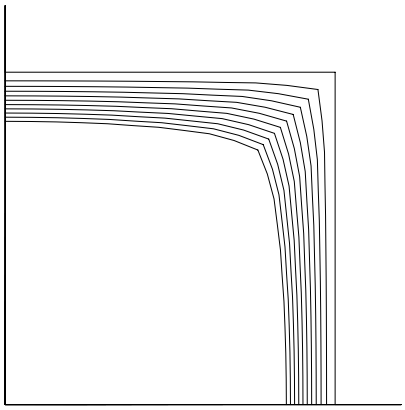

Figure 4.11: Free boundary during the first 10 even time-steps using Algorithm 4; mesh-size and time-step have been halved

Chapter 5

Experiments

An algorithm has been developed, suitable for use to investigate the dissolution kinetics for a two-dimensional case. One of the goals of the algorithm is to determine the dissolution kinetics of disk-like particles. Before two-dimensional problems are considered, we compare the outcomes of our algorithm to one-dimensional results. In all our examples we have chosen the following parameters:

diffusion coefficient	$\mathbb{D}$	=	0.1,
concentration at the interface	c_{sol}	=	1,
initial concentration	c_0	=	0,
	$\frac{\mathbb{D}}{c_{part}-c_{sol}}$	=	0.0101.

5.1 Axial symmetric problems

The finite element results are compared to the finite difference answers [15] obtained for the one-dimensional case with axial symmetry. The two-dimensional finite element method is applied to a circular particle with radius 1, dissolving in a circular cell with radius 5. For both algorithms the concentration profiles at $t = 20s, 40s, 60s$ and $80s$ are displayed in Figures 5.1 and 5.2. The differences between the results obtained from the one-dimensional finite difference method and the two-dimensional finite element method are very small. This indicates that the algorithm based on the finite element method is reliable for the axi-symmetric problem.

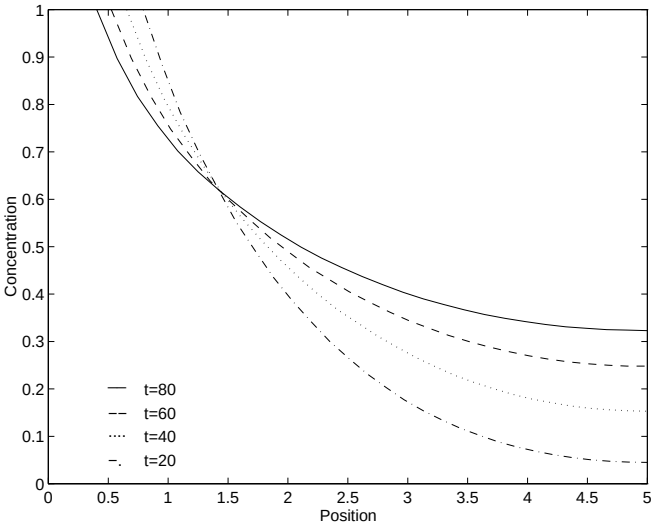

Figure 5.1: Concentration profiles at various times from the 1D FDM

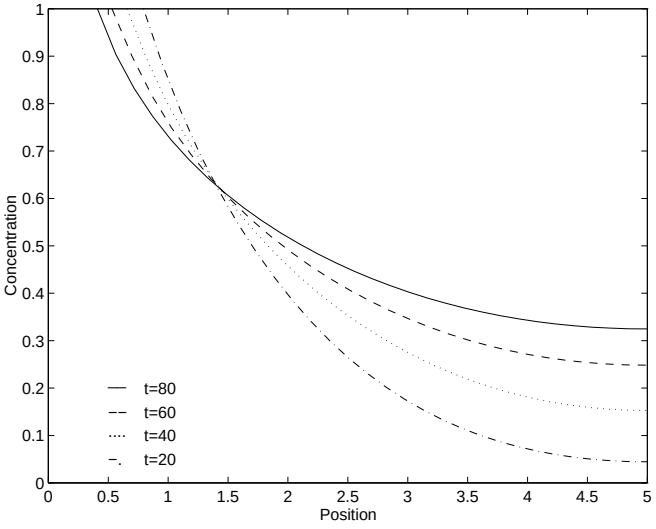

Figure 5.2: Concentration profiles on a line perpendicular to the cylinder at various times from the 2D FEM

5.2 Disk-like problems

Next we consider the dissolution of disk-like particles. This problem is in fact three-dimensional, but due to the axi-symmetry it can be solved as a two-dimensional problem. Figure 5.3 shows the configuration of the particle in the (r, z) plane. The disk length is D and the cell length L. Various values of D are considered. The radius of disk and cell are respectively 1 and 5, whereas L has the fixed value of 5. The finite element method uses axi-symmetrical elements.

Firstly we consider the special case of $D = L$. The resulting problem is again one-dimensional.

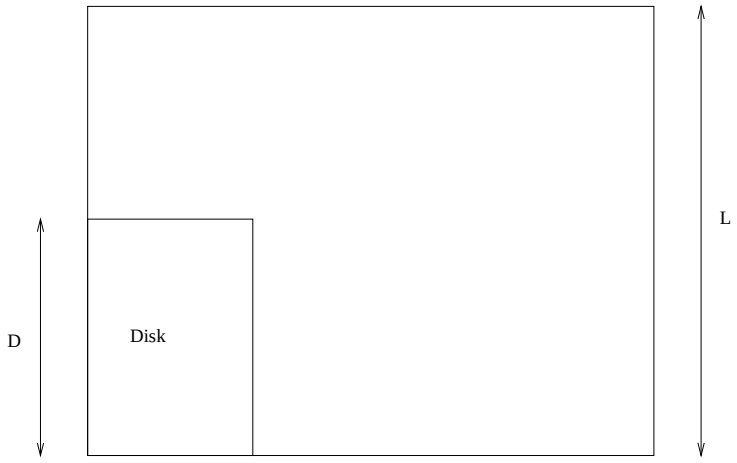

Figure 5.3: Configuration of the axial-symmetric disk; Disk length D; Cylinder length L

In Figure 5.4 Curve I corresponds to the 1D axi-symmetric finite difference code. The one-

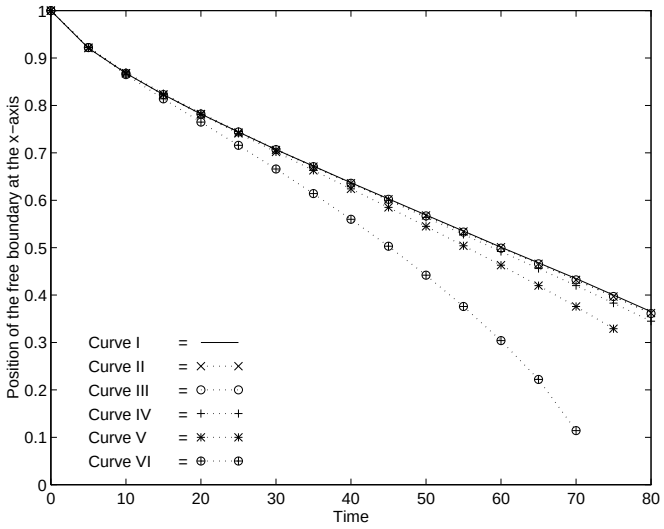

Figure 5.4: Radius of cross-section of the dissolving cylinder with the OXY-plane

dimensional difference method can be applied for planar, cylindrical and spherical geometries.

From a comparison of the 1D finite difference method (fdm) with an analytical method ([4]) during the first iterations for spherical geometry, the fdm method used, may be considered as reliable. The results from the finite element method are shown as Curve II. Both results are nearly the same.

Secondly we consider the position of the intersection of the free boundary and Γ_4 as a function of time for various disk-heights. Curves III, IV, V and VI correspond to a disk-length of $0.99L$, $0.95L$, $0.90L$ and $0.75L$, respectively. In the limit (Curves III and IV), the position approaches that of the one-dimensional case with axial symmetry. Note that for a small value of D ($D = 0.75L$), the final position of the intersection is close to zero, whereas the final position in the one-dimensional problem is approximately equal to 0.35. So we may conclude that the behaviour of a disk-like particle is different from a cylindrical one, especially when the time increases.

Figure 5.5 gives the concentration c at $r = 5, z = 0$, as a function of the disk length for differ-

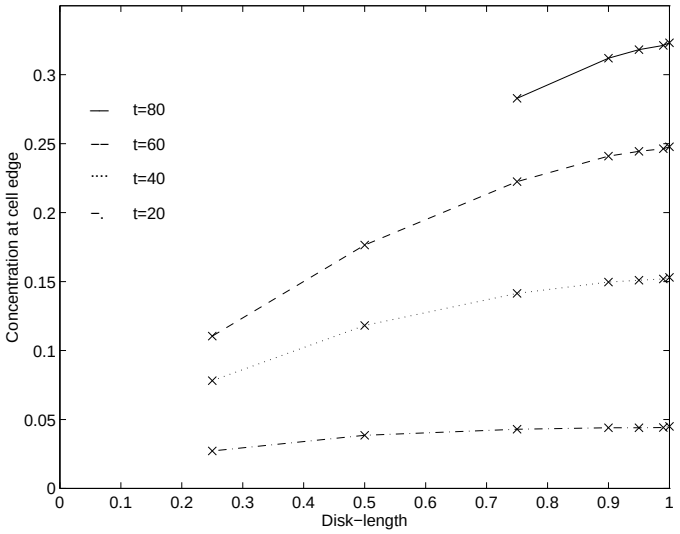

Figure 5.5: Concentration at $r = 5, z = 0$ as function of the disk-length

ent dissolution times. It can be seen in Figure 5.5 that the influence of the disk length on the concentration at $r = 5$ increases with increasing dissolution time. This is in accordance with what one expects physically, because at early stages of the dissolution process the diffusion fields are small, so the the shape of the particle hardly influences the concentration at the intersection of Γ_3 and Γ_4.

5.3 A cylinder dissolving in a bar

In metallurgical literature only one-dimensional algorithms are used to investigate dissolution kinetics. This requires the use of an equally shaped cell in which the particle dissolves such that the cell volume equals the real volume around the particle. Since the two-dimensional algorithm has been developed, the error of the last mentioned approach may be analysed. The

dissolution kinetics of a cylinder, dissolving in a bar, has been compared to the dissolution of a cylinder dissolving in a cylinder with an equal volume. As can be expected for cases in which the cell size is large, the approach with an equally shaped cell is most accurate. The average distance between the free boundary and the origin does not differ much for both geometries of the cell. However, for the case of a bar-like cell, the movement of the free boundary near the intersections with the co-ordinate axes, will be smaller than the movement of the free boundary near the intersection with the line y = x, causing a shape change during dissolution. These effects will be more pronounced as dissolution proceeds. For both approaches iso-concentration lines have been sketched in Figures 5.6 and 5.7. In these examples the following data have been used:

radius particle = 1,
radius outer cylinder = 5,
edge of square = 4.43 (hence the area of the square and the cylinder are equal).

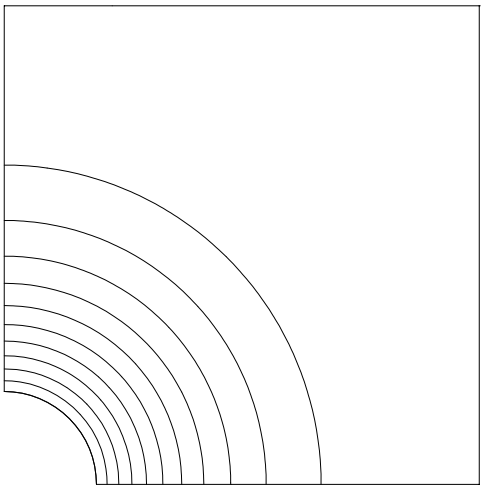

Figure 5.6: Iso-concentration lines for a cylinder dissolving in a bar

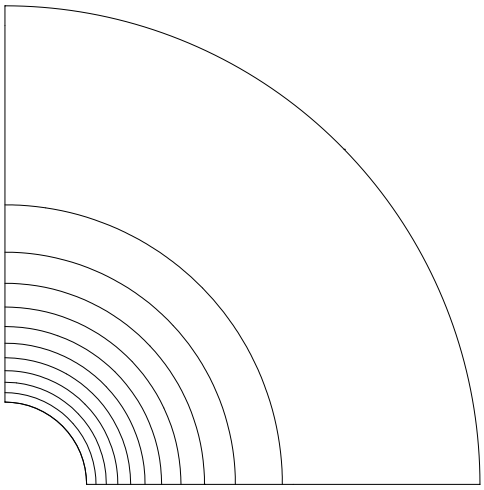

Figure 5.7: Iso-concentration lines for a cylinder dissolving in a cylinder

Reducing the cell size with respect to the particle size, would reveal a larger difference between both approaches. However, such small cell sizes are not likely to occur in metallurgy, so the approach made in literature can be considered as reasonable.

5.4 A bar dissolving in a bar

The free boundary of a square dissolving in a square has been sketched at different stages of the dissolution process in Figure 5.8. The edge of the particle has length 1; the edge of the cell has length 5. One sees that the shape of the free boundary becomes more rounded as dissolution proceeds. The shape of the free boundary even becomes almost circular at later stages of the dissolution process. The same has been done for the dissolution of a disk in a

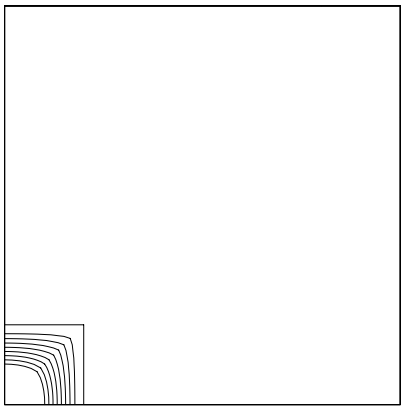

Figure 5.8: Free boundary of a bar dissolving in a bar at various stages of the dissolution process.

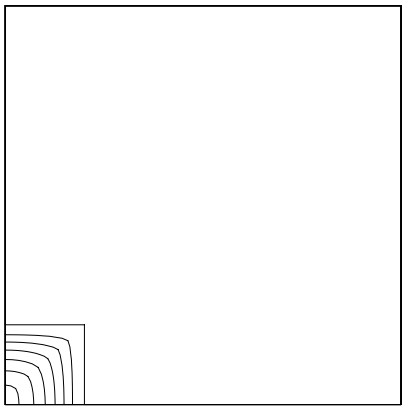

Figure 5.9: Free boundary of a disk dissolving in a disk at various stages of the dissolution process.

disk in Figure 5.9. In this axi-symmetric case the inner radius is equal to 1 and the outer radius equal to 5. Furthermore we used $D = 1$ and $L = 5$. The same shape transition is observed.

5.5 Two particles dissolving in one cell

In metallic systems the particle size is nonuniform. Most authors in literature incorporate a particle size distribution using the assumptions

- all particles dissolve in an equally shaped cell,
- the average concentration of alloying element is equal in each cell.

This implies a unique correspondence between the cell size and the particle size. Moreover most authors assume that there is no mass transfer between the cells, except Tundal and Ryum [14]. When the smallest particle is dissolved completely, the cells related to the residual particles are enlarged by them such that the volume of the cells equal the sum of the volumes of the original cells.

In our two-dimensional numerical method, these assumptions can be released. Therefore we compute the dissolution of two particles in one cell. In this problem the length of the edge of the square is equal to 5, whereas the radii of the particles are equal to 2 and 0.5 respectively. The movement of the free boundaries is visualised in Figure 5.10. As soft-impingement occurs, i.e. the interaction of the diffusion fields around the particles, movement of the free boundary is influenced by the presence of different particles. Iso-concentration lines for this case have been sketched in Figure 5.11. These lines can be compared to the iso-concentration lines for one particle in a bar-like cell, as have been sketched in Figure 5.6. It is a straightforward exercise to extend these calculations to a system with more cells.

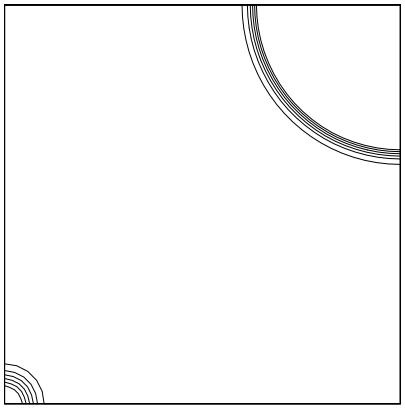

Figure 5.10: the movement of two circular free boundaries in a cell in which two particles dissolve.

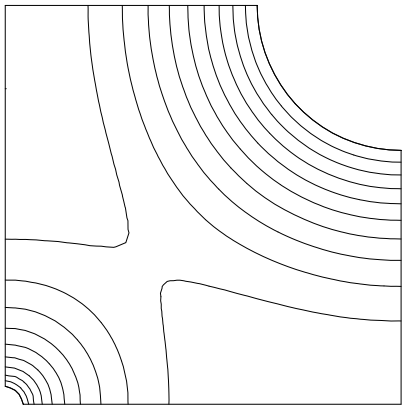

Figure 5.11: Iso-concentration lines in a cell in which two particles dissolve.

Chapter 6

Conclusions

In this report we have investigated particle dissolution in rigid media by numerical techniques. The mathematical model used is that of a free boundary problem of Stefan type. These equations are solved by a two-dimensional finite element method. It has been shown that this approach leads to an accurate solution of the problem.

With respect to the adaptation of the free boundary during time-stepping it has been demonstrated that sharp corners require a special algorithm. Several algorithms have been developed. From these algorithms, the non-linear approach, based on the discretization of the integral balance between dissolution and diffusion, has proven to be superior.

The finite element method applied is based upon a displacement of all nodes. The free boundary is approximated by a spline and the nodes are redistributed in order to maintain the original coarseness of the nodes. If necessary remeshing is applied.

It has been shown that for some types of particles two-dimensional effects can not be neglected. For those cases it is not sufficient to run a one-dimensional code and our 2D finite element method gives a very attractive alternative.

Appendix A Detailed description of Algorithm 4

Algorithm 4 "Exact" satisfaction of balance between dissolution and diffusion

Compute the constant gradient of the concentration in the elements connected to the free boundary.
Define the normal in the vertices by
for all i **do**
$\hat{\mathbf{n}}_i = \frac{1}{2}(\mathbf{n}_{i-\frac{1}{2}} + \mathbf{n}_{i+\frac{1}{2}})$; $\mathbf{n}_i = \frac{\hat{\mathbf{n}}_i}{||\hat{\mathbf{n}}_i||}$.
end for
for all i **do**
$\frac{\partial c}{\partial \mathbf{n}}(\mathbf{x}_{i\pm\frac{1}{2}}) = \nabla c(\mathbf{x}_{i\pm\frac{1}{2}}) \cdot \mathbf{n}_{i\pm\frac{1}{2}}$
$\delta x_{i\pm\frac{1}{2}} = \lambda \frac{\partial c}{\partial \mathbf{n}}(\mathbf{x}_{i\pm\frac{1}{2}})\Delta t$
$\hat{\mathbf{x}}_{i\pm\frac{1}{2}} = \mathbf{x}_{i\pm\frac{1}{2}} + \delta x_{i\pm\frac{1}{2}} \mathbf{n}_{i\pm\frac{1}{2}}$
Compute Δx_i from $\delta x_{i-\frac{1}{2}}$ and $\delta x_{i+\frac{1}{2}}$ according to *Algorithm 3*.
The equivalence of the areas gives:

$$\begin{aligned} \frac{1}{2}(l_i \delta x_{i-\frac{1}{2}} + l_{i+1}\delta x_{i+\frac{1}{2}}) &= \frac{F_1}{2} \\ F_1 &= (x_i - \hat{x}_{i-\frac{1}{2}})(y_{i-\frac{1}{2}} - \hat{y}_i) - (y_i - \hat{y}_{i-\frac{1}{2}})(x_{i-\frac{1}{2}} - \hat{x}_i) + \\ &\quad (x_{i+\frac{1}{2}} - \hat{x}_i)(y_i - \hat{y}_{i+\frac{1}{2}}) - (y_{i+\frac{1}{2}} - \hat{y}_i)(x_i - \hat{x}_{i+\frac{1}{2}}), \end{aligned} \quad (7.1)$$

where x and y denote the components of the vector $\mathbf{x}$.
Substitution of $\hat{\mathbf{x}}_i = \mathbf{x}_i + \Delta x_i \mathbf{n}_i$ gives:

$$\begin{aligned} \Delta x_i &= \frac{F_2}{F_3} \\ F_2 &= (l_i \delta x_{i-\frac{1}{2}} + l_{i+1}\delta x_{i+\frac{1}{2}}) - (x_i - \hat{x}_{i-\frac{1}{2}})(y_{i-\frac{1}{2}} - y_i) + (y_i - \hat{y}_{i-\frac{1}{2}})(x_{i-\frac{1}{2}} - x_i) - \\ &\quad (x_{i+\frac{1}{2}} - x_i)(y_i - \hat{y}_{i+\frac{1}{2}}) + (y_{i+\frac{1}{2}} - y_i)(x_i - \hat{x}_{i+\frac{1}{2}}) \\ F_3 &= n_i^x(\hat{y}_{i+\frac{1}{2}} - \hat{y}_{i-\frac{1}{2}}) + n_i^y(\hat{x}_{i-\frac{1}{2}} - \hat{x}_{i+\frac{1}{2}}), \end{aligned} \quad (7.2)$$

with n_i^x and n_i^y the components of $\mathbf{n}_i$.
$\hat{\mathbf{x}}_i = \hat{\mathbf{x}}_i + \omega(\mathbf{x}_i + \Delta x_i \mathbf{n}_i - \hat{\mathbf{x}}_i)$.
end for
while not converged **do**
for all i **do**
$\hat{\mathbf{x}}_{i-\frac{1}{2}} = \frac{1}{2}(\hat{\mathbf{x}}_{i-1} + \hat{\mathbf{x}}_i)$.
Compute Δx_i from equation (7.2).
end for
end while

Appendix B Various implementations of the implicit Euler method in combination with the free boundary problem

In this appendix we consider three possible implementations of the combination of Euler backward with the moving boundary. First we start with the most explicit formulation, next we consider a somewhat more implicit formulation and finally the complete implicit method is treated.

Explicit treatment of the moving boundary problem

In this approach we start in step n by updating the free boundary explicitly using the concentration c at level n. By interpolating the co-ordinates of the mesh, we get the new co-ordinates $\mathbf{x}^{n+1}$. The mesh velocity at level $n+1$ is now defined by Equation (3.3) using $\mathbf{x}^{n+1}$ and $\mathbf{x}^n$.

$$\mathbf{u}_{mesh} = \frac{\mathbf{x}^{n+1} - \mathbf{x}^n}{\Delta t}, \tag{8.3}$$

Furthermore the convective term is treated explicitly, hence the time discretization may be written as:

$$\frac{c^{n+1}(x^{n+1}) - c^n(x^n)}{\Delta t} - \Delta c^{n+1}(x^{n+1}) - \mathbf{u}^n_{mesh} \cdot \nabla c^n = 0. \tag{8.4}$$

The motivation of this approach is as follows:
Once we have updated the mesh we can solve the diffusion equation (2.7) by Euler implicit. Hence:

$$\frac{c^{n+1}(x^{n+1}) - c^n(x^{n+1})}{\Delta t} - \Delta c^{n+1}(x^{n+1}) = 0. \tag{8.5}$$

The problem is however, that $c^n(x^{n+1})$ is not known. Since we do not want to interpolate we use a Taylor expansion:

$$c^n(x^{n+1}) = c^n(x^n) + \nabla c^n(x^n) \cdot (\mathbf{x}^{n+1} - \mathbf{x}^n). \tag{8.6}$$

Substitution of (8.6) into (8.5) leads to expression (8.4).

A disadvantage of this approach is that the boundary update in the first step may be incorrect since boundary conditions and initial conditions do not match. Another problem might be that remeshing can only be performed after the solution at t^{n+1} has been computed. Hence, it may be possible that the mesh has been distorted so much, that solution of the convection diffusion equation is not possible anymore. The consequence is that remeshing must be performed with more care, since either the new mesh must be acceptable or the old mesh be remeshed first.

Semi implicit treatment of the moving boundary problem

To prevent the disadvantages of the previous approach, we make a slight modification. In the first step of the process we solve the diffusion equation at the original mesh. Hence we start with $\mathbf{u}_{mesh} = 0$. In this way the transient problem is avoided.

At time level $t = t^n$ the mesh is updated using c^n. Thereafter $\mathbf{u}^n_{mesh}$ is computed. If remeshing is required we do not only interpolate c^n to the new nodes but also $\mathbf{u}^n_{mesh}$. Finally we solve the convection diffusion equation (3.2) implicitly, including the convective terms. However, the mesh velocity just computed is used. The time discretization may be written as:

$$\frac{c^{n+1}(x^{n+1}) - c^n(x^n)}{\Delta t} - \Delta c^{n+1}(x^{n+1}) - \mathbf{u}^n_{mesh} \cdot \nabla c^{n+1} = 0. \tag{8.7}$$

In this case we do not have the remeshing problem. However, one might consider the usage of the implicit convection in combination with the explicit mesh velocity as inconsequent. An experiment in which we changed the convection term from implicit to explicit showed negligible differences. For example the difference between the positions of the free boundaries in the explicit and implicit case was less than 0.1%.

Fully implicit treatment of the moving boundary problem

If one expects stability problems it might be a good idea to use a complete implicit scheme. In that case we have to solve

$$\frac{c^{n+1}(x^{n+1}) - c^n(x^n)}{\Delta t} - \Delta c^{n+1}(x^{n+1}) - \mathbf{u}^{n+1}_{mesh} \cdot \nabla c^{n+1} = 0. \tag{8.8}$$

This is a nonlinear problem since we do not know the new region nor the mesh velocity at the new time-level. Hence we must start with an estimate of the new region and accordingly the mesh velocity. The final region and mesh velocity at t^{n+1} is found by iteration.
The fully implicit treatment of the free boundary applied to the test example of Chapter 3, showed a difference less than 0.25% in the free boundary. Hence for our computations it is not necessary to use the fully implicit method.

Bibliography

[1] H.B. Aaron and G.R. Kotler. Second phase dissolution. *Metallurgical Transactions*, 2:393–407, 1971.

[2] M.J. Baines. *Moving finite elements.* Monographs on numerical analysis. Clarendon, Oxford, 1994.

[3] J. Crank. *Free and Moving Boundary Problems.* Clarendon Press, Oxford, 1984.

[4] P. van Mourik F.J. Vermolen and S. Van der Zwaag. An analytical approach to particle dissolution in a finite medium. *Materials Science and Technology.* to appear.

[5] P.G. Shewmon F.V. Nolfi Jr. and J.S. Foster. The dissolution and growth kinetics of spherical precipitates. *Transactions of the Metallurgical Society of AIME*, 245, 1969.

[6] D.R. Lynch. Unified approach to simulation on deforming elements with application to phase change. *J. Comp. Phys.*, 47:387–411, 1982.

[7] D.R. Lynch and K. O'Neill. Continuosly deforming finite elements for the solution of parabolic problems , with and without fase change. *Int. J. Num. Meth. Engng.*, 17:81–96, 1981.

[8] W.D. Murray and F. Landis. Numerical and machine solutions of transient heat-conduction problems involving melting or freezing. *Trans. ASME (C) J. Heat Transfer*, 81:106–112, 1959.

[9] A.A. Samarskii, P.N. Vabishchevich, O.P. Iliev, and A.G. Churbanov. Numerical simulation of convection/diffusion phase change problems - a review. *Int. J. Heat Mass Transfer*, 36:4095–4106, 1993.

[10] V.R.B. Santos. A direct method for solving two-dimensional one phase Stefan problems. *Comp. Meth. Appl. Mech. Engng*, 25:51–64, 1981.

[11] J. Stefan. Über die Theorie der Eisbildung, insbesondere über die Eisbildung im Polarmeere. *Annalen der Physik und Chemie*, 42:269–286, 1891.

[12] U.L. Baty R.A. Tanzilli and R.W. Heckel. Dissolution kinetics of cual2 in an al-4cu alloy. *Metallurgical Transactions*, 1:1651–1656, 1970.

[13] D.A. Tarzia. A bibliography on moving-free boundary problems for the heat-diffusion equation. The Stefan problem. Technical report, Istituto Matematico "Ulisse Dini", University of Florence, Florence, 1988.

[14] U.H. Tundal and N. Ryum. Dissolution of particles in binary alloys: Part i. computer simulations. *Metallurgical Transactions A*, 23A:433–449, 1992.

[15] F.J. Vermolen and S. Van der Zwaag. A numerical model for the dissolution of spherical particles in binary alloys under mixed mode control. *Materials Science and Engineering A*. to appear.

[16] Fred Vermolen. Dissolution in ternary alloys: a mathematical study. Technical report, Laboratory of Materials Science, Delft University of Technology, 1996.

[17] C. Vuik. *The solution of a one-dimensional Stefan problem.* CWI-tract 90. CWI, Amsterdam, 1993.

[18] C. Vuik. Some historical notes about the Stefan problem. *Nieuw Archief voor Wiskunde, 4e serie*, 11:157–167, 1993.

[19] C. Vuik and C. Cuvelier. Numerical solution of an etching problem. *J. Comp. Physics*, 59:247–263, 1985.

[20] M.J. Whelan. On the kinetics of particle dissolution. *Metal Science Journal*, 3:95–97, 1969.